ALMA MATER STUDIORUM
UNIVERSITÁ DI BOLOGNA

DEIS – DIPARTIMENTO DI INFORMATICA,
ELETTRONICA E SISTEMISTICA

METHODOLOGIES FOR VISUAL CORRESPONDENCE

Federico Tombari

TUTOR
Professor
Tullio Salmon Cinotti

COORDINATOR
Professor
Paola Mello

PHD. THESIS
January, 2006 – December, 2008

PHD PROGRAM IN ELECTRONIC, COMPUTER SCIENCE AND TELECOMMUNICATIONS ENGINEERING

CYCLE XXI – ING-INF/05

Contents

Chapter 1

Introduction

Computer vision is a young research discipline that has been intensively studied during the second half of the 20th century until today. Computer vision aims at building artificial systems able to infer information on reality through the processing of data obtained from optical sensors. Among the several problems investigated within this discipline, *visual correspondence* surely represents one of the most important and most studied. As for a general definition, the goal of visual correspondence can be described as finding corresponding (*homologous*) tokens belonging to two sets of pixels. As a matter of fact, to compare together two or more images usually an instance of the visual correspondence problem has to be solved.

Some examples of applications that require solving the visual correspondence problem are as follows:

- robot navigation and autonomous vehicle navigation, that allow a robot or a vehicle to navigate within an unknown environment based on the information coming from two (or more) optic sensors;
- automatic recognition and categorization of objects and faces;
- defect detection and quality control, a typical industrial application aimed at automatic inspection and detection of defective or missing parts of a product;
- video surveillance and video analytics, which include intelligent systems for the automatic detection of motion in a monitored scene (*change detection*), the analysis of the behavior of the subjects moving in a scene (*behavior analysis*) and the recognition of events that might be of particular interests (*event detection*);
- motion estimation in video sequences;
- 3D reconstruction;

- people tracking and counting;
- $\cdots$

There are different tokens on which visual correspondence can be applied, i.e. points, patches and images. In particular, visual correspondence on *keypoints* (also referred to as *interest points*) aims at finding corresponding points within two images that may differ by translation, rotation, scale, occlusions and/or photometric distortions. Interest points on images are generally extracted by means of a feature detector [85, 94]. Then, possible correspondences are evaluated using a representation of each keypoint by means of appropriate descriptors [93]. This problem often occurs for applications such as object categorization and recognition, scene categorization [85, 94].

On another level lies the problem of finding corresponding patches within images: this occurs in problems such as pattern matching [35], stereo matching [112], change detection [106], motion estimation [63]. As for this case, patches usually are represented by square or rectangular windows of pixels, and similarities between patches are computed by means of area-based measures. Corresponding patches are those maximizing the similarity measure within a set of possible candidates. Visual correspondence can even be image-based, e.g. for image registration and alignment by direct methods [54, 149].

This work presents several visual correspondence problems that have been analyzed within the research activity carried out during the Ph.D. course. In particular, three main topics have been investigated. The first concerns efficient visual correspondence, dealing in particular with the proposal of novel techniques for rapidly finding correspondences between image patches based on similarity measures such as those derived from the $L-p$-norm and the Normalized Cross-Correlation. This part of the work will be presented in Chapter 2. The second topic deals with accurate stereo correspondence: a classification and performance evaluation of local stereo correspondence algorithms is proposed in Chapter 3, together with new approaches for accurate stereo matching. Then, Chapter 4 deals with applications of stereo vision to the context of 3D reconstruction and video surveillance. The third main topic concerns robust visual correspondence and regards the proposal of a new measure for visual correspondence, together with its application to the fields of pattern matching, change detection, videosurveillance. This will be illustrated in Chapter 5. Finally, conclusions are drawn in Chapter 6.

Chapter 2

Fast and exhaustive pattern matching

2.1 Introduction

Pattern matching is a classical image analysis task that aims at locating the instances of a given template into a reference set. This task occurs in numerous image analysis applications and consists in determining the regions of the reference image that are *similar* to the template according to a given criterion, and discarding those that are *dissimilar*. Pattern matching is widely deployed for tasks such as quality control, defect detection, robot navigation, face and object recognition, edge detection [54].

The *Full Search* (FS) pattern matching algorithm consists in calculating a correlation, or distortion, function at each position of a search area within an image measuring the degree of similarity, or dissimilarity, between a given template and the portion of the image currently under examination, referred to as *image subwindow*. This determines a set of maximum-correlation, or minimum-distortion, positions that locate the template in the examined image.

Functions typically employed to carry out pattern matching can be divided into two classes: similarity and dissimilarity measures. Similarity measures are often based on correlation, such as the *Normalized Cross-Correlation* (NCC) and the *Zero-mean Normalized Cross-Correlation* (ZNCC). These two measures are typically adopted when robustness with regards to photometric variations is required. On the other hand, a popular class of dissimilarity measures derives from the L_p norm. The two most used L_p norm-based dissimilarity functions are the *Sum of Squared Differences* (SSD) and the *Sum of Absolute Differences* (SAD).

Since the FS algorithm is often unacceptably slow with respect to the application requirements, many faster approaches have been proposed in literature. Among these approaches, *non exhaustive* algorithms yield computational savings by reducing the search space [7,54,73,107,128,149] or by decomposing the template or the image into rectangular regions and approximating each as a polynomial [16,56,115,118].

Conversely, *exhaustive* algorithms speed up the template matching process and yield exactly the same result as the FS. In the case of a dissimilarity-based search, one of the first approaches has been the Partial Distortion Elimination (PDE) [9] and consists in terminating the evaluation of the current distortion measure as soon as it rises above the current minimum. Another approach suitable for dissimilarity-based searches consists in defining a rapidly computable lower bounding function of the adopted distortion measure, so as to check quickly one or more sufficient conditions to detect mismatching candidates without carrying out the heavier computations required by the evaluation of the actual distortion measure. Examples of such an approach include the Successive Elimination Algorithm (SEA) [83], [132], [57] and Gharavi-Alkhansari's algorithm [46]. Analogously, in the case of correlation-based measures such as the NCC and the ZNCC rapidly computable upper bounding functions can be deployed [35,87,88,102].

This chapter investigates on the use of successions of increasingly tighter bounding functions to rapidly detect mismatching candidates both for dissimilarity measures derived from the L_p norm (Section 2.2.2) and similarity measures based on correlation (Section 2.4.5). The goal is to speed up pattern matching yet maintaining the property of exhaustiveness. Experimental results are provided that demonstrate how the proposed techniques are able to achieve notable speed-ups compared to the state-of-the-art approaches.

It is interesting to point out that the proposed exhaustive algorithms can be deployed also to speed up similar tasks such as *motion estimation* and *vector quantization*, given their affinity with the pattern matching problem. This is investigated in section 2.3.

2.2 Fast exhaustive pattern matching based on L_p-norm dissimilarity measures

This section presents a novel algorithm, referred to as *Incremental Dissimilarity Approximations* (IDA), that aims at speeding up pattern matching based on measures derived from the L_p norm. Experimental results concerning more than 6000 pattern matching instances prove that IDA significantly outperforms state-of-the-art approaches and can yield substantial speed-ups with respect to the FS.

2.2.1 Previous Work

For what concerns the SSD, though the typical alternative to the naive FS algorithm is represented by the FFT-based approach, a novel fast FS-equivalent method [57], referred to here as *Projection Kernels (PK)*, was recently proposed in literature. This method was shown to be much more efficient compared to the naive FS-approach as well as to the FFT. As regards the SAD, a well known classical approach is the *Sequential Similarity Detection Algorithm (SSDA)* [7].

The L_p-norm of a M-dimensional vector $X = [x_1, \cdots, x_M]^T$ is defined as:

$$||X||_p = (\sum_{i=1}^{M} |x_i|^p)^{\frac{1}{p}} \tag{2.1}$$

where p is any positive real number [108].

Let now X be the template vector and $Y_1, \cdots, Y_N$ the N candidates (corresponding to the image subwindows) against whom X must be matched, each candidate having the same cardinality as the template vector (i.e. $Y_j = [y_{j,1}, \cdots, y_{j,M}]^T$).

The generic function based on the L_p-norm measuring the dissimilarity between X and Y_j can be written as:

$$||X - Y_j||_p^p = \sum_{i=1}^{M} \left|x_i - y_{j,i}\right|^p \tag{2.2}$$

If $p = 1$ then (2.2) coincides with the SAD function, while $p = 2$ yields the SSD function.

We will now briefly review the FFT-based, PK and SSDA approaches for fast FS-equivalent pattern matching with L_p norm-based dissimilarity functions.

Fast Fourier Transform A common approach for speeding-up the FS pattern matching process based on the SSD function relies on the *Fast Fourier Transform* (FFT). The SSD function can be written as:

$$||X - Y_j||_2^2 = ||X||_2^2 + ||Y_j||_2^2 - 2 \cdot \Theta(X, Y_j) \tag{2.3}$$

where:

$$\Theta(X, Y_j) = \sum_{i=1}^{M} x_i \cdot y_{j,i} \tag{2.4}$$

represents the *dot product* between X and Y_j. In order to achieve computational savings, the FFT approach calculates Θ in the frequency domain according to the correlation theorem. As it would be inefficient to compute $||Y_j||_2^2$ in the frequency domain, this term is usually calculated directly by means of efficient incremental techniques

([24], [91], [131]), as described in [81], while $||X||_2^2$ is computed once for all at initialization time.

Compared to the FS algorithm, the FFT-based approach is more efficient when the template size is large enough compared to the image size. The FFT-based approach cannot be adopted for SAD-based pattern matching since the correlation theorem does not apply to the L_1 norm case.

Projection Kernels The PK method [57] carries out fast full-search equivalent SSD-based pattern matching in the signal domain. With PK, each basis vector U of the *Walsh-Hadamard* transform is used as a projection vector. Then, projecting the template vector X and the candidate vector Y_j onto each of these projection vectors yields a *projected distance* B_j:

$$B_j = U^T X - U^T Y_j \tag{2.5}$$

that can be used to determine a lower bound of the SSD function:

$$||X - Y_j||_2^2 \geq \frac{B_j^2}{||U||_2^2} \tag{2.6}$$

Therefore, if D is the threshold that discriminates between matching and mismatching candidates, it is possible to establish the condition:

$$D < \frac{B_j^2}{||U||_2^2} \tag{2.7}$$

which allows for safely pruning Y_j from the list of candidates. Furthermore, the lower bound can be tightened by using a collection of projection vectors along with the corresponding projected distances. Hence, an iterative algorithm is proposed in [57]: at each step the lower bound is tightened so as to increase the effectiveness of the current pruning condition for those candidates that were not pruned by the previous one.

According to [57], PK is almost two orders of magnitude faster than the FS and FFT-based approaches, but it is more demanding in terms of memory requirements. It is worth pointing out that the size of the template is constrained to be a power of 2. The experimental results reported in [57] show also that PK is more effective with very small templates (i.e. of size 16 × 16 or 32 × 32).

Sequential Similarity Detection Algorithm The SSDA method is a classical approach originally introduced to determine simple inequalities to speed-up SAD-based pattern matching. Let D be a threshold and X, Y_j the template-candidate pair under evaluation. During the computation of the SAD function, at each new element pair x_b, $y_{j,b}$ condition

$$\sum_{i=1}^{b} \left| x_i - y_{j,i} \right| > D \tag{2.8}$$

is tested. As soon as (2.8) is satisfied, the evaluation process is terminated and the value of the last vector index, $\tilde{b}_j$, recorded. Once this is done for all candidates, the best matching candidates correspond to those having high $\tilde{b}_j$. Typically, D is much lower than the global minimum and SSDA turns out not equivalent to the FS (i.e. non exhaustive). In particular, the choice of D determines a cost-performance trade-off: the higher D, the higher the mean number of calculations needed to evaluate the current candidate, and the higher the chance the resulting matching candidates will coincide with those yielded by FS. In order to better deal with this issue D is not kept constant, but increases along with b. Moreover, to obtain a more regular behavior the order of processed vector elements is randomly scrambled. However, it is practically unfeasible to determine a varying D which yield a FS-equivalent algorithm. Therefore, similarly to the other methods considered throughout the paper, we set D to a constant threshold higher than the global minimum: this turns SSDA into a FS-equivalent method.

2.2.2 Incremental Dissimilarity Approximations Algorithm

This subsection describes a novel signal domain method, referred to as *Incremental Dissimilarity Approximations* (IDA), aimed at speeding-up full-search equivalent pattern matching based on the L_p-norm. IDA relies on partitioning the template vector, X, and each candidate vector, Y_j, into a certain number of sub-vectors in order to determine a succession of pruning conditions characterized by increasing tightness and computational weight.

Given an M-dimensional vector, we establish a partition of the vector into r disjoint sub-vectors (not necessarily with the same number of components) by defining a partition, P, of set $S = \{1, 2, \ldots M\}$ into r disjoint sub-sets:

$$\begin{cases} P = \{S_1, S_2 \ldots S_r\}, r \in S \\ \bigcup_{u=1}^{r} S_u = S \\ S_u \cap S_v = \phi, \forall u \neq v, u, v \in \{1, 2, \ldots r\} \end{cases}$$

The minimum number of sub-vectors is 1, that is the vector is actually not partitioned into smaller sub-vectors, the maximum number is M, the vector partitioned into M one-dimensional disjoint sub-vectors. Details concerning an efficient implementation of such partitioning will be discussed later.

Given P, we define the *partial* L_p-norm of vectors X, Y_j restrained to the sub-vectors associated with $S_t \in P$ as:

$$||X||_{p,S_t} = (\sum_{i \in S_t} |x_i|^p)^{\frac{1}{p}} \tag{2.9}$$

$$||Y_j||_{p,S_t} = (\sum_{i \in S_t} |y_{j,i}|^p)^{\frac{1}{p}} \tag{2.10}$$

and the *partial* L_p-dissimilarity between X and Y_j restrained to the sub-vectors associated with $S_t \in P$ as:

$$||X - Y_j||^p_{p,S_t} = \sum_{i \in S_t} |x_i - y_{j,i}|^p \tag{2.11}$$

Then, by virtue of the *triangular inequality* applied on corresponding sub-vectors we establish the following r inequalities:

$$||X - Y_j||^p_{p,S_t} \geq \left| ||X||_{p,S_t} - ||Y_j||_{p,S_t} \right|^p, \ \ t = 1, \ldots r \tag{2.12}$$

and summing up both members of the inequalities attains a *lower bound* of the function measuring the dissimilarity between X and Y_j :

$$||X - Y_j||^p_p \geq \sum_{t=1}^{r} \left| ||X||_{p,S_t} - ||Y_j||_{p,S_t} \right|^p \tag{2.13}$$

This inequality provides a sufficient condition that allows for pruning those candidates which cannot represent a matching position. In fact, if the lower-bound of the dissimilarity function exceeds the threshold D that discriminates between matching and non-matching candidates:

$$\sum_{t=1}^{r} \left| ||X||_{p,S_t} - ||Y_j||_{p,S_t} \right|^p > D \tag{2.14}$$

then from (2.13) and (2.14) Y_j cannot be a matching pattern.

If (2.14) does not hold, rather than computing from scratch the term $||X - Y_j||^p_p$, we can obtain another pruning condition based on a tighter lower-bound by considering a sub-vectors pair and replacing in the left-hand term of (2.14) the difference between the *partial* norms with the corresponding *partial* L_p-dissimilarity:

$$||X - Y_j||^p_{p,S_i} + \sum_{t=1, t \neq i}^{r} \left| ||X||_{p,S_t} - ||Y_j||_{p,S_t} \right|^p > D \tag{2.15}$$

Since the following relation holds as a consequence of the triangular inequality

$$||X - Y_j||_p^p \geq ||X - Y_j||_{p,S_i}^p + \sum_{t=1,t\neq i}^{r} \left| ||X||_{p,S_t} - ||Y_j||_{p,S_t} \right|^p \geq$$
$$\sum_{t=1}^{r} \left| ||X||_{p,S_t} - ||Y_j||_{p,S_t} \right|^p \tag{2.16}$$

the lower bound appearing at the left-hand side of (2.15) is tighter compared to that of (2.14) and hence the associated pruning condition is potentially more effective in skipping non-matching candidates.

Should condition (2.15) fail, the tightness of the lower-bounding function can be further increased by taking another sub-vectors pair and, again, replacing the difference between the *partial* norms with the corresponding *partial* L_p-dissimilarity. This process can be iteratively applied to all the r sub-vectors pairs resulting from P, so as to determine up to r sufficient conditions that can be sequentially checked when matching each candidate vector Y_j. These r conditions are based on the following succession of increasingly tighter lower-bounds:

$$\begin{aligned}
&\sum_{t=1}^{r} \left| ||X||_{p,S_t} - ||Y_j||_{p,S_t} \right|^p \leq \\
&\leq ||X - Y_j||_{p,S_i}^p + \sum_{t=1,t\neq i}^{r} \left| ||X||_{p,S_t} - ||Y_j||_{p,S_t} \right|^p \leq \\
&\leq ||X - Y_j||_{p,S_i}^p + ||X - Y_j||_{p,S_k}^p + \sum_{t=1,t\neq i,k}^{r} \left| ||X||_{p,S_t} - ||Y_j||_{p,S_t} \right|^p \leq \cdots \\
&\cdots \leq \sum_{t=1,t\neq l}^{r} ||X - Y_j||_{p,S_t}^p + \left| ||X||_{p,S_l} - ||Y_j||_{p,S_l} \right|^p
\end{aligned} \tag{2.17}$$

Hence, throughout the matching process, each vector Y_j undergoes checking a succession of sufficient conditions starting from (2.14), until either it is pruned or the following last condition is reached:

$$\sum_{t=1,t\neq l}^{r} ||X - Y_j||_{p,S_t}^p + \left| ||X||_{p,S_l} - ||Y_j||_{p,S_l} \right|^p > D \tag{2.18}$$

Should the last condition be not verified, the process ends up in computing the dissimilarity $||X - Y_j||_p^p$ by replacing $\left| ||X||_{p,S_l} - ||Y_j||_{p,S_l} \right|^p$ with $||X - Y_j||_{p,S_l}^p$ in the left-hand term of (2.18). Then, Y_j is classified as a valid pattern if:

$$||X - Y_j||_p^p < D \tag{2.19}$$

The key point of the IDA algorithm is that it achieves computational savings since, compared to $||X - Y_j||_{p,S_t}^p$, the term $\left| ||X||_{p,S_t} - ||Y_j||_{p,S_t} \right|^p$ can be computed much more rapidly, and independently from the sub-vectors cardinality, by calculating the partial norms using well-known fast incremental calculation schemes [24, 91, 131]. Consequently, since replacing differences of partial norms with the corresponding partial dissimilarities yields tighter bounding functions, the tighter is the bounding function the higher is its calculation time. Therefore, IDA establishes a succession of increasingly tighter bounding functions, with the next computationally more demanding function calculated only when required, i.e. when a candidate has not been pruned by the previous condition.

In order to efficiently compute the partial norms by means of incremental calculation schemes, the adopted partitioning scheme for X and Y_j must follow certain rules of regularity [119]. In particular, we propose to partition X and Y_j according to a splitting of template and image subwindows into r rectangular regions and calculate the partial norms by the one-pass *box-filtering* method proposed in [91]. In our implementation, a box-filtering function fills-in an array of partial norms by computing the norm of each rectangular region of given dimensions belonging to the reference image. As described in [91], this is done by exploiting a double recursion on the rows and columns of the reference image, which requires only 4 elementary operations per image point independently of the sizes of the rectangular region. Hence, to obtain the required partial norms we need to run as many box-filters as the number of differently sized regions corresponding to sub-vectors. In the particular case of r equally sized regions a single box-filter is needed by IDA. This results in a memory footprint on the order of N, that compares favorably with the PK technique which requires a memory footprint on the order of $N \log M$ [57].

It is worth pointing out that the idea of partitioning the vectors in order to deploy tighter bounding functions has been already proposed in other fields such as motion estimation [43, 77] and vector quantization [103]. Nevertheless, our approach differs from these proposals since it incorporates the idea of successively refining the bounding functions by means of the partial dissimilarity concept and it is not based on a multiresolution scheme.

2.2.3 Hybrid IDA algorithm

The main drawback of techniques such as IDA, PK and SSDA is data dependency, that results in unpredictable response times. In fact, the computational efficiency of these techniques relies on the ability of pruning mismatching candidates by means of the adopted sufficient conditions, which in turn depends on the data. Conversely, with the FS approach response time depends only on the image and pattern sizes and with the FFT approach only on image size. Moreover, as it will be shown in subsection 2.2.4, although IDA turns out generally faster than the FS and FFT approaches, in some cases it happens to be slower than the FFT approach.

We observed that the overall behavior of the IDA algorithm can be predicted with a high degree of reliability by evaluating the pruning efficiency of its sufficient conditions on a small subset of points uniformly distributed over the image. This task requires a fixed and small computation time and it is particularly meaningful when images have high spatial similarity within large neighborhoods, as occurs in most cases. Hence, in those pattern matching instances where IDA is predicted to be not particularly effective, the matching process may be carried out using the faster between the FS and FFT approach, this choice made upon image and template sizes. Such an approach requires a small overhead with respect to the basic IDA algorithm and, as generally the prediction turns out to be correct, it guarantees in most cases a deterministic upper bound on response time.

Based on these considerations, we have devised the following variation to the basic IDA algorithm, referred to as *hybrid IDA* (hIDA). Given a fixed and small subset of points uniformly distributed over the image, hIDA evaluates the percentage of points within this subset where the first sufficient condition (i.e. (2.14)) succeeds in pruning the corresponding candidate. In case this percentage is higher than a certain threshold (typically between 50%, for small images, and 85%, for bigger images) hIDA carries out the matching process using the IDA algorithm, conversely it switches to the fastest between the FS and FFT algorithms. As it will shown in subsection 2.2.4, thanks to the computational efficiency and reliability of the prediction step, in most cases hIDA guarantees that in problem instances favorable to the IDA approach the performance is substantially equivalent to that of the basic IDA algorithm, while in those few cases less favorable to IDA the performance is substantially equivalent to that of the faster between FS and FFT.

2.2.4 Experimental results

This subsection is aimed at assessing the performance of IDA and hIDA by comparing them with the FS algorithm as well as with the fast exhaustive algorithms presented in Sec. 2.2.1, i.e. the PK, FFT-based and SSDA algorithms. The IDA, hIDA, FS and SSDA algorithms were implemented in *C*. As for PK, we compiled and ran the original authors' *C* code (available at their web site [104]) which refers to the case of the SSD function. With regards to the FFT-based algorithm, we used the very efficient implementation (*cvMatchTemplate* function) provided by *OpenCV* library [26]. Hence, we compared IDA and hIDA to the FS, PK and FFT algorithms in the case of the SSD function ($p = 2$), and then IDA to the FS and SSDA algorithms in the case of the SAD function ($p = 1$)[1]. The benchmarking platform was an *AMD Athlon* processor with 3

[1] *hIDA has not been considered in the case* $p = 1$ *since it deploys the FFT.*

GB RAM running *Windows XP*.

Two different kinds of experiments were carried out. In *Experiment 1* we individually compare all the speed-up values yielded by the considered algorithms on an indoor sequence of 3 images acquired by means of a digital camera. This dataset is affected by real distortions since each image was taken at a slightly different pose with respect to that where the templates were extracted from. Instead, *Experiment 2* aims at evaluating the global performance of the algorithms on a large dataset of 120 images, with artificial noise at 5 different levels added on each image. In this latter experiment, results are shown by means of statistical indicators.

In order to evaluate the performance of the algorithms with different image dimensions, for both experiments 4 different scales $S1, \cdots, S4$ of patterns and images have been used:

- S1) Images: 160×120; Templates: 16×16 *(M=256)*
- S2) Images: 320×240; Templates: 32×32 *(M=1024)*
- S3) Images: 640×480; Templates: 64×64 *(M=4096)*
- S4) Images: 1280×960; Templates: 128×128 *(M=16384)*

This choice is suitable to both PK and FFT, since PK requires power of 2 dimensions for the template size and the FFT optimally fits into power of 2 image sizes. Since the proposed approach can be applied to any L_p-norm based dissimilarity measure, although of limited practical relevance, at the end of this section we also compare IDA to the FS in the case $p = 3$ for $S1$.

Parameters of the algorithms For what concerns Experiment 1, for each pattern matching instance the threshold D was set to two different values, referred to as *th* and *th*2. The value *th* is chosen to be very close to the global minimum, i.e. to the value $||X - Y_W||_p^p$ where Y_W is the best matching image subwindow.

In the case $p = 2$, since the authors' code of the PK algorithm requires parameter MMD (*Maximum Mean Difference*)

$$MMD = \sqrt{\frac{SSD_{min}}{M}}, \tag{2.20}$$

so that the threshold D is computed as

$$D = MMD^2 \cdot M, \tag{2.21}$$

we set *th* as

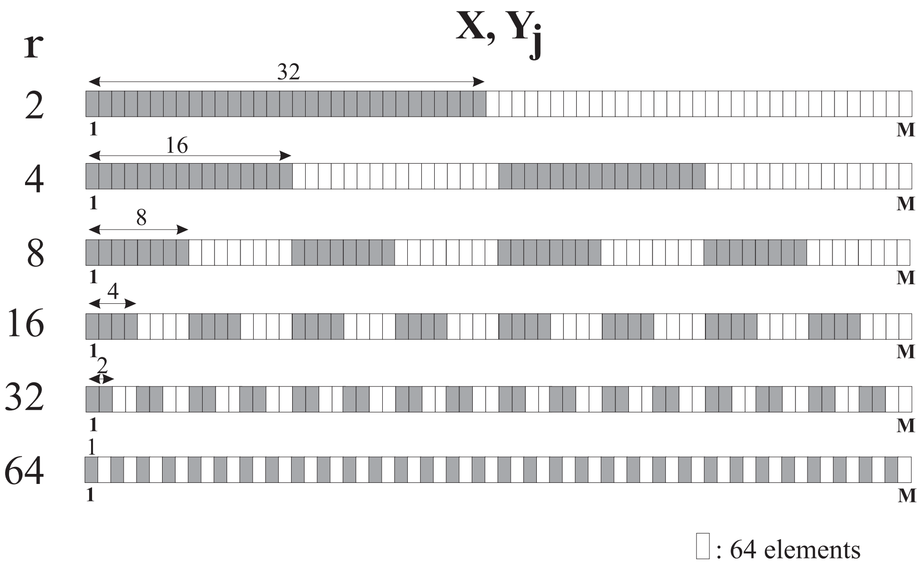


Figure 2.1: The adopted partitioning schemes of vectors X and Y_j as a function of parameter r in the case $M = 64 \times 64$.

$$th = \left\lceil \sqrt{\frac{\|X - Y_W\|_2^2}{M}} \right\rceil \tag{2.22}$$

The second value, $th2$, was chosen to be less selective than th, i.e. 10% higher:

$$th2 = th \cdot 1.10 \tag{2.23}$$

In the case $p = 1$, the threshold values th and $th2$ were set as follows:

$$th = \|X - Y_W\|_1^1 + 1 \tag{2.24}$$

$$th2 = th \cdot 1.05, \tag{2.25}$$

with $th2$ tighter than in (2.23) in order to compensate for the reduced dynamic of the dissimilarity function. Instead, in Experiment 2 only the case $D = th$ was considered.

The parameters of the algorithms were kept constant throughout all experiments. In particular, for what means the PK algorithm, the number of Walsh-Hadamard kernels was set to the default value suggested by the authors in their code. As for IDA and hIDA, we partitioned templates and image subwindows into r equally sized sub-vectors of adjacent elements, so that, as pointed out in Section 2.2.2, only a single incremental calculation process is required by the algorithms. With such a choice r is the only parameter of the IDA algorithm. In order to further limit the degrees of freedom of the

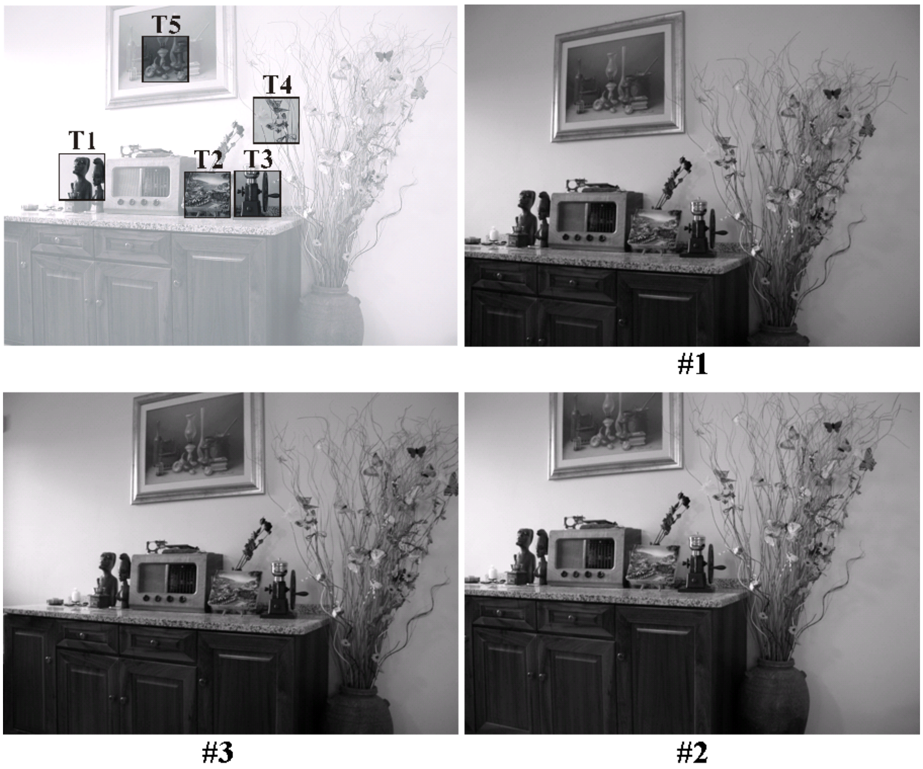


Figure 2.2: Experiment 1: the 5 templates (top, left) and the 3 test images.

adopted partitioning scheme, we constrained r to be a power of 2 ranging from 2 up to the template side (i.e. $16, 32, 64, 128$ in both experiments), as described graphically in Fig. 2.1 in the case of a 64×64 pixels template. In both experiments the results yielded by the IDA algorithm with the choice of parameter r yielding the best performance are referred to as *IDA opt*. We show also the results yielded by IDA and hIDA using some given r values which can be regarded as generally good default choices for the considered template sizes. In particular, parameter r was set to $\{4, 4, 8, 8\}$ for template side equal respectively to $\{16, 32, 64, 128\}$, as in most cases the IDA approach is more efficient if a higher r is used with bigger templates. For what concerns the hIDA algorithm, it requires the setting of an additional parameter, i.e. the threshold on the percentage of candidates pruned within the prediction step that determines whether the search process is carried out using IDA or the fastest between the FS and the FFT. This parameter was set to $50\%, 50\%, 70\%, 85\%$ for template side equal respectively to $\{16, 32, 64, 128\}$. The prediction step analyses a subset of points obtained by selecting one point out of 20 along both the directions within the search area.

Experiment 1 In this experiment we first extracted 5 templates from an image, then we took 3 other shots of the same scene from slightly different positions (templates and images are shown in Fig. 2.2). All templates and images were scaled according to scales $S1$, $S2$, $S3$, $S4$. Hence, for each of the 4 scales we obtained 5 templates and 3 images, resulting overall in 60 pattern matching instances. Hereinafter, each instance will be denoted by the pair *test image number- template number* (e.g. the pair $1-2$ denotes template 2 matched into test image 1). Experimental results are given out as ratios of execution times (i.e. speed-ups) measured on the benchmarking platform.

Fig. 2.3 and Fig. 2.4 report the speed-ups yielded by IDA, hIDA, PK and FFT with respect to the FS SSD-based algorithm setting respectively $D = th$ and $D = th2$. For each pattern matching instance of each scale the first two bars concern IDA: the leftmost regards the value of parameter r providing the highest speed-up (i.e. *IDA opt*), the other, tagged as *IDA r,* $r \in \{4, 8\}$, the default value of r. Then, the third bar, tagged as *hIDA r,* $r \in \{4, 8\}$, regards hIDA, with r set to the same default value as IDA. Finally, the last 2 bars show the speed-up yielded respectively by the PK and FFT-based algorithm.

As far as Fig. 2.3 is concerned, the IDA algorithm, using the optimal r as well as the default r, turns out to be very effective in most instances of $S1$, $S2$ and $S3$. As a matter of fact, with these scales IDA *opt* and IDA r are both always much faster than the FFT-based algorithm. Furthermore, IDA *opt* does not outperform PK in only 5 instances out of 45 (i.e. $1-1$, $2-1$ at $S1$ and $1-1$, $3-1$, $3-4$ at $S2$) while IDA r in only 6 instances out of 45 (the previous 5 plus $3-5$ at $S1$).

For what concerns $S4$, though the computational efficiency of the FFT algorithm is very high due to the image and template sizes (speed-up=20.5), IDA algorithms run notably faster in 9 instances out of 15 (reaching a maximum speed-up as high as 184.7 in instance $2-1$). As for hIDA, it is almost as fast as IDA in the former 9 instances and provides substantially the same speed-up as the FFT-based algorithm in the remaining 6. Hence, at this scale the effectiveness of the prediction step is clearly shown, since hIDA allows for deploying the template matching algorithm more suited to the data by correctly selecting the faster between IDA and the FFT. This is also demonstrated at $S1$, $S2$ and $S3$, where IDA clearly outperforms the FFT and hIDA provides substantially the same computational savings as IDA. It is also interesting to note that at $S4$ the average speed-up yielded by hIDA is 77.8, with a lowest speed-up equal to 20.0, which is very similar to the constant speed-up yielded by the FFT. As a result of these considerations, it turns out that IDA is particularly suited to small size images, while hIDA provides the best overall performance.

Moreover, for what concerns a comparison between IDA *opt* and IDA r, it can be noticed that the choice of a default r in most instances does not affect notably the

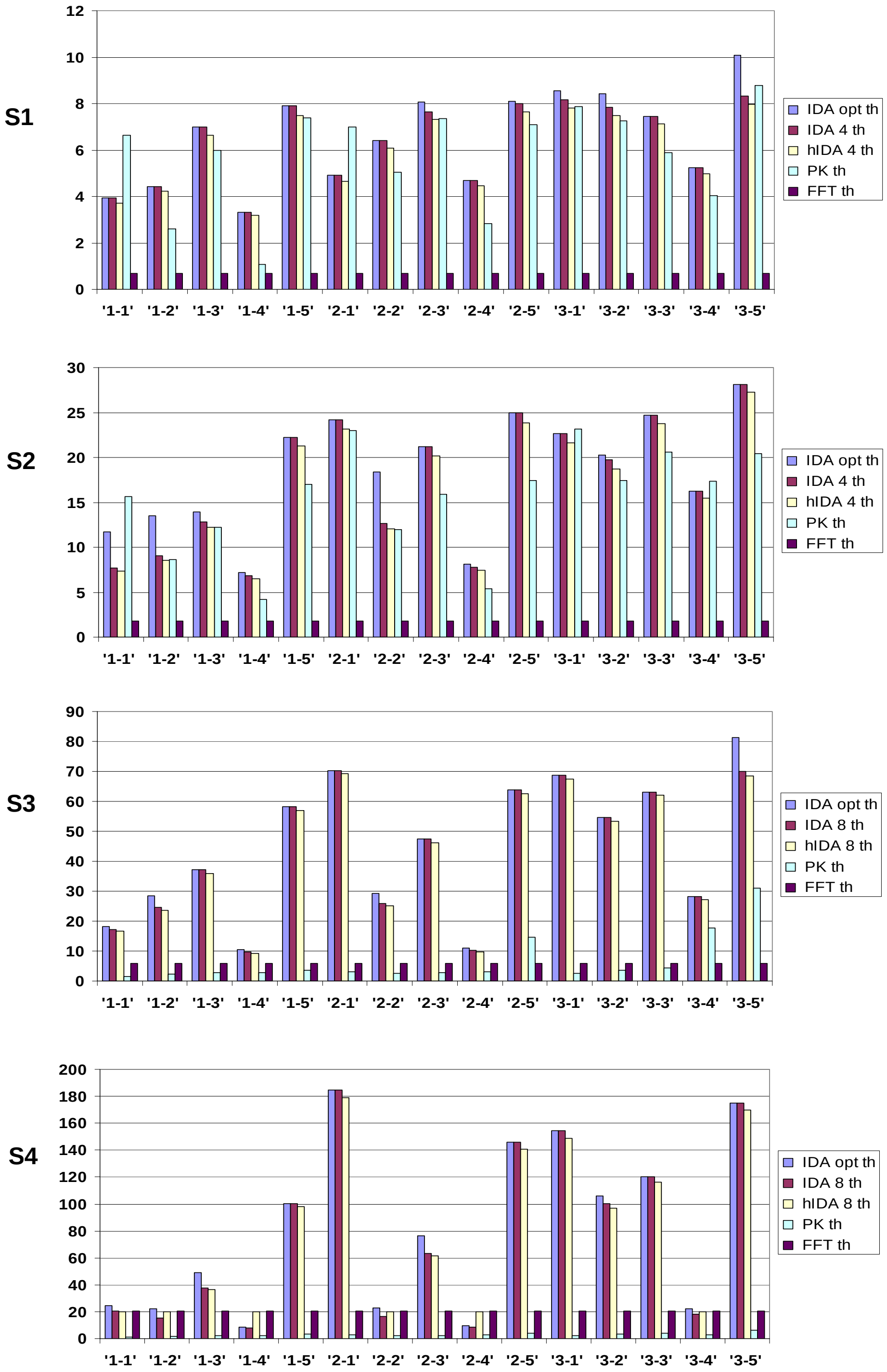


Figure 2.3: Experiment 1: measured speed-ups in the SSD case, $D = th$.

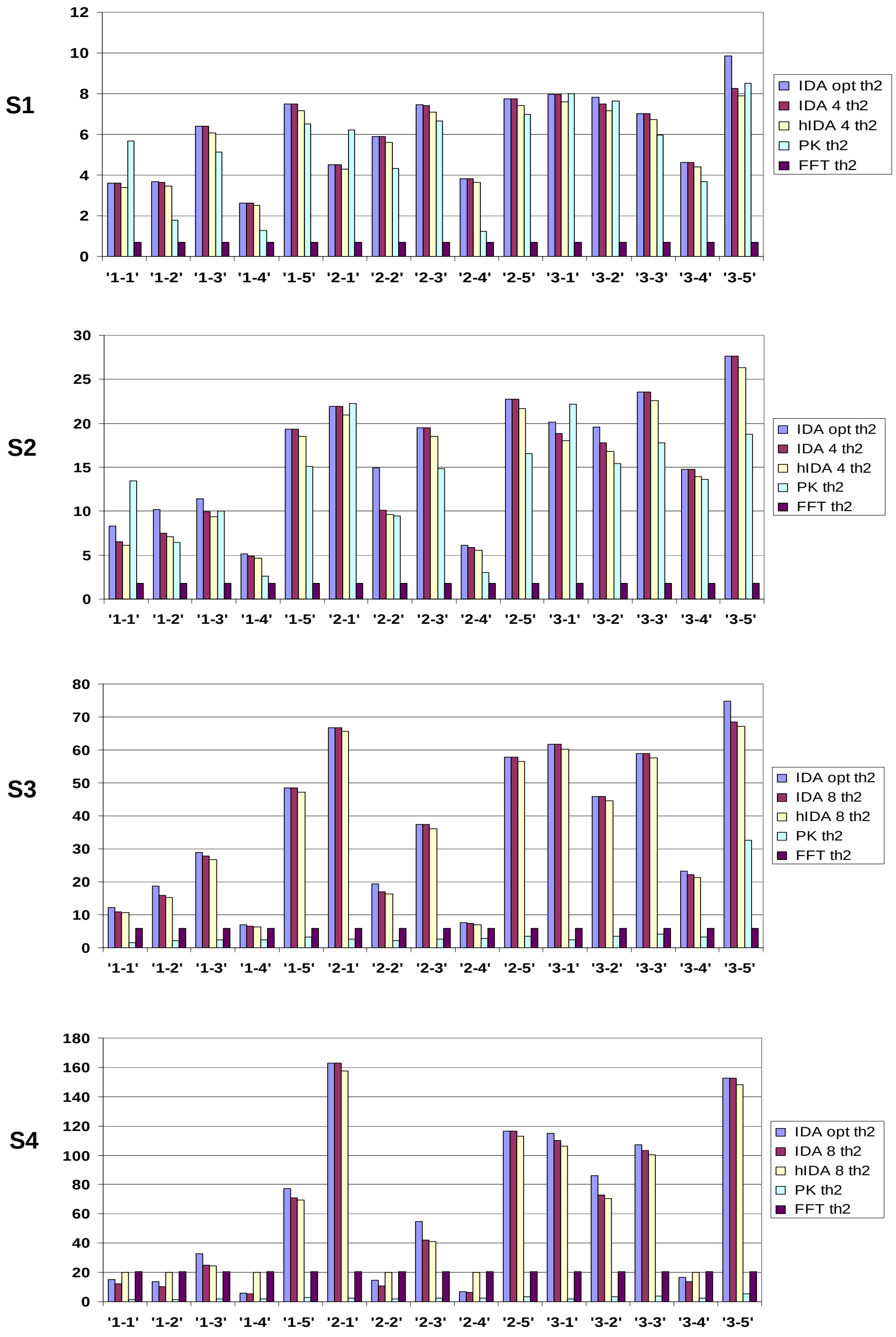


Figure 2.4: Experiment 1: measured speed-ups in the SSD case, $D = th2$.

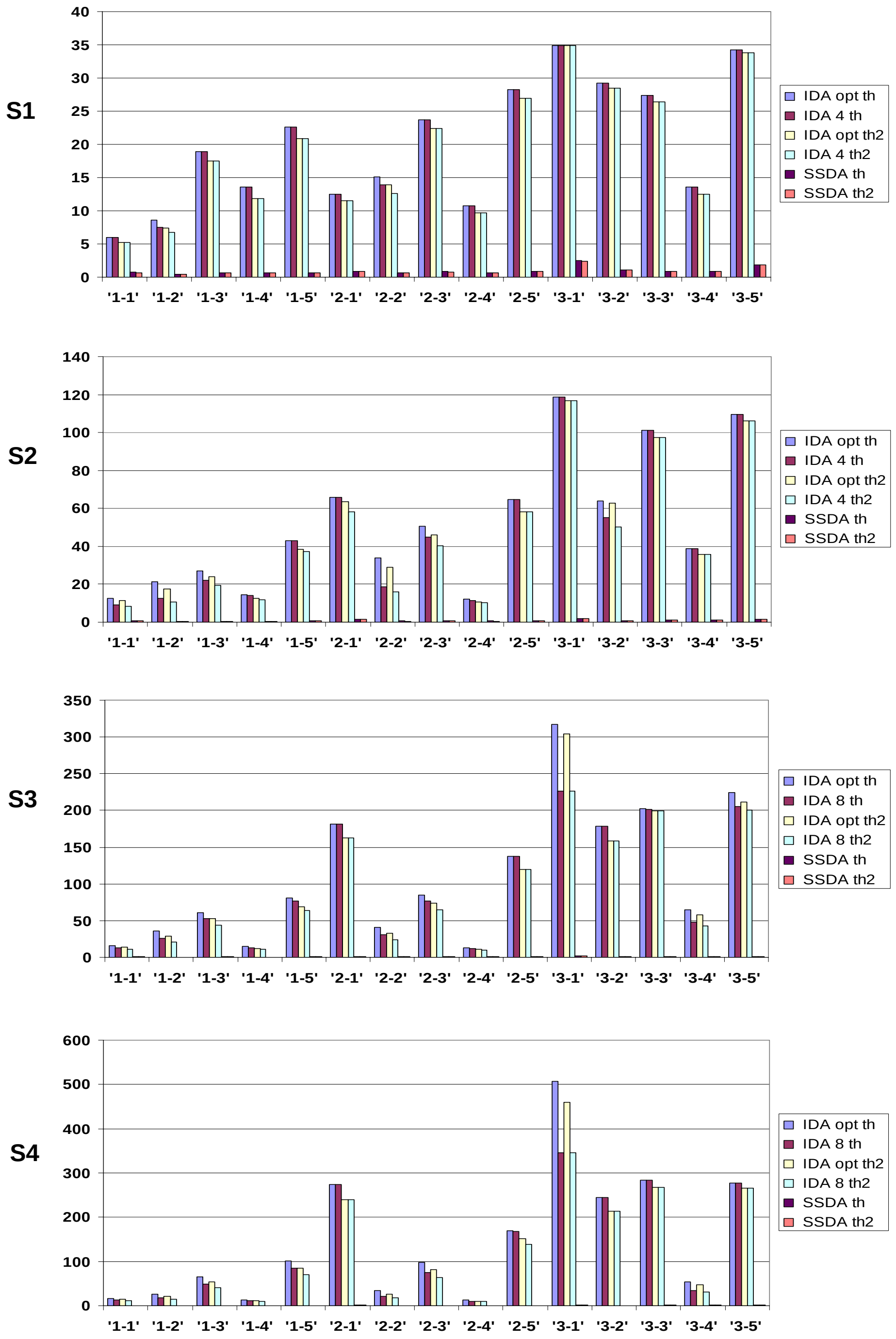


Figure 2.5: Experiment 1: measured speed-ups in the SAD case.

performance compared to the optimal choice, the speed-ups yielded by IDA r being generally very close to those of IDA *opt*.

As regards the PK algorithm, at $S1$ and $S2$ it turns out slower than IDA and hIDA in most instances, but always notably faster than the FFT algorithm. At $S3$ and $S4$ PK is significantly outperformed by FFT in most instances, and turns out always slower than IDA and hIDA.

The results reported in Fig. 2.4 substantially confirm the outcomes of the previous comparative analysis. Focusing our attention on $S4$, the hIDA algorithm also with threshold value $th2$ is able to yield in the best cases speed-ups comparable to IDA and in the worst cases speed-ups similar to the FFT. Furthermore also in Fig. 2.4 it can be noticed that at $S3$ and $S4$ PK is always slower that IDA (30 out of 30) and in most instances slower that the FFT (29 out of 30).

Finally, Fig. 2.5 shows the speed-ups yielded by IDA and SSDA with respect to the FS SAD-based algorithm (i.e. $p = 1$) with $D = th$ and $D = th2$. For each instance of this experiment the first and third bars refer to IDA with the optimal value of parameter r (respectively for $D = th$ and $D = th2$), the second and fourth bars to IDA with the default choice of r (respectively for $D = th$ and $D = th2$). The last two bars refer to SSDA, respectively for $D = th$ and $D = th2$. The Figure shows that IDA is always much faster than the FS algorithm, with speed-ups ranging from about 5 (worst case) up to more than 500 (best case). It is worth pointing out the ranges of the measured speed-ups with the less favorable parameter settings (i.e. default r and less selective threshold $D = th2$, fourth bar of each instance): from 5.2 to 34.9 at $S1$, from 8.4 to 116.6 at $S2$, from 10.4 to 226.7 at $S3$ and from 9.0 to 346.4 at $S4$. For what means SSDA, the reported speed-ups are always dramatically lower than those yielded by IDA algorithms, with the algorithm being sometimes even slower than the FS. This has to be ascribed to the significant number of test operations (as high as M) performed by SSDA, which slow down the method particularly at large scales. Furthermore, this is also due to the fact that, as explained in Section 2.2.1, in order to guarantee the exhaustiveness of the search the pruning threshold for SSDA must be set to a constant value higher than the global minimum (i.e. th or $th2$), while this algorithm was originally conceived to perform best with a varying D much lower than the global minimum (i.e. in a non-exhaustive scenario).

Experiment 2 Experiment 2 was aimed at assessing the performance of the examined algorithms on a larger dataset. This experiment includes a total of 120 images chosen between 3 databases: MIT [29], medical [30] and remote sensing [31]. The MIT database concerns mainly indoor, urban and natural environments, plus some object categories such as cars and fruits. The two other databases are composed respectively

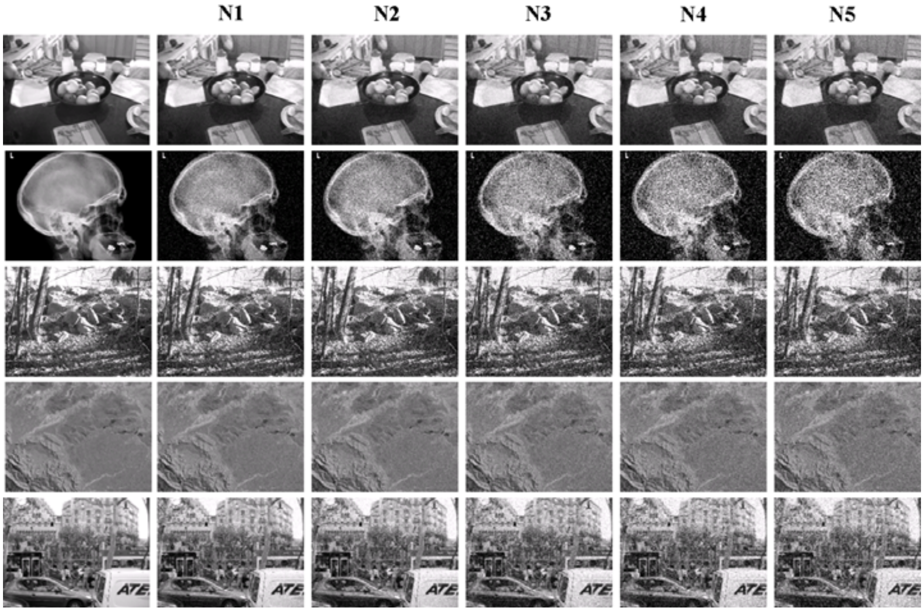


Figure 2.6: 5 images of the dataset used in Experiment 2. Each row shows the noise-free image (leftmost) where templates were extracted from, together with the 5 corresponding noisy images.

of medical (radiographs) and remote sensing (Landsat satellite) images. All images have been subdivided into 4 groups of 30 images, each group being characterized by a different scale and with scales being the same as in Experiment 1 (i.e. $S1, \cdots, S4$). For each image 10 templates were randomly selected among those showing a standard deviation of pixel intensities higher than a threshold (i.e. 45). Then, 5 different levels of i.i.d. zero-mean Gaussian noise, referred to as $N1, \cdots, N5$, were added to each image. The 5 noise levels range from very low noise to very high noise, the variances of the Gaussian distribution being respectively 1.3, 2.6, 5.1, 7.7, 10.2 [2]. Hence, overall each algorithm was tested against 6000 pattern matching instances. Figure 2.6 shows 5 images of the dataset. For each of them, the 5 corresponding images with increasing (from left to right) noise levels are also shown.

Due to the large size of the dataset, for each scale and noise level we provide a global indication (in terms of mean μ and standard deviation σ) of the measured speed-ups with respect to the FS algorithm on the same benchmark platform as in Experiment 1. Moreover, in order to better assess the behavior of the algorithms, we show two additional descriptors that allow for measuring the asymmetry of the distribution. These descriptors, referred to as σ_- and σ_+, represent the square root of the mean square error

[2] Corresponding to 0.005, 0.01, 0.02, 0.03, 0.04 on normalized pixel intensities ranging within [0, 1].

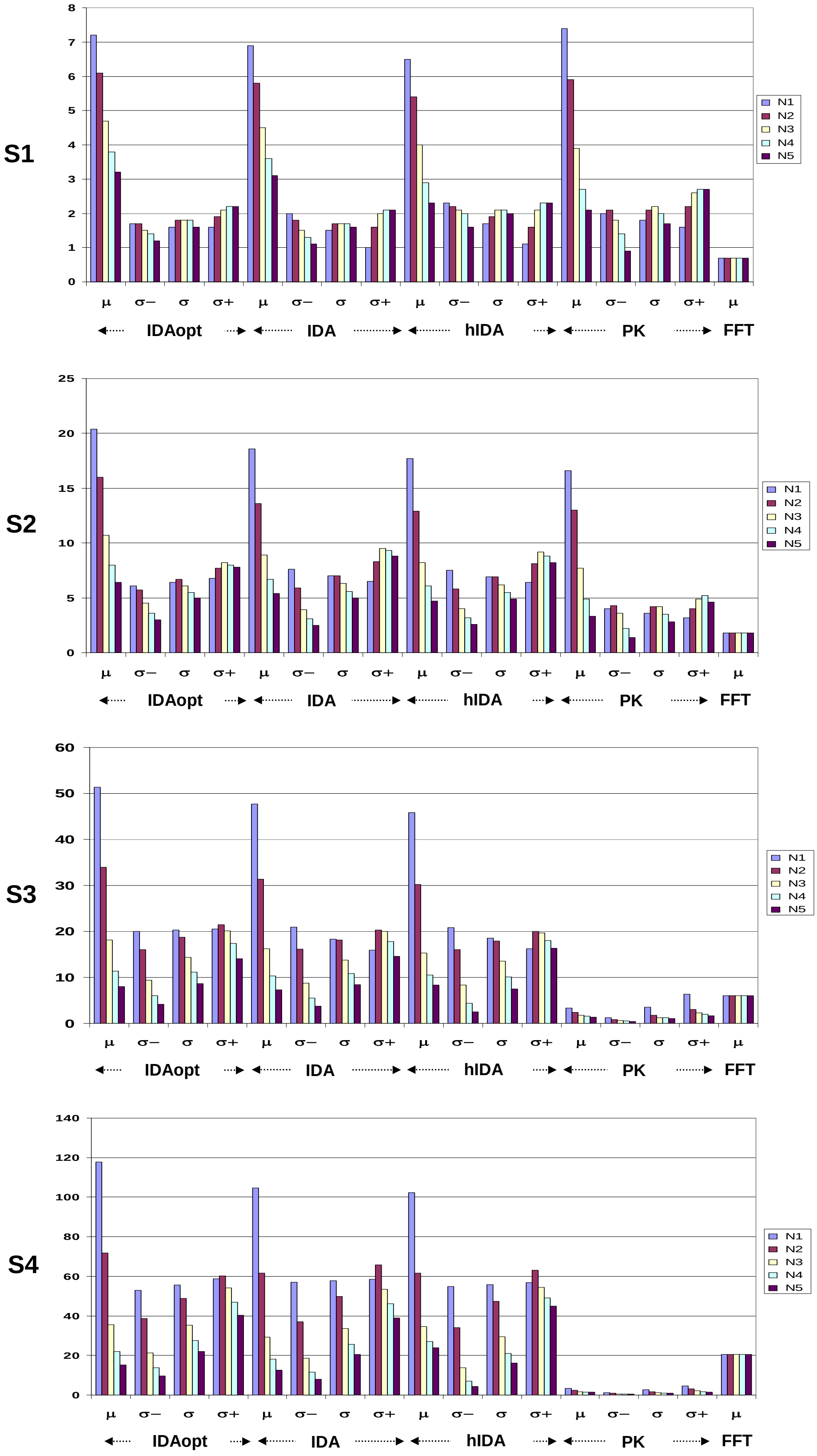


Figure 2.7: Experiment 2: speed-ups yielded by the exhaustive techniques vs. FS algorithm at the 4 scales, SSD case ($p = 2$).

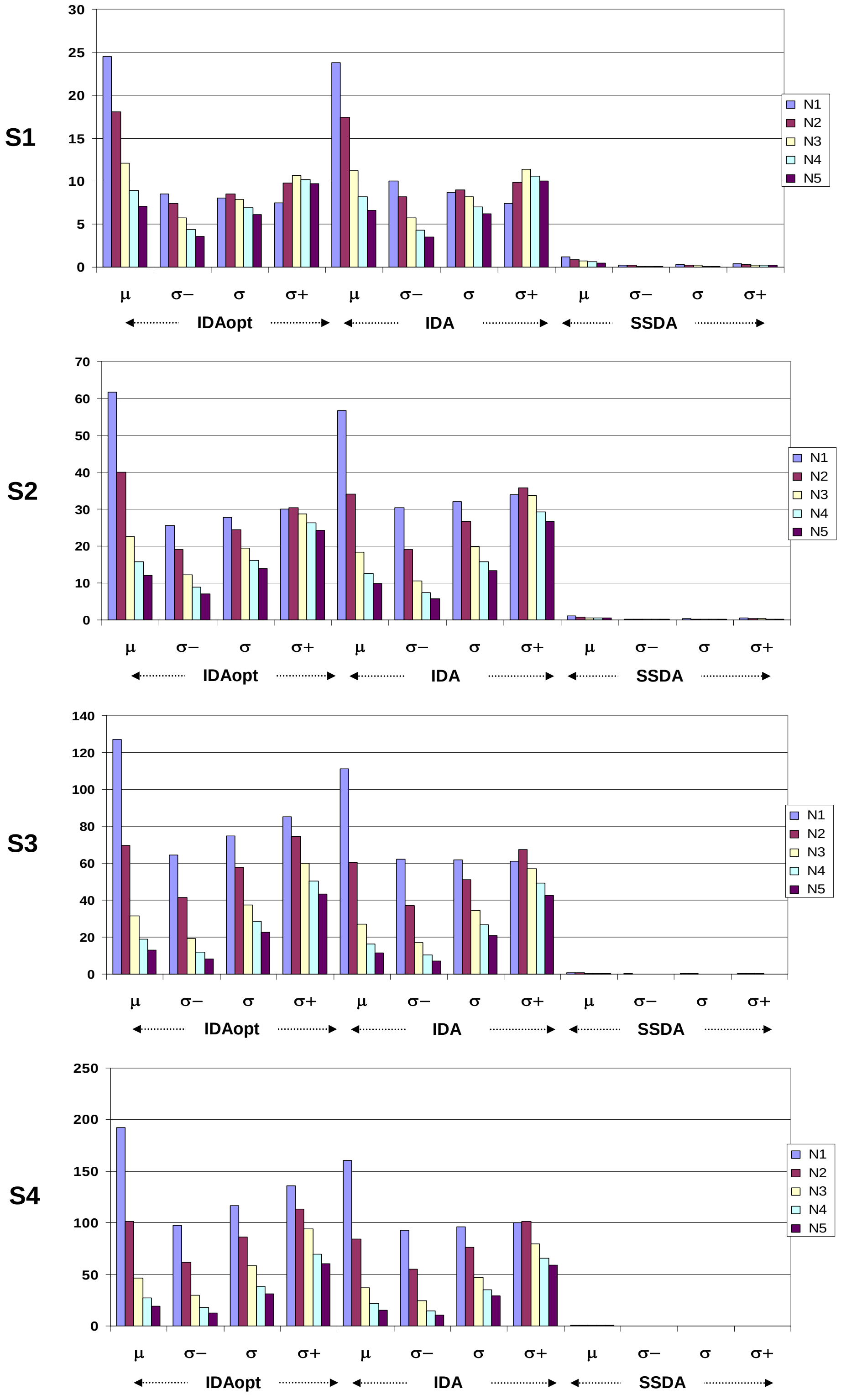


Figure 2.8: Experiment 2: speed-ups yielded by the exhaustive techniques vs. FS algorithm at the 4 scales, SAD case ($p = 1$).

with respect to μ of the population - respectively - below and above μ.

Fig. 2.7 reports for $p = 2$ and $D = th$ the performance of IDA *opt*, IDA, hIDA, PK and FFT. The figure shows that at $S1$ and $S2$ IDA, hIDA and PK yield, substantially, comparable speed-ups. The algorithms are always notably faster than the FFT although their efficiency decreases significantly with increasing noise. Nevertheless, on average, IDA *opt*, IDA and hIDA turn out to be more robust to noise than PK. Furthermore, it is worth pointing out that at $S2$ IDA *opt* and IDA always provide mean speed-ups higher than PK. Moreover, in both scales IDA *opt* and IDA are always slightly more efficient than hIDA. For what concerns $S3$, IDA *opt*, IDA, hIDA always yield much higher speed-ups than PK, which in turn is clearly outperformed also by the FFT. Conversely, our algorithms always perform better than the FFT. However, though all data dependent algorithms are significantly affected by noise, our algorithms, according to the larger dynamic of the mean speed-up, show a more substantial decrease of the computational efficiency with increasing noise. The comparison between our algorithms indicates that hIDA tend to perform slightly better at higher noise levels. As for $S4$, IDA *opt* and IDA always dramatically outperform PK and, at noise levels $N1$,$N2$,$N3$, they also result much faster than FFT. However, at higher noise levels (e.g. $N5$ for IDA *opt*, $N4$ and $N5$ for IDA) the FFT turns out more effective. Nevertheless, it is worth observing that hIDA always outperforms PK and the FFT, resulting the best choice for large images.

Overall, the results of Experiment 2 confirm the trend inferable from Experiment 1: at $S1$ IDA and PK perform best, at $S2$ IDA is the best choice, at $S3$ IDA and hIDA are comparable and yield the best results, at $S4$ hIDA is the best performing algorithm.

The standard deviation, σ, reported in Fig. 2.7 confirms on this larger dataset the notable data dependency of IDA *opt*, IDA, hIDA and PK highlighted in experiment 1. Although for these algorithms σ is significantly high at high noise levels, it is worth observing that in all such cases the distribution of the speed-up is clearly asymmetric, with the right tail more pronounced (that is, σ_+ is sensibly greater than σ_-). Hence, the values which differ most from the mean occur for speed-ups above μ, while speed-ups lower than the mean show less dispersion with regards to μ.

For what concerns $p = 1$, Fig. 2.8 reports the speed-ups yielded by IDA *opt*, IDA and SSDA with respect to the FS SAD-based algorithm with $D = th$. Similarly to Experiment 1, at each scale IDA *opt* and IDA dramatically outperform SSDA and FS, yielding always substantial speed-ups, thus confirming the efficiency of the proposed approach on this larger dataset. Nevertheless, it is worth observing that the speed-ups are significantly affected by noise. The figure also confirms the significant data dependency of IDA *opt*, IDA and SSDA. Similarly to $p = 2$, the distributions of the speed-up values are clearly asymmetric and right-tailed, this behavior getting more pronounced as the noise level increases.

Table 2.1: Speed-ups yielded by IDA vs. FS in the case $p = 3$, measured on the images of Experiment 2 at $S1$.

S	N	IDA opt		IDA 4	
		μ	σ	μ	σ
S1	N1	5.2	0.8	5.0	0.8
	N2	4.6	0.9	4.4	0.9
	N3	3.9	0.9	3.7	1.0
	N4	3.4	0.9	3.2	0.9
	N5	3.0	0.9	2.9	0.9

Experiment with $p > 2$ We report here the results of an experiment addressing the case $p = 3$. In particular, Table 2.1 shows the mean and standard deviation of the speed-ups yielded by IDA *opt* and IDA 4 with regards to the FS algorithm on the dataset used in experiment 2 at $S1$. As it can be noted, also in this case both the considered algorithms run significantly faster than the FS approach.

2.3 Fast exhaustive block matching

Block matching is a common approach adopted in computer vision, in particular to carry out motion estimation for video compression, whose aim is to reduce temporal redundancy in video sequences. Let $\{I_r, I_t\}$ be two consecutive frames of a video sequence, and let I_r be subdivided into non-overlapping *blocks*, i.e. subwindows of size $N \times N$. Block matching aims at finding for each block of frame I_r the most similar block in frame I_t. Usually each block is not sought on the whole frame, but on a *search area* centered at the position of the block in frame I_r.

Similarly to the FS approach for pattern matching, also the FS approach for block matching relies on comparing each block of I_r with all possible *candidate blocks* belonging to the corresponding search area in I_t by computing a distance between blocks. The most commonly used distance for this scope is the SAD. Once the distances between the reference block and the candidate blocks have been computed, the best matching block is selected as the one corresponding to the minimum distance value found within the search area. Since this approach turns out to be computationally expensive, many techniques have been proposed in the last two decades with the aim of accelerating the FS (see [63] for a survey), that, as it is the case of pattern matching, can be either *exhaustive* or *non-exhaustive*.Non-exhaustive techniques [82, 96] usually reduce the search area in order to save computations, hence they don't guarantee the requirement of finding, for each block, the candidate block at the globally minimum distance within the search area. As a result, they tend to increase the distortion of the compressed video signal. Conversely, exhaustive techniques aim at accelerating the FS by selecting rapidly and safely many non-matching candidate blocks, so as to discard them without the need of computing the distance function. Analogously to the pattern matching case, non-matching candidate blocks are usually selected by means of lower bounds of the distance function [43, 77, 83].

This section proposes experimental results concerning the extension to block matching of the IDA technique proposed for pattern matching. In particular, as for the new approach, the same succession of increasingly tighter lower bounding functions of the SAD measure is deployed in order to rapidly detect mismatching candidates. The partitioning parameter, r, turns out to be a parameter of the proposed technique. Increasing r would mean having lower bounding functions which better approximate the distance term, but would also need more computations for their calculations. As for block matching the size of the blocks is typically $N = \{8, 16\}$, we experimentally found out that in most cases the best results are obtained by choosing r equal to $\{4, 8\}$.

It is important to point out that the performances of this method are affected by the initial value of the minimum *found so far*, D_m. In fact, if D_m is initialized with a value close to the global minimum of the distance function to be found in the search area, the sufficient conditions embodied in the proposed approach have good chances to discard a high number of non-optimal candidate blocks. Conversely, if D_m is initialized e.g. to the maximum value which the distance can assume, no blocks can be discarded within the initial positions until a local minimum is found. For this reason, a simple but effective improvement can be obtained by initializing D_m to $D_{m,0}$, that is the distance value corresponding to the candidate block at offset $(u, v) = (0, 0)$. We also propose an alternative approach, that is to initialize D_m to the global minimum corresponding to the best matching offset position found at the previous frame for the same block, $D_{m,mp}$. In order to deploy this, the technique has to keep trace of the motion flow of the previous frame. This approach can be seen as a very basic motion predictor, as it assumes as most probable motion offset that found in the previous frame.

It is important to note that, although other optimal techniques deploying lower bounding functions obtained by summations over partitioned blocks have been proposed [18, 43, 77], only our technique exploits the concept of partial distance in order to further refine the bounding functions.

2.3.1 Experimental results

In this subsection some experimental results are presented which compare the proposed block matching technique with the FS. The distance used is the SAD (i.e. $p = 1$), the block size is 16×16 (i.e. $N = 16$) and the search is performed on both directions on an

Table 2.2: Speed-ups of the proposed algorithm vs. FS, $r = 4$

Sequence	$D_{m,0}$	$D_{m,mp}$
Claire	12.9	13.0
Miss America	2.1	2.0
Salesman	12.6	13.4
Flower garden	2.9	7.3
Table tennis	2.4	3.0
Grandmother	5.3	4.9
Mr. Chest	12.4	11.9
Trevor	5.9	5.8
Surfside	3.9	3.9
Football	5.2	5.3
Average	6.6	7.1

Table 2.3: Speed-ups (ratios of operations) of the proposed algorithm vs. FS, $r = 8$

Sequence	$D_{m,0}$	$D_{m,mp}$
Claire	12.3	12.4
Miss America	2.0	2.0
Salesman	11.5	12.5
Flower garden	3.9	7.4
Table tennis	2.4	3.2
Grandmother	4.8	4.6
Mr. Chest	11.9	11.7
Trevor	5.5	5.6
Surfside	3.6	3.7
Football	5.0	5.1
Average	6.3	6.8

offset equal to $[-16, +16]$, hence $M = 32$. For what means the proposed technique, we show the results obtained by choosing $r = \{4, 8\}$, which, as said before, turns out to be the best choice in most cases. The testing sequences used for the comparison are typical video sequences used for benchmarking motion estimation algorithms All algorithms have been implemented in C on a Linux workstation with a 1.5 GHz Pentium M CPU.

Tables 2.2 and 2.3 show the speed-ups in terms of ratios of measured execution time of the proposed algorithm versus the FS, the former referring to $r = 4$, the latter to $r = 8$. In both tables, the second column ($D_{m,0}$) refers to the initialization of D_m as

Table 2.4: Speed-ups (ratios of operations) of the proposed algorithm vs. FS, $r = 4$

		$D_{m,0}$			$D_{m,mp}$	
Sequence	If	$+/-$	Abs	If	$+/-$	Abs
Claire	0.8	21.5	23.9	0.8	21.4	23.8
Miss America	0.4	2.6	2.7	0.4	2.7	2.8
Salesman	0.8	23.2	26.4	0.8	27.0	31.2
Flower garden	0.5	3.7	3.9	0.6	11.6	12.7
Table tennis	0.4	3.0	3.1	0.4	4.2	4.4
Grandmother	0.6	6.8	7.2	0.5	6.7	7.0
Mr. Chest	0.8	21.7	25.0	0.7	21.7	25.0
Trevor	0.7	8.3	8.8	0.6	8.5	9.1
Surfside	0.5	4.7	4.9	0.5	4.7	5.0
Football	0.6	7.8	8.4	0.5	7.8	8.3

Table 2.5: Speed-ups in terms of reduction of N. elementar Ops, proposed algorithm vs. FS, $r = 8$

		$D_{m,0}$			$D_{m,mp}$	
Sequence	If	$+/-$	Abs	If	$+/-$	Abs
Claire	0.8	25.3	30.1	0.7	25.2	29.9
Miss America	0.3	2.8	3.0	0.3	2.9	3.1
Salesman	0.8	28.8	36.0	0.8	32.1	41.2
Flower garden	0.4	4.3	4.6	0.6	15.2	18.1
Table tennis	0.3	3.6	3.9	0.4	5.6	6.2
Grandmother	0.5	7.2	7.9	0.5	7.1	7.7
Mr. Chest	0.8	29.9	39.2	0.7	29.9	39.2
Trevor	0.6	9.7	10.9	0.5	10.0	11.3
Surfside	0.4	5.1	5.6	0.4	5.2	5.6
Football	0.5	9.5	10.8	0.5	9.6	10.9

the distance corresponding to candidate block at offset $(0, 0)$, while the third column ($D_{m,mp}$) refers to the initialization of D_m by means of the motion predictor as explained in Section 2.3. As it can be inferred from the tables, the proposed technique can speed-up notably the FS along the whole dataset. In the $D_m = D_{m,0}, r = 4$ case the speed-ups range from 2.1 to 12.9, in the $D_m = D_{m,mp}, r = 4$ case they range from 2.0 to 13.4. Similarly, in the $D_m = D_{m,0}, r = 8$ case the speed-ups range from 2.0 to 12.3, in the $D_m = D_{m,mp}, r = 8$ case they range from 2.0 to 12.5. For what means the method for initializing D_m, the motion predictor approach seems to bring more benefits, as it yields to almost the same results as the other approach in all cases except for the *Flower garden* sequence, where the speed-up obtained is more than doubled.

Table 2.4 and 2.5 show the speed-ups in terms of ratios of number of elementary operations of the proposed algorithm versus the FS and, as previously, the former refers to $r = 4$, the latter to $r = 8$. The elementary operations considered refer to the high-level code, and are subdivided into three groups: *"If"* for branch instructions, *"+/-"* for additions and subtractions, *"Abs"* for absolute values. Similarly to Table 2.2, columns $2, 3, 4$ ($D_{m,0}$) refers to the initialization of D_m as the distance corresponding to candidate block at offset $(0, 0)$, while columns $5, 6, 7$ ($D_{m,mp}$) refers to the initialization of D_m by means of the motion predictor. As it can be seen, the proposed technique allows for a significant reduction in terms of operations for what regards additions, subtractions and absolute values. Obviously, the number of branch operations is always increased compared to the FS, due to the high number of tests performed when applying the sufficient conditions for discarding candidate blocks. Nevertheless, it is worth noting that for all the tested video sequences, the percentage of branch instructions never accounts for more than 0.13% of the total number of operations performed by the FS.

2.4 Fast exhaustive template matching based on the NCC

A common alternative formulation of the pattern matching problem deals with locating the most similar instance of a template within a reference image. For the sake of clarity, hereinafter we will refer to this formulation as *template matching*. It is worth pointing out that the IDA technique can be straightforwardly modified to deal with template matching. In such a case, the term D is not a constant but represents the best similarity score *found so far*. In particular, D might be conveniently and rapidly initialized by selecting an initial guess for the best matching candidate through a fast non-exhaustive algorithm [149], [54]. Then, for each candidate Y_j, the algorithm is the same as described previously, the only difference being that if condition (2.19) holds then $||X - Y_j||_p^p$ is assigned to D (i.e. the best score *found so far* is updated).

This Section proposes a novel algorithm, referred to as EBC (*Enhanced Bounded Correlation*), that significantly reduces the number of computations required to carry out template matching based on Normalized Cross Correlation (NCC) and yields exactly the same result as the FS algorithm. Analogously to IDA, the algorithm relies on the concept of bounding the matching function: finding an efficiently computable upper bound of the NCC rapidly prunes those candidates that cannot provide a better NCC score with respect to the current best match. In this framework, we apply a succession of increasingly tighter upper bounding functions based on Cauchy-Schwarz inequality. Moreover, by including an on-line parameter prediction step into EBC we obtain a parameter free algorithm that in most cases affords computational advantages very similar to those attainable by optimal off-line parameter tuning. Experimental results show that the proposed algorithm can significantly accelerate a Full-Search equivalent template matching process and outperforms state-of-the-art methods.

2.4.1 Previous Work

As far as template matching based on the NCC function is concerned, it is well known that a faster exhaustive algorithm can be obtained by computing the correlation in the frequency domain by means of the FFT (e.g. see [81], [65]). Given some conditions on the dimensions of template and image, this approach yields notable computational savings with respect to the FS in the signal domain. Moreover, we have shown that NCC-based template matching can also be accelerated by deploying sufficient conditions to skip mismatching image positions based on properly defined upper bounding functions. This algorithm, known as *Bounded Partial Correlation* (BPC), requires calculation of a given portion of the cross-correlation term and bounds the remaining portion by means of a proper inequality. BPC initial formulation was based on Jensen inequality [35]. In [36] we subsequently proposed an improved BPC formulation that deploys Cauchy-Schwarz inequality.

The main novelty of EBC with respect to BPC algorithms [35], [36] consists in the use of a new and more effective bounding strategy based on the deployment of a succession of increasingly tighter upper bounds. Thanks to the new bounding strategy and unlike BPC algorithms, EBC can skip many unmatching positions without calculating any portion of the cross-correlation term. Moreover, the bounding functions including a cross-correlation term are guaranteed to be tighter than those deployed by BPC algorithms.

2.4.2 Notation

Let T be a template of size $M \times N$, and I the image under examination. NCC-based template matching locates T into I by searching for the maximum of the NCC function. Denoting the current template position as (x, y) in the image and the current image subwindow as $I_c(x, y)$, the NCC function can be written as:

$$\eta(x, y) = \frac{\psi(x, y)}{||I_c(x, y)|| \cdot ||T||} \tag{2.26}$$

where the numerator, $\psi(x, y)$, represents the *cross correlation* between the template and the current image subwindow

$$\psi(x,y) = \sum_{j=1}^{N}\sum_{i=1}^{M} I(x+i, y+j) \cdot T(i,j) \tag{2.27}$$

while the terms at the denominator represent the ℓ_2-norm of the current image subwindow

$$||I_c(x,y)|| = \sqrt{\sum_{j=1}^{N}\sum_{i=1}^{M} I^2(x+i, y+j)} \tag{2.28}$$

and the ℓ_2-norm of the template

$$||T|| = \sqrt{\sum_{j=1}^{N}\sum_{i=1}^{M} T^2(i,j)} \tag{2.29}$$

The value of the NCC function is between -1 and 1; however, when dealing with images, it ranges between 0 and 1 since pixels always have positive values.

Computing $\psi(x,y)$ turns out to be the bottleneck in the evaluation of $\eta(x,y)$. In fact, $||I_c(x,y)||$ can be obtained very efficiently using incremental calculation schemes (i.e. [91], [24]) while $||T||$ can be computed once at initialisation time.

2.4.3 Related Work

BPC techniques ([35], [36]) rely on appropriately chosen upper-bounding functions of the numerator of the NCC. Let us assume that a function $\beta(x,y)$ exists such that $\beta(x,y)$ is an upper-bound of $\psi(x,y)$:

$$\beta(x,y) \geq \psi(x,y) \tag{2.30}$$

then by normalising $\beta(x,y)$ we obtain an upper-bound of the NCC:

$$\frac{\beta(x,y)}{||I_c(x,y)|| \cdot ||T||} \geq \frac{\psi(x,y)}{||I_c(x,y)|| \cdot ||T||} = \eta(x,y) \tag{2.31}$$

Indicating with η_M the maximum correlation *found so far*, if the following inequality holds at image point (x,y):

$$\frac{\beta(x,y)}{||I_c(x,y)|| \cdot ||T||} < \eta_M \tag{2.32}$$

then the matching process can proceed with the next position without calculating $\eta(x,y)$, for the point is guaranteed not to correspond to the new correlation maximum. Hence (2.32) is a sufficient condition for skipping points that cannot improve the current best degree of matching without carrying out the computation of the actual cross correlation score. Conversely, if (2.32) holds then it is necessary to compute $\eta(x,y)$ and check the condition:

$$\eta(x,y) \geq \eta_M \tag{2.33}$$

It is intrinsic to this approach that using bounding functions more closely approximating the cross correlation (i.e. *tighter* bounds) increases the chance of skipping a higher number of image points, thus resulting in a more efficient algorithm. As far as BPC is concerned, $\beta(x,y)$ was obtained initially based on Jensen inequality [35]. Subsequently, a tighter bound was derived in [36] by deploying the Cauchy-Schwarz inequality as follows.

Given two p-dimensional vectors **a** and **b**, the Cauchy-Schwarz inequality can be written as

$$\sum_{k=1}^{p} a_k \cdot b_k \leq \sqrt{\sum_{k=1}^{p} a_k^2} \cdot \sqrt{\sum_{k=1}^{p} b_k^2} \tag{2.34}$$

Applying (2.34) to vectors T and $I_c(x,y)$ yields

$$\beta(x,y) = ||I_c(x,y)|| \cdot ||T|| \geq \psi(x,y) \tag{2.35}$$

Unfortunately, plugging (2.35) into (2.32) does not yield a useful sufficient condition since (2.32) turns out to be always false. However, as described in [36], an effective sufficient condition can be obtained by computing only a given portion of the actual correlation function referred to as *partial correlation* (i.e. the correlation associated with rows $[1 \ldots n], 1 < n < N$), and bounding the residual portion of the correlation function with the term derived from the application of Cauchy-Schwarz inequality:

$$\beta(x,y) = \sum_{j=1}^{n} \sum_{i=1}^{M} I\,(x+i, y+j) \cdot T\,(i,j) +$$

$$\sqrt{\sum_{j=n+1}^{N} \sum_{i=1}^{M} I^2\,(x+i, y+j)} \cdot \sqrt{\sum_{j=n+1}^{N} \sum_{i=1}^{M} T^2\,(i,j)} \tag{2.36}$$

Hereinafter we will describe the EBC approach, which yields higher computational savings than BPC thanks to the use of more effective sufficient conditions that in most cases do not require computation of the partial correlation term at all.

2.4.4 Mathematical Framework

This section establishes the mathematical properties that lead to determination of the EBC algorithm.

Lemma Let $\mathbf{a}, \mathbf{b} \in \mathbb{R}^p$ and $S = \{1, 2, \cdots, p\}$. Hence, $\forall S_1, S_2 \mid$

$$\begin{cases} S_1 \cup S_2 = S \\ S_1 \cap S_2 = \phi \end{cases}$$

then the following inequality holds:

$$\sqrt{\sum_{k\in S_1} a_k^2} \cdot \sqrt{\sum_{k\in S_1} b_k^2} + \sqrt{\sum_{k\in S_2} a_k^2} \cdot \sqrt{\sum_{k\in S_2} b_k^2} \leq \sqrt{\sum_{k\in S} a_k^2} \cdot \sqrt{\sum_{k\in S} b_k^2} \tag{2.37}$$

Proof. See *Appendix B*.

□

Property I *Let* $\mathbf{a}, \mathbf{b} \in \mathbb{R}^p$ *and* $S = \{1, 2, \cdots, p\}$. *Hence,* $\forall r \in \{1, \cdots, p\} \mid$

$$\begin{cases} S_1 \cup S_2 \cdots \cup S_r = S \\ S_i \cap S_j = \phi, \forall i \neq j, i, j \in \{1 \cdots r\} \end{cases}$$

then the following inequalities hold:

$$\sum_{k\in S} a_k \cdot b_k \leq \sum_{t=1}^{r} \left(\sqrt{\sum_{k\in S_t} a_k^2} \cdot \sqrt{\sum_{k\in S_t} b_k^2} \right) \leq \sqrt{\sum_{k\in S} a_k^2} \cdot \sqrt{\sum_{k\in S} b_k^2} \tag{2.38}$$

Proof. The left inequality can be easily derived from the application of the Cauchy-Schwarz inequality to each sub-vector pair defined by subsets $S_t, t \in \{1 \cdots r\}$. The right inequality can be obtained directly from successive applications of the previous lemma.

□

Property I states that an upper-bound of ψ, the cross correlation between vectors $\mathbf{a}, \mathbf{b}$, can be obtained by applying r-times the Cauchy-Schwarz inequality to the sub-vector pairs defined by subsets S_t, and that this bound is tighter than the upper bound attainable by applying the inequality to the original vectors $\mathbf{a}, \mathbf{b}$.

Property II *Let* $\mathbf{a}, \mathbf{b} \in \mathbb{R}^p$ *and* $S = \{1, 2, \cdots, p\}$. *Hence,* $\forall r \in \{1, \cdots, p\} \mid$

$$\begin{cases} S_1 \cup S_2 \cdots \cup S_r = S \\ S_i \cap S_j = \phi, \forall i \neq j, i, j \in \{1 \cdots r\} \end{cases}$$

then the following inequalities hold:

$$\begin{aligned} \sum_{k \in S} a_k \cdot b_k &\leq \sum_{k \in S_i} a_k \cdot b_k + \sum_{t=1, t \neq i}^{r} \left(\sqrt{\sum_{k \in S_t} a_k^2} \cdot \sqrt{\sum_{k \in S_t} b_k^2} \right) \\ &\leq \sum_{t=1}^{r} \left(\sqrt{\sum_{k \in S_t} a_k^2} \cdot \sqrt{\sum_{k \in S_t} b_k^2} \right) \end{aligned} \tag{2.39}$$

Proof. Similar to Property I, the left inequality derives from the application of the Cauchy-Schwarz inequality to each sub-vector pair defined by subsets $S_t, t \in \{1 \cdots r\} - \{i\}$. The right inequality is easily proved by applying the Cauchy-Schwarz inequality to the sub-vector pair defined by subset S_i.

□

Property II tells us that given an upper bound of ψ obtained by partitioning $\mathbf{a}, \mathbf{b}$ into r sub-vectors as defined in Property I, a tighter upper bound can be obtained by replacing the product-of-norms term related to the sub-vector pair defined by S_i with the corresponding cross correlation term. Successive applications of Property II yield increasingly tighter upper-bounding functions of ψ, each step of the succession requiring the computation of a new cross correlation term associated with a sub-vector pair so as to replace the corresponding product-of-norms term.

2.4.5 Core EBC algorithm

This subsection describes the core EBC algorithm, which relies on the mathematical properties presented in section 2.4.4. First of all, both the template T and the current image subwindow $I_c(x, y)$ are seen as vectors belonging to a $M \times N-$dimensional space and Fig. 2.9 shows a generic partitioning of such vectors into r sub-vectors, as required by Properties I and II. In general a sub-vector can consist of disjoint sets of pixels, as is the case of sub-vector 4 in Fig. 2.9.

To deploy the properties of section 2.4.4 within the bounded correlation framework outlined in section 2.4.3, we define a partitioning of vectors T, $I_c(x, y)$ and apply Property I at first, to obtain an initial upper bound $\beta(x, y)$ at each image position (x, y) and check the associated skipping condition (2.32) that does not require calculation of any partial correlation term. If such initial condition is not verified, we then apply Property II in successive steps. At each step the product-of-norms term of a sub-vector pair is replaced by the corresponding cross-correlation term, to obtain a tighter bounding function and associated skipping condition (2.32).

For reasons of computational efficiency, in a practical deployment of the EBC principle it is preferable to adopt a kind of "regular" partitioning scheme to be applied to T and $I_c(x, y)$. In our implementation T and $I_c(x, y)$ are partitioned into sub-vectors made out of successive rows, as shown in Fig. 2.10. All sub-vectors are chosen to have the same number of rows, n, except for the last one (e.g. in Fig. 2.10, sub-vector r). Hence, with our partitioning scheme the first $r - 1$ sub-vectors have $M \times n$ elements and the last one has $M \times (N - (r - 1) \cdot n)$ elements. In fact, EBC requires evaluation of the norms of all the sub-vectors resulting from the partitioning of T and $I_c(x, y)$. Like $||T||$ and $||I_c(x, y)||$, the former norms can be computed once for all at initialisation time, while the latter can be calculated efficiently at run-time by means of incremental techniques. In particular, we adopted the one-pass *box-filtering* method proposed in [91]. In our implementation, a box-filtering function fills in an array of norms by computing the norm of each rectangular window of given dimensions belonging to image I. As described in [91], this is done by exploiting a double recursion on the rows and columns of image array I, which requires only four elementary operations per image point irrespective of the size of the rectangular window. Hence, it is readily inferred that to obtain the required sub-vector norms we need to run as many box-filters as the number of differently shaped rectangular windows corresponding to sub-vectors. Therefore, the choice of using two different shapes of sub-vectors allows us to run only two distinct box-filters, thereby also requiring a relatively small memory footprint (i.e. twice the image size). In the particular case $n = N/r$, all the r sub-vectors have the same shape and the computational efficiency is even higher, with the need for only one box-filter instance.

Having shown EBC basic principles and the partitioning scheme adopted, we proceed herein with a detailed description of the core algorithm.

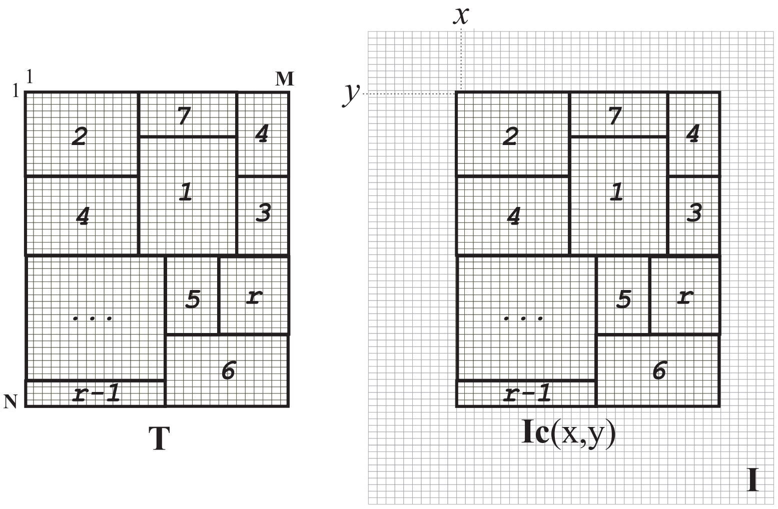


Figure 2.9: Generic partitioning of template and current image subwindow.

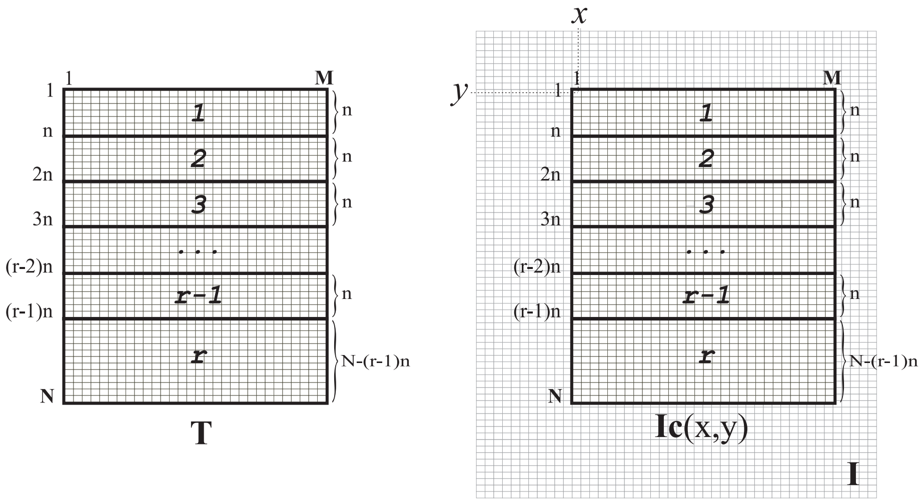


Figure 2.10: Partitioning scheme adopted in our current EBC implementation.

The initial upper bounding function based on Property I can be expressed as:

$$\beta^r(x,y)|_1^N = \sum_{t=1}^{r-1}\sqrt{\sum_{j=(t-1)\cdot n+1}^{t\cdot n}\sum_{i=1}^{M} I^2(x+i,y+j)}\cdot\sqrt{\sum_{j=(t-1)\cdot n+1}^{t\cdot n}\sum_{i=1}^{M} T^2(i,j)}+ + \sqrt{\sum_{j=(r-1)\cdot n+1}^{N}\sum_{i=1}^{M} I^2(x+i,y+j)}\cdot\sqrt{\sum_{j=(r-1)\cdot n+1}^{N}\sum_{i=1}^{M} T^2(i,j)} \tag{2.40}$$

This upper bound gives the initial sufficient condition for skipping the current image point:

$$\frac{\beta^{r}(x,y)|_{1}^{N}}{||I_c\,(x,y)\,||\cdot||T||} < \eta_M \tag{2.41}$$

The right hand inequality of Property I guarantees the potential effectiveness of the initial sufficient condition (i.e. the left-hand term in (2.41) is always ≤ 1).

If the initial condition holds, EBC skips the current image point. Instead, if it does not hold, we attain a tighter bounding function by deploying Property II. That is, denoting the generic *partial correlation* term associated with rows (ρ, θ) as:

$$\psi(x,y)|_{\rho}^{\theta} = \sum_{j=\rho}^{\theta}\sum_{i=1}^{M} I\,(x+i, y+j)\cdot T\,(i,j), \tag{2.42}$$

the next bounding function can be expressed as

$$\gamma^{r-1}(x,y) = \psi(x,y)|_{1}^{n} + \beta^{r-1}(x,y)|_{n+1}^{N}, \tag{2.43}$$

$\beta^{r-1}(x,y)|_{n+1}^{N}$ representing a function defined as in (2.40) with the summation starting from $t = 2$ instead of $t = 1$, and the corresponding sufficient condition as

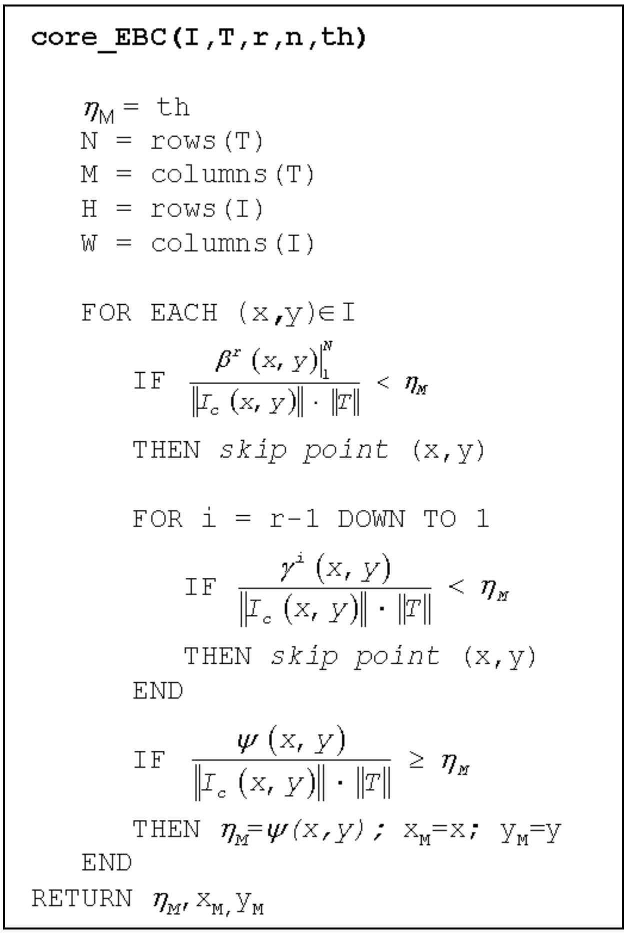
$$\frac{\gamma^{r-1}(x,y)}{||I_c\,(x,y)\,||\cdot||T||} < \eta_M \tag{2.44}$$

Should also (2.44) not be satisfied, the method would proceed by successive applications of Property II: at each step a bounding term (product-of-norms) is replaced with the corresponding partial correlation term and a new skipping condition is checked. With this approach, EBC can check up to r sufficient conditions (including the initial one), the last upper bounding function and associated condition given by:

$$\gamma^{1}(x,y) = \psi(x,y)|_{1}^{(r-1)\cdot n} + \beta^{1}(x,y)|_{(r-1)\cdot n+1}^{N} \tag{2.45}$$

$$\frac{\gamma^{1}(x,y)}{||I_c\,(x,y)\,||\cdot||T||} < \eta_M \tag{2.46}$$

Should the last condition not be verified, the process completes the computation of the actual cross-correlation value (i.e. $\psi(x,y)$) that is used to check condition (2.33) by replacing $\beta^{1}(x,y)|_{(r-1)\cdot n+1}^{N}$ with $\psi(x,y)|_{(r-1)\cdot n+1}^{N}$. The pseudo-code for the core EBC algorithm is shown in Fig. 2.11, with parameter *th* (always < 1) representing the initialization value for η_M.



```
core_EBC(I,T,r,n,th)

    $\eta_M$ = th
    N = rows(T)
    M = columns(T)
    H = rows(I)
    W = columns(I)

    FOR EACH (x,y)∈I
        IF $\frac{\beta^r(x,y)|_1^N}{\|I_c(x,y)\| \cdot \|T\|} < \eta_M$
        THEN skip point (x,y)

        FOR i = r-1 DOWN TO 1
            IF $\frac{\gamma^i(x,y)}{\|I_c(x,y)\| \cdot \|T\|} < \eta_M$
            THEN skip point (x,y)
        END

        IF $\frac{\psi(x,y)}{\|I_c(x,y)\| \cdot \|T\|} \geq \eta_M$
        THEN $\eta_M$=$\psi(x,y)$; $x_M$=x; $y_M$=y
    END
RETURN $\eta_M$, $x_M$, $y_M$
```

Figure 2.11: Pseudo-code describing the core EBC algorithm.

As previously mentioned, if the initial condition (2.41) holds, EBC skips the current image point without calculating any partial correlation term. This is not the case for previous bounded correlation algorithms ([35], [36]), that always require calculation of a certain fraction of the actual correlation score (indeed, they are referred to as bounded *partial* correlation algorithms). As shown in Section 2.4.7 (Table 2.8), the initial condition very frequently allows a substantial fraction of the total image points to be skipped.

As regards the higher effectiveness of the EBC bounding strategy, by comparing (2.43) to (2.36) it can also be observed that, by virtue of the right inequality of Property I, given the same amount of partial correlation (i.e. with the same *n*), the bounding function used by EBC to check condition (2.44) is tighter than that used in [36] and consequently in [35]. Moreover, if (2.32) is not satisfied the algorithm in [36] carries on with the computation of the remaining fraction of the correlation, while in case of failure of condition (2.44) EBC increasingly tightens the bounding function by computation of smaller partial correlation terms.

The idea of obtaining increasingly tighter bounds through computation of small partial correlation terms was also suggested in [35] in the framework of the initial BPC algorithm based on Jensen inequality. However, even applying the incremental method discussed in [35] to a BPC algorithm based on Cauchy-Schwarz inequality [36] would yield bounds less tight than those attainable with the novel bounding strategy proposed in this paper by virtue of the right inequality of Property I. This will also be proved by the much higher computational efficiency reported in the experimental results of section 2.4.7.

Finally, it can readily be inferred from (2.40) and (2.44) that should the elements of I or T be multiplied by a constant factor, then each of the sufficient conditions checked by EBC would not change. Hence, the computational benefits provided by EBC are independent from any possible intensity scaling occurring between the image under examination and the template. This is an important property since in template matching applications the NCC is often preferred to other functions, such as the SAD or SSD, due to its invariance to intensity scaling.

2.4.6 Overall EBC algorithm

The core EBC algorithm, as in the case of [46], [83], [132], [35], [36], [57], is a data-dependent computational optimization technique, with one major factor that impacts on performance being the goodness of some initial match. In fact, the sufficient conditions checked by the algorithm become more effective as the correlation between the template and the current best matching image subwindow (i.e. η_M) becomes higher. Therefore, the algorithm provides a higher computational efficiency when, given the scan order, the search process rapidly finds a good matching position.

To deal with this issue EBC would benefit from a strategy aimed at rapidly finding a suitable initialization value for η_M. In our current implementation we enforce the following coarse-to-fine approach. An initial search for the best match is carried out using sub-sampled versions of I and T, then the best matching position is mapped at full resolution and a second search is carried out in a small neighbourhood of this candidate position. The sub-sampling factor $\overline{k}$ depends on the image size and is chosen automatically as that minimizing the total number of operations required by the two searches. More precisely, in our implementation:

$$\overline{k} = \arg\min_{k}\{\frac{(W-M)\cdot(H-N)\cdot M\cdot N}{k^4} + (4\cdot k)^2\cdot M\cdot N\} \tag{2.47}$$

where the first term of the function to be minimized represents the number of operations executed using the sub-sampled images, the second that carried out in a $4k \times 4k$ neighbourhood of the full resolution image. The outcome of the second search is used as the initial best match that provides the initialization value for η_M. By determining the sub-sampling factor from (2.47) and the neighbourhood size accordingly (i.e. $4\bar{k} \times 4\bar{k}$), this match turns out to be generally a good one, so that, as regards overall efficiency, the increased effectiveness of the sufficient conditions largely pays-off with respect to the negligible computational overhead due to the initial coarse-to-fine search. Since the initial value for η_M is the actual NCC score computed at a certain position of the full resolution image, it is guaranteed to be less than or equal to the global maximum. Hence, the coarse-to-fine stage does not affect the optimality of the solution found by the overall EBC algorithm.

Furthermore, as is clear from section 2.4.5, performance of the core algorithm depends inherently on the choice of parameters (r, n), as different partitioning schemes yield different bounding functions and associated sufficient conditions. In this regard, our experiments have shown notable variations of EBC execution time in some cases as a consequence of different choices of the partitioning parameters. Hence, the determination of a good (perhaps optimal) choice of parameters (r, n) plays an important role with respect to practical deployments of the EBC principle. Since most applications do not allow for an off-line training process aimed at parameter choice, we have developed a general predictive approach that enables rapid run-time estimation of a good parameter pair for the current image and template.

The run-time prediction step relies on several empirical observations derived from an off-line analysis of the algorithm behaviour carried out on a large dataset. First of all, we limited the maximum r value to be taken into consideration (i.e. $r_{max} = 40$) due to the observation that, generally, increasing r above this limit does not provide additional computational savings. Then, for each problem instance (i.e. image and template) in the dataset, we ran the core algorithm with all possible parameter pairs (i.e. $r = 2 \dots r_{max}$ and, given r, $n = 1 \dots \lfloor \frac{(N-1)}{(r-1)} \rfloor$). Indeed, to deal with parameter pairs independent of the actual template size we normalized the size of the partitions with respect to the template size, thus taking into considerations the pairs $(r, \frac{n}{N})$. For each problem instance we recorded the pair $(r, \frac{n}{N})$ that minimizes the execution time. All such pairs were fed into a vector quantization process to select a small subset of reference parameter pairs (i.e. the 7 pairs shown in Table 2.6). The selection was based on minimization of the error associated with representing the whole set of the recorded best pairs with a smaller subset of given cardinality (i.e. only the 7 pairs).

Once the reference pairs in Table 2.6 had been determined off-line, the task left to the run-time prediction step was to guess the most appropriate reference pair for

Table 2.6: Reference parameter pairs used in the predictive approach.

r	$\frac{n}{N}$
34	0.03
25	0.04
20	0.05
16	0.06
12	0.09
8	0.13
4	0.18

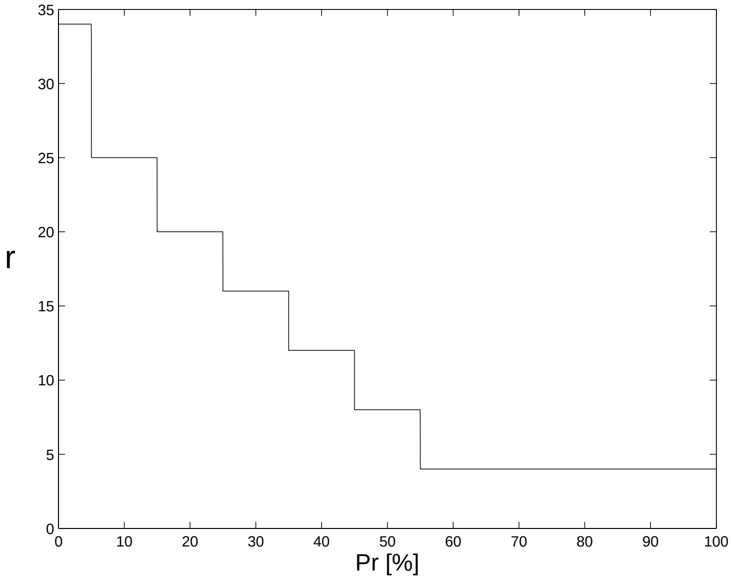
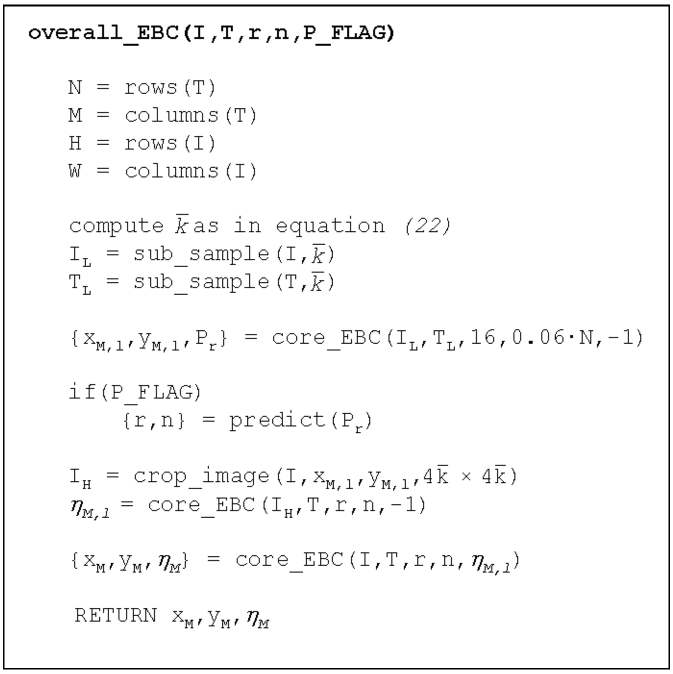


Figure 2.12: Mapping function used to determine r from P_r.

the actual problem instance as rapidly as possible. As it is in the case of initialization of η_M, the run-time parameter prediction step works on sub-sampled versions of I, T, so as to require a low computational overhead under the statistical assumption that a heterogeneous subset holds the characteristics of the whole set.

Hence, the first sufficient condition (i.e. (2.41)) is applied on the sub-sampled I, T using the reference parameter pair $r = 16$, $\frac{n}{N} = 0.06$ to calculate the fraction of skipped image points, denoted as P_r. We have observed empirically that P_r can be regarded as an indicator of the efficiency of the first sufficient condition of the algorithm and that this provides guidelines on the choice of parameters (r, n). More precisely, if P_r turns out close to 1, then it is highly probable that EBC sufficient conditions are also



```
overall_EBC(I,T,r,n,P_FLAG)

    N = rows(T)
    M = columns(T)
    H = rows(I)
    W = columns(I)

    compute $\bar{k}$ as in equation (22)
    $I_L$ = sub_sample(I,$\bar{k}$)
    $T_L$ = sub_sample(T,$\bar{k}$)

    {$x_{M,1}$,$y_{M,1}$,$P_r$} = core_EBC($I_L$,$T_L$,16,0.06·N,-1)

    if(P_FLAG)
        {r,n} = predict($P_r$)

    $I_H$ = crop_image(I,$x_{M,1}$,$y_{M,1}$,$4\bar{k} \times 4\bar{k}$)
    $\eta_{M,1}$ = core_EBC($I_H$,T,r,n,-1)

    {$x_M$,$y_M$,$\eta_M$} = core_EBC(I,T,r,n,$\eta_{M,1}$)

    RETURN $x_M$,$y_M$,$\eta_M$
```

Figure 2.13: Pseudo-code for the overall EBC algorithm with on-line parameter estimation.

effective with a much coarser partitioning scheme: this suggests the use of a much smaller r, so as to avoid the computational overhead associated with an unnecessarily fine partitioning scheme. Conversely, if a low P_r is found, then the number of partitions must be increased accordingly to attain effective sufficient conditions. Based on such empirical observations we designed a mapping function from the calculated P_r value to the r value belonging to the reference parameter set (see Fig. 2.12). More precisely, the mapping function was obtained as the decreasing step function that best fitted a cloud of (P_r, r) points found experimentally. Thus, the run-time prediction step quickly calculates P_r, then finds r according to the mapping function in Fig. 2.12 and finally $\frac{n}{N}$ from Table 2.6. This will be denoted in the pseudo-code of Fig. 2.13 as function `predict`.

Eventually, our current implementation of the overall EBC algorithm provides two options. The first requires (r, n) as input parameters and runs only the coarse-to-fine search aimed at initializing η_M. The second also runs the prediction stage by integrating it seamlessly with the coarse-to-fine search. More precisely, the coarse step aimed at finding the initial value for η_M consists in running the core algorithm on the sub-sampled versions of I, T with parameters $(r = 16, \frac{n}{N} = 0.06)$, so that P_r can be calculated by applying the first sufficient condition. The pseudo-code for the overall

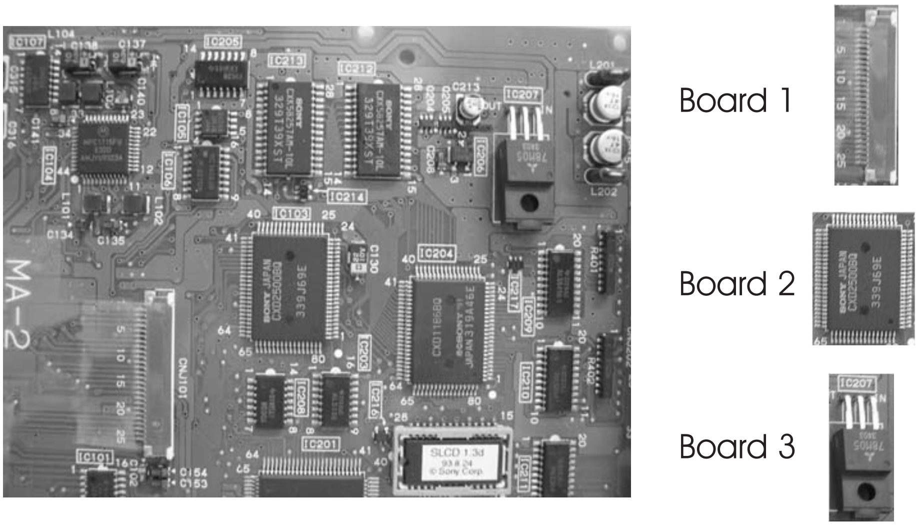


Figure 2.14: *Board* image and templates (rows 8-10 of Tables 2.7, 2.8 and 2.9).

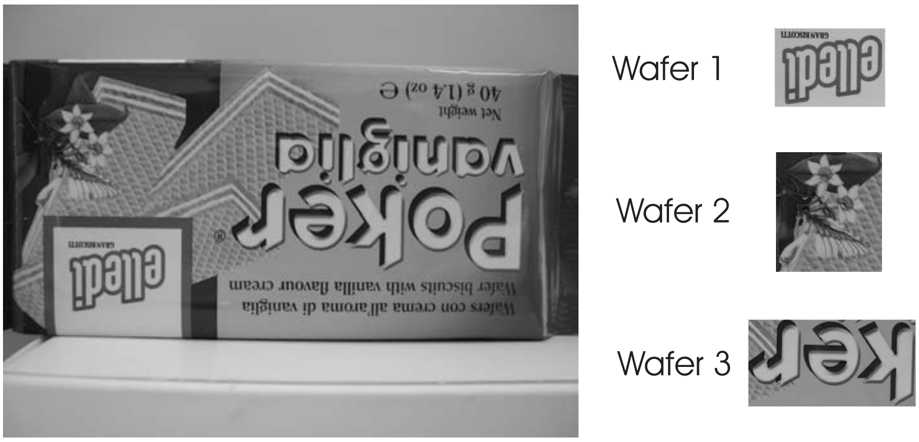


Figure 2.15: *Wafer* image and templates (rows 11-13 of Tables 2.7, 2.8 and 2.9).

EBC algorithm is shown in Fig. 2.13. We point out that *P_FLAG* allows switching between the two operating modes (with and without on-line parameter prediction), and that the core EBC function, unlike that shown in Fig. 2.11, also has to calculate and return P_r, which is needed for the on-line parameter estimation stage.

2.4.7 Experimental Results

This subsection compares the computational advantages of the overall EBC algorithm with respect to the other state-of-the-art exhaustive template matching algorithms, i.e.

Table 2.7: Dataset used in the experiments.

	Image Size	Template Size	η_M	(x_M, y_M)
Paint 1	1152×864	164×161	0.9956	(142,258)
Paint 2	1152×864	128×152	0.9953	(160,434)
Paint 3	1152×864	118×162	0.9980	(175,725)
Pcb	384×288	72×73	0.9970	(65,268)
Plants	512×400	104×121	0.9859	(66,333)
Ringo 1	640×480	126×144	0.9747	(149,102)
Ringo 2	640×480	118×162	0.9762	(159,161)
Board 1	640×480	63×179	0.9887	(265,113)
Board 2	640×480	106×138	0.9789	(198,239)
Board 3	640×480	65×149	0.9829	(61,500)
Wafer 1	640×480	119×84	0.9942	(259,65)
Wafer 2	640×480	109×123	0.9932	(108,32)
Wafer 3	640×480	189×98	0.9882	(198,256)

Table 2.8: Measured speed-ups: EBC Vs. FS.

	EBC_{opt}				EBC_{est}			
	$(r, \frac{n}{N})$	Speed-up	$P_{tot}[\%]$	$P_1[\%]$	$(r, \frac{n}{N})$	Speed-up	$P_{tot}[\%]$	$P_1[\%]$
Paint 1	(34,0.03)	49.5	≈ 100.0	63.1	(34,0.03)	47.6	≈ 100.0	63.1
Paint 2	(24,0.04)	97.4	≈ 100.0	95.2	(25,0.04)	91.7	≈ 100.0	95.4
Paint 3	(14,0.06)	139.7	≈ 100.0	98.9	(34,0.03)	90.3	≈ 100.0	99.7
Pcb	(4,0.18)	47.0	≈ 100.0	99.0	(4,0.18)	45.9	≈ 100.0	99.0
Plants	(12,0.08)	80.6	≈ 100.0	98.5	(8,0.13)	75.8	≈ 100.0	98.1
Ringo 1	(34,0.03)	19.2	≈ 100.0	70.3	(16,0.06)	14.7	≈ 100.0	60.6
Ringo 2	(34,0.03)	25.6	≈ 100.0	79.3	(16,0.06)	20.0	≈ 100.0	72.0
Board 1	(34,0.03)	9.1	≈ 100.0	29.3	(34,0.03)	9.0	≈ 100.0	29.3
Board 2	(34,0.03)	10.5	≈ 100.0	12.4	(34,0.03)	10.0	≈ 100.0	12.4
Board 3	(16,0.05)	63.2	≈ 100.0	97.4	(4,0.18)	41.0	≈ 100.0	92.6
Wafer 1	(25,0.04)	25.5	≈ 100.0	64.3	(20,0.05)	25.4	≈ 100.0	65.6
Wafer 2	(19,0.05)	65.0	≈ 100.0	93.1	(34,0.03)	59.4	≈ 100.0	93.9
Wafer 3	(25,0.04)	15.1	≈ 100.0	31.0	(25,0.04)	15.2	≈ 100.0	31.0

FS, BPC, and FFT-based. All the compared algorithms have been implemented in *C* and run on a Linux workstation based on a *Pentium 4 3.056 GHz* processor. The implementations of EBC, BPC and FS algorithms deploy the box-filtering technique [91] to compute the norms of all vectors and sub-vectors involved in the calculations. Moreover, the implementation of BPC deploys the same coarse-to-fine initialization as EBC. The dataset used for the experiments consist of grayscale images and templates of various sizes, as shown in Table 2.7. The Table also lists the coordinates of the best matching position and the corresponding NCC score. Since templates are not extracted from the image under examination itself but from another image taken with the same camera from a slightly different viewpoint, the NCC scores reported in the Table are always < 1. Hence, the impact of real distorsions typically occurring in pattern matching applications, such as camera noise and slight changes in viewpoint, is accounted for in the proposed experiments. Some samples of images and templates belonging to the dataset are shown in Figs. 2.14 and 2.15[3].

Table 2.8 compares the overall EBC algorithm to the FS algorithm. The table is split in two parts. Columns $2-5$ refer to the EBC algorithm without the on-line prediction stage and with parameters (r, n) chosen optimally (EBC_{opt}), that is determined by means of a thorough off-line training session carried out on each instance of the dataset. Instead, columns $6-9$ refer to the case where parameters have been estimated on-line by means of the prediction algorithm described in section 2.4.6 (EBC_{est}). In this case the time measurements for EBC include the on-line parameter prediction stage that, on average, requires 3.2% of the overall execution time. The table reports in each part the parameter pair $(r, \frac{n}{N})$ used by the algorithm (optimally selected in column 3, estimated on-line in column 7), the speed-ups (i.e. ratios of measured execution times) with respect to the FS algorithm, P_{tot}, i.e. the percentage of points skipped by all the applications of the sufficient conditions involved in the template matching process, and P_1, i.e. the percentage of points skipped by the application of the first sufficient condition (inequality (2.41)).

Table 2.8 shows that with an appropriate choice of the parameters the overall EBC algorithm can yield significant computational savings with respect to the FS algorithm, with measured speed-ups ranging from 9.1 up to 139.7 in the considered dataset. Moreover, it also points out that the predictive approach to parameter selection described in section 2.4.6 rapidly finds a very good $(r, \frac{n}{N})$ pair, so that the parameter-free EBC version can achieve notable computational savings, the speed-ups being in most cases very similar to those attained with an optimal parameter tuning [4]. By looking at both parts

[3] The whole dataset can be found at the following url: *www.vision.deis.unibo.it/smatt/PatternMatching.html*

[4] The slight advantage given by EBC_{est} with regards to EBC_{opt} in the *Wafer 3* instance of Table 2.8, is to be ascribed only to a faster fine resolution step of the coarse-to-fine search aimed at estimating η_M. This is due the fact that, at that stage, EBC_{opt} (*P_FLAG* switched off) uses a default parameter pair, while EBC_{est}

Table 2.9: Measured Speed-ups: EBC Vs. BPC and FFT-based algorithms.

	EBC vs BPC [35]		EBC vs BPC [36]		EBC vs FFT	
	EBC_{opt}	EBC_{est}	EBC_{opt}	EBC_{est}	EBC_{opt}	EBC_{est}
Paint 1	11.8	11.4	14.9	14.3	1.8	1.5
Paint 2	28.4	26.7	29.1	27.4	4.9	4.4
Paint 3	39.6	25.6	41.8	27.0	9.0	4.1
Pcb	17.9	17.5	14.5	14.2	7.3	7.1
Plants	27.3	25.7	24.4	22.9	8.5	8.2
Ringo 1	12.8	9.8	6.1	4.7	2.8	2.4
Ringo 2	13.8	10.8	8.0	6.2	3.2	2.8
Board 1	4.2	4.2	3.5	3.5	1.2	1.2
Board 2	4.0	3.8	3.2	3.0	1.1	1.1
Board 3	20.6	13.4	19.4	12.6	5.3	4.4
Wafer 1	12.3	12.2	7.6	7.5	3.5	3.3
Wafer 2	23.4	21.4	19.1	17.5	4.7	3.2
Wafer 3	4.9	5.0	4.6	4.6	1.4	1.4

of the table we can observe that EBC always runs almost one order of magnitude faster than the FS.

Comparing EBC with BPC [35], [36], it can be noticed that EBC theoretically outperforms BPC significantly due to its tighter bounds and more effective bounding strategy. In fact, BPC has a fixed *correlation-ratio* (i.e. $\frac{n}{N}$), which places a theoretical upper bound on the maximum attainable speed-up. For example, taking $\frac{n}{N} = 0.3$, as proposed in [36], implies that a fraction of the cross-correlation term as high as 30% must be computed at each point under examination. As a result, whatever the dataset, the maximum attainable speed-up with respect to the FS algorithm cannot exceed 3.3. Similar considerations apply for results in [35], where $\frac{n}{N} = 0.2$ is used. Conversely, the proposed approach has the capability of calculating at each point the proper fraction of the cross-correlation that enables to compare accurately the point to the current best matching position (i.e. from no correlation at all, when condition (2.41) holds, up to the whole cross-correlation, when even condition (2.46) is not satisfied). Hence, EBC

(*P_FLAG* switched on) can already exploit the predicted parameter pair has the potential for much higher speed-ups: e.g. in the dataset considered in this paper the minimum speed-up yielded by the parameter-free EBC algorithm (i.e. 9.0) is nearly three times the theoretical upper bound on the speed-up for the BPC algorithm proposed in [36].

These theoretical considerations were experimentally confirmed by the results shown in columns $1 - 4$ of Table 2.9, reporting the measured speed-ups of EBC with regard to BPC algorithms [35], [36], with the optimal choice of parameter pairs (EBC_{opt}) and with the on-line estimation of parameters (EBC_{est}). For a fair comparison, both BPC algorithms deployed the same coarse-to-fine strategy aimed at initializing η_M as EBC. BPC parameters were chosen according to [35] (i.e. $Cr1 = 0.2, Cr2 = 0.4$) and [36] (i.e. $Cr = 0.3$). These results clearly demonstrate that the proposed approach dramatically outperform BPC algorithms, with measured speed-ups ranging from 3.0 up to 41.8. Moreover, since the same coarse-to-fine initialization strategy was used for EBC and BPC algorithms, this comparison demonstrates that the computational efficiency attained by the overall EBC algorithm is closely associated with its novel and effective bounding strategy.

To further validate the proposed method we compared EBC to a FFT-based template matching approach. As already mentioned, the FFT is quite popular for NCC-based template matching, even though the FFT requirement for floating-point arithmetic turned out to be a serious drawback in a number of applications (especially those based on embedded architectures) [123]. Among the various FFT-based algorithms we chose as term of comparison the *cvMatchTemplate* algorithm, belonging to the well known and highly optimized *OpenCV* computer vision library, written in *C* and developed by *Intel*. To compare the two algorithms as fairly as possible, the EBC implementation was also optimized by deploying the parallel multimedia-oriented instructions (i.e. MMX) available on state-of-the-art processors based on Intel Architecture. Columns 6 and 7 of Table 2.9 show the speed-ups provided by EBC with respect to the FFT-based algorithm in the considered dataset. Column 6 refers to the case of optimal choice of EBC parameters. Column 7 shows the results in case of on-line parameter estimation, that now requires on average 3.4% of the overall execution time. As shown clearly by the Table, in the dataset considered EBC was always faster than the FFT-based algorithm, yielding on average substantial computational savings and, in some instances, quite remarkable speed-ups (e.g. up to 8.2 in the parameter-free version).

2.4.8 Experiments with small templates and artificial noise

This subsection provides further experimental results aimed at assessing the performance of EBC in case of smaller template sizes and increasing artificial noise. To this purpose, ten templates of size 64×64 were hand-selected uniformely from an image

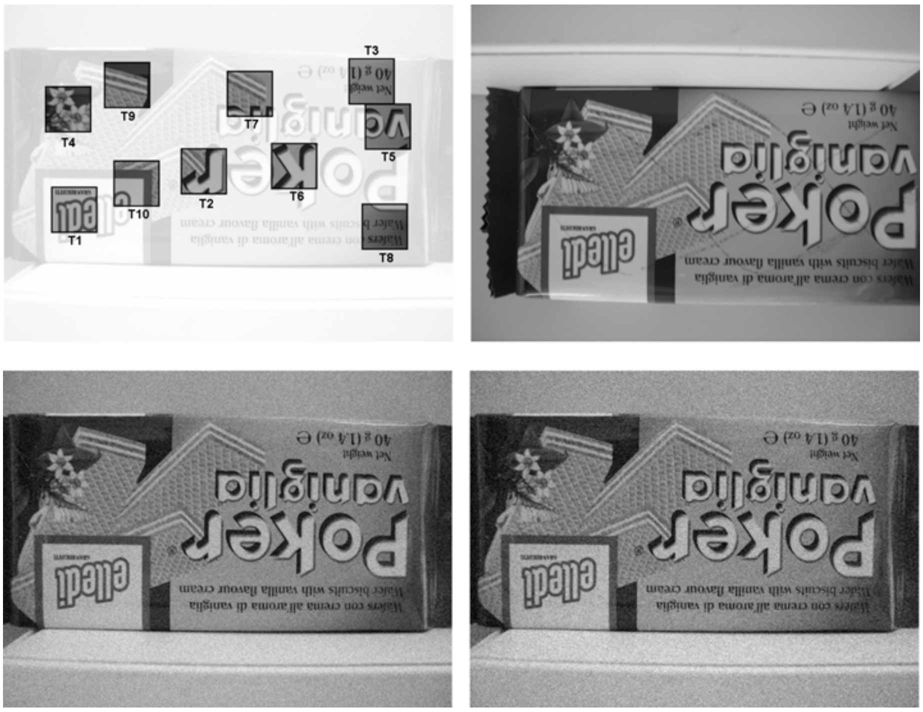


Figure 2.16: Images and templates used for the experiments with small templates and artificial noise.

Table 2.10: Measured Speed-ups in case of smaller templates and artificial noise

	Artificial noise, σ_1		Artificial noise, σ_2		Real distorsions	
	EBC_{est} vs FS	EBC_{est} vs FFT	EBC_{est} vs FS	EBC_{est} vs FFT	EBC_{est} vs FS	EBC_{est} vs FFT
T1	23.3	2.8	12.8	2.0	15.8	2.4
T2	19.3	3.2	11.4	2.2	37.9	4.3
T3	5.6	1.2	4.6	1.1	4.8	1.1
T4	10.7	2.0	7.8	1.6	29.4	2.8
T5	27.7	3.0	10.9	1.8	7.3	1.6
T6	35.3	4.1	23.9	3.3	30.1	3.1
T7	5.2	1.2	3.8	0.9	9.8	2.0
T8	45.6	6.0	24.8	4.1	33.4	5.0
T9	29.8	4.6	17.7	3.3	46.5	6.0
T10	39.7	4.9	29.8	4.2	45.8	5.3

belonging to our dataset (Fig. 2.16, top left). These were then matched into the image itself, after addition of two different levels of i.i.d. zero-mean-gaussian artificial noise (i.e. $\sigma_1 = 0.003$ and $\sigma_2 = 0.005$ [5], Fig. 2.16 bottom left and bottom right respectively), and into another image taken with the same camera from a slightly different position (Fig. 2.16, top right). The speed-ups yielded by the overall EBC algorithm with online parameter estimation (EBC_{est}) with respect to the FS and FFT algorithms are shown in Table 2.10. To investigate on the impact of a smaller template size, the results in the two rightmost columns of Table 2.10 can be compared directly to those reported in the last eight rows (column 7) of Tables 2.8 and 2.9, as they refer to pattern matching instances under real distorsions and characterized by the same image size but significantly larger template sizes. By comparing the three Tables it can readily be seen that the computational advantages made by EBC do not change sharply as a result of a significant decrease of the template size. Also with smaller templates EBC generally runs notably faster than the FS and the FFT, the speed-up ranges now being $[4.8 \div 46.5]$ and $[1.1 \div 6.0]$ respectively.

As to the impact of increasing noise, comparison of columns 2,3 with 6,7 of Table 2.10 indicates that with the smaller level of artificial noise the performance of EBC is substantially equivalent to that measured in the addressed real distortions scenario. However, columns 3,4 of the Table show clearly that the computational benefits tend to decrease with increasing noise, although with σ_2 they are still notable (i.e. up to 29.8 and 4.1 respectively compared to the FS and FFT algorithms).

Finally, as regards this dataset, the on-line prediction stage requires on average 2.7% of the overall execution time, which increases to 3.1% in the case of the EBC implementation deploying MMX-optimization.

2.5 Fast template matching based on the ZNCC

This section proposes a novel fast and exhaustive technique for template matching, referred to as *Zero-mean Enhanced Bounded Correlation* (ZEBC), which is based on the ZNCC measure and it is inspired by EBC.

2.5.1 Previous work

The ZNCC function at pixel position (x, y) is given by:

$$ZNCC(x,y) = \frac{(I_c(x,y) - \mu_{I_c}(x,y)) \circ (T - \mu_T)}{||I_c(x,y) - \mu_{I_c}(x,y)|| \cdot ||T - \mu_T||} \tag{2.48}$$

[5] with respect to normalized pixel intensities ranging within [0, 1].

with $\mu_{I_c}(x, y)$ and μ_T being respectively the mean intensity value computed over $I_c(x, y)$ and T, $\circ$ representing the dot product between two vectors, and the two terms at the denominator being respectively the L_2 norm of zero-mean image candidate and zero-mean template vectors. Hereinafter the numerator of (2.48) will be referred to as $\eta(x, y)$.

Similarly to the case of the SSD and NCC, also for the ZNCC a common alternative [81] to the FS algorithm computes the cross-correlation in the frequency domain by means of the well-known Fast Fourier Transform (FFT). Another exhaustive approach aimed at speeding up ZNCC-based template matching is the *Zero-mean Bounded Partial Correlation* (ZBPC) technique [37]. This method relies on two computationally efficient upper-bounding functions for $\eta(x, y)$, $\beta'_{ZBPC}(x, y)$ and $\beta''_{ZBPC}(x, y)$, that allow for rapidly pruning mismatching candidates by testing:

$$\frac{min(\beta'_{ZBPC}(x, y), \beta''_{ZBPC}(x, y))}{||I_c(x, y) - \mu_{I_c}(x, y)|| \cdot ||T - \mu_T||} \leq ZNCC_{max} \tag{2.49}$$

with $ZNCC_{max}$ being the ZNCC maximum found among previously evaluated candidates. If (2.49) holds then the current candidate is guaranteed not to be the global maximum and ZNCC computation need not being carried out. Nevertheless, since the computation of both $\beta'_{ZBPC}(x, y)$ and $\beta''_{ZBPC}(x, y)$ involves calculating a partial cross-correlation term on a subset of template and candidate vectors (i.e. on n_{ZBPC} rows), then the speed-ups yielded by ZBPC on FS are upper-bounded by $\frac{N}{n_{ZBPC}}$.

Moreover, very recently a fast exhaustive scheme for ZNCC-based block matching was proposed in [87]. By determining a monotonically decreasing equivalent expression of (2.48), a Partial Distortion Elimination [9] approach is applied in order to safely terminate the computation of ZNCC as soon as it gets below $ZNCC_{max}$.

The two main novelties of ZEBC with respect to ZBPC are represented by the use of two bounding functions which do not require any partial cross-correlation term computation at all and which can be demonstrated being tighter to $\eta(x, y)$ than $\beta'_{ZBPC}(x, y)$ and $\beta''_{ZBPC}(x, y)$, and by the definition of an additional set of increasingly tighter bounding functions.

2.5.2 The ZEBC algorithm

We now devise two novel bounding functions, $\beta'(x, y)$ and $\beta''(x, y)$, which allow to rapidly detect mismatching candidates without the need to compute any partial correlation term. By means of a partitioning scheme similar to that deployed for EBC in Section 2.4, each candidate and template vectors are subdivided into r non-overlapping rectangular regions $R_1, \cdots, R_r$ of size $M \times n$, with $n = \frac{N}{r}$. We will refer to $I_{c,t}(x, y)$ and T_t as, respectively, the candidate and template subvectors corresponding to region

R_t, and to A_t as their cardinality (i.e. the number of pixel in each region, $A_t = n \cdot M$). Then, $\eta(x, y)$ can be seen as the sum of r partial terms $\eta_t(x, y)$ each one computed on its corresponding region R_t:

$$\eta(x,y) = \sum_{t=1}^{r} \eta_t(x,y) \tag{2.50}$$

where:

$$\eta_t(x,y) = \left(I_{c,t}(x,y) - \mu_{I_{c,t}}(x,y)\right) \circ \left(T_t - \mu_{T_t}\right) \tag{2.51}$$

By means of the application of the Cauchy-Schwarz inequality on (2.51) we can devise an upper-bound for term $\eta_t(x, y)$:

$$\begin{aligned} \beta'_t(x,y) &= \|I_c(x,y) - \mu_{I_c}(x,y)\| \cdot \|T - \mu_T\| = \\ &= \sqrt{\|T_t\|^2 + A_t \cdot \mu_T \left(\mu_T - 2 \cdot \mu_{T_t}\right)} \cdot \sqrt{\|I_{c,t}\|^2 + A_t \cdot \mu_{I_c}(x,y)\left(\mu_{I_c} - 2 \cdot \mu_{I_{c,t}}\right)} \end{aligned} \tag{2.52}$$

All terms in $\beta'_t(x, y)$ relative to the image candidate can be efficiently computed by means of incremental techniques such as [91], [24], while the others, relative to the template, can be computed once for all at start-up.

Additionally, equation (2.51) can be algebraically manipulated as follows:

$$\eta_t(x,y) = I_{c,t} \circ T_t + A_t \left(\mu_{I_c}(x,y) \cdot \mu_T - \mu_{T_t} \cdot \mu_{I_c}(x,y) - \mu_T \cdot \mu_{I_{c,t}}(x,y)\right) \tag{2.53}$$

Hence, by applying the Cauchy-Schwarz inequality on the cross-correlation between $I_{c,t}$, T_t term in (2.53) we get an additional upper-bound for $\eta(x, y)$:

$$\beta''_t(x,y) = \|I_{c,t}\| \cdot \|T_t\| + A_t \left(\mu_{I_c}(x,y) \cdot \mu_T - \mu_{T_t} \cdot \mu_{I_c}(x,y) - \mu_T \cdot \mu_{I_{c,t}}(x,y)\right) \tag{2.54}$$

Though different from $\beta'_t(x, y)$, also this term can be computed very efficiently, partly at start-up and partly by means of incremental schemes. It is worth pointing out that, since both $\beta'_t(x, y)$ and $\beta''_t(x, y)$ are computed on the same region R_t, and since all regions are equally sized, their calculation requires a reduced number of incremental scheme instances, with benefits for what concerns efficiency and memory requirements. Moreover, their computational complexity is independent from image and template sizes.

Thus, we propose a very effective upper-bound for $\eta(x, y)$ by choosing, for each region R_t, the term between $\beta'_t(x, y)$ and $\beta''_t(x, y)$ that better approximates $\eta_t(x, y)$:

$$\beta^B(x,y) = \sum_{t=1}^{r} min\,(\beta'_t(x,y), \beta''_t(x,y)) \tag{2.55}$$

By comparing (2.50) and (2.55) it is easy to infer the bounding property of $\beta_B(x,y)$. Hence, for each candidate $\beta^B(x,y)$ can be used to reliably detect mismatching candidates previously to the computation of the ZNCC term. If condition

$$\frac{\beta^B(x,y)}{||I_c(x,y) - \mu_{I_c}(x,y)|| \cdot ||T - \mu_T||} < ZNCC_{max} \tag{2.56}$$

is verified, then candidate $I_c(x,y)$ is guaranteed not to be the global maximum and its ZNCC score does not need to be computed.

For the sake of efficiency, since the computation of $\beta^B(x,y)$ requires the computation of both $\beta'_t(x,y)$, $\beta''_t(x,y)$ terms on all regions, based on experimental evidence we suggest to compute first

$$\beta''(x,y) = \sum_{t=1}^{r} (\beta''_t(x,y)) \tag{2.57}$$

and then to use it to detect a first set of mismatching candidates. The computation of (2.55) is carried out only for those candidates that are not rejected by means of (2.57). Though experimentally it seemed more favourable to choose $\beta''_t(x,y)$ rather than $\beta'_t(x,y)$ as the bounding terms to be computed first, a deeper study concerning a more advanced scheme aimed at exploiting these terms more effectively is currently under development.

Now, for all candidates not rejected by means of either (2.55) or (2.57) we propose to refine the search for mismatching candidates by means of a set of increasingly tighter bounding functions. First, a bounding function can be determined by substituting in $\beta^B(x,y)$ the bounding term computed on region R_1 with its corresponding

$\eta_1(x,y)$ term:

$$\begin{aligned}\gamma^1(x,y) = \eta_1(x,y) + \sum_{t=2}^{r} min\,(\beta'_t(x,y), \beta''_t(x,y)) = \\ \beta^B(x,y) - min\,(\beta'_1(x,y), \beta''_1(x,y)) + \eta_1(x,y)\end{aligned} \tag{2.58}$$

where the right hand term in (2.58) shows how to efficiently compute $\gamma^1(x,y)$ from $\beta^B(x,y)$. $\gamma^1(x,y)$ represents a tighter approximation of $\eta(x,y)$ compared to $\beta''(x,y)$ and $\beta^B(x,y)$, though computationally more expensive, hence can be used to classify as mismatching those candidates which were previously not rejected by $\beta''(x,y)$ and $\beta^B(x,y)$.

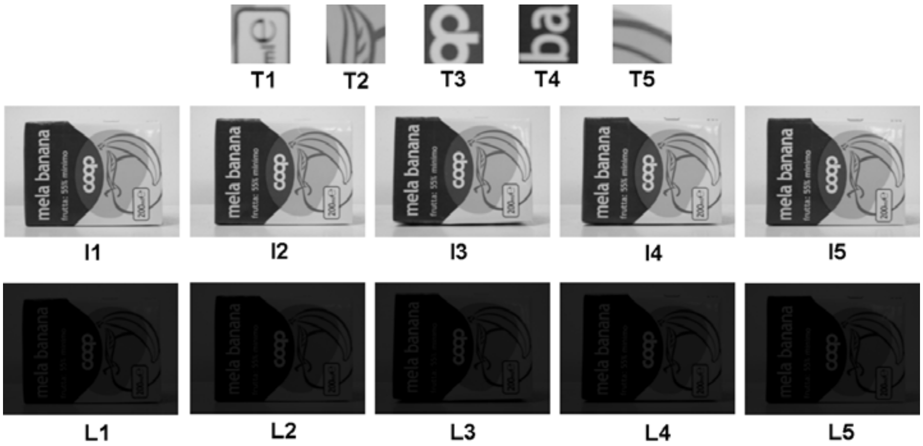


Figure 2.17: Dataset used for the experimental results.

Following this approach, by substituting at each step the current bounding term with its corresponding $\eta_t(x,y)$ term, up to $r-1$ additional upper bounding functions can be overall deployed for candidate rejection, that is $\gamma^1(x,y)\cdots\gamma^{r-1}(x,y)$, the last one being:

$$\gamma^{r-1}(x,y) = \gamma^{r-2}(x,y) - min\left(\beta'_{r-1}(x,y), \beta''_{r-1}(x,y)\right) + \eta_{r-1}(x,y) \tag{2.59}$$

If not even $\gamma^{r-1}(x,y)$ is able to reject the current candidate, then the computation of the ZNCC is completed by calculating $\eta_r(x,y)$.

It is worth pointing out that similarly to ZBPC, also ZEBC would benefit of the use of a proper strategy to initialize $ZNCC_{max}$ with an initial guess aimed at increasing the efficiency of the bounding functions applied. Hence, we propose to use the same coarse-to-fine strategy adopted in [37]. It is worth pointing out that this initialization strategy can not violate the exhaustivity of the search, since the initialized ZNCC maximum can never be higher than the real maximum. Thus, ZEBC is always FS-equivalent.

2.5.3 Experimental results

In this section we propose an experimental evaluation aimed at assessing the benefits brought in by the proposed method, ZEBC, by comparing it to the other state-of-the-art fast exhaustive template matching approaches. As a benchmark for evaluation we propose a typical quality assessment setup, where 5 templates, $T1\cdots T5$ of size 64×64 were uniformly extracted from a reference image of a product item and then searched in the images of different items, as if on a production belt. In particular, 5 different

Table 2.11: Measured speed-ups: ZEBC Vs. FS, ZBPC and FFT-based algorithms.

	ZEBC vs FS					**ZEBC vs ZBPC**					**ZEBC vs FFT**				
	I1	**I2**	**I3**	**I4**	**I5**	**I1**	**I2**	**I3**	**I4**	**I5**	**I1**	**I2**	**I3**	**I4**	**I5**
T1	17.2	15.1	14.7	24.5	23.5	1.5	1.3	1.5	2.1	2.1	2.4	2.1	2.1	3.0	2.9
T2	15.2	15.6	11.6	11.5	13.8	1.4	1.4	2.3	2.3	1.4	2.2	2.3	1.8	1.8	2.0
T3	19.3	21.4	17.7	22.8	15.9	1.7	1.8	1.6	2.0	1.4	2.6	2.8	2.4	2.9	2.2
T4	14.3	15.4	19.0	17.5	15.7	1.3	1.4	1.6	1.6	1.4	2.1	2.2	2.6	2.4	2.2
T5	27.7	27.8	28.5	28.0	12.4	3.9	4.9	3.1	3.4	3.9	3.3	3.4	3.4	3.4	1.8
Mean	18.6					2.1					2.5				

Table 2.12: Measured speed-ups on dataset affected by affine photometric distortions.

	ZEBC vs FS					**ZEBC vs ZBPC**					**ZEBC vs FFT**				
	L1	**L2**	**L3**	**L4**	**L5**	**L1**	**L2**	**L3**	**L4**	**L5**	**L1**	**L2**	**L3**	**L4**	**L5**
T1	13,7	12,8	12,2	19,7	18,2	1,3	1,2	1,3	1,8	1,7	2,0	1,9	1,9	2,7	5,3
T2	12,7	13,4	10,6	10,4	11,8	1,3	1,3	2,2	2,2	1,3	2,0	2,0	1,7	1,7	1,9
T3	15,2	16,7	14,1	17,5	19,1	1,4	1,5	1,3	1,6	0,8	2,2	2,4	2,1	2,5	2,0
T4	8,5	9,1	9,5	10,6	10,8	3,6	2,4	2,5	1,7	1,9	1,4	1,5	1,5	1,8	1,7
T5	23,2	23,0	23,6	23,6	10,4	3,4	4,1	2,8	3,0	3,3	3,0	3,0	2,0	3,1	1,7
Mean	14,8					2,0					2,2				

images of as many items, $I1 \cdots I5$, are used, each one sized 640×480. This dataset is shown in Fig. 2.17.

As for the comparison, ZEBC is tested against FS, FFT-based and ZBPC. FS deploys incremental calculation schemes to efficiently compute the candidate norms and mean values in (2.48). ZBPC parameter $\frac{n_{ZBPC}}{N}$ was optimally tuned to 0.07. As for FFT, we used the implementation proposed in well-known OpenCV library, optimized with SIMD instructions. For what regards ZEBC, parameter r was set to 8. In addition, for fairness of comparison, when tested against the FFT also for ZEBC a SIMD optimization is used. Finally, the sampling factor k used for the initial multi-resolution scheme [37] employed by both ZBPC and ZEBC was set to 4.

Table 2.11 reports the speed-ups (ratios of measured execution times) of the ZEBC algorithm against, from left to right, FS, ZBPC and FFT, obtained on a PC running Linux with 3.06 GHz clock AMD CPU. The last row of the table reports the mean speed-up reported by ZEBC against the three algorithms. From the table it can be noted that ZEBC is always able to notably speed-up the FS algorithm, speed-ups ranging between 11.5 and 28.5. Moreover, ZEBC is always faster than ZBPC and FFT, mean speed-ups being respectively 2.1 and 2.5. It is also worth noting that ZEBC, despite being a data-dependent technique, showed a rather limited range of variations of the measured speed-ups.

In addition, we also propose a further experiment where synthetic illumination distortions are applied between images and templates. In particular, all images are transformed according to an affine mapping function ($gain = 0.25$, $bias = -20$), as shown in Fig. 2.17, images $L1 \cdots L5$. This is motivated by the fact that ZNCC is typically employed in those cases where photometric distortions which can be assimilated to affine illumination changes are present between image and template, since ZNCC is invariant to this kind of transformations. Table 2.12 shows the speed-ups reported with this dataset by ZEBC against FS, ZBPC, FFT. By comparing the two tables it can be noted that, despite the notable distortions affecting the images, ZEBC is always the fastest algorithm, its speed-ups being lightly affected by the introduced distortions. From a theoretical point of view this can be explained since the effectiveness of all bounding functions applied by ZEBC, though not demonstrated here for lack of space, is robust to the presence of constant multiplicative and additive factors within I and T. Hence the decrease of speed-ups between the two tables has to be mainly ascribed to the distortions due to intensity quantization and saturation arising when such kind of synthetic transformation is applied.

Chapter 3

Stereo correspondence

3.1 Introduction

Stereo vision represents one of the most active areas of computer vision. The problem of stereo correspondence can be formulated as follows: given a pair of images taken from two viewpoints with overlaps, we need to find for each point p_r on one image, i.e. the *reference* image, its correspondent p_t on the other image. By means of the *epipolar constraint* [126], it is possible to reduce the search space from the whole image area to one single line (i.e., going from two to one dimension). In particular, given p_r, it is possible to define two lines, called *conjugate epipolar lines*, one on each image: on one line lies p_r, while on the other one lies p_t.

If the stereo setup is calibrated, it is possible to apply two homographies, one for each image, that allow each pair of conjugate epipolar lines to be collinear [126]. This process, called *rectification*, simplifies the problem of stereo correspondence from an algorithmic point of view, since p_t now lies on the same vertical coordinate as p_r.

Once a correspondence between p_r and p_t has been determined, the coordinates of P, the 3D point whose projections on the two images are p_r and p_t, can be estimated via triangulation. In particular, by denoting as d, *disparity*, the difference between the horizontal coordinates of p_r and p_t:

$$d(p_r) = x_{p_r} - x_{p_t} \tag{3.1}$$

the z coordinate (depth) of P can found as:

$$z(P) = \frac{b \cdot f}{d(p_r)} \tag{3.2}$$

with b the *baseline* (distance between the views) and f the focal length. For further details see e.g. [126].

Generally speaking, stereo correspondence can be *feature-based* or *dense*. In the first case, depth is estimated only for salient points of the reference image such as segments, edges, corners. The resulting techniques are robust and efficient, but do not yield depth at all points of the image. Conversely, dense techniques try to find correspondences for all points of the reference image, tend to be computationally more expensive and tend to fail along depth borders and low-textured areas. For further details, see, e.g., [126]. This chapter will deal with dense stereo algorithms.

A notable number of approaches has been proposed in the last years to attack the problem of stereo correspondence. Dense stereo techniques are currently divided [112] into two main categories: local approaches and global approaches. Local algorithms [130], [45], [141], [34], [49], [14], [100], [60], [32], [20], [138] are traditionally characterized by efficient and simple approaches. Despite being able of achieving real-time frame rate performance [34], as previously said they typically fail on low-textured areas as well as along depth borders and over occluded regions. In order to increase the accuracy of disparity estimations, particularly along depth borders, state-of-the-art algorithms deploy a *variable support* to compute the local matching cost rather than using, as in the traditional approach, a fixed squared window. Some proposed approaches exploit segmentation and excellent results have been recently obtained on the Middlebury [92] dataset (the reference dataset for testing effectiveness of stereo algorithms) although these approaches are not currently suited for real-time applications.

Conversely, most global methods [120], [62], [135], [148], [70], [13], [71], [121], [122], [142], [139] attempt to minimize an energy functional computed on the whole image area by mean of a Pairwise Markov Random Field model (P-MRF). Since this task turns out to be a NP-hard problem, approximate but efficient strategies such as Graph Cuts (GC) and Belief Propagation (BP) have been proposed. Currently, global approaches employing segmentation provide the most accurate results on the Middlebury dataset. Global approaches are computational demanding and hence currently not suitable for real-time application. Nevertheless, a promising framework for efficient energy minimization was recently proposed [40].

Finally, a third category of methods [58], [51], [78], [33] which lies in between local and global approaches refers to those techniques based on the minimization of an energy function computed over a subset of the whole image area, i.e. typically along epipolar lines or scanlines. The adopted minimization strategy is usually based on the Dynamic Programming (DP) or Scanline Optimization (SO) techniques. The global energy function to be minimized includes a pointwise matching cost and a smoothness term which enforces constant disparity e.g. on untextured regions by means of a discontinuity penalty. These approaches provide a trade-off between accuracy and computational requirements.

3.2 A classification of variable support methods

This section focuses on stereo cost aggregation techniques, and addresses methods that perform cost aggregation on a *variable support*. The basic local stereo algorithms, referred to as *Fixed Window*, relies on rectangular windows to determine the correspondence for each point p_r on the reference image. In particular, given p_r and a set of candidate points that lie on the same line as p_r on the other image and within a *disparity range* $D = [d_{min}, d_{max}]$, a matching cost is computed on a window centered on p_r and on each candidate. Then, the candidate reporting the minimum cost value represents the correspondence for point p_r.

The idea at the basis of the variable support concept is to determine the best set of pixels on which to compute the matching cost (i.e. the *support*) at each pair of candidates under evaluation (i.e. the *correspondence*). Hence, unlike the basic approach that relies on a fixed static support, these methods deploy a support which varies along the potential correspondences in order to adapt itself to the local characteristics of each correspondence. This allows for obtaining higher accuracy along depth borders and lower matching ambiguity, especially within low textured regions.

Although works dealing with cost aggregation on a variable support date back to the 70s, 80s and early 90s [4,44,80,100], only in the last years a broad research activity has provided effective ideas allowing local algorithms based on a variable support to yield an accuracy comparable to that of many global methods. Moreover, though typically performed by local algorithms, cost aggregation on a variable support proved to be very effective in improving the performance of global algorithms such as based on Belief Propagation (BP) [142], Dynamic Programming (DP) [133], Scanline Optimization (SO) [90].

We believe that the variety of approaches, as well as the excellent results achieved, deserve a specific classification, highlighting similarities and differences between the main cost aggregation strategies, together with a comparative performance evaluation of the different methods. Recent surveys on stereo matching [17,50,61,112] do not address the above topics since they consider the whole class of stereo methods [112], review advances in computational stereo with particular emphasis on occlusion detection and real-time methods [17], focus on matching functions robust to photometric distortions and noise [61] or address only those cost aggregation methods that are suited to real-time implementation on a GPU [50]. Differently, the work proposed in this section (and, succesively, in Section 3.5 is specifically focused on classifying the main variable support-based cost aggregation strategies and comparing them experimentally within a plain vanilla *Winner-Take-All* (WTA) framework.

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

As for the matching (or error) function employed, this is typically based on the L_p distance between the two vectors representing the supports in the stereo images, such as the *Sum of Absolute Differences* (SAD) or *Sum of Squared Differences* (SSD). Often M-estimators are used to achieve better robustness toward outliers. The basic one simply truncates the values of the matching measure up to a threshold (e.g. *Truncated SAD* [124, 141]), while sometimes other more complex M-estimators are used [45]. Another popular solution regards the use of a measure insensitive to image sampling [12] (e.g. used in [130]). Moreover, a promising similarity function based on point distinctiveness has been recently proposed in [143].

Finally, it is worth pointing out that this work addresses aggregation strategies based on fronto-parallel variable supports only, thus not considering proposals that account for three-dimensional supports (e.g. [147]).

3.2.1 Cost aggregation based on rectangular windows

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

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

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

Varying window size and/or offset

One of the first algorithms exploiting the idea of using a set of windows to improve the accuracy of stereo correspondence is *Shiftable windows* [112]. In this case, the set of windows $S(p, q)$ is defined as:

$$S(p, q) = \{W_n(i, j, d) : i \in [x - n, x + n], j \in [y - n, y + n]\} \tag{3.3}$$

where n is a parameter of the algorithm representing the chosen window size. The support at each correspondence, $S_V(p, q)$, is given by the window minimizing the error function over $S(p, q)$. This approach is useful along depth borders, since it aims at determining the most appropriate displacement with respect to p on which to center the window in order to aggregate points lying at the same depth plane as p. A variation of this basic strategy concerns including in $S(p, q)$ only 9 squared windows in symmetrical positions with respect to the central point [14, 42].

An alternative approach [1, 100] is to vary the size of the window rather than its displacement by properly selecting n between a minimum value N_{min} and a maximum value N_{max}:

$$S(p, q) = \{W_n(x, y, d) : n \in [N_{min}, N_{max}]\} \tag{3.4}$$

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

These schemes can be generalized by selecting the best support between a set of window pairs having different sizes and different displacements. In [67] the best displacement is selected by means of a shiftable window approach, while, to determine the size of the support, starting from $n = N_{min}$ the window is iteratively enlarged until a given minimum variance of the error function is reached.

A slightly more general approach is represented by the method proposed in [130], which selects as support the window minimizing a matching cost over a set of windows $S(p, q)$ defined as:

$$\begin{aligned} S(p, q) = \{W_n(i, j, d) : n \in [N_{min}, N_{max}], \\ i \in [x - n, x + n], j \in [y - n, y + n]\} \end{aligned} \tag{3.5}$$

The 3 criteria on which the matching cost is based include minimization of the error function and its variance, plus the use of a biasing weight to favor the choice of large windows within low-textured regions, where the error function and its variance might not vary significantly along the evaluated window sizes. Moreover, this method explicitly proposes an incremental scheme aimed at efficiently computing (3.5) at each new correspondence.

An analogous approach was proposed in [20]. As for this method, the best displacement is selected out of 9 using the shiftable windows approach. Then, the window size is iteratively decreased until either the error function gets worse or the minimum window size is reached.

In [32] the displacements considered at each correspondence are 4, disposed on the four window corners. As for the window size, starting from an initial value, the window horizontal and vertical sides are iteratively increased until either the error variance on a direction gets higher than a certain threshold or the error function gets worse. Differently from previous approaches, this allows to obtain rectangular supports.

Selecting more than one window

All previous schemes select, for each correspondence, one window on each image representing the best support over $S(p,q)$. A generalization of this approach is represented by $S_V(p,q)$ being not one single window pair, but a subset of window pairs. In [101], $S(p,q)$ is the same as in (3.3) and the outcome of the error measure computation on the various window pairs is used to assess whether each point is close or not to a depth edge. Based on that, a variable support strategy is deployed on all points detected as close to a depth edge, where the final matching cost assigned to each correspondence is obtained by averaging the error function along those displacement positions detected as lying on the same border side as p and q.

In one version of method [60], $S(p,q)$ is defined as a set of 5 squared windows

$$S(p,q) = W_n(x,y,d) \cup \{W_n(x \pm n, y \pm n, d)\} \tag{3.6}$$

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

Associating different weights to window points

It is worth pointing out that with the methods described in Section 3.2.1 the resulting shape of the support is no longer constrained to a rectangle. Moreover, getting a support made out of several partially overlapping windows having uniform weights is equivalent to getting a support made out of a single rectangular window where each point is weighted differently. This latter strategy concerns the explicit assignment of different weights to the points of each window belonging to $S(p,q)$. In method [69], the aggregation stage defines $S(p,q)$ as a set of 108 rod-shaped windows. Each window is characterized by a specific orientation and weight set, and the support at each correspondence is determined by the window minimizing the error function. Each point is then classified as *homogeneous* or *heterogeneous* based on the outcome of the application of a LoG filter, and on those points denoted as homogeneous the minimum error score determined on $S(p,q)$ is also compared to that yielded by a basic shiftable windows approach.

In the strategy proposed in [49], $S(p,q)$ is defined as a set of 5×5 window pairs centered on (p,q), each window point being characterized by a weight belonging to the set $\{0,1,2,4\}$. For each correspondence a window pair is selected according to the local structure of I_r, which is extracted by means of either edge detection or segmentation. The authors propose also to iterate the process k times, and suggest to use $k = 4$ or $k = 2$ respectively in a local or global framework.

3.2.2 Cost aggregation based on unconstrained shapes

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

The first method exploiting this idea was proposed in [15]. At each correspondence (p,q), each pixel p_i on I_r is classified either as plausible or not-plausible based on an estimation of the photometric relation between p_i and its correspondent on I_q at the same disparity as (p,q). The best disparity for p is simply selected as that yielding the largest set of connected plausible pixels. This allows to have variable supports which can ideally extend to all pixels of the image. Differently, in [129] the support shape at each correspondence is represented by a polygonal line around p, which is extracted by applying the *minimum ratio cycle* technique.

Finally, in [45] the support shape is represented by the intersection between the segment on which p lies, G_p, and a squared window centered on p, $w_n^r(x,y)$. This approach is intrinsically based on the assumption, introduced by [122], that disparity is constant over each segment obtained from a segmentation process. Moreover, this approach relies only on segmentation information concerning I_r (i.e. it is not symmetrical). Those points belonging to $w_n^r(x,y)$ and not to G_p are included in the error function by means of a small constant weight.

3.2.3 Use of adaptive weights

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

The method proposed in [138] can be regarded as the first proposal for stereo exploiting this idea. It was inspired by [27], which proposed a method to segment a foreground object from its background in an image based on the radial propagation of similarity starting from a foreground point. In [138] 3 different cues are deployed to determine the support weights for points belonging to the reference view I_r. The first one (the *certainty*) is based on the variance of the error function: since weights are propagated radially starting from p, each point weight depends from the error variance of previous points along its ray. With increasing variances, the assigned weight is lower since it corresponds to a low certainty. The two other cues are color and disparity distribution correlation: the weight assigned to a point p_i increases as the difference in the color space between p_i and p decreases and as the correlation between p_i and p disparity distribution increases. Each cue is weighted by means of a gaussian function in the final weight formulation, the 3 gaussian variances being 3 parameters of the method.

In [141] this approach is enhanced by symmetrically extending the weight computation to points on I_t. Weights are computed based on the two cues of distance in the color space and distance in the coordinate space (proximity) by means of gaussian functions, this approach being motivated by the Gestalt principles of similarity. Then, a final weighted error function is proposed including normalization by means of the weight coefficients. Points farther than a certain distance from p and q are not evaluated. A more detailed analysis of the method proposed in [141] is presented in the next section. An efficient though simplified asymmetric version of this method is proposed within a Dynamic Programming framework in [133], so to allow a GPU implementation with real-time capabilities.

3.3 Accurate stereo based on adaptive support and segmentation

This section proposes a novel adaptive support aggregation strategy which deploys segmentation information in order to increase the reliability of the stereo matches. The proposed approach aims at improving the method proposed in [141], that, as will be illustrated, tends to produce errors in presence of highly textured regions, where the support can shrink to a few pixels thus dramatically reducing the reliability of the matches. Unreliable matches can be found also near depth discontinuities, as well as in presence of low textured regions and repetitive patterns. Compared to the classification of local stereo algorithms based on a variable support presented in the previous section, the proposed approach falls into the class of methods that are based on adaptive weights.

3.3.1 Previous work

The basic idea of [141] is to extract an adaptive support for each possible correspondence by assigning a weight to each pixel which falls into the current correlation window W_r in the reference image and, correspondingly, in the correlation window W_t in the target image. Let p_c and q_c being respectively the central points of W_r and W_t, whose correspondence is being evaluated. Thus, the pointwise score, which is selected as the Truncated Absolute Difference (TAD), for any point $p_i \in W_r$ corresponding to $q_i \in W_t$ is weighted by a coefficient $w_r(p_i, p_c)$ and a coefficient $w_t(q_i, q_c)$, so that the total cost for correspondence (p_c, q_c) is given by summing up all the weighted pointwise scores belonging to the correlation windows and normalized by the weights sum:

$$C(p_c, q_c) = \frac{\sum\limits_{p_i \in W_r, q_i \in W_t} w_r(p_i, p_c) \cdot w_t(q_i, q_c) \cdot TAD(p_i, q_i)}{\sum\limits_{p_i \in W_r, q_i \in W_t} w_r(p_i, p_c) \cdot w_t(q_i, q_c)} \tag{3.7}$$

Each point in the window is weighted on the basis of its spatial distance as well as of its distance in the CIELAB colour space with regards to the central point of the window. Hence, each weight $w_r(p_i, p_c)$ for points in W_r (and similarly each weight $w_t(q_i, q_c)$ for points in W_t) is defined as:

$$w_r(p_i, p_c) = exp\left(-\frac{d_p(p_i, p_c)}{\gamma_p} - \frac{d_c\left(I_r(p_i), I_r(p_c)\right)}{\gamma_c}\right) \tag{3.8}$$

where d_c and d_p are respectively the euclidean distance between two CIELAB triplets and the euclidean distance between two coordinate pairs, and the constants γ_c, γ_p are two parameters of the algorithm.

This method provides excellent results but has also some drawbacks, which will be highlighted in the following by analysing the results obtained by [141][1] on stereo pairs belonging to the Middlebury dataset and shown in Fig. 3.1.

Depth discontinuities The idea of a variable support is mainly motivated by depth discontinuities: in order to detect accurately depth borders, the support should separate ”good” pixels, i.e. pixels at the same disparity as the central point, from ”bad” pixels, i.e. pixels at a different disparity from the central point. It is easy to understand that within these regions the concept of spatial distance is prone to lead to wrong separations, as due to their definition border points always have close-by pixels belonging to different depths. Therefore ”bad” pixels close to the central point might receive higher weights than ”good” ones far from the central point, this effect being more significant the more the chromatic similarities between the regions at different disparities increase. Moreover, as for ”good” pixels, far ones might receive a significantly smaller weight than close ones while ideally one should try to aggregate as many ”good” pixels as possible. Generally speaking, weights based on spatial proximity from the central point are constant for each correlation window, hence drive toward fixed - not anymore variable - supports, with all negatives consequences of such an approach.

Fig. 3.2 shows a typical case where the use of spatial distance would determine wrongly the correct support. Imagine that the current point (the blue point in figure) is on the border of two planes at different depths and characterized by a slightly different colour or brightness. The central image shows the correlated pixels (circles coloured from red - high correlation - to yellow - low correlation) on the basis of spatial proximity, where it can be seen that many ”bad” pixels would receive a high weight because of the close spatial distance from the central point. Right image depicts in red the correct support that should be ideally extracted. This effect leads to mismatches on some depth borders of the *Tsukuba* and *Venus* datasets, as indicated by the blue boxes of Fig. 3.1 (groundtruth is shown in Fig. 3.6).

Low textured surfaces A further drawback of [141] deals with matching ambiguities which apply when trying to match points belonging to low textured areas on constant depths. When considering the correspondence of points on these areas, the support should ideally enlarge itself as much as possible in order to maximize the signal-to-noise ratio. Instead, the combined use of the spatial and colour proximities force the

[1] The results shown in this paper were obtained running the authors' code available at: *http://cat.middlebury.edu/stereo/code.html*.

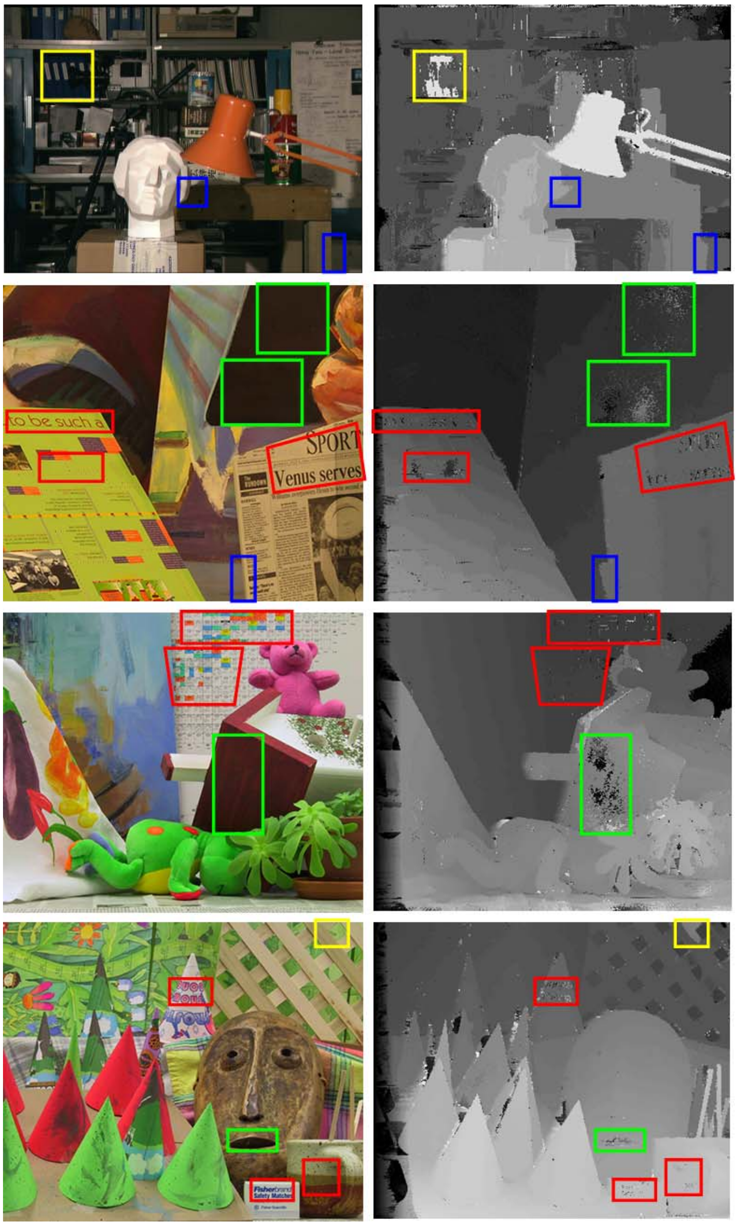


Figure 3.1: Some typical artifacts caused by the cost function adopted by [141] on high textured regions (red), depth discontinuities (blue), low textured regions (green), repetitive patterns (yellow). [This image is best viewed with colors]

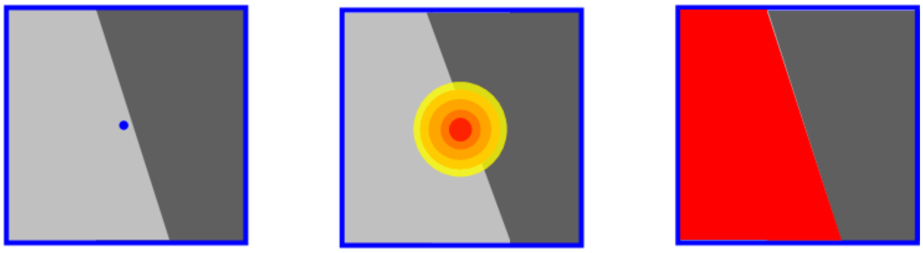


Figure 3.2: Example of a correlation window along depth borders (left), correspondent weights assigned by [141] on the basis of spatial proximity (center) and ideal support (right).[This image is best viewed with colors].

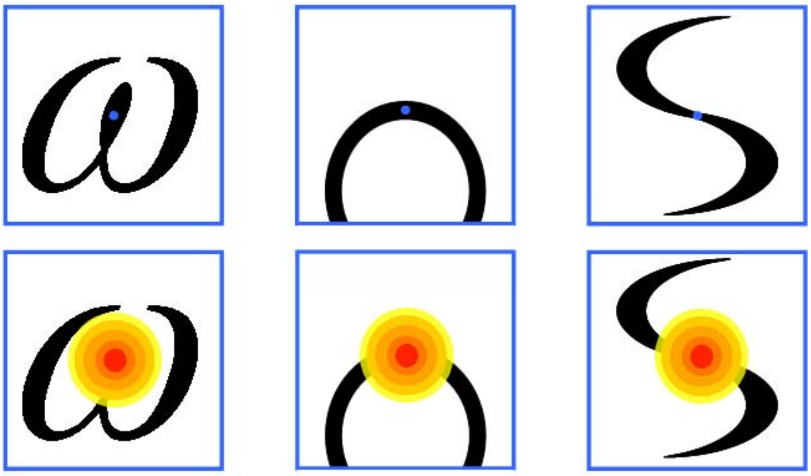


Figure 3.3: Examples where the support shrinks to a few elements due to the combined use of spatial and colour proximity. The coloured circles indicate the region correlated to the central pixels on the basis of the spatial proximity.

support to be smaller than the correlation window. This effect is particularly evident in datasets *Venus*, *Cones* and *Teddy*, where the low textured regions denoted by the green boxes of Fig. 3.1 lead to remarkable artifacts in the correspondent disparity map.

High textured surfaces Suppose to have a high textured region laying on a constant disparity plane. Then, for all those points having not enough chromatic similarities in their surroundings the aggregated support tends to reduce to a very small number

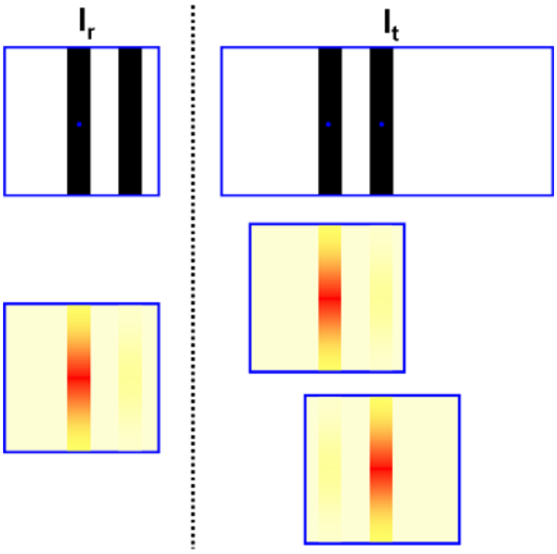


Figure 3.4: Typical example of a repetitive pattern along epipolar lines where the aggregation step of [141] would lead to ambiguous match. Red-to-yellow colours are proportional to the weights assigned to the supports.

of points. This effect is due to the weights decreasing exponentially with the spatial and colour distances, and it tends to reduce notably the robustness of the matching as the support tends to become pointwise. It is important to note that in these situations the support should ideally enlarge itself and aggregate many elements in the window because of the constant depth.

In order to have an idea of the behaviour of the aggregated support, consider the situation of Fig. 3.3, where some particular shapes are depicted. In the upper row, the blue point represents the current element for which the support aggregation is computed and the blue square represents the window whose elements concur in the computation of the support. In the lower row the coloured circles denote the points correlated to the central point on the basis of the spatial proximity criterion, where red corresponds to high correlation and yellow to low correlation. As it can be clearly seen the combined use of spatial and colour proximity would lead in these cases to very small aggregated supports compared to the whole area of the shapes as well as to the correlation window area.

Typical artifacts induced by this circumstance are evident in datasets *Venus*, *Cones* and *Teddy* as highlighted by the red boxes in Fig. 3.1, where it is easy to see that they are often induced by the presence of coloured writings on objects in the scene and that they produce notable mistakes in the correspondent regions of the disparity maps.

Repetitive patterns Finally, a further problem due to the use of the weight function (3.7) applies in presence of repetitive patterns along the epipolar lines. As an example consider the situation depicted in Fig. 3.4. In this case, the blue point in top left image has to be matched with two candidates at different disparities, centered on two similar patterns and shown in top right image. In this situation, the combined use of spatial and colour proximities in the weight function would extract supports similar to the ones shown in the bottom part of the figure, where red corresponds to high weight values and yellow to low weight values. It is easy to see that the pixels belonging to both candidate supports are similar to the reference support, hence would lead to an ambiguous match. This would not happen, e.g., with the use of the common fixed square support which includes the whole pattern.

In Fig. 3.1 a typical case of a repetitive pattern along epipolar lines is shown by the yellow box in dataset *Tsukuba*, which lead to mismatches in the disparity map. Also the case depicted by the yellow box in dataset *Cones* seems due to a similar situation.

3.3.2 Proposed approach

The basic idea beyond our approach is to employ information obtained from the application of segmentation within the weight cost function in order to increase the robustness of the matching process. Several methods have been recently proposed based on the hypothesis that disparity varies smoothly on each segment yielded by an (over-)segmentation process applied on the reference image [45], [70], [13]. As the cost function (3.7) used to determine the aggregated support is symmetrical, i.e. it computes weights based on the same criteria on both images, we propose to apply segmentation on both images and to include in the cost function the resulting information. The use of segmentation allows for including in the aggregation stage also information dealing with the connectiveness of pixels and the shape of the segments, rather than only relying blindly on colour and proximity. Because our initial hypothesis is that each pixel lying on the same segment of the central pixel of the correlation window must have a similar disparity value, then its weight has to be equal to the maximum value of the range(i.e. 1.0). Hence we propose a modified weight function as follows:

$$w'_r(p_i, p_c) = \begin{cases} 1.0 & p_i \in S_c \\ exp\left(-\frac{d_c(I_r(p_i), I_r(p_c))}{\gamma_c}\right) & otherwise \end{cases} \tag{3.9}$$

with S_c being the segment on which p_c lies. It is important to note that for all pixels outside segment S_c, the proximity term has been eliminated from the overall weight computation and all pixels belonging to the correlation window have the same importance independently from their distance from the central point, because of the negative drawbacks of the use of such a criterion shown in the previous section. Instead, the use of segmentation plays the role of an intelligent proximity criterion.

It is easy to see that this method is less subject to the negative aspects of method [141] outlined in the previous section. The problem of having very small supports in presence of shapes such as the ones depicted in Fig. 3.3 is improved by segmentation. In fact, as segmentation allows segments to grow as long as chromatic similarity is assessed, the aggregated supports extracted by proposed approach are likely to correctly coincide with the shapes depicted in the figure. Moreover, the use of segmentation in spite of the spatial proximity would allow to extract correctly the support also for border points such as the situation described in Fig. 3.2, with the extracted support tending to coincide with the one shown on the right of that figure. Improvements are yielded also in presence of low textured areas: as they tend to correspond to a single segment because of the low texture, the support correctly enlarges to include all points of these regions. Finally, in presence of repetitive patterns such as the ones shown in Fig. 3.4 the exclusion of the spatial proximity from the weights computation allows only the correct candidate to have a support similar to the one of the reference point.

Moreover, from experimental results it was found that the use of a colour space such as the CIELAB helps the aggregation of pixels which are distant chromatically but which are closer in the sense of the colour space. Unfortunately this renders the colour distance measure less selective, and tends to produce more errors along depth discontinuities.Conversely, the use of the RGB colour space appeared more picky, decreasing the chance that pixels belonging to different depths are aggregated in the same support, but also increasing the number of artifacts along textured regions which lie at the same depth. As the use of segmentation implies adding robustness to the support, we found more convenient to operate in the RGB space in order to enforce smoothness over textured planes as well as to increase the accuracy of depth borders localization.

Finally, it is worth pointing out that there are two main differences between our method and that proposed in [45]: first we apply segmentation on both reference and target images, hence the support aggregation strategy is symmetric. Besides, rather than using two constant weights, we exploit the concept of colour proximity with all benefits of such an approach shown in [141].

3.3.3 Experimental results

In this section we present some experimental results of the proposed method. First we compare our results on the Middlebury dataset with those yielded by [141] using a *Winner-Take-All* (WTA) strategy. The parameter set is kept constant for all image pairs: the set used for the algorithm by Yoon and Kweon is the one proposed in the experimental results in [141], while the set used for the proposed approach is: γ_c =

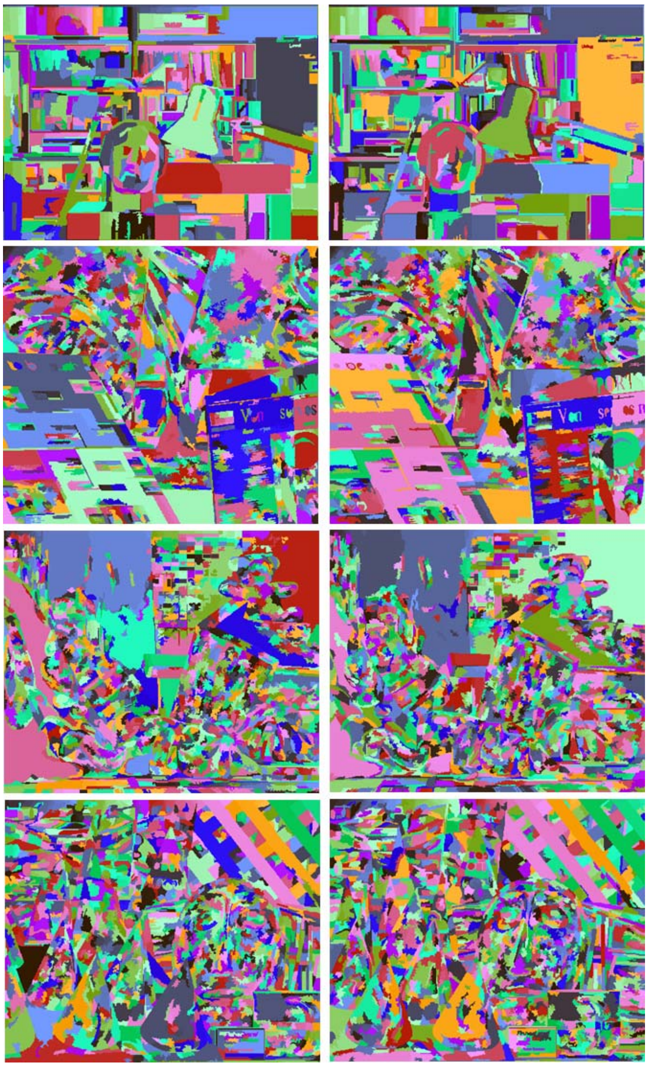


Figure 3.5: Output of the segmentation stage on the 4 stereo pairs of the Middlebury dataset.

22.0, window size = 51 × 51, T (parameter for TAD) = 80. For what means the segmentation step in the proposed approach, we use the *Mean-Shift* algorithm [21] with the same constant parameter set, that is: $\sigma_S = 3$ (spatial radius), $\sigma_R = 3$ (range radius), $min_R = 35$ (minimum region size). Figure 3.5 shows the output of the segmentation stage on both images of each of the 4 stereo pairs used for testing.

Fig. 3.6 compares the disparity maps obtained by [141] with the proposed approach. Significant improvements can be clearly noticed since the artifacts highlighted in Fig. 3.1 are less evident or no longer present. In particular, errors within the consid-

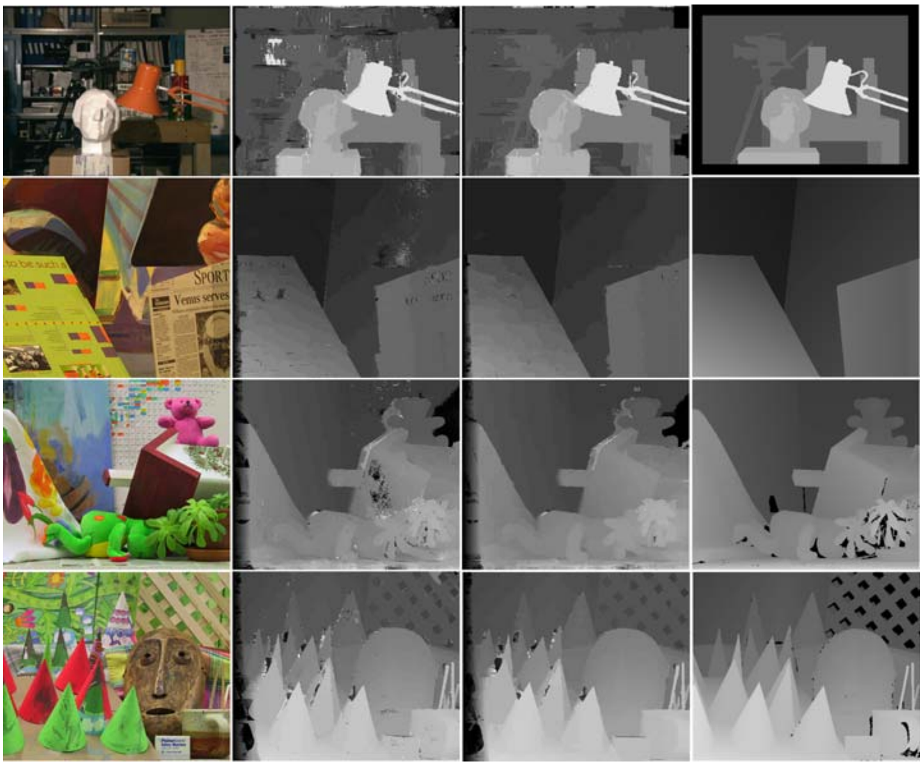


Figure 3.6: Reference images (first column), disparity maps computed by [141] (second column) and our approach (third column), ground truth (last column).

Table 3.1: Comparison between proposed approach and method [141] on the *Middlebury* dataset using a WTA strategy.

	Tsukuba	Venus	Teddy	Cones
	N.O. - DISC	N.O. - DISC	N.O. - DISC	N.O. - DISC
Proposed	2,05 - 7,14	1,47 - 10,5	10,8 - 21,7	5,08 - 12,5
[141]	4.66 - 8.25	4.61 - 13.3	12.7 - 22.4	5.50 - 11.9

ered high textured regions on *Venus* and *Teddy* are greatly reduced and almost disappear on *Cones*. Accuracy along depth borders of *Tsukuba* is significantly enhanced while the error along the depth border in *Venus* shrinks to the true occluded area. Moreover, highlighted artifacts present on low textured regions notably decrease on *Venus* and disappear on *Teddy* and *Cones*. Finally, also the artifacts due to the presence of repetitive patterns as shown on *Tsukuba* and *Cones* definitely disappear.

In addition, Table 3.1 shows the error percentages with regards to the groundtruth,

Table 3.2: Disparity error rates and rankings obtained on Middlebury website by the proposed approach (referred to as *SegmentSupport*) compared to method [141] (referred to as *AdaptWeight*) and (where available) [45].

	Rank	Tsukuba	Venus	Teddy	Cones
		N.O. - ALL - DISC	N.O. - ALL - DISC	N.O. - ALL - DISC	N.O. - ALL - DISC
SegmentSupport	♯ 9	1.25-1.62-6.68	0.25-0.64-2.59	8.43-14.2-18.2	3.77-9.87-9.77
AdaptWeight	♯ 13	1.38-1.85-6.90	0.71-1.19-6.13	7.88-13.3-18.6	3.97-9.79-8.26
[45]	n.a.	n.a.-2.27-n.a.	n.a.-1.22-n.a.	n.a.-19.4-n.a.	n.a.-17.4-n.a.

with the error threshold set to 1, computed on the maps of Fig. 3.6. For each image pair two error measures are proposed: the former is relative to all image area except for occlusions (*N.O.*), the latter only to discontinuities except for occlusions (*DISC*). The error on all image area including occlusions has not been reported because occlusions are not handled by WTA strategy. As it can be seen from the table, the use of the proposed approach yields notable improvements for what concerns the error measure on all *N.O.* area. Moreover, by looking only at discontinuities, we can see that generally the proposed approach allows for a reduction of the error rate (all cases except for *Cones*). Benefits are mostly evident on *Venus* and *Tsukuba*.

Finally, we show the results obtained by our method after application of the *Left-Right* consistency check and interpolation of those points which were determined as inconsistent. The obtained disparity maps were submitted and are available at the Middlebury website. We report, in Tab. 3.2, the quantitative results of our method (referred to as *SegmentSupport*) compared to the submitted results of method [141] (referred to as *AdaptWeight*), together with the overall ranking assigned by Middlebury to the two approaches. The table reports also the results published in [45] which consist only of the error rates on the *ALL* groundtruth maps (all image area including occlusions), since no submission has been done so far on Middlebury. As it is clear from the table and the Middlebury website, currently our approach is the best performing known local method ranking 9th overall (as of July 2007).

3.4 Accurate near-real time stereo

The idea which motivates the work presented in this section relates to a novel aggregation strategy deploying segmentation aiming at high efficiency and at the same time as accurate as to improve the results of fast local stereo algorithms. This lead us to devise a method which improves significantly the performance-cost trade-off, yielding a level of accuracy comparable to that of segmentation-based methods and capable to meet near-real time processing requirements.

It is interesting to note that recently many stereo matching algorithm relying on image segmentation and aimed at improved accuracy have been proposed [13, 45, 59, 62, 70, 120, 122, 124, 135, 139, 148]. The great majority of these methods are global, and a subset of them [13, 70, 120, 139] represents currently the most accurate methods on the Middlebury Stereo Evaluation website [2], which is the standard benchmark platform for the stereo community. Anyway, the computational burden they require is far from meeting real-time or near real-time requirements.

Local approaches that are state-of-the-art in terms of accuracy are based on segmentation (see previous section) or adaptive weights [141], but are far from being computationally efficient. Indeed, apart from GPU or hardware-based implementation, typically only aggregation strategies based on sets of rectangular windows [14, 34, 60, 130] can afford real-time or near-real-time processing, this implying a notably reduced accuracy of retrieved disparities. Exceptions are represented by methods [45, 49], whose aggregation strategies rely on segmentation and that exhibit interesting trade-offs between accuracy and computational efficiency. Moreover, between those methods for which a GPU implementation has been proposed [50], no one so far deploys segmentation.

3.4.1 Cost aggregation strategy

Let I_r and I_t be respectively a reference and a target image of a stereo pair, and let $p \in I_r, q \in I_t$ be a pair of points at disparity d for which correspondence is being evaluated. The proposed aggregation scheme deploys a *variable support*, that is at each correspondence (p, q) the set of points around p and q on which the local similarity measure (or local cost) is computed depends on the local characteristics of the images. Similarly to most stereo matching algorithms deploying segmentation, the proposed aggregation strategy relies on the assumption that disparity varies smoothly within points lying on the same segment (this is true in practice especially if images are over-segmented). Thus, the idea is to shape the variable support at each correspondence based on information derived from image color segmentation. This is achieved by computing for each correspondence (p, q) at disparity d an aggregation cost defined as:

$$C_s(p, q, d) = \sum_{p_i \in S_p} min\left(\delta\left(p_i, q_{i,d}\right), T_r\right) \tag{3.10}$$

where S_p is the segment on which p lies, $\delta(p, q)$ is the computationally efficient L_1 distance between the RGB components of p and q:

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

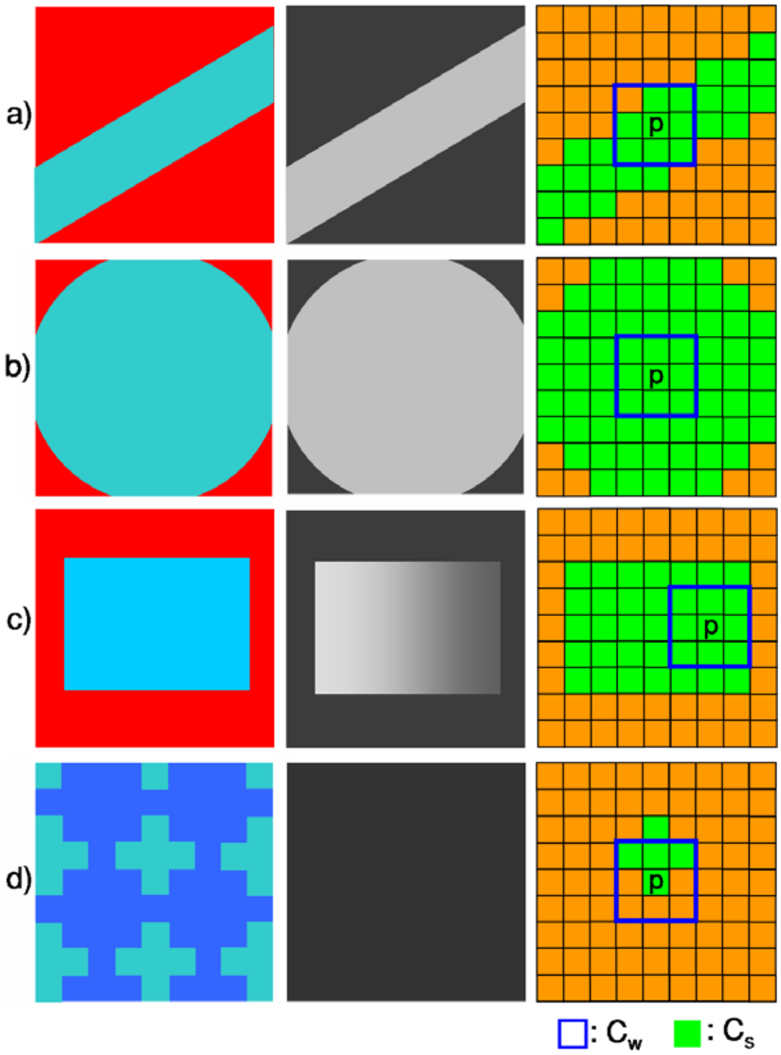


Figure 3.7: Examples of the behaviour of the proposed aggregation cost

$$\delta(p, q) = |R_p - R_q| + |G_p - G_q| + |B_p - B_q| \tag{3.11}$$

and T_r is a fixed threshold. In practice, C_s represents the sum of the truncated absolute differences (TAD) over the segment on which p lies. The use of the truncation value T_r is a very basic M-estimator to enhance robustness toward outliers (in our experiments, T_r is set to 35).

C_s can be efficiently pre-computed by means of a single image scan for each possible disparity within the disparity range. Moreover, it tends to be notably accurate along depth borders since disparity edges tend to coincide with color edges on real images. Furthermore, within low-textured regions segments tends to be very big, which results in a high SNR and hence good robustness of C_s toward matching ambiguities. However, relying only on the segmentation cue might lead to mistakes, since C_s tends to assign the same disparity value to all points belonging to the same segment. This leads to mistakes for those points lying at slightly different depths from the majority of elements of a segment, e.g. on slanted surfaces. Furthermore, it also tends to decrease matching distinctiveness along highly-textured regions, where segments tend to be particularly small. Hence, we modify (3.10) to include also a corrective term based on a squared correlation window:

$$C_{aggr}(p, q, d) = \frac{C_s(p, q, d)}{n(S_p)} + \alpha \cdot \frac{C_w(p, q, d)}{(2r + 1)^2} \tag{3.12}$$

where C_w is the TAD over the squared window $W_p(r)$ of radius r and centered on p:

$$C_w(p, q, d) = \sum_{p_i \in W_p(r)} min\left(\delta\left(p_i, q_{i,d}\right), T_r\right) \tag{3.13}$$

Cost C_{aggr} includes a normalization of the two terms C_s, C_w by the total number of points in, respectively, S_p and W_p. This is useful because, while the area of $W_p(r)$ is fixed, the number of points in each segment, $n(S_p)$, varies with p: thus, the normalization stage allows to weight equally each pixel included in C_{aggr}. It is important to point out that, thanks to the use of incremental schemes [24, 91] the complexity of the calculation of term C_w amounts to only 4 elementary operations for each point and disparity, and it is independent on the choice of parameter r. Overall this results in a particularly efficient aggregation strategy.

Fig. 3.7 depicts graphically the behaviour of the proposed aggregation strategy in 4 different cases. In the figure, the first column shows the reference colour image, the second column shows the expected disparity map and the third column illustrates the behaviour of the proposed aggregation strategy. In particular, cost C_s assures that the variable support is shaped according to local chromatic cues. This is particularly useful along depth borders (case a) and within low-textured regions (b). Cost C_w, instead, adds a further weight for those points that are close to p (i.e. spatially more correlated). Generally the role of cost C_w is to increase the robustness of term C_s for those points violating the segmentation assumption, e.g. for bordering regions along slanted surfaces (case c). In addition, it is particularly effective along highly-textured regions (case d), where segments tend to reduce to a few pixels.

3.4.2 Further comments

The proposed aggregation strategy bears some resemblances with that proposed in [45], where for each correspondence (p, q) the variable support is defined as the intersection between the points lying on the same segment as p and those belonging to the current correlation window. Nevertheless, if the working assumption that disparity varies smoothly within points lying on the same segment is verified, then the use of all points lying on S_p, rather than just those included in the current correlation window shall yield improved matching robustness and thus less ambiguity. Moreover, to avoid matching ambiguities due to few intersection points, method [45] requires the use of big correlation windows and the inclusion in the local cost with a smaller weight also of the remaining points in the window, which tends to increase inaccuracy. Furthermore, the efficient incremental implementation of the aggregation strategy proposed in [45] sacrifices accuracy for speed and tends to deteriorate the accuracy of the results. Conversely, our proposal can be directly implemented in an efficient way without any loss in accuracy. This results in significant improvements in accuracy and speed, as shown in next section.

The proposed aggregation strategy might be usefully deployed either by a local algorithm or as the initial stage of a global process based on e.g. Scanline Optimization [59] or Belief Propagation [120]. Moreover, it is interesting to note that this aggregation strategy could be symmetrically extended to include information also from the color segmentation of the target image I_t, rather than only that from I_r. This is not investigated here for lack of space, but there are hints that it would result in improved accuracy and lower computational efficiency.

Finally, in our implementation we use Mean Shift [21] to perform segmentation. This method yields accurate segmentation but is not extremely fast: overall in our experiments it accounts for a percentage between 40 and 80% of the total time. As a consequence, the proposed method could be further speeded-up using a faster segmentation method.

3.4.3 Experimental results

This section presents a comparison between the proposed method and other state-of-the-art aggregation strategies. Methods are evaluated within the same plain WTA (Winner-Take-All) stereo matching framework. In particular, as a term of comparison we selected state-of-the-art efficient aggregation strategies based on variable support, that is, *Segmentation Based* [45], *Shiftable Windows* [14], *Variable Windows* [130], *Multiple Windows* [60]. For what regards this last method, the version based on 9 correlation window is used as representative of the best accuracy-speed trade-off [60].

Algorithm	**Accur.**	Tsukuba	Venus	Teddy	Cones	Art	Books	Dolls	Laundry	Moeb.	Reind.	**MDS**
Variable Wind.	86.7	96.23	91.99	87.4	94.34	80.81	80.04	87.22	76.68	87.29	84.63	0.3
Proposed	86.4	97.04	96.47	89.33	95.08	78.72	81	85.64	74.89	84.88	80.48	18.9
Segm. Based	83.3	94.3	93.92	90.35	92.69	76.22	79.86	84.75	61.7	81.09	77.75	5.9
Multiple Wind.	82.1	94.42	95.82	85.46	91.18	72.68	78.31	81.36	64.23	80.79	76.66	2.7
Grad. Guided	79.4	92.99	87.66	80.46	88.03	72.17	72.86	83.93	61.48	76.15	78.27	3.2
Shiftable Wind.	79.4	93.46	93.4	83.84	90.45	68.08	75.6	77.42	60.03	77.06	74.6	1.2

Table 3.3: Comparison of accuracy and MDS yielded by the proposed approach with respect to different state-of-the-art local stereo algorithms.

Methods [141], though being state-of-the-art in accuracy among local algorithms, have not been included in our comparison since this paper focuses on real-time or near real-time methods while these methods are far from compelling these requirements (e.g. on the same platform and on Teddy, author's code of [141] runs in 18 minutes against 0.6 seconds of our approach).

For fairness of comparison, algorithms do not employ neither pre-processing nor post-processing such as consistency check and interpolation. Moreover, the local cost function is for all methods the TAD on RGB values, except for *Segmentation based* which deploys the Sum of Absolute Differences on RGB values plus a more complex M-estimator, as originally proposed in [45]. For what concerns the choice of parameters, all parameter values of the algorithms were optimally tuned on the dataset. In particular, for the proposed method the two parameters of the aggregation stage were set as $\alpha = 0.9$, $r = 6$. Finally, all algorithms were implemented in *C*, without any kind of optimization based, e.g., on SIMD instructions and tested on Intel Core Duo 2.14 GHz CPU.

Table 3.3 shows the results in terms of accuracy and computational requirements yielded by the evaluated algorithms on 10 stereo pairs belonging to the Middlebury dataset [112]. Accuracy is calculated as the percentage of retrieved disparities whose difference with the ground truth is ≤ 1. Evaluated disparities relate to all points of the disparity map except for occluded regions, since local WTA methods do not explicitly handle occlusions. As for computations, we report the millions of computed disparity per second (MDS) averaged on the whole tested dataset (for those algorithms deploying segmentation, the MDS includes also the overhead time spent for segmentation). To allow for a qualitative evaluation, all stereo pairs and disparity maps can be found online [3].

[3] Available at: *www.vision.deis.unibo.it/Stereo-FS.asp*

From the table it can be inferred that *Variable Windows* and the proposed approach outperform neatly all the other methods in terms of accuracy on the evaluated dataset, yielding comparable results. Nevertheless, for what concerns computations *Variable Windows* results to be the slowest method, while our approach is the fastest one, being almost two orders faster than *Variable Windows* and more than 3 times faster than *Segmentation based*. Hence it is clear that overall our approach yields the best accuracy-speed trade-off. It is interesting to point out that processing time for our method is around 0.2 s for *Tsukuba* (320 × 240, 16 disp., i.e. working at 5 fps.) and around 0.6 s for *Teddy* and *Art* (respectively 450 × 675, 60 disp. and 463 × 370, 75 disp.), thus achieving near real-time performance.

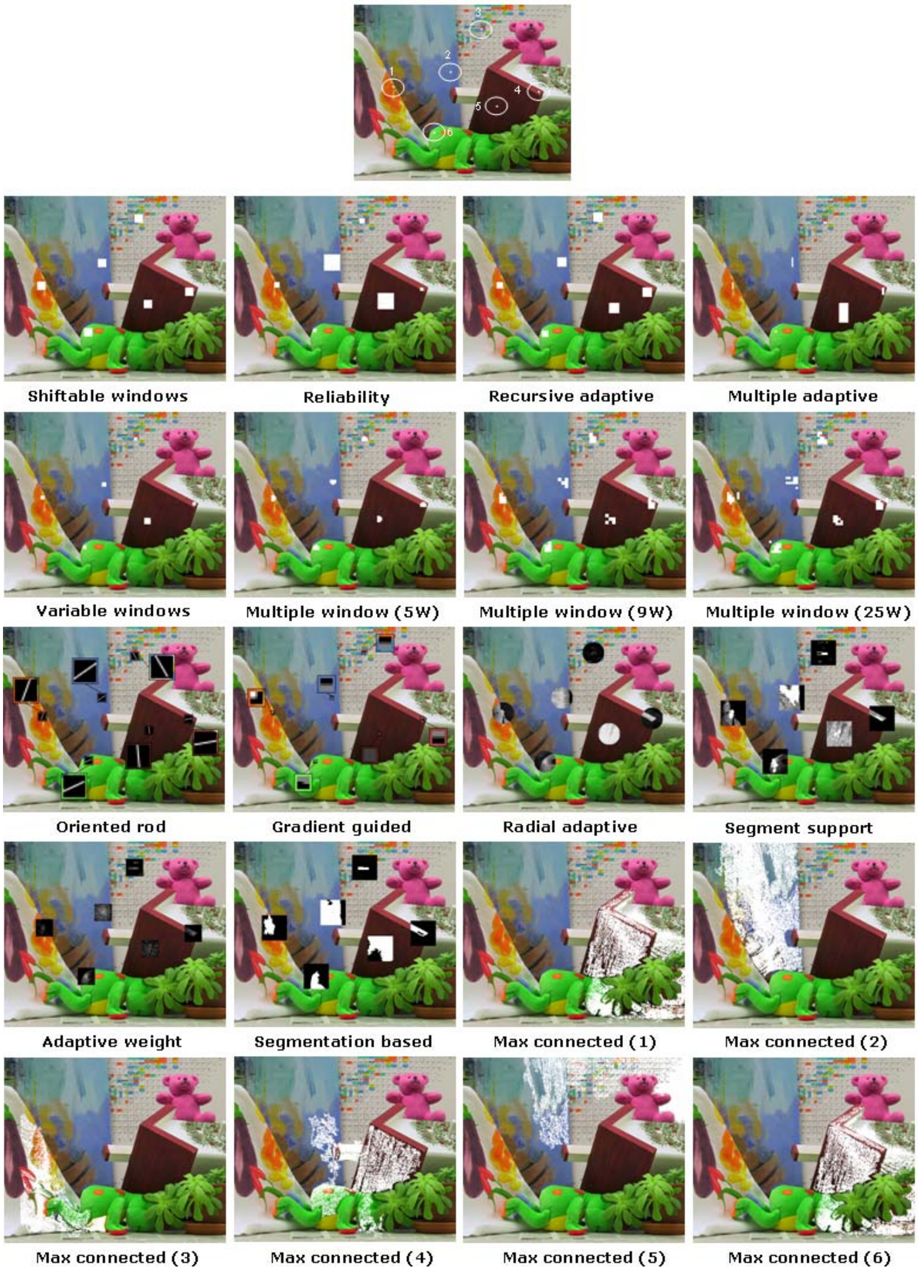
3.5 A performance evaluation of variable support methods

As introduced in section 3.2, in this section we propose a comparison of variable support strategies in terms of accuracy of the retrieved disparity maps based on the methodology proposed in [112]. Then, we extend the evaluation methodology in [112] by comparing strategies also in terms of computational complexity. Moreover, by evaluating jointly the two parameters of comparison we highlight the methods that better trade-off between accuracy and computational complexity.

In particular, since the aim is to specifically evaluate the effectiveness and efficiency of the various aggregation methods, all the considered strategies have been embodied into the same plain WTA framework. The two criteria used for the evaluation are accuracy and computational cost. Evaluation according to the first criterion is accomplished by using the Middlebury Stereo Evaluation Dataset[4] [112]. Computational cost is assessed by measuring for each method the execution time needed to process a reference stereo pair on the same machine (Intel Core Duo 2.14 GHz CPU, 2 GB RAM).

The selected approaches are those that represent the state-of-the-art for the different classes of cost aggregation strategies identified in Section 3.2. In particular, 15 methods are compared, which are now listed together with the nickname used hereinafter to refer them to. The basic method that uses a fixed square window is referred to as *Fixed window*. As for the approaches based on a selection over a set of windows, we evaluated *Shiftable window* [14], *Reliability* [67], *Variable windows* [130], *Recursive adaptive* [20], *Multiple adaptive* [32], *Multiple windows* [60] (tested in the 3 versions based respectively on 5, 9 and 25 *supporting windows*), *Oriented rod* [69], *Gradient guided* [49]. With regards to the approaches that allow for unconstrained

[4] `http://vision.middlebury.edu/stereo`



Shiftable windows Reliability Recursive adaptive Multiple adaptive

Variable windows Multiple window (5W) Multiple window (9W) Multiple window (25W)

Oriented rod Gradient guided Radial adaptive Segment support

Adaptive weight Segmentation based Max connected (1) Max connected (2)

Max connected (3) Max connected (4) Max connected (5) Max connected (6)

Figure 3.8: Qualitative comparison of the supports obtained by different strategies on 6 points of stereo pair *Teddy*. The 6 points are depicted in the top-most image. The images corresponding to *Gradient guided* and *Oriented rod* display also enlarged pictures of the supports to enable better visual analysis of the results. Method *Fast Aggreg* is not included in the comparison.

support shapes we evaluated *Max connected* [15], *Segmentation based* [45] and *Fast Aggreg*, i.e. the method presented in section 3.4. Finally, within the methods based on adaptive weights we considered *Radial adaptive* [138], *Adaptive weight* [141] and *Segment support*, i.e. the method presented in section 3.3.

In order to carry out an as fair as possible comparison, all method were implemented using the same criteria, except in the case of *Adaptive weight* [141] for which the authors' source code is publicly available. For each method, only the proposed aggregation stage was implemented and tested. In particular, neither pre-processing stages nor typical post-processing stages, such as median filtering and left-right consistency check, were applied. In order to better assess the performance of *Oriented rod* and *Multiple windows*, where the proposed pre-processing stage is intrinsically connected with the aggregation stage, for both methods we considered two versions, that is with and without pre-processing. Those versions where pre-processing was excluded will be denoted hereinafter with the symbol $*$. Since for *Oriented rod* pre-processing served as a way to discriminate between homogeneous and heterogeneous points, in the version without pre-processing all points are considered homogeneous and thus compared with the result of a shiftable filter. This generally implies higher computation times and in some case better accuracy. Moreover, for the sake of fairness the cost function is the same for all methods. In particular, since many aggregation strategies rely on colour information [45,49,138,141], the selected cost function is the *Truncated sum of Absolute Differences* (TAD) on RGB pixels, exception made for method *Max connected*, which was implemented as originally proposed by the authors since it is not explicitly based on a cost function. Finally, for each method which was not originally proposed with this cost function or where parameter values were not explicitly specified by authors, parameter values were selected by means of a tuning process ran on the considered dataset.

As far as execution times are concerned, in our implementations we took into account all the guidelines and details originally proposed by the authors, including e.g. the use of incremental schemes for a more efficient implementation (e.g. Variable windows [130]). Our implementations of the basic *Fixed window* and of *Shiftable windows* also deploy incremental schemes. When implementation details were not explicitly provided by the authors we adopted the same plain criteria across the considered algorithm, so as to the render the comparison of execution times as fair as possible. However, it is clear that by extensively optimizing each algorithm according to its own structure one would get different and perhaps much faster execution times. Therefore, the reported measurements should be interpreted only as useful indicators aimed at comparing the computational costs of the considered methods.

3.5.1 Analysis of extracted supports

Fig. 3.8 allows for a qualitative evaluation and comparison of the variable supports extracted by the considered methods on 6 representative hand-chosen points of the stereo pair *Teddy*. The selected points, highlighted in the top picture of Fig. 3.8, refer to regions where stereo methods based on fixed or variable support are often ambiguous due to one or more of the following causes: presence of depth borders (points 1, 2, 4, 6), low-textured areas (points 2, 5), highly textured areas (point 3). For those methods associating weights to points belonging to the support, higher weights are represented by brighter grayscale values. Furthermore, since method *Max connected* can have supports extending to the whole image area, only for this method, for each of the evaluated point, each extracted support is shown on a different picture (indicated in brackets in the figure). Moreover, for the sake of simplicity the supports displayed for [49] are relative to $k = 1$ (although in our implementation we use $k = 4$, as originally proposed in [49], to obtain the experimental results shown in Subsection 3.5.2). Finally, supports for method *Fast Aggreg* are not shown.

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

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

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

	Rank	Tsukuba		Venus		Teddy		Cones		Rank	Time	Average
Algorithm	Accur.	NonOcc		NonOcc		NonOcc		NonOcc		Time	(mm:ss)	Rank
Fast Aggreg	4,25	3,77	2	8,01	10	12,60	3	5,66	2	2	00:00:01	3,13
Segmentation based [45]	4,00	4,53	4	6,91	6	10,94	2	7,67	4	3	00:00:02	3,50
Fixed Window	7,25	6,94	9	7,47	8	16,81	6	8,79	6	1	00:00:00	4,13
Variable Windows [130]	4,25	4,28	3	5,99	4	13,48	5	7,87	5	9	00:00:15	6,63
Multiple Windows (9W)* [60]	10,25	8,39	14	8,05	11	17,51	9	10,17	7	4	00:00:04	7,13
Multiple Windows (5W) [60]	12,00	7,09	10	12,96	14	20,72	14	12,33	10	3	00:00:02	7,50
Adaptive weight [141]	3,25	4,66	5	4,61	3	12,70	4	5,50	1	12	00:17:01	7,63
Shiftable Windows [14]	9,25	7,58	12	7,79	9	17,19	8	10,27	8	6	00:00:12	7,63
Segment support	1,50	2,15	1	1,38	1	10,54	1	5,83	3	14	00:30:38	7,75
Gradient Guided [49]	10,75	6,68	8	11,10	13	19,17	10	12,58	12	5	00:00:05	7,88
Multiple Windows (25W)* [60]	9,25	6,51	7	6,85	5	19,21	11	13,55	14	7	00:00:13	8,13
Multiple Windows (5W)* [60]	13,50	9,56	17	13,32	15	19,56	13	12,11	9	3	00:00:02	8,25
Multiple Windows (9W) [60]	14,50	7,80	13	14,74	18	21,06	16	12,52	11	5	00:00:05	9,75
Reliability [67]	7,75	5,71	6	2,87	2	16,82	7	14,40	16	13	00:18:07	10,38
Multiple Windows (25W) [60]	14,00	7,40	11	14,72	17	21,01	15	12,96	13	8	00:00:14	11,00
Recursive Adaptive [20]	14,25	9,90	18	10,76	12	19,26	12	13,67	15	12	00:17:01	13,13
Radial Adaptive [138]	15,00	9,55	16	7,27	7	23,46	18	21,77	19	15	00:56:14	15,00
Oriented Rod [69]	20,25	15,14	20	28,70	19	34,80	21	37,03	21	10	00:06:46	15,13
Oriented Rod* [69]	20,00	17,21	21	29,26	20	32,70	19	32,51	20	11	00:06:48	15,50
Multiple Adaptive [32]	16,50	9,40	15	13,61	16	21,19	17	20,19	18	16	01:13:26	16,25
Max Connected [15]	19,25	11,81	19	42,46	21	34,46	20	17,70	17	17	01:56:22	18,13

Table 3.4: Performance evaluation in terms of accuracy and execution times of the considered variable-support based strategies.

while *Segment support* also handles better the considered high-textured region (point 3) compared to the other two approaches which, conversely, at this point retrieve very small supports.

3.5.2 Accuracy and computational cost

For what concerns accuracy we rely on a testbed and evaluation methodology analogous to that adopted on the Middlebury Stereo Evaluation site. In particular, as it can be seen from Table 3.4, we use 4 reference stereo pairs (Tsukuba, Venus, Teddy and Cones) and for each of them evaluate the error rates on the ground truth maps *NonOcc* (all points except for occluded areas). Each single error rate is also denoted by its respective ranking along the considered methods. Error rates on all image points including occlusions have not been taken into account since the tested algorithms do not explicitly handle disparity retrieval for occluded points due to the adopted WTA framework. Besides, an overall accuracy ranking obtained by averaging the single rankings of each method along the dataset is shown in the second column.

As for computational costs, Table 3.4 reports for each method the measured execution time on the stereo pair *Teddy*. Similarly to the evaluation of accuracy, the Table shows the ranking of methods according to the measured execution times. Finally, the Table reports in the rightmost column the ranking obtained by averaging the overall accuracy ranking and the time ranking, so as to highlight the methods that better trade-off between accuracy and computational efficiency.

Coherently with the qualitative analysis based on Fig. 3.8, the Table shows that the most accurate methods are those deploying adaptive weights. In particular, *Segment support* [124] and *Adaptive weight* [141] outperform the other methods almost on the whole datasetConversely, the fastest methods are those based on the evaluation of the support over a set of windows or based on unconstrained shapes. It is worth observing that, apart from the basic *Fixed window* approach, methods such as *Fast Aggreg*, *Segmentation based* [45], *Gradient guided* [49], *Multiple window* [60], *Shiftable window* [14] and *Variable windows* [130] can run in seconds or tens of seconds, while some methods, i.e. *Max connected* [15] and *Radial adaptive* [138] may require hours. As regards the accuracy/efficiency trade-off, the best method turned out to be *Fast Aggreg*, followed by *Segmentation based* [45]

The disparity maps obtained by the various methods on the Middlebury dataset as well as the qualitative comparison of the extracted supports concerning the *Tsukuba* and *Teddy* stereo pairs can be found on-line[5]. In addition, this web site includes the results dealing with other cost measures (SAD, SSD) and the program used for generating the supports depicted in Fig. 3.8.

3.6 Accurate stereo matching based on Scanline Optimization

A category of stereo methods which lies in between local and global approaches refers to those techniques based on the minimization of an energy function computed over a subset of the whole image area, i.e. typically along epipolar lines or *scanlines*. The adopted minimization strategy is usually based on *Dynamic Programming* (DP) or *Scanline Optimization* (SO) [58], [59], [51], [69] techniques, and some algorithms also

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

exploit DP on a tree [78], [33]. The global energy function to be minimized includes a pointwise matching cost C_M (see [112] for details) and a smoothness term which enforces constant disparity e.g. on untextured regions by means of a discontinuity penalty π:

$$E\left(d\left(A\right)\right) = \sum_{i \in A} C_M\left(p_r^i, p_{t,d(A)}^i\right) + N\left(d\left(A\right)\right) \cdot \pi \tag{3.14}$$

with A being the image subset (e.g. a scanline) and N being the number of times the smoothness constraint is violated within the region where the cost function has to be minimized. These approaches achieved excellent results in terms of accuracy in the disparity maps [58] and in terms of very fast, near real-time, computational performances [51].

In order to increase robustness against outliers a fixed support (typically a 3×3 window) can be employed instead of the pointwise matching score. Nevertheless, this approach embodies all the negative aspects of a local window-based method, which are especially evident near depth discontinuities: object borders tend to be inaccurately detected.

In this section we propose to deploy an SO-based algorithm which embodies, as matching cost C_M, a function based on a variable support. The SO framework allows to handle effectively low-textured surfaces while the variable support approach helps preserving accuracy along depth borders. In order to determine the variable support, we adopt the effective technique based on colour proximity and segmentation proposed in section 3.3. The accuracy of the SO-based process is also improved by the use of a symmetrical smoothness penalty which depends on the pixel intensities of both stereo images. It will be shown that this approach allows to obtain notable accuracy in the retrieved disparities.

Moreover, we propose a refinement step which allows to further increase the accuracy of the proposed method. This step relies on a technique that, exploiting symmetrically the relationship between occlusions and depth discontinuities on the disparity maps obtained assuming alternatively as reference the left and the right image, allows for accurately locating borders. This is shown to be particularly useful to assign the correct disparity values to those points violating the cross-checking constraint. Finally, experimental results show that the proposed approach is able to determine accurate dense stereo maps and it is state-of-the-art for what means approaches which do not rely on a global framework.

3.6.1 A symmetric Scanline Optimization framework

The first step of the proposed technique computes a matching cost $C_{M,v}(p_r, p_{t,d})$ based on a variable support strategy proposed in section 3.3. Then, the matching cost is embodied in a simplified SO-based framework similar to that proposed in [58]. Hence, in the first stage of the algorithm the matching cost matrix $C_{M,v}(p_r, p_{t,d})$ is computed for each possible correspondence $(p_r, p_{d,t})$. Then, in the second stage, 4 SO processes are used: 2 along horizontal scanlines on opposite directions and 2 similarly along vertical scanlines. The *j-th* SO computes the current global cost between p_r and $p_{t,d}$ as:

$$\begin{aligned} C_G^j(p_r, p_{t,d}) = C_{M,v}(p_r, p_{t,d}) + min(C_G^j(p_r^p, p_{t,d}^p), C_G^j(p_r^p, p_{t,d-1}^p) + \pi_1, \\ C_G^j(p_r^p, p_{t,d+1}^p) + \pi_1, c_{min} + \pi_2) - c_{min} \end{aligned} \tag{3.15}$$

with p_r^p and $p_{t,d}^p$ being respectively the point in the previous position of p_r and $p_{t,d}$ along the considered scanline, π_1 and π_2 being the two smoothness penalty terms (with $\pi_1 \leq \pi_2$) and c_{min} defined as:

$$c_{min} = min_i(C_G^j(p_r^p, p_{t,i}^p)) \tag{3.16}$$

For what means the two smoothing penalty terms, π_1 and π_2, they are dependent on the image local intensities similarly to what proposed in [142] within a global stereo framework. This is due to the assumption that often a depth discontinuity coincides with an intensity edge, hence the smoothness penalty must be relaxed along edges and enforced within low-textured areas. In particular, we apply a symmetrical strategy so that the two terms depend on the intensities of both I_r and I_t. If we define the intensity difference between the current point and the previous one along the considered scanline on the two images as:

$$\begin{aligned} \triangledown\,(p_r) = |I_r(p_r) - I_r(p_r^p)| \\ \triangledown(p_{t,d}) = |I_t(p_{t,d}) - I_t(p_{t,d}^p)| \end{aligned} \tag{3.17}$$

then π_1 is defined as:

$$\pi_1(p_r, p_{t,d}) = \begin{cases} \Pi_1 & \triangledown(p_r) < P_{th}, \triangledown(p_{t,d}) < P_{th} \\ \Pi_1/2 & \triangledown(p_r) \geq P_{th}, \triangledown(p_{t,d}) < P_{th} \\ \Pi_1/2 & \triangledown(p_r) < P_{th}, \triangledown(p_{t,d}) \geq P_{th} \\ \Pi_1/4 & \triangledown(p_r) \geq P_{th}, \triangledown(p_{t,d}) \geq P_{th} \end{cases} \tag{3.18}$$

where Π_1 is a constant parameter of the algorithm, and π_2 is defined in the same manner based on Π_2. Finally, P_{th} is a threshold which determines the presence of

Table 3.5: Error rates using $C_{M,v}$ within the SO-based framework proposed (first row), a pointwise matching cost ($C_{M,p}$) within the same SO-based framework (second row), and $C_{M,v}$ in a local WTA approach (last row).

	Tsukuba	Venus	Teddy	Cones
	N.O. - DISC	N.O. - DISC	N.O. - DISC	N.O. - DISC
$C_{M,v}$, SO	1.63 - 6.80	0.97 - 9.03	9.64 - 19.35	4.60 - 11,52
$C_{M,p}$, SO	3.70 - 13.38	4.19 - 19.27	12.28 - 20.40	5.99 - 13.96
$C_{M,v}$, local	2,05 - 7,14	1,47 - 10,5	10,8 - 21,7	5,08 - 12,5

an intensity edge. Thanks to this approach, horizontal/vertical edges are taken into account along corresponding scanline directions (i.e. horizontal/vertical) during the SO process, so that edges orthogonal to the scanline direction can not influence the smoothness penalty terms.

Once the 4 global costs C_G are obtained, they are summed up together and a *Winner-Take-All* approach on the final cost sum assigns the disparity:

$$d_{p_r,best} = \arg\min_{d\in D}\{\sum_{j=1}^{4} C_G^j(p_r, p_{t,d})\} \tag{3.19}$$

3.6.2 A first experimental evaluation of the proposed approach

We now briefly show some results dealing with the use of the approach outlined so far. In particular, in order to demonstrate the benefits of the joint use of the SO-based framework with the variable support-based matching cost $C_{M,v}$, we compare the results yielded by our method to those attainable by the same SO framework using the pointwise TAD matching cost on RGB triplets, as well as by $C_{M,v}$ in a local WTA approach.

The dataset used for experiments is available at the Middlebury website[6]. Parameter set is constant for all runs. Truncation parameter for TAD in both approaches is set to 80. For what means the variable support, segmentation is obtained by running the Mean Shift algorithm [21] with a constant set of parameters (spatial radius $\sigma_S = 3$, range radius $\sigma_R = 3$, minimum region size $min_R = 35$), while maximum radius size of the support is set to 51, and parameter γ_c is set to 22. Finally, for what means the SO framework, our approach is run with $\Pi_1 = 6, \Pi_2 = 27, P_{th} = 10$, while the pointwise cost-based approach is run with $\Pi_1 = 106, \Pi_2 = 312, P_{th} = 10$ (optimal parameters for both approaches).

[6] *vision.middlebury.edu/stereo*

Table 3.5 shows the error rates computed on the whole image area except for occlusion (*N.O.*) and in proximity of discontinuities (*DISC*). Occlusions are not evaluated here since at this stage no specific occlusion handling approach is adopted by any of the algorithms. As it can be inferred, the use of a variable support in the matching cost yields significantly higher accuracy in all cases compared to the pointwise cost-based approach, the highest benefits being on *Tsukuba* and *Venus* datasets. Moreover, benefits are significant also by considering only depth discontinuities, which demonstrate the higher accuracy in retrieving correctly depth borders provided by the use of a variable support within the SO-based framework. Finally, benefits of the use of the proposed SO-based framework are always notable if we compare the results of our approach with those yielded by using the same cost function within a local WTA strategy.

3.6.3 Symmetrical detection of occluded areas and depth borders

By respectively assuming as reference I_r the left and the right image of the stereo pair, it is possible to obtain two different disparity maps, referred to as D_{LR} and D_{RL}. Our idea is to derive a general method for detecting depth borders and occluded regions by enforcing the symmetrical relationship on both maps between occlusions and depth borders resulting from the stereo setup and the scene structure.

In particular, due to the stereo setup, if we imagine to scan any epipolar line of D_{LR} from left side to right side, each sudden depth decrement corresponds to an occlusion in D_{RL}. Similarly, scanning any epipolar line of D_{RL} from right side to left side, each sudden depth increment corresponds to an occlusion in D_{LR}. Moreover, the occlusion width is directly proportional to the amount of each depth decrement and increment along the correspondent epipolar line, and the two points composing a depth border on one disparity map respectively correspond to the starting point and ending point of the occluded area in the other map.

Hence, in order to detect occlusions and depth borders, we deploy a symmetrical cross-checking strategy, which detects the disparities in D_{LR} which violate a *weak* disparity consistence constraint by tagging as *invalid* all points $p_d \in D_{LR}$ for which:

$$|D_{LR}(p_d) - D_{RL}\,(p_d - D_{LR}\,(p_d))\,| \leq 1 \tag{3.20}$$

and analogously detects invalid disparities on D_{RL}. Points referring to disparity differences equal to 1 are not tagged as invalid at this stage as we assume that occlusions are not present where disparity varies smoothly along the epipolar lines, as well as to handle slight discrepancies due to the different view points. The results of this symmetrical cross-checking are shown, referred to *Tsukuba* and *Cones*, on the left and center images of Fig. 3.9, where colored points in both maps represent the disparities

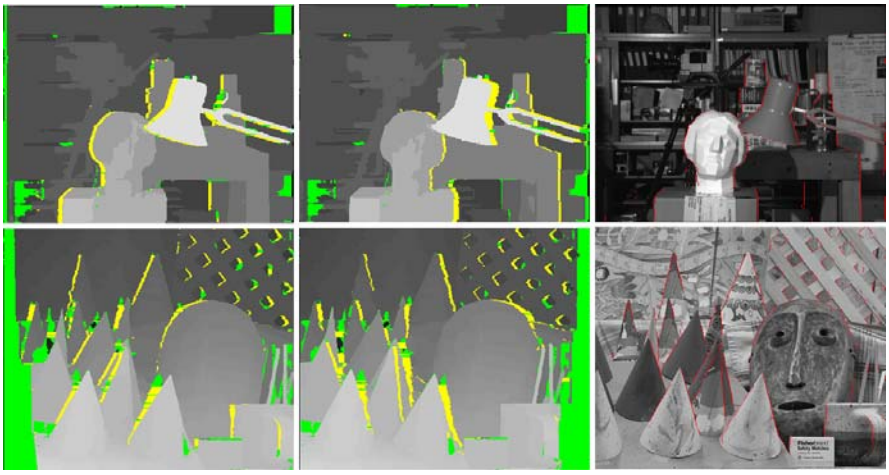


Figure 3.9: Points violating (3.20) on D_{LR} and D_{RL} (colored points, left and center) are discriminated between occlusions (yellow) and false matches (green) on *Tsukuba* and *Cones* datasets. Consequently depth borders are detected (red points, right) [This Figure is best viewed with colors]

violating (3.20). It is easy to infer that only a subset of the colored regions of the maps is represented by occlusions, while all other violating disparities denote mismatches due to outliers.

Hence, after cross-checking the two disparity maps D_{LR} and D_{RL}, it is possible to discriminate on both maps occluded areas from incorrect correspondences (respectively yellow and green points on left and center image, Fig. 3.9) by means of application of the constraints described previously. Then, putting in correspondence occlusions on one map with homologous depth discontinuities in the other map, it is possible to reliably localize depth borders generated by occlusions on both disparity maps (details of this method are not provided here due to the lack of space). Right images on Fig. 3.9 show the superimposition of the detected borders referred to D_{LR} (in red color) on the corresponding grayscale stereo image. As it can be seen, borders along epipolar lines are detected with notable precision and very few outliers (detected borders which do not correspond to real borders) are present.

3.6.4 Refinement by means of detected depth borders and segmentation

Depth border detection is employed in order to determine the correct disparity values to be assigned to points violating cross-checking. In particular, a two-step refinement process is now proposed, which exploits successively segmentation and depth border information in order to fill-in, respectively, low textured areas and regions along depth

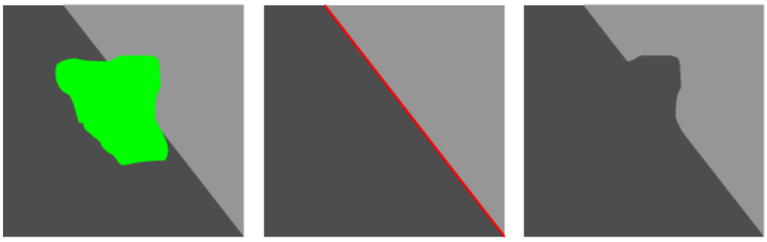


Figure 3.10: The reliability of assigning disparities to points violating the strong cross-checking (3.21) along depth borders (green points, left) is increased by exploiting information on depth borders location (red points, center) compared to a situation where this information is not available (right)

discontinuities.

First of all, the following *strong* cross-checking consistency constraint is applied on all points of D_{LR}:

$$D_{LR}(p_d) = \begin{cases} D_{LR}(p_d) & D_{LR}(p_d) \neq D_{RL}\left(p_d - D_{LR}(p_d)\right) \\ invalid & otherwise \end{cases} \tag{3.21}$$

The first step of the proposed refinement approach employs segmentation information in order to fill-in regions of D_{LR} denoted as invalid after application of (3.21). In particular, for each segment extracted from the application of the Mean Shift algorithm, a disparity histogram is filled with all valid disparities included within the segment area. Then, if a unique disparity value can be reliably associated with that segment, i.e. if there is a minimum number of valid disparities in the histogram and its variance is low, the mean disparity value of the histogram is assigned to all invalid points falling within the segment area. This allows to correctly fill-in uniform areas which can be easily characterized by mismatches during the correspondence search.

As this first step is designed to fill-in only invalid points within uniform areas, then a second step allows to fill-in the remaining points by exploiting the previously extracted information on border locations, especially along depth border regions which usually are not characterized by uniform areas. In particular, the assigned disparity value for all invalid points near to depth discontinuities is chosen as the minimum value between neighbours which do not lie beyond a depth border. This allows to increase the reliability of the assigned values compared to the case of no information on borders location, where e.g. the minimum value between neighbouring disparities is selected, as shown in Fig. 3.10.

Table 3.6: Disparity error rates and rankings obtained on Middlebury website

	Rank	Tsukuba	Venus	Teddy	Cones
		N.O.-ALL-DISC	N.O.-ALL-DISC	N.O.-ALL-DISC	N.O.-ALL-DISC
AdaptingBP [70]	♯1	1.11-1.37-5.79	0.10-0.21-1.44	4.22-7.06-11.8	2.48-7.92-7.32
DoubleBP [139]	♯2	0.88-1.29-4.76	0.14-0.60-2.00	3.55-8.71-9.70	2.90-9.24-7.80
SymBP+occ	♯3	0.97-1.75-5.09	0.16-0.33-2.19	6.47-10.7-17.0	4.79-10.7-10.9
SO+border	**♯4**	**1.29-1.71-6.83**	**0.25-0.53-2.26**	**7.02-12.2-16.3**	**3.90-9.85-10.2**
Segm+visib [13]	♯5	1.30-1.57-6.92	0.79-1.06-6.76	5.00-6.54-12.3	3.72-8.62-10.2
C-SemiGlob [59]	♯6	2.61-3.29-9.89	0.25-0.57-3.24	5.14-11.8-13.0	2.77-8.35-8.20
RegionTreeDP [78]	♯10	1.39-1.64-6.85	0.22-0.57-1.93	7.42-11.9-16.8	6.31-11.9-11.8

3.6.5 Experimental results

This section shows an experimental evaluation obtained by submitting on the Middlebury site the results yielded by the proposed algorithm. The parameter set of the algorithm is constant for all runs and is the same as for the experiments in Sec. 3.6.2. As it can be seen from Table 3.6, our algorithm (*SO+border*), which ranked 4*th* (as of May 2007), produces overall better results compared to [59], which employs a higher number of scanlines during the SO process, and also compared to the other SO and DP-based approaches and most global methods, for higher accuracy is only yielded by three BP-based global algorithms. Obtained disparity maps, together with corresponding reference images and groundtruth are shown in Fig. 3.11 and are available at the Middlebury website. The running time on the examined dataset is of the order of those of other methods based on a variable support [141] (i.e. some minutes) since the majority of time is required by the local cost computation, while the S.O. stage and the border refinement stage only account for a few seconds and are negligible compared to the overall time.

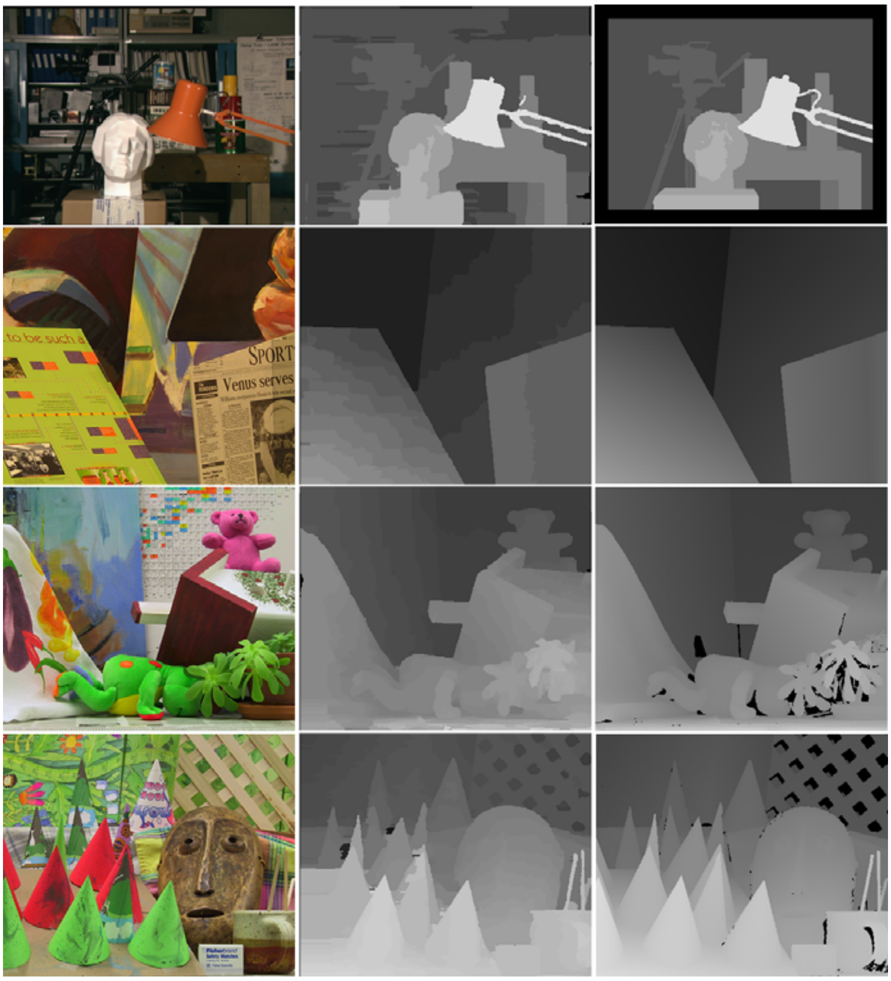


Figure 3.11: Disparity maps obtained after the application of all steps of the proposed approach

Chapter 4

Stereo applications

This chapter presents some applications of stereo vision that were investigated during the Ph.D. activity. In particular, the tasks that will be illustrated concern the fields of 3D reconstruction and video-surveillance. The common point is that we always attack the problem using a stereo camera. It will be also highlighted how an accurate - and often fast - stereo correspondence algorithm is of key importance for the performance of the ilustrated systems.

4.1 3D reconstruction

This section concerns the research activity carried out at Willow Garage [1], a privately-funded research center based in Menlo Park (USA). In particular, the Personal Robot project ran by the center aims at building a mobile robot with manipulators (PR2) for ordinary household tasks such as setting or clearing a table.

An important sensing technology for object recognition and manipulation is short-range (30cm – 200cm) 3D perception. Criteria for this device include:

- Good spatial and depth resolution (1/10 degree, 1 mm).
- High speed (>10 Hz).
- Ability to deal with moving objects.
- Robust to ambient lighting conditions.
- Small size, cost, and power.

Current technologies fail on at least one of these criteria. Flash ladars [2] lack depth and, in some cases, spatial resolution, and have non-gaussian error characteristics that

[1] `www.willowgarage.com`

are difficult to deal with. Line stripe systems [25] have the requisite resolution but cannot achieve 10 Hz operation, nor deal with moving objects. Structured light systems [110] are achieving reasonable frame rates and can sometimes incorporate motion, but still rely on expensive and high-powered projection systems, while being sensitive to ambient illumination and object reflectance. Standard block-matching stereo, in which small areas are matched between left and right images [72], fails on objects with low visual texture.

An interesting and early technology is the use of stereo with *u*nstructured light [97]. Unlike structured light systems with single cameras, stereo does not depend on the relative geometry of the light pattern – the pattern just lends texture to the scene. Hence the pattern and projector can be simplified, and standard stereo calibration techniques can be used to obtain accurate 3D measurements.

Even with projected texture, block-matching stereo still forces a tradeoff between the size of the match block (larger sizes have lower noise) and the precision of the stereo around depth changes (larger sizes "smear" depth boundaries). One possibility is to use smaller matching blocks, but reduce noise by using many frames with different projection patterns, thereby adding information at each pixel. This technique is known as *Spacetime Stereo* (STS) [28], [145]. It produces outstanding results on static scenes and under controlled illumination conditions, but moving objects create obvious difficulties (see Figure 4.1, bottom-left). While there have been a few attempts to deal with motion within a STS framework [145], [136], the results are either computationally expensive or perform poorly, especially for fast motions and depth boundaries.

This section proposes the use of regularization methods to attack the problem of motion in spacetime stereo. One proposed contribution is to enforce not only spatial, but also temporal smoothness constraints that benefit from the texture-augmented appearance of the scene. Furthermore, we propose a new regularization method, *local smoothing*, that yields an interesting efficiency-accuracy trade-off. Finally, the section also aims at comparing STS with regularization methods, since a careful reading of the spacetime stereo literature [28, 145] shows that this has not been addressed before. Experimentally we found that, using a projected texture, regularization methods applied on single frames perform better than STS on dynamic scenes (see Figure 4.1) and produces interesting results also on static scenes.

In the next subsection we review several standard regularization methods, and introduce a novel method, *local smoothness*, which is more efficient and almost as effective. We then show how regularization can be applied across time as well as space, to help alleviate the problem of object motion in STS. In the experimental subsection, the considered methods are compared on static scenes and in the presence of moving objects.

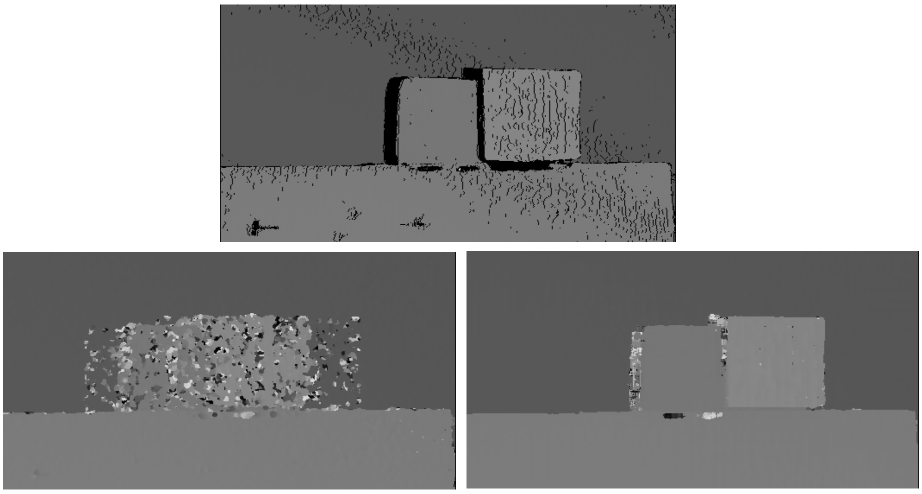


Figure 4.1: The top figure shows the disparity surface for a static scene; disparities were computed by integrating over 30 frames with varying projected texture using block-matching (3x3x30 block). The bottom-left figure is the same scene with motion of the center objects, integrated over 3 frames (5x5x3 block). The bottom-right figure is our local smoothing method for a single frame (5x5 block).

4.1.1 Smoothness constraints in stereo matching

As discussed in Chapter 3, stereo matching is difficult in areas with low texture and at depth boundaries. Regularization methods add a smoothness constraint to model the regularity of surfaces in the real world. The general idea is to penalize those candidates lying at a different depth from their neighbors. A standard method is to construct a disparity map giving the probability of each disparity at each pixel, and compute a global energy function for the disparity map as a multi-class Pairwise Markov Random Field. The energy is then minimized using approximate methods such as Belief Propagation (BP) [70], [139] or Graph Cuts (GC) [23]. Even though efficient BP-based algorithms have been proposed [140], [40], overall the computational load required by global approaches does not allow real-time implementation on standard PCs.

Rather than solving the full optimization problem over the disparity map, scanline methods enforce smoothness along a line of pixels. Initial approaches based on Dynamic Programming (DP) and Scanline Optimization (SO) [112] use only horizontal scanlines, but suffer from streaking effects. More sophisticated approaches apply SO over multiple, variably-oriented scanlines [58] or use multiple horizontal and vertical passes [69], [86], [51]. These methods tend to be faster than global regularization, though the use of several DP or SO passes tends to increase the computational load of the algorithms. Another limit to the applicability of these approaches within a mobile robotic platform is their fairly high memory requirements. This subsection we review scanline methods and proposes a new method called *local smoothness*.

Global scanline methods Let I_L, I_R be a rectified stereo image pair sized $M \cdot N$ and $W(p)$ a vector of points belonging to a squared window centered on p. The *standard* block-matching stereo algorithm computes a local cost $C(p,d)$ for each point $p \in I_L$ and each possible correspondence at disparity $d \in D$ on I_R:

$$C(p,d) = \sum_{q \in W(p)} e(I_L(q), I_R(\delta(q,d))). \tag{4.1}$$

where $\delta(q,d)$ is the function that offsets q in I_R according to the disparity d, and e is a (dis)similarity function. A typical dissimilarity function is the L_1 distance:

$$e(I_L(q), I_R(\delta(q,d))) = \left| I_L(q) - I_R\left(\delta(q,d)\right)\right|. \tag{4.2}$$

In this case, the best disparity for point p is selected as:

$$d^* = \arg\min_d \{C(p,d)\}. \tag{4.3}$$

In the usual SO or DP-based framework, the global energy functional being minimized along a scanline S is:

$$E\left(d(\cdot)\right) = \sum_{p \in S} C\left(p, d(p)\right) + \sum_{p \in S} \sum_{q \in \mathcal{N}(p)} \rho\left(d(p), d(q)\right) \tag{4.4}$$

where $d(\cdot)$ denotes now a function that picks out a disparity for its pixel argument, and $q \in \mathcal{N}(p)$ are the neighbors of p according to a pre-defined criterion. Thus to minimize (4.4) one has to minimize two different terms, the first acting as a local evidence and the other enforcing smooth disparity variations along the scanline, resulting in a non-convex optimization problem. The smoothness term ρ is usually derived from the Potts model [105]:

$$\rho\left(d(p), d(q)\right) = \begin{cases} 0 & d(p) = d(q) \\ \pi & d(p) \neq d(q) \end{cases} \tag{4.5}$$

π being a penalty term inversely proportional to the temperature of the system. Usually for stereo a Modified Potts model is deployed, which is able to handle slanted surfaces by means of an additional penalty term $\pi_s << \pi$:

$$\rho\left(d(p), d(q)\right) = \begin{cases} 0 & d(p) = d(q) \\ \pi_s & |d(p) - d(q)| = 1 \\ \pi & elsewhere \end{cases} \tag{4.6}$$

Thanks to (4.6), smooth variations of the disparity surface are permitted at the cost of the small penalty π_s. Usually in SO and DP-based approaches the set of neighbours for a point p includes only the previous point along the scanline, p_{-1}. From an algorithmic perspective, an aggregated cost $A(p, d)$ has to be computed for each $p \in S$, $d \in D$:

$$A(p, d) = C(p, d) + \min_{d'}\{A(p_{-1}, d') + \rho(d, d')\} \tag{4.7}$$

Because of the nature of (4.7) the full cost for each disparity value at the previous point p_{-1} must be stored in memory. If a single scanline is used, this typically requires $O(M \cdot D)$ memory, while if multiple passes along non-collinear scanlines are concerned, this usually requires $O(M \cdot N \cdot D)$ memory [58].

Local smoothness Keeping the full correlation surface over $M \cdot N \cdot D$ is expensive; we seek a more local algorithm that aggregates costs incrementally. In a recent paper [146], a penalty term is added in a local fashion to improve post-processing of the disparity image based on left-right consistency check. Here, we apply a similar penalty during the construction of the disparity map and generalize its use for multiple scanlines. Given a scanline S, we can modify (4.7) as follows:

$$A_{LS}(p, d) = C(p, d) + \rho(d, \tilde{d}) \tag{4.8}$$

where

$$\tilde{d} = \arg\min_{d}\{C(p_{-1}, d)\} \tag{4.9}$$

is the best disparity computed for the previous point along the scanline. Hence, each local cost is penalized if the previously computed correspondence along the scanline corresponds to a different disparity value. In this approach, there is no need to keep track of an aggregated cost array, since the aggregated cost for the current point only depends on the previously computed disparity. In practice the computation of (4.8) for the current disparity surface might be performed simply by subtracting π from $C(p, \tilde{d})$ and $\pi - \pi_s$ from $C(p, \tilde{d} - 1)$, $C(p, \tilde{d} + 1)$.

Enforcing smoothness in just one direction helps handle low-textured surfaces, but tends to be inaccurate along depth borders, especially in the presence of negative disparity jumps. Using two scans, e.g. horizontally from left to right and from right to left, helps to reduce this effect, but suffers from the well-known streaking effect [112]. In order to enforce inter-scanline consistency, we run local smoothness over 4 scans, 2 vertical and 2 horizontal (see Figure 4.8). In this case, which we will refer to as *Spatial Local Smoothness* (LS_s), the aggregated cost (4.8) is modified as follows:

$$A_{LS_s}(d) = C(p, d) + \sum_{q \in \mathcal{N}(p)} \rho\,(d, d(q))\,. \tag{4.10}$$

Here $\mathcal{N}$ refers to the 4 disparities previously computed on p. The computation of d^* benefits from propagated smoothness constraints from 4 different directions, which

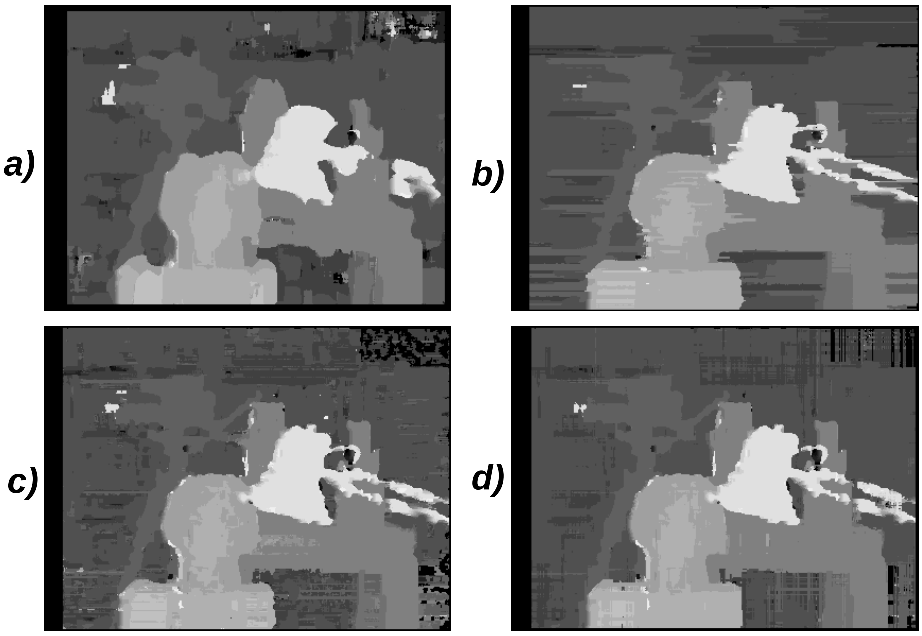


Figure 4.2: Qualitative comparison of different algorithms based on the smoothness constraint: a)standard b)SO-based c)local smoothness (2 horizontal scanlines) d)local smoothness (4 scanlines).

reduces noise in low-textured surfaces, and also reduces streaking and smearing effects typical of scanline-based methods.

It is worth pointing out that the LS_s approach can be implemented very efficiently by means of a two-stage algorithm. In particular, during the first stage of the algorithm, the forward-horizontal and forward-vertical passes are computed, and the result is stored into two $M \cdot N$ arrays. Then, during the second pass, the backward-horizontal and backward-vertical passes are processed, and within the same step the final aggregated cost (4.10) is also computed. Then the best disparity is determined as in (4.9). Overall, computational cost is between 3 and 4 times that of the standard local stereo algorithm. Memory requirements are also small – $O(2 \times M \times N)$.

Experimental evaluation We now briefly present some experimental results showing the capabilities of the previously introduced regularization methods on stereo data by comparing them to a standard block-correlation stereo algorithm. In particular, in addition to the LS_s algorithm, we consider a particularly efficient approach based only on one forward and one backward horizontal SO pass [86]. This algorithm accounts for low memory requirements and fast performance, though it tends to suffer the streaking effect. We will refer to this algorithm as SO_s.

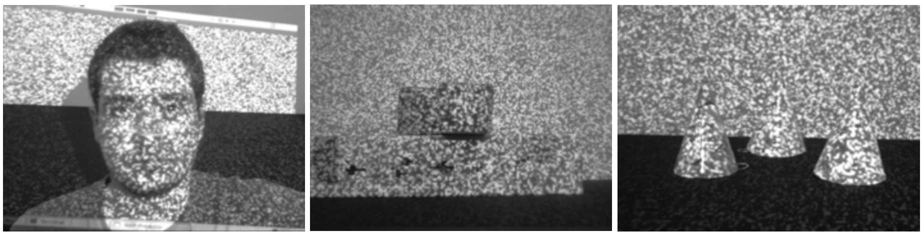


Figure 4.3: Dataset used for experiments: from left to right, *Face*, *Cubes*, *Cones* sequences.

Fig. 4.2 shows some qualitative results on the *Tsukuba* dataset [112]. The standard local algorithm is in (a), SO_s (b) and the LS_s algorithm in (d). Also, the figure shows the disparity map obtained by the use of the Local Smoothness criterion over only 2 horizontal scanlines in (c). It can be noticed that, compared to the standard approach, regularization methods allow for improved accuracy along depth borders. Furthermore, while methods based only on horizontal scanlines (b, c) present typical horizontal streaking effects, these are less noticeable in the LS_s algorithm (d). In our implementation, using standard incremental techniques but no SIMD or multi-thread optimization, time requirements on a standard PC for the standard, SO_s and LS_s algorithms are 18, 62 and 65 ms, respectively.

In addition, we show some results concerning images where a pattern is projected on the scene. As for the pattern, we use a randomly-generated grayscale chessboard, which is projected using a standard video projector. Fig. 4.3 shows 3 frames taken from 3 stereo sequences used here and in Section 4.1.2 for our experiments. Sequence *Face* is a static sequence, while *Cubes* and *Cones* are dynamic scenes where the objects present in the scene rapidly shift towards one side of the table. All frames of all sequences are 640×480 in resolution.

Figure 4.4 shows experimental results for the standard algorithm as well as SO_s and LS_s over different window sizes. Similarly to what done in [28], ground truth for this data is the disparity map obtained by the spacetime stereo technique (see next Section) over all frames of the sequence using a 5×5 window patch. A point in the disparity map is considered erroneous if the absolute difference between it and the groundtruth is higher than one.

From the figure it is clear that, even on this real dataset, regularization methods allow for improved results compared to standard methods since the curve concerning the standard algorithm is always above the other two. It is worth pointing out that both SO_s and LS_s achieve their minimum with a smaller spatial window compared to the standard algorithm, allowing for reduced smearing effect along depth borders.

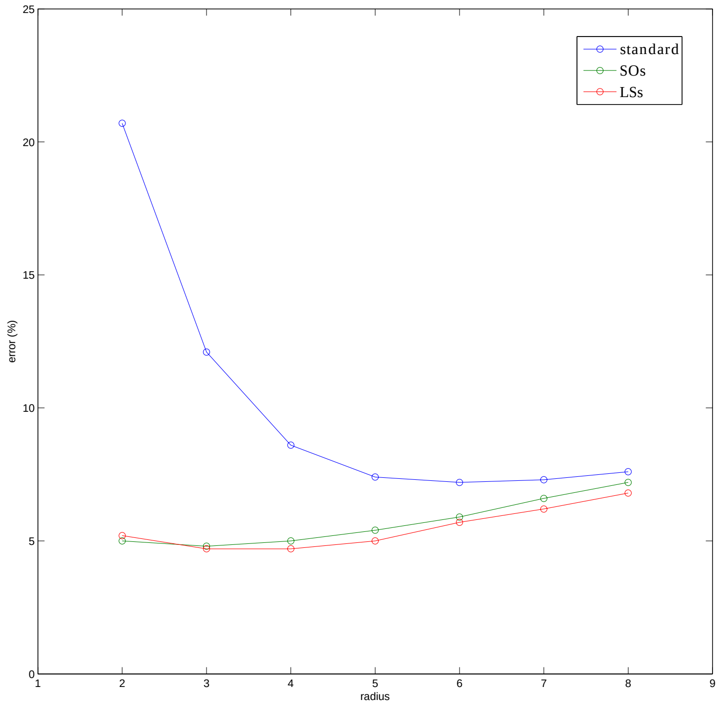


Figure 4.4: Quantitative comparison between different spatial approaches: standard algorithm, SO_s, LS_s.

Conversely, the use of regularization methods with big windows increase the error rate which tends to converge to the one yielded by the standard method. It is also worth pointing out that overall the best result is yielded by the proposed LS_s algorithm. Finally, Figure 4.5 shows the 3D point cloud of the face profile obtained by using the LS_s algorithm over one frame on the *Face* dataset. From the Figure it can be noted that despite being fast and memory-efficient, this algorithm is able to obtain good accuracy in the reconstructed point cloud.

4.1.2 Spacetime stereo

Block-correlation stereo uses a spatial window to smooth out noise in stereo matching. A natural extension is to extend the window over time, that is, to use a spatio-temporal window to aggregate information at a pixel [145], [28] (Figure 4.6). The intensity at position $I(p,t)$ is now dependent on time, and the block-matching sum over a set of frames F and a spatial window W can be written as

$$C(p,d) = \sum_{t \in F} \sum_{q \in W(p)} e(I_L(q,t), I_R(\delta(q),t)). \tag{4.11}$$

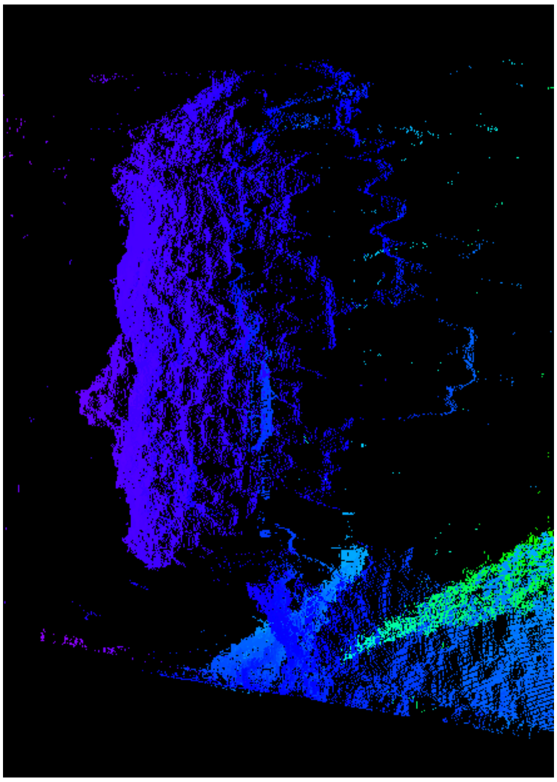


Figure 4.5: Point cloud showing the 3D profile of the face in Fig. 4.3 (left), computed using a single frame and LS_s algorithm.

Minimizing C over d yields an estimated disparity at the pixel p. Note that we obtain added information only if the scene illumination changes within F.

As pointed out in [145], block matching in Equation (4.11) assumes that the disparity d is constant over both the local neighborhood W and the frames F. Assuming for the moment that the scene is static, by using a large temporal window F we can reduce the size of the window W while still reducing matching noise. This strategy has the further salutary effect of minimizing the smearing of object boundaries. Figure 4.1 (top) shows a typical result for spacetime block matching of a static scene with small spatial windows.

Moving objects In a scene with moving objects, the assumption of constant d over F is violated. A simple scheme to deal with motion is to trade off between spatial and temporal window size [28]. In this method, a temporal window of the last k frames is kept, and when a new frame is added, the oldest frame is popped off the window, and $C(p,d)$ is calculated over the last k frames. We will refer to this approach as *sliding*

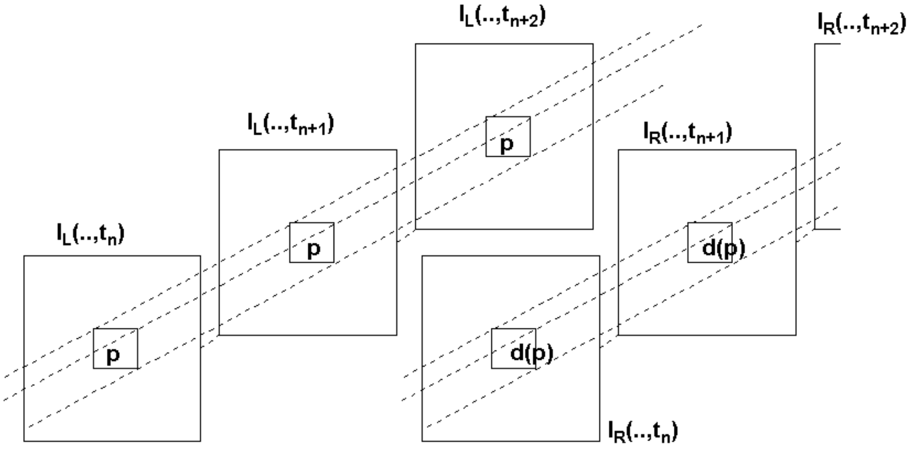


Figure 4.6: Spacetime window for block matching. Spatial patches centered on p are matched against corresponding patches centered on $d(p)$, and the results summed over all frames.

windows (STS-SW). The problem is that any large image motion between frames will completely erase the effects of temporal integration, especially at object boundaries (see Figure 4.1, bottom-left). It is also suboptimal, since some areas of the image may be static, and would benefit from longer temporal integration.

A more complex method is to assume locally linear changes in disparity over time, that is, $d(p,t)$ is a linear function of time [145]:

$$d(p,t) \approx d(p,t_0) + \alpha(p)(t - t_0). \tag{4.12}$$

For smoothly-varying temporal motion at a pixel, the linear assumption works well. Unfortunately, searching over the space of parameters $\alpha(p)$ makes minimizing the block-match sum (4.11) computationally difficult. Also, the linear assumption is violated at the boundaries of moving objects, where there are abrupt changes in disparity from one frame to the next (see Figure 4.7). These temporal boundaries present the same kind of challenges as spatial disparity boundaries in single-frame stereo.

A more sophisticated strategy would be to detect the temporal boundaries and apply temporal smoothness only up to that point. In this way, static image areas enjoy long temporal integration, while those with motion use primarily spatial information. Hence, we propose a novel method with the aim of efficiently dealing with dynamic scenes and rapidly-varying temporal boundaries. In particular, the main idea is to avoid using the spacetime stereo formulation as in (4.11) which blindly averages all points of the scene over time, instead enforcing a temporal smoothness constraint similarly to what is done spatially. In particular, this can be done either modelling the spatio-

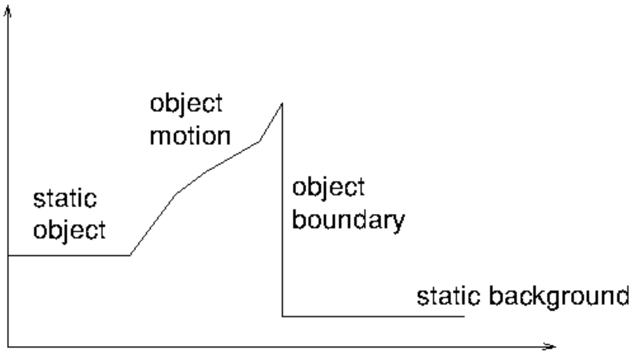


Figure 4.7: Disparity at a single pixel during object motion. Initially disparity is constant (no motion); then varies smoothly as the object moves past the pixel. At the object boundary there is an abrupt change of disparity.

temporal structure with a MRF and solving using an SO or DP-based approach, or enforcing a local smoothness constraint as described in Section 4.1.1.

Temporal regularization using SO The idea of looking for temporal discontinuities was first discussed in [136], which proposed an MRF framework that extends over three frames. The problem with this approach is that the cost in storage and computation is prohibitive, even for just 3 frames. Here we propose a much more efficient method that consists in defining a scanline over time, analogous to the SO method over space. Given a cost array for each point and time instant $C(p, d, t)$ being computed by means of any spatial method (local, global, DP-based, $\cdots$), a SO-based approach is used for propagating forward a smoothness constraint over time:

$$A_{SO}(p, d, t) = C(p, d, t) + \min_{d'}\{A_{SO}(p, d', t-1) + \rho(d, d')\} \qquad (4.13)$$

Instead of backtracking the minimum cost path as in the typical DP algorithm, here it is more convenient to compute the best disparity over space and time as follows:

$$d^*(p, t) = \arg\min_{d}\{A_{SO}(p, d, t)\} \qquad (4.14)$$

so that for each new frame its respective disparity image can be readily computed. As shown in Figure 4.8, accumulated costs from previous frames $t_{i<n}$ are propagated forward to influence the correlation surface at time t_n. Here we propose to use as spatial algorithm the SO-based approach deploying two horizontal scanlines as discussed in Section 4.1.1. This algorithm is referred to as $SO_{s,t}$.

Temporal regularization using local smoothness In a manner similar to applying SO across frames, we can instead use local smoothness. The key idea is to modify the

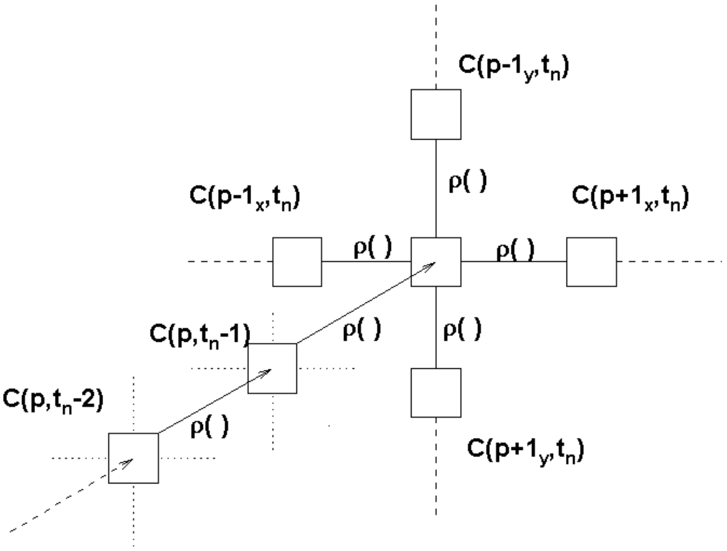


Figure 4.8: Local smoothing applied in the temporal domain. Disparity values influence the center pixel at time t_n from vertical and horizontal directions, and also from previous frames $t_i, i < n$.

correlation surface at position p and time t according to the best disparity found at the same point p at the previous instant $t-1$. This does not require storing and propagating a cost array, only the correspondences found at the previous time instant.

The local temporal smoothness criterion is orthogonal to the strategy adopted for solving stereo over the spatial domain, hence any local or global stereo techniques can be used together with it. Here we propose to use local spatial smoothness described in Section 4.1.1. The cost function at pixel p and time t becomes:

$$\begin{gathered} A_{LS_{s,t}}(p,d,t) = C(p,d,t) + \\ \sum_{q \in \mathcal{N}} \rho\left(d, d(q,t)\right) + \rho\left(d, d(p,t-1)\right), \end{gathered} \tag{4.15}$$

That is, the penalty terms added to the local cost are those coming from the 4 independent scanline-based processes at time t plus an additional one that depends on the best disparity computed at position p at the previous time instant (see Figure 4.8). This algorithm will be referred to as $LS_{s,t}$.

Table 4.1: Percentage of errors, *Cubes* stereo sequence

Radius	STS-SW	Standard	SO_s	$SO_{s,t}$	LS_s	$LS\,s,t$
2	12.8	12.1	1.1	1.0	1.1	0.7

Table 4.2: Percentage of errors, *Cones* stereo sequence

Radius	STS-SW	Standard	SO_s	$SO_{s,t}$	LS_s	$LS\,s,t$
1	46.9	49.9	5.3	5.2	14.8	12.2
3	35.4	15.9	4.2	4.1	8.2	6.9
5	31.9	9.6	4.6	4.5	7.0	6.1

It is possible to propagate information both forwards and backwards in time, but there are several reasons for only going forwards. First, it keeps the data current – previous frames may not be useful for a realtime system. Second, the amount of computation and storage is minimal for forward propagation. Only the previous image local costs have to be maintained, which is $O(M \cdot N)$. In contrast, to do both forwards and backwards smoothing we would need to save local costs over k frames ($O(k \cdot M \cdot N)$), and worse, recompute everything for the previous k frames, where k is the size of the temporal window for accumulation.

Experiments We now present experimental results over two stereo sequences with moving objects and a projected pattern, referred to as *Cubes* and *Cones* (see Fig. 4.3). To obtain ground truth for the stereo data, each different position of the objects is captured over 30 frames with a 3 × 3 spatial window, and stereo depths are averaged over time by means of spacetime stereo. Then, a sequence is built up by using only one frame for each different position of the objects.

As a comparison, we compute spacetime stereo using the sliding window approach (STS-SW). This approach is compared with regularization techniques based only on spatial smoothness (i.e. SO_s, LS_s) as well as with those enforcing temporal regularization (i.e. $SO_{s,t}$, $LS_{s,t}$).

Figure 4.9 shows the error rates of each algorithm for each frame of the *Cubes* dataset, with a fixed spatial window of radius 2. Table 4.1 reports the average error over the whole sequence. In addition, Figure 4.1 shows the ground truth for one frame of the sequence as well as the results obtained by $STS - SW$ and $LS_{s,t}$. As can be seen, due to the rapid shift of the objects in the scene, the approach based on spacetime stereo is unable to improve the results compared to the standard algorithm. Instead, approaches based on spatial regularization yield very low error rates, close to those obtained by the use of spacetime stereo over the same scene but with no moving objects. Furthermore, Figure 4.9 shows that the error variance of the methods enforcing the smoothness constraint is notably lower than that reported by the standard and STS-SW algorithms. It is worth pointing out that the use of the proposed LS regularization technique both in space and time yields the best results over all the considered frames.

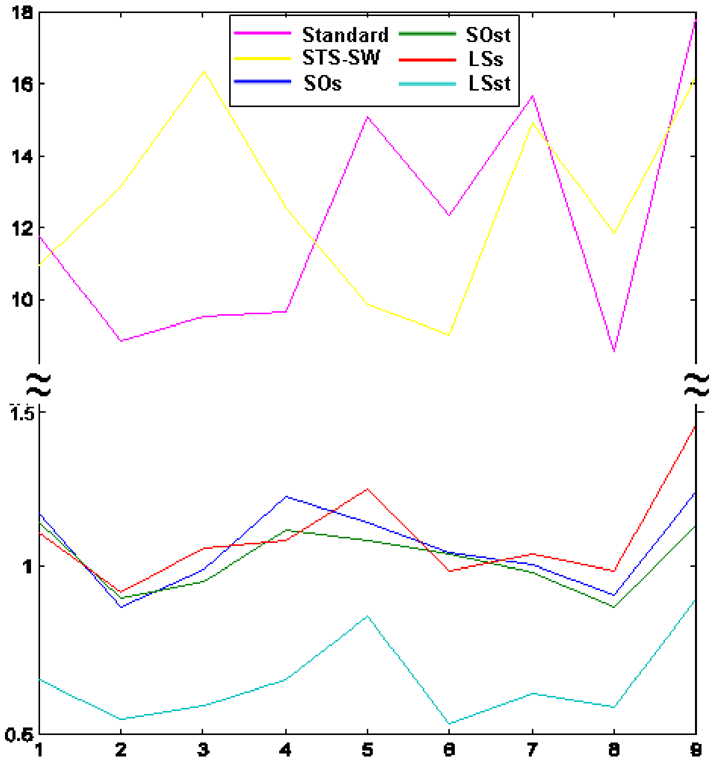


Figure 4.9: Comparison of error percentages between different approaches for the *Cubes* sequence at each frame of the sequence. [The graph uses two different scales for better visualization]

As in the previous experiment, Table 4.2 shows the mean error percentages over the *Cones* dataset with different spatial windows (i.e., radius 1, 3, and 5). Also in this case, regularization approaches achieve notably lower error rates compared to standard and spacetime approaches. From both experiments it is possible to observe that the introduction of temporal smoothness always helps improving the performance of the considered regularization methods.

The code concerning the regularization methods and the STS algorithms used in this paper is open source and available online [2].

[2] available at:
`https://prdev.willowgarage.com/trac/personalrobots/browser/pkg/trunk/vision/spacetime_stereo`
and at:
`https://prdev.willowgarage.com/trac/personalrobots/browser/pkg/trunk/vision/stereo_image_proc`

4.2 Multi-view vandal act detection

This section presents a novel video surveillance approach designed to detect vandal acts occurring on the background of the monitored scene, such as graffiti painting on walls and surfaces, public and private property defacing or etching, unauthorized post sticking. The aim of our approach is to detect this class of events rapidly and robustly. We propose to use two synchronized views to deploy synergically depth and intensity information concerning the monitored scene.

Nowadays vandal acts represent a serious problem in urban areas, with thousands of public and private properties being damaged daily all around the world. Costs issued by this problem are huge: e.g. for the problem of graffiti, that relates to the wide range of markings, etchings and paintings that deface public and private properties, an estimate $ 12 billion a year is spent for cleaning and prevention in the United States [3]. Beside the expenses related to repairing, cleaning and/or substituting a vandalized property, indirect costs arise due to the perceived insecurity associated with the occurrence of vandal acts in a certain area. This typically results in a decrease of revenues for commercial activities or services taking place in the area, such as shops, house tenures, and public transport, for which the uncleanness and perceived insecurity lower passenger confidence in the transport system and consequently tend to decrease ridership. Not less important are the social consequences that repetitive vandal acts in a certain urban area imply on the dwellers.

The effort pushed to tackle - or at least control - the diffusion of vandal acts in urban areas worldwide has often resorted to the use of automatic monitoring system due to the huge amount of public and properties in cities. Commercial products based on audio sensors [6, 8, 79, 125] try to detect vandal acts by analyzing the sounds typically occurring during these actions. These devices present notable limitations, since they detect specific sounds and can hardly generalize to different or noiseless vandal acts. Moreover they can be easily tricked by the presence of environmental noise, and they typically need to stand very close to the monitored region.

Due to these reasons, vision-based approaches relying on automatic video analysis have been recently driving increasing attention. On one hand, all the proposed state-of-the-art vision-based systems [3, 47, 109] rely on a single-view approach, that is they try to recognize vandal acts by processing a video sequence obtained from a single camera. On the other hand, two different classes of algorithms can be outlined.

One approach consists in applying behaviour and gesture analysis techniques for recognizing the high-level spatio-temporal pattern corresponding to the person perpetrating the vandal act. For example, detection of graffiti is carried out in [109] by searching for the pattern corresponding to a person writing on a monitored surface. However, such techniques require accurate training of classifiers and generally perform much better when a certain degree of cooperation from the subject can be achieved, which is obviously not the case of vandal acts.

Another approach relies on comparing the current appearance of the monitored object with that of a background model of the scene. Hence, this class of methods can detect only vandal acts which produce visible and stationary changes of the appearance of the monitored scene. We will refer to this class of events as *Stationary Visible Changes* (SVC), which includes paintings on walls and surfaces, public and private property defacing, etching or stealing, unauthorized post sticking. This also concerns other scenarios such as, e.g. for cultural heritage environments or museums, criminal acts such as tearing, dirtying, defacing, stealing of parts of an artwork. Detection of vandal acts by recognizing SVC is carried out in [3, 47].

In particular, in [3] the authors focus on graffiti and propose a low-level approach for SVC recognition based on single-view change detection. This approach inherently suffers from a false positives problem when deployed as vandal acts detectors. In fact, SVC events include most of the common aforementioned vandal acts, but also other frequent events such as people standing still, parked vehicles (such as cars, motorbikes, bicycles), abandoned objects. The same issue arises with method [47], which concerns a fast single-view SVC detector based on the analysis of higher-level events occurring in the monitored scene. This problem is partially dealt with by limiting the detection only on a subset of the camera field-of-view and by assuming that the monitored scene is not crowded. Finally, robustness with regards to sudden illumination changes occurring in the scene is not investigated.

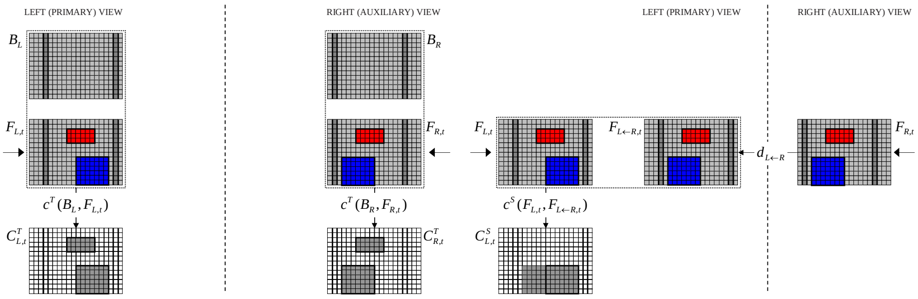
Our idea is to go beyond the visibility and stationarity cues in order to obtain a finer classification of SVC. This should allow for a more effective detection of specific vandal acts based on the recognition of their peculiar effects. To this purpose, we propose to deploy also depth information, so that the class of SVC events can be partitioned into the two following mutually-exclusive sub-classes:

a) *Stationary Appearance Changes* (SAC): stationary visible changes due to variations of the appearance but not the 3D geometry of the scene.

b) *Stationary Geometric Changes* (SGC): stationary visible changes due to variations of the 3D geometry of the scene;

It is clear that most of the vandal acts that previous proposals try to detect as SVC are indeed SAC, for they determine no (or small) variation of the scene 3D geometry, while most false positives have to be ascribed to SGC.

Hence, we propose to effectively detect vandal acts based on the ability of distinguishing between SAC and SGC. In particular, we present a real-time SAC detection algorithm based on the use of two synchronized views of the monitored scene and on



(a) temporal change detection: only the VC super-class can be recognized

(b) spatial change detection: only the GC sub-class can be recognized

Figure 4.10: Temporal and spatial change detection

a novel multi-view change detection approach. This deploys on-line intensity information coming from the two image sensors together with knowledge of the 3D structure of the monitored scene, which is obtained once at initialisation time by means of a stereo matching process. This enables to detect effectively SAC events even in presence of static subjects that produce SGC. The proposed method can work within unstructured environments and does not pose any constraint on the appearance and geometry of the background of the monitored scene. Moreover, by means of a specific stage which is robust with respect to non-linear photometric distortions, our approach can also handle strong sudden illumination changes and shadows. Finally, our system can alert the occurrence of vandal acts while they are being committed.

4.2.1 Principles of multi-view change detection

The input information to our approach is represented by two synchronized video sequences of a scene characterized by a considerable overlap of field-of-views. Moreover, we assume stationarity of the capturing devices as well as of the scene background geometry, so that geometric registration of the background over the two views, hereinafter denoted as left view (L) and right view (R), can be computed only once at initialization time. Apart from stationarity, no further assumption is made about geometry of the background surface which, in particular, is not constrained to be planar.

The goal of our approach is to compute in one of the two views, referred to as *primary*, a binary mask highlighting the pixels which are sensing a SAC, that is, at the event level, the effect of a vandal act. To this purpose, we use a novel multi-view change detector to carry out the twofold task of detecting VC and, among these, discriminating between AC and GC. Then, a simple procedure is used to evaluate stationarity of AC. To better illustrate our proposal, in this section we outline some basic principles concerning multi-view change detection and, contextually, review the state-of-the-art in the field.

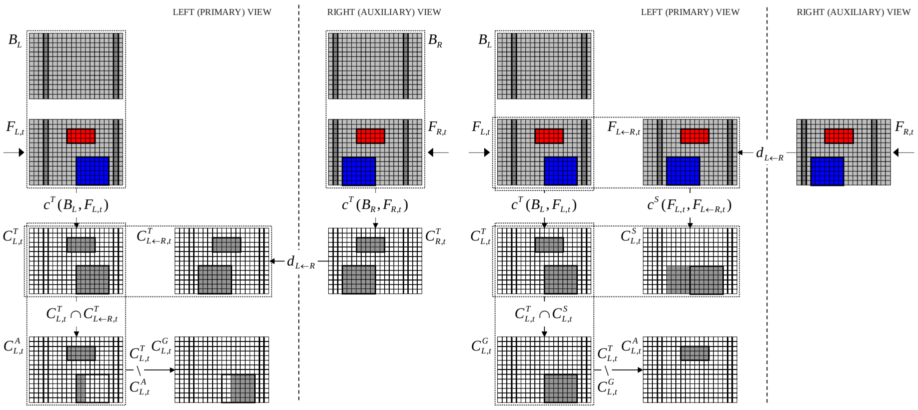
As regards the way the input information (i.e. the two synchronized video sequences) can be exploited for detecting changes with respect to a reference scene, we define:

a) *temporal consistency constraint*: for a given view-point, the processed frames are images of the same scene taken at different times;

b) *spatial coherence constraint*: for a given elaboration time instant, the processed frames are images of the same scene taken from different view-points;

The temporal consistency constraint can be exploited to perform a *temporal change detection* independently in each view by a classical background subtraction procedure. That is, at each time t the current frames $F_{L,t}$ and $F_{R,t}$ are compared with as many off-line generated view-dependent appearance models B_L and B_R of the reference scene, that we call *temporal backgrounds*. Two *temporal change masks* are thus obtained, that is two binary masks $C^T_{L,t}$ and $C^T_{R,t}$ comprising the pixels which are currently sensing a violation of the temporal consistency constraint. Temporal change detection is illustrated in Figure 4.10(a) by means of a toy example consisting of a planar background (light grey with two darker vertical strips), a parallel-axis stereo sensor with the two optical axes perpendicular to the background, and AC (red) as well as GC (blue) events being sensed in the current frames. As one can easily understand and as pointed out in Figure 4.10(a), generally speaking temporal change detection allows for detecting, independently in each view, the super-class of VC events but not for discriminating between the AC and GC sub-classes. This is due to the fact that recoverying depth information from a single view is in principle an ill-posed problem.

Exploitation of the spatial coherence constraint yields the simplest multi-view change detection approach, proposed in [64], that we call *spatial change detection*. The *spatial background*, unlike temporal ones, does not store appearance but geometric information about the monitored scene. In fact, it consists in the disparity map $D_{L \leftarrow R}$ (computed off-line) warping the monitored scene from the auxiliary to the primary view. Spatial background subtraction is thus performed by a background disparity verification. That is, at each time the auxiliary frame $F_{R,t}$ is warped into the primary view by the background disparity map and then compared with the primary frame $F_{L,t}$ to obtain a *spatial change mask*, a binary mask $C^S_{L,t}$ highlighting the pixels which are currently sensing a violation of the spatial coherence constraint. As illustrated in Figure 4.10(b), only the GC sub-class can be recognized by spatial change detection. In fact, AC events occur on the background surface and, hence, are coherent with respect to the background disparity map. Moreover, the method suffers from an intrinsic false positives problem, called *occlusion shadows*. In fact, the background pixels in the primary view which are occluded by a foreground object in the auxiliary view are in-



(a) joint use of temporal and spatial change detection as proposed in [8,10]

(b) joint use of temporal and spatial change detection as proposed in our approach

Figure 4.11: Joint exploitation of temporal and spatial change detection

herently detected as changed. To deal with this problem, in [64] the authors propose to exploit more than one auxiliary view and to compute the intersection of the binary masks obtained by comparing the primary with each of the auxiliary views. In [84] the problem is addressed from a sensor planning perspective. In particular, it is shown how occlusion shadows can be removed by using just two views if a suitable sensors configuration is adopted.

The combined exploitation of both the temporal consistency and the spatial coherence constraints is proposed in [68] and [76]. Essentially, both approaches rely on the idea, illustrated in Figure 4.11(a), of first performing temporal change detection in each view and then carrying out spatial change detection based on the obtained temporal change masks. This should allow to obtain the AC mask $C^A_{L,t}$ and, by subtraction from the temporal change mask, the GC mask $C^G_{L,t}$. However, as pointed out in Figure 4.11(a), these methods inherently suffer from missed detection in the GC mask and, as discussed in detail in Section 4.2.2, from false detections in the AC mask.

4.2.2 The proposed algorithm

The proposed graffiti detection algorithm relies on a novel multi-view change detection approach. The novelty consists in a simple yet clever way of combining temporal and spatial change detection so as to perform an effective discrimination between AC and GC. To better illustrate the approach we will distinguish between off-line and on-line elaboration steps. Once the AC events are detected, the proposed procedure for the recognition of stationary AC (SAC) can be regarded as a post-processing step. Hence, it will be described in a separate section together with a simple binary morphology stage applied on the final change mask.

Off-line elaboration

The very first step concerns the calibration of the stereo sensor, which aims at estimating, for each view, the calibration parameters, a set of optical distortion parameters and a rectification homography. This information is condensed into two geometrical transformations $g_L(\cdot)$ and $g_R(\cdot)$ that will be used at each processing time - both off-line and on-line - to compute the undistorted and rectified versions $F_{L,t}$ and $F_{R,t}$ of the captured frames $F^c_{L,t}$ and $F^c_{R,t}$, respectively. In formulas:

$$F_{L,t} = g_L(F^c_{L,t}) \qquad F_{R,t} = g_R(F^c_{R,t}) \tag{4.16}$$

Hence, for each view a short bootstrap sequence of N frames (N in the order of tens) is used to infer an appearance model of the reference scene, i.e. the temporal background:

$$B_L = b(F_{L,1}, \ldots, F_{L,N}) \qquad B_R = b(F_{R,1}, \ldots, F_{R,N}) \tag{4.17}$$

with b denoting a generic - possibly robust - pixel-wise statistical estimator. In the experiments shown in Section Section 4.2.3 we have used the median operator. The two temporal backgrounds are thus fed to a dense stereo matching algorithm so as to compute the disparity map warping the reference scene from the auxiliary (right) to the primary (left) view, i.e. the spatial background:

$$D_{L\leftarrow R} = m(B_L, B_R) \tag{4.18}$$

It is worth pointing out here that this operation aimed at obtaining the spatial background needs to be obtained once and for all at initialisation time, hence on-line stereo matching is not required by our method. Therefore, with our approach one should deploy an as accurate as possible, even though slow, stereo matching algorithm, so as to maximize the accuracy of the warping function. In the experiments shown in Section Section 4.2.3 we have used the algorithm described in [58].

On-line elaboration

The main on-line processing steps performed by the proposed algorithm are illustrated in Figure 4.11(b) by means of the same toy example used in the previous section.

First of all, temporal change detection is performed in the primary view by background subtraction, that is by comparing the current frame $F_{L,t}$ with the off-line generated temporal background B_L, so as to compute the temporal change mask $C^T_{L,t}$:

$$C^T_{L,t} = c^T(B_L, F_{L,t}) \tag{4.19}$$

In particular, to achieve robustness with respect to strong photometric distortions we apply at pixel-level the block-level approach presented in [75]. This algorithm is able to filter-out illumination changes yielding locally order-preserving transformations of pixel intensities.

Spatial change detection is then performed. To this purpose, the auxiliary frame $F_{R,t}$ is warped into the primary view by the $D_{L\leftarrow R}$ background disparity map:

$$F_{L\leftarrow R,t} = d_{L\leftarrow R}(F_{R,t}) \tag{4.20}$$

The spatial change mask $C^S_{L,t}$ is thus obtained by comparing the primary frame $F_{L,t}$ with the warped auxiliary frame $F_{L\leftarrow R,t}$:

$$C^S_{L,t} = c^S(F_{L,t}, F_{L\leftarrow R,t}) \tag{4.21}$$

Differently from temporal change detection, here the compared frames are synchronized. Hence, under the assumtpion of lambertian surfaces, illumination changes occurring in the monitored scene affect in the same way the amount of radiation incident onto the two sensors. Nevertheless, in general the two sensors can produce different measures (i.e. image intensities) due to the presence of non-lambertian surfaces, to a different foreshortening of the objects in the two views and different camera parameters (e.g. gain, exposure).

For this reason, also in this case a robust change detection algorithm is desirable. We propose to use a block-based approach and the well-known Normalized Cross-Correlation (NCC) measure, due to its simplicity and its constant complexity. This measure is invariant to linear photometric distortions. It is also worth to point out that the computation of $C^S_{L,t}$ by means of the NCC measure can be efficiently performed using incremental schemes [24, 91], so that complexity turns out independent on block size.

As discussed in the previous section and clearly outlined in Figure 4.11(b), on one hand the temporal change mask comprises the super-class of pixels sensing a VC, while on the other hand the spatial change mask contains the sub-class of GC pixels and the false positives corresponding to occlusion shadows. Hence, by computing the intersection of the two masks the geometric change mask $C^G_{L,t}$ containing GC pixels can be easily obtained:

$$C^G_{L,t} = C^T_{L,t} \cap C^S_{L,t} \tag{4.22}$$

Finally, it is straightforward to compute the appearance change mask $C^A_{L,t}$ by subtracting the geometric from the temporal change mask:

$$C^A_{L,t} = C^T_{L,t} \setminus C^S_{L,t} \tag{4.23}$$

Summarizing, in principles this mask should include only those pixels that are currently sensing a change of the background appearance related neither to an illumination change nor to a variation of the background geometry. Given the addressed application domain, such changes can be ascribed to acts of vandalism.

It is worth pointing out that in the method of Figure 4.11(a) disparity verification occurs on the binary temporal change masks. As a result, the method will definitely yield false AC in correspondence of the overlapping areas between GC regions found in the primary view and in the warped auxiliary view. Conversely, with our approach disparity verification is carried out between the original frames, as in 4.10(b). Hence, for the pixels belonging to the above mentioned overlapping areas a decision is taken based on photometric similarity according to the NCC measure. Since in such areas overlapping between different parts of a foreground object is likely to occur (e.g. the left and right shoulder of a person perpetrating a vandal act), unless the object is untextured, it is likely that photometric dissimilarity will allow for a correct classification as spatial changes. As a consequence, our method will unlikely yield false AC.

Stationarity and morphology

Since the addressed acts of vandalism yield permanent and static modifications of scene appearance, we can exploit the further constraint that AC detected by the proposed multi-view change detector have to be stationary. To this purpose, we propose to use a simple procedure based on a *post-processing* and *pixel-wise* approach. That is, at each time t a pixel sensing an AC is classified as a SAC if the the appearance change is persistent over a given interval of previous frames. In formulas:

$$C_t^{SA}(\boldsymbol{p}) = C_t^A(\boldsymbol{p}) \wedge C_{t-1}^A(\boldsymbol{p}) \ldots \wedge C_{t-k}^A(\boldsymbol{p}) \tag{4.24}$$

where C_t^{SA} denotes the obtained SAC binary mask and the subscript L is drop for simplicity. Similarly, a persistent absence of AC is required to switch off a SAC pixel in C_t^{SA}.

Finally, to refine the computed SAC binary mask and remove small false positives and false negatives we apply a simple two-steps morphological filtering consisting of an area-opening and a morphological closing. The obtained graffiti blobs are then labelled and their bounding-boxes are extracted.

4.2.3 Experimental results

This subsection presents experimental results aimed at evaluating the capabilities of the proposed approach to detect typical SAC events under real conditions. In particular, we have implemented the proposed algorithm in C code using off-the-shelf hardware which includes a PC with an AMD Athlon 2.21 GHz core processor and a very cheap stereo setup represented by two web-cams.

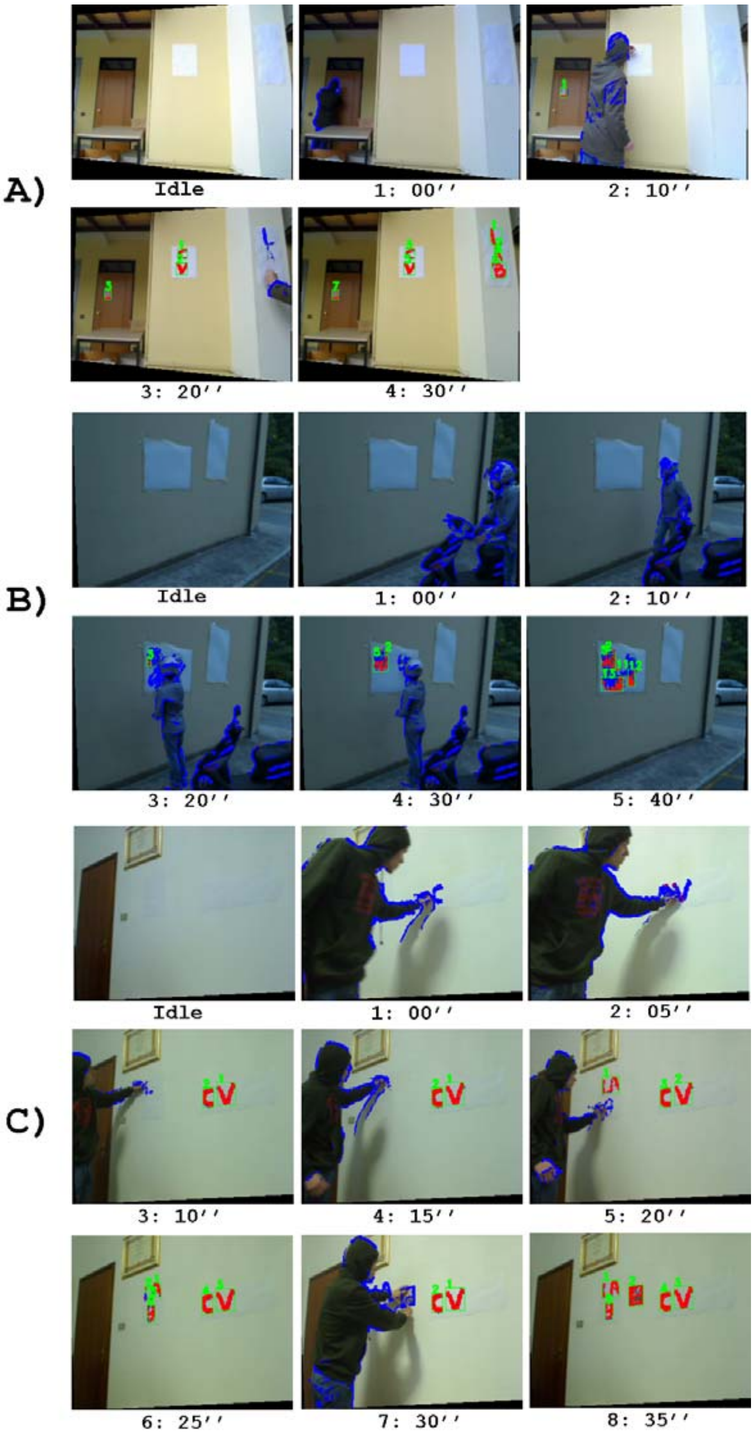


Figure 4.12: Results dealing with the 3 *Graffiti* sequences (to be viewed in color)

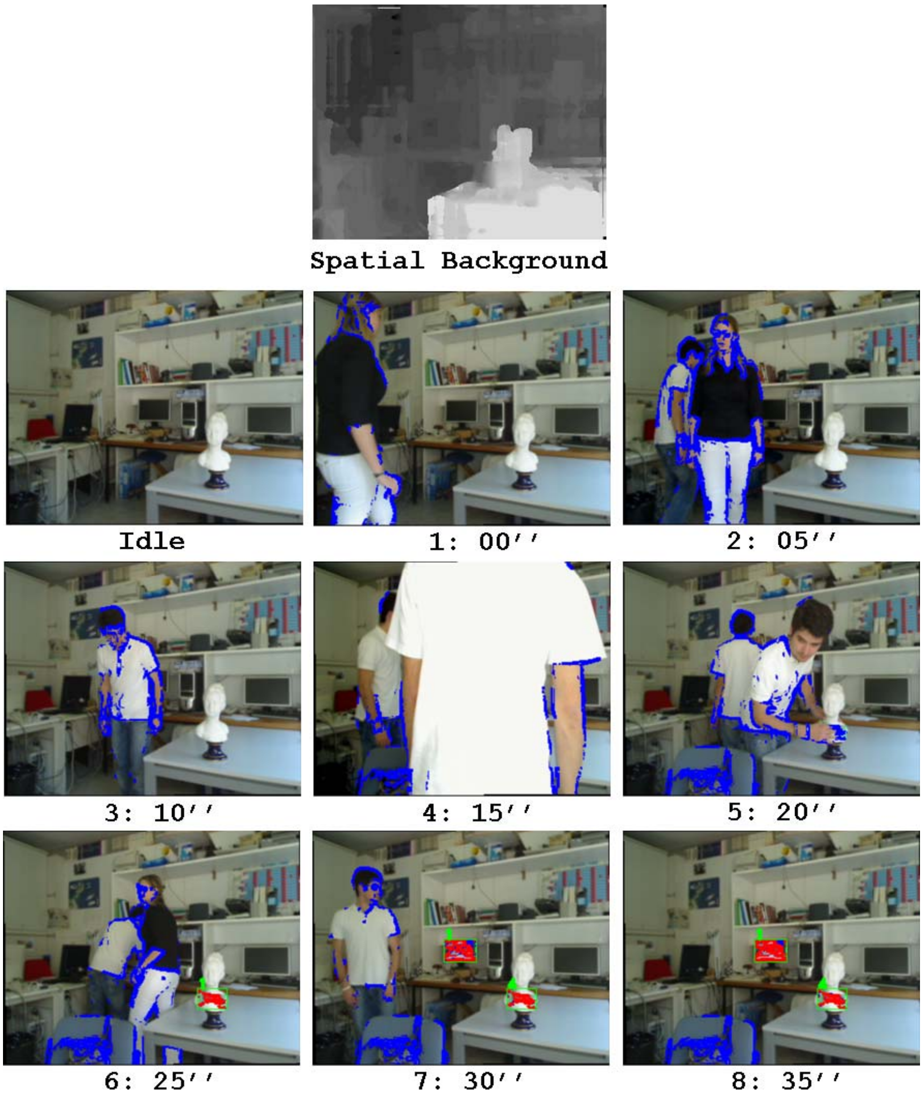


Figure 4.13: Results dealing with the *Statue* sequence (to be viewed in color)

Figure 4.12 shows the results dealing with three video sequences (*A*, *B* and *C*) concerning the particular case of graffiti detection. Sequences *A*, *B* refer to an outdoor environment, while sequence *C* refers to an indoor scene. For all sequences, the top left frame shows the idle appearance of the scene (i.e. the background), while remanining frames show the output of the system sampled every 10 or 5 seconds (depending on the dynamics of the event) starting from the beginning of the vandalic action. In particular, the output depicts with blue pixels those points currently detected as GC, while in red those points currently detected as AC. Finally, when a SAC event is detected (i.e. after post-processing) a green bounding box with a numbered label highlights the area where the action is taking place.

In the *Graffiti A* sequence the background is represented by three textureless slanted walls at different depths on which a person posts up a flyer and draws some graffiti, while in the *Graffiti B* sequence the background is mainly composed by a textureless slanted wall. In both sequences it is worth to note that our approach is able to accurately detect the graffiti events at different depths while the action is occurring. Moreover it is worth pointing out the notable absence of false positives throughout the whole sequences. It is also interesting to note that SGC events are currently discriminated from SAC events (e.g. in *Graffiti B* sequence, the motorbike which is parked in front of the background during the vandal act).

For what concerns the *Graffiti C* sequence, the background is represented by a white slanted wall on which a person draws some graffiti and posts up a flyer. Also in this case, graffiti are correctly and on-line discriminated from GC. Similarly to the previous sequence, false positives are absent along all frames despite the notable presence of shadows on the background. In this case, the adopted robust temporal change detection algorithm allows to reject the majority of shadow points as visible changes (frames 1-5, 7), the remaining ones being discarded by stationarity and morphology (frames 1-5).

Fig. 4.13 refers to a more general case of vandal acts detection over a complex background. In this case, referred to as *Statue* sequence, the background is constituted by a table and a small statue close to the sensor, plus a variegated group of objects at a further distance. The background models for the two views together with the corresponding disparity maps are shown on the top of the figure. Similarly to the previous cases, frames 1-9 show the output of the system sampled every 5 seconds.

Beside the complex background and the not perfectly synchronized stereo sensor, challenges are also introduced by the events taking place in the scene. That is, different people are moving simultaneously (frames 3, 5-7) even close to the camera (frame 5). Furthermore, a chair is placed in the scene (frames 5-8), this event being correctly not classified as SAC since it represents a SGC. SAC events are represented by defacing of the statue (between frames 6 and 7) and by switching on a monitor (between frames 7 and 8). These events are correctly and accurately detected (frames 7-9). Besides, a person standing still (frame 8) does not produce any false positive since, again, it correspond to a SGC. As for computational requirements, the proposed approach can efficiently process video frames at an average rate of 10 fps.

4.2.4 Graffiti detection using a Time-of-Flight camera

This subsection proposes a system based on a TOF (Time-of-Flight) camera to perform automatic graffiti detection. The rationale of this study is that since a TOF camera senses both brightness and depth at each pixel location, it may be deployed to de-

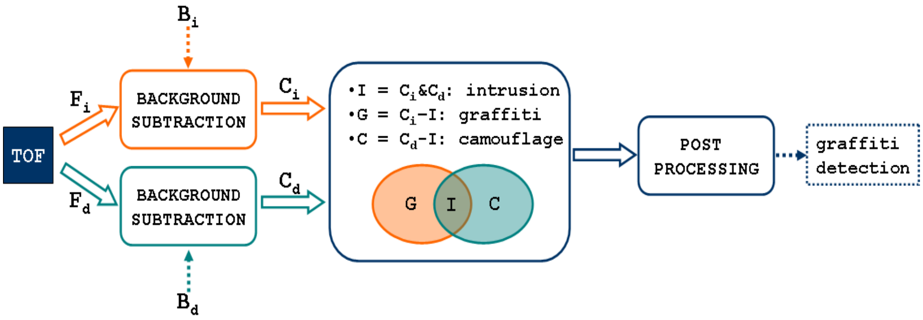


Figure 4.14: Outline of the proposed algorithm

tect graffiti by looking for stationary changes of brightness that do not correspond to changes in depth. It is clear that, analogously to the use of a stereo camera, also the use of a TOF camera holds the potential to overcame the false positives issue of the method in [3], for most still objects other than graffiti would yield both brightness and depth changes. Furthermore, it is worth noting that the same idea can be usefully employed to detect further events rather than only graffiti, such as modifications to the background surfaces. This can be useful, e.g., for cultural heritage environments or museums, to detect acts of vandalism such as painting, dirtying, etching or defacing of parts of an artwork.

The proposed algorithm for graffiti detection jointly deploys depth and intensity information to detect events such as changes in the appearance of the visible surfaces in the monitored scene. The basic idea of our algorithm can be outlined as follows. First, by means of an intensity-based analysis visible changes can be detected by comparing the current intensity information with a model of the background of the scene. Then, the use of depth information can discriminate between changes occurring in the space between the background and the camera (e.g. intrusion) and those occurring directly on the background surface (e.g. graffiti). The outline of the proposed algorithm, described hereinafter, is shown in Fig. 4.14.

The proposed approach is based on background subtraction [106]. In order to compare the background model with each frame, we adopt a basic background subtraction approach, i.e. we compute a change mask C_i by thresholding the absolute difference between each pixel intensity in the background B_i and in the current frame F_i:

$$C_i(x,y) = \begin{cases} false & |B_i(x,y) - F_i(x,y)| < T \\ true & elsewhere \end{cases} \tag{4.25}$$

Thus, C_i is a binary mask denoting all points which result *changed* after the comparison with the background. In the basic model, T is a fixed parameter of the algorithm.

Nevertheless, when dealing with TOF cameras, the amount of noise perceived by each pixel is lower at the center of the image, where the power of the reflected light signal is higher, and increases as we get far from it. Hence, T should depend on the position (x, y) where the change mask is being evaluated. In particular, we assume that noise can be modeled as a zero-mean Gaussian distribution:

$$F_i(x, y) = F_i^T(x, y) + N_i(x, y) \tag{4.26}$$

with $F_i^T(x, y)$ being the noise-free version of $F_i(x, y)$ and

$$N_i(x, y) = \frac{1}{2\pi \cdot \sigma_i^2(x, y)} \cdot exp\{-\frac{(x^2 + y^2)}{2 \cdot \sigma_i^2(x, y)}\} \tag{4.27}$$

Hence, parameter T in (4.25) is chosen to be proportional to the standard deviation of the Gaussian distribution of each pixel, which is estimated during the initialization sequence. This leads to

$$C_i(x, y) = \begin{cases} false & |B_i(x, y) - F_i(x, y)| < k_i \cdot \sigma_i(x, y) \\ true & elsewhere \end{cases} \tag{4.28}$$

with k_i typically ranging within $[1, \cdots, 3]$ ($k_i = 1$ in our experiments).

The same approach can be carried out also for what concerns the depth information coming from the TOF sensor. In particular, a depth background model B_d can be built by averaging the depth value of each point of the depth map over an initialization sequence, assuring the background is static along that sequence. Moreover, for each point (x, y) the standard deviation $\sigma_d(x, y)$ of the depth values over the initialization sequence is also computed. Then, similarly to (4.28), the current depth map F_d can be compared at run-time to the depth background model B_d :

$$C_d(i, j) = \begin{cases} false & |B_d(i, j) - F_d(i, j)| < k_d \cdot \sigma_d(x, y) \\ true & elsewhere \end{cases} \tag{4.29}$$

with k_d typically ranging within $[1, \cdots, 3]$ ($k_d = 1.5$ in our experiments).

Once C_i and C_d are computed, they are compared so to determine the presence of graffiti in the scene. In particular, the event of a point (x, y) resulting *changed* in either one of the two masks refers to one of the following three possible circumstances:

1. C_i = true, C_d=true: a change in intensity corresponds to a change in depth. This means that an intrusion by something/someone is currently going on.

2. C_i = true, C_d=false: a change in intensity does not correspond to a change in depth. Thus, a change of the appearance of the background surface has occurred: a graffiti event is triggered.

3. C_i = false, C_d=true: a change in depth does not correspond to a change in intensity. In this case, an intrusion has been performed by something/someone having an intensity similar to that of the background (i.e. *camouflage*).

Thus, graffiti are detected simply by choosing all points marked as *changed* on C_i and as *unchanged* on C_d.

In addition, we carry out a final post-processing stage in order to improve the reliability of our detector. Along this last stage, three different conditions are checked in order to eliminate false positives from the final graffiti mask. First, a stationarity check is performed, that is a point is detected as graffito only if it was positively detected in the last t frames. This is a necessary measure against the high amount of noise of the camera sensor, which otherwise would produce a high number of flickering points in the final change mask. Successively, a labeling algorithm is applied on the detected graffiti regions. This allows to eliminate from the final change mask all graffiti whose area is less than a certain number of pixels, which are as well typically generated by noise. Finally, the last check eliminates all graffiti blobs having any of their 8-connected neighboring points detected as 3D intrusions in C_d, since they are most probably generated by the parts of an intruding object/person laying close to the background surface.

Experimental results

We now show some preliminary results dealing with the application of our graffiti detector to real video sequences. Unfortunately, due to the current limits of the TOF technology and, above all, due to the characteristics of TOF camera (Canesta DP205, Field of View: 55 deg.) available to us for the experiments, which is not state-of-the-art, resolution is limited to 64×64 for both intensity and depth. Furthermore, the power of the infrared illuminator limited the maximum depth range during our tests, forcing the camera to stand not farther than 1.5~2 meters away from the background walls, otherwise the sensor is unable to detect the majority of details appearing on the background surface. In fact, at a farther distance, the intensity image tends to appear very dark.

Thus, we now show some footage dealing with some typical acts of vandalism which can be detected by our system. These acts include graffiti (e.g. writings on a surface, Video 1), object stealing (e.g. stealing a painting or a drawing hung up on the wall, Video 2), surface defacing or damaging (e.g. tearing apart a drawing on the wall, Video 3). For each sequence, we show some qualitative results by uniformly taking some snapshots of the outputs of the various stages of the algorithm along the whole sequence. In particular, for each snapshot we show the current intensity frame F_i, the current depth frame F_d, the intensity change mask C_i, the depth change mask

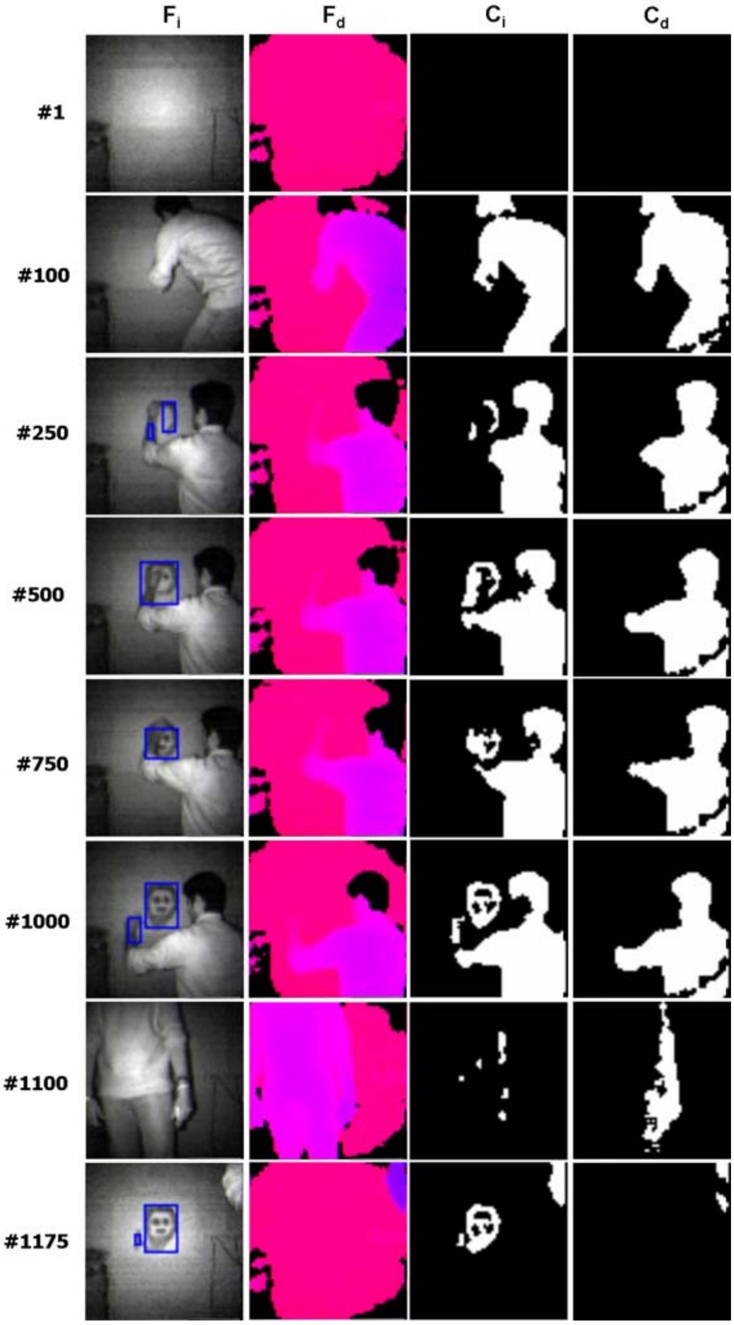


Figure 4.15: Video 1: graffiti sequence

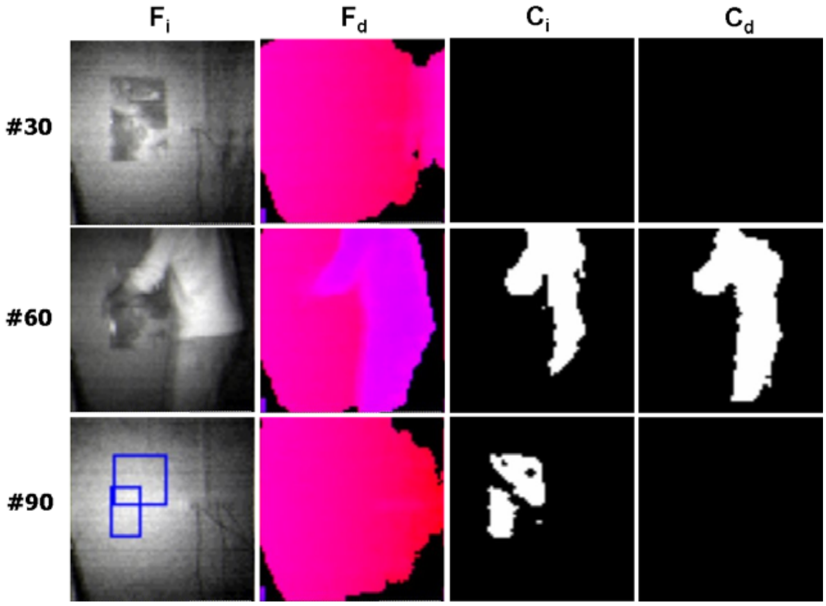


Figure 4.16: Video 2: object stealing sequence

C_d. Besides, we also superimpose on F_i the final output of our algorithm, which is a bounding box around each graffiti blob detected by our system.

Fig. 4.15 shows the results dealing with the graffiti sequence (Video 1). At the very beginning of the sequence, the scene is empty, with a white wall on the background (Frame ♯1). Then, a person enters the scene and starts making some drawings on the wall (Frames ♯100, 250, 500, 750, 1000). As soon as the graffiti start being visible on the scene (Frame ♯250), the system detects their presence and localizes them quite accurately. It is worth noting that in Frames ♯250 and ♯1000 2 false positives arise due to the fact that the person's arm, laying on the wall, is almost at the same depth as the background and is recognized as a graffito. Then, when the person stands in front of the drawing, no graffito is detect in the output (Frame ♯1100). Finally, Frame ♯1175 shows the output of the system at the very end of the sequence.

As said before, the proposed algorithm can be usefully deployed also to detect other events rather than just graffiti. In Fig. 4.16 results are showed concerning a sequence where a painting hanging on the wall is stolen (Video 2). The object appears at the beginning of the sequence (Frame ♯30). While a person is stealing the painting, no output is raised since an intrusion is present but the background has not structurally changed yet (Frame ♯60). Finally, when the object is removed the event is correctly detected by our system (Frame ♯90).

Finally, we show a sequence concerning the defacing of an object hanging on the

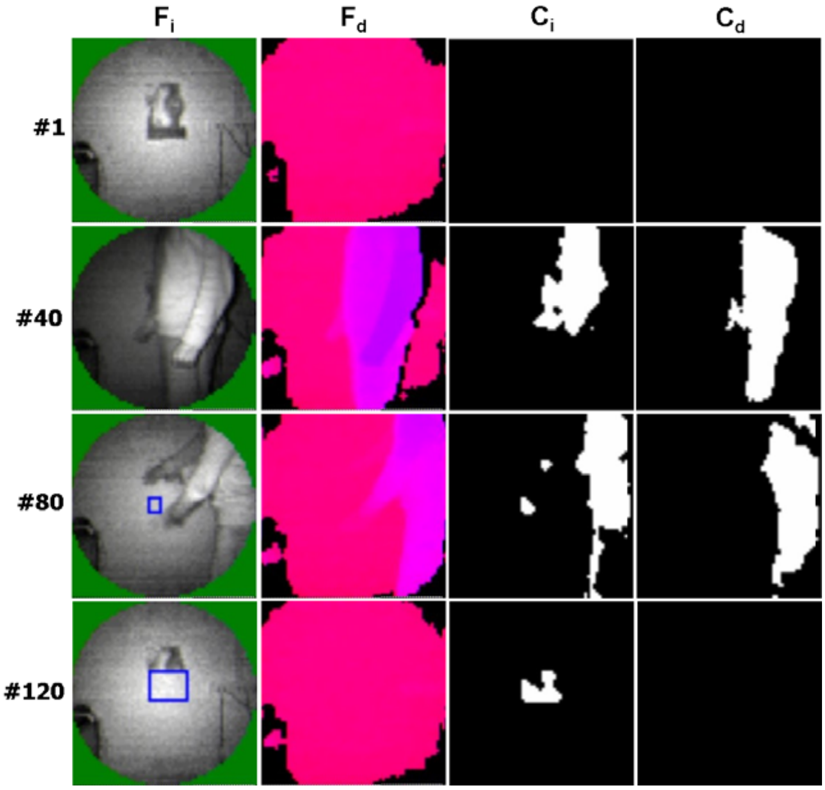


Figure 4.17: Video 3: object defacing sequence

wall (Video 3). In this last case, we also propose a slight modification to the output of the algorithm. It is easy to note, from the various depth frames F_d shown in Figg. 4.15, 4.16, that the depth map computed by the TOF sensor is rarely able to determine a good depth estimation of the scene on the regions around the 4 corners of the map. This is mainly due, as previously said, to the amount of noise which increases as the distance of point from the image center increases, and it is maximum around the 4 corners, which are the points laying farthest from the center. This phenomenon is also evident for points belonging to the intensity frames. Hence, as a consequence, regions around the four corners of the image are highly unreliable, their depth and intensity variances being extremely high. In practice, this increases the chances of having false positives around those regions. Hence, we propose to use a binary mask which excludes the graffiti detection over these points, which can be regarded as peripheral regions where detection can not be performed. The output frames of Video 3 (Fig. 4.17, left column) show in green this mask.

The first frame of the sequence (♯1) shows a painting hanging on a white background wall. Then a person enters the room (Frame ♯40) and starts tearing apart the painting (Frame ♯80). Correctly, only when defacing is being performed, the algorithm produces an alarm (Frame ♯80). At the very end of the sequence (Frame ♯120) defacing is correctly detected, as only the lower half of the painting (the part which has been torn apart) is being highlighted by the bounding box.

4.3 Video surveillance

Detecting motion in video sequences is a fundamental requirement for many higher-level vision tasks such as object classification, tracking, event detection (e.g. stolen or abandoned object). A common domain of application for such tasks is video-surveillance, e.g. with the aim of detecting intrusions. Major issues related to the motion detection process are as follows. It is difficult to correctly segment out moving objects when they look similar to the background of the scene (*camouflage*). The motion detection process is very sensitive to sudden illumination variations of the scene. It is difficult to filter out shadows from the detected foreground. By using more than one view, e.g. by means of a stereo camera, it is possible to obtain 3D information on the surveyed scene. This allows to exploit, in addition to scene radiance information, also depth information concerning scene 3D structure, so as to achieve higher robustness with regards to one or more of the above mentioned issues [14, 39, 53, 84]. In the next we propose and compare two change detection strategies based on two views which exploits 2D and 3D information in order to deal with typical change detection issues.

4.3.1 Proposed approach

This subsection describes a novel change detection approach which jointly exploits depth information coming from a 3D device and 2D brightness information. Information on scene changes is recovered by means of two different strategies. The former, referred to as *3D Output*, mainly relies on depth information, and aims at being robust to camouflage, shadows and sudden illumination changes. The latter, referred to as *2D Output*, aims at obtaining robustness with regards to sudden illumination changes as well as accuracy in foreground segmentation. The final change masks determined by the two outputs will be referred to as, respectively, C_{3D} and C_{2D}.

As depicted in Fig. 4.18, the proposed approach can be outlined as a 4-stage algorithm. The imaging sensor used is a stereo camera. In particular, we assume that a calibrated stereo setup is available, so that the image pairs retrieved from the camera can be properly rectified.

Stereo Matching For each new rectified image pair coming from the stereo device, a stereo matching algorithm (see [112] for a survey on this topic) is used in order to

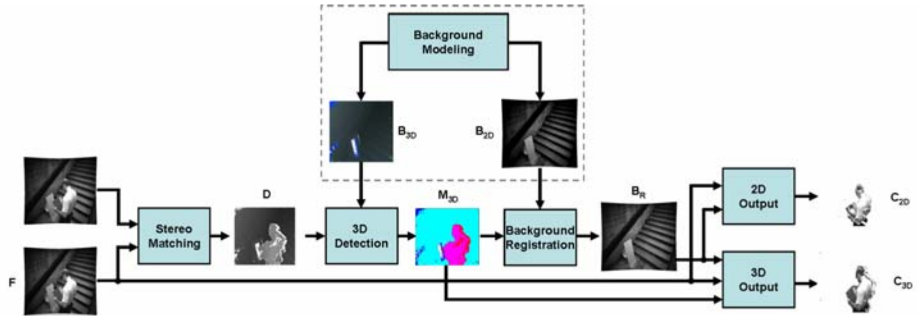


Figure 4.18: Flow diagram of the proposed approach

retrieve a dense disparity map relative to the observed scene. In particular, in our experiments we have used two different local stereo matching algorithms. The former, referred to as SMP (*Single Matching Phase*) [34], is an algorithm that allows to obtain dense disparity maps in real-time. The latter, referred to as *Variable Windows* [130], is an algorithm that holds the potential to retrieve more accurate depth borders compared to SMP thanks to the use of a variable aggregation stage, even if at a higher computational cost which renders it slower than SMP (near real-time).

We first describe briefly how SMP computes disparities for each point p_r of the reference image. Let I_r, I_t be respectively the reference image and the ”other” image. Given a disparity range D, for each point $p_{t,d}$ on I_t belonging to the disparity range induced by p_r a similarity measure is applied between a squared window centered on p_r and all squared windows centered on $p_{t,d}$. The adopted similarity measure is the SAD (*Sum of Absolute Differences*). The selected disparity d_r for point p_r is that relative to the point $p_{t,d}$ that yielded the lowest SAD score on its window. Then *uniqueness*, *distinctiveness* and *sharpness* constraints (see [34] for details) are used to eliminate ambiguous disparity values. Hence, the pixels of the final disparity maps obtained by SMP are labeled either as valid disparity values, or as points violating the constraints (referred here to as non-matched points, NM). SMP relies on incremental calculations techniques [91] and delivers disparity maps in real-time.

Differently to SMP, Variable Windows uses the Birchfield-Tomasi [12] measure as a point wise matching cost and selects the more appropriate aggregation support evaluating a useful range of window sizes/shapes. Although slower than SMP, also this approach relies on incremental calculation techniques (i.e. integral images [24, 131]) for efficient disparity maps computation. The matching selection is based on a Winner Takes All (WTA) strategy and hence all points are labeled as valid.

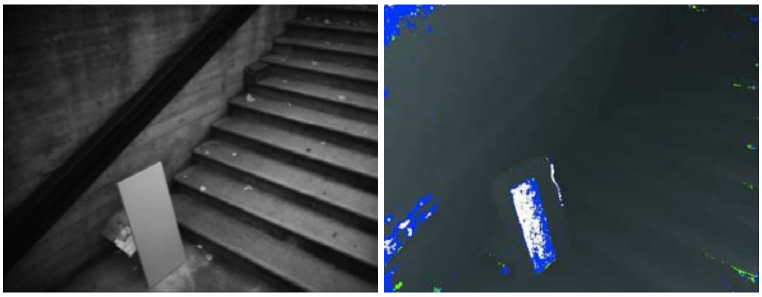


Figure 4.19: An example of B_{2D} and B_{3D}

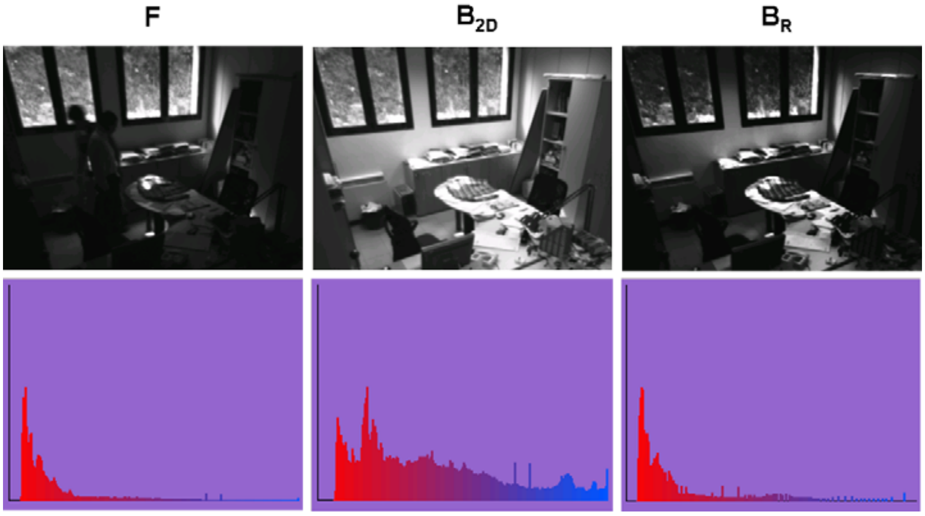


Figure 4.20: During the background registration stage, the histogram of the background model B_{2D} (center) is registered according to the specification given by the histogram of the frame F (left), yielding the new background model B_R (right)

Background Modeling The proposed approach requires that, at initialization, two background models of the scene are built: the former, B_{2D}, is determined from the brightness values of the reference view, the latter, B_{3D}, is determined from the disparity values provided by the stereo matching stage. Both models are built by processing a short initialization sequence of frames. While B_{2D} captures the radiance information of the scene background, B_{3D} represents a model of the scene 3D structure. In order to obtain B_{2D}, a classical method is used, that is each value in B_{2D} represents the mean brightness of a pixels over the initialization sequence. Conversely, by means of three different thresholds T_1,T_2,V, each pixel of B_{3D} is associated to 4 different classes:

1. Valid disparity: if a valid disparity is retrieved by the matching algorithm for

more than T_1 frames during the initialization sequence, and the variance of disparities is less than V. The final disparity assigned to the pixel in B_{3D} is the mean disparity over the initialization sequence.

2. High-variance disparity: if a valid disparity is retrieved by the matching algorithm for more than T_1 frames during the initialization sequence, but the variance of disparities is equal or higher than V. The pixel is depicted in green in B_{3D}.
3. Non-match: if a NM is retrieved by the matching algorithm for more than T_2 frames during the initialization sequence. This pixel is depicted in white in B_{3D}.
4. Unreliable point: if a valid disparity is retrieved by the matching algorithm for equal or less than T_1 frames during the initialization sequence, and a NM for equal or less than T_2 frames. The pixel is depicted in blue in B_{3D}.

An example of B_{2D} and B_{3D} is shown in Fig. 4.19.

It is worth observing that when a WTA strategy is adopted (as in [130]) the last two conditions are not meaningful.

3D detection Once the background models are built, at each time instant a new frame from the reference view, F, together with its disparity map, D, is obtained. At this stage B_{3D} and D are deployed to compute a mask, M_{3D}, which encodes with different colors the various correspondences between D and B_{3D}. In particular, as shown in Figure 4.18:

1. $\mathbf{b_1}$ (*light blue*): a valid disparity point in B_{3D} corresponding to a valid disparity point in D, the difference between the two disparity values being less than a certain threshold.
2. $\mathbf{b_2}$ (*pink*): a valid disparity point in B_{3D} corresponding to a valid disparity point in D, the difference between the two disparity values being equal or higher than a certain threshold.
3. $\mathbf{b_3}$ (*blue*): an unreliable point in B_{3D}.
4. $\mathbf{b_4}$ (*green*): a high-variance disparity in B_{3D}.
5. $\mathbf{b_5}$ (*white*): a non-match point in B_{3D} corresponding to a NM point in D.
6. $\mathbf{b_6}$ (*red*): a non-match point in B_{3D} corresponding to a valid disparity point in D.
7. $\mathbf{b_7}$ (*yellow*): a valid disparity point in B_{3D} corresponding to a NM point in D.

The information encoded in mask M_{3D} is useful to perform the tonal alignment procedure performed in the background registration stage, as well as in the generation of the final change mask C_{3D}.

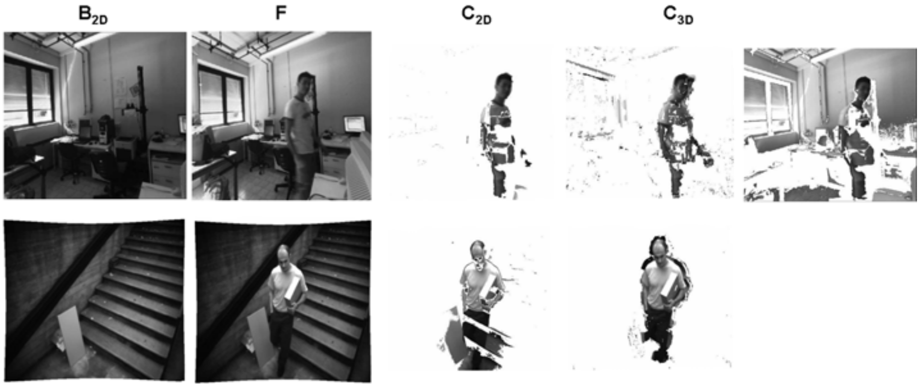


Figure 4.21: Experimental results on an indoor (above) and outdoor stereo sequence. Top right picture: output of a classical background subtraction algorithm

Background registration A further stage of the algorithm, which will be particularly useful for the generation of C_{2D}, deals with the elimination of photometric distortions between F and B_{2D} by tonally registering B_{2D} with respect to F. In particular, the evaluation of the Intensity Mapping Function that tonally aligns B_{2D} to F is done by applying the *histogram specification* method [52]. For this aim a set of pixels belonging to F which reliably belongs to the scene background has to be extracted. This can be easily done by exploiting the information included in M_{3D}: in particular, the set of pixels chosen as representative of the background of the scene are selected as those tagged as $\mathbf{b_1}$ (i.e. in *light blue* color) on M_{3D}, as they denote unchanged valid disparities between D and B_{3D}. The output of this stage is a novel background model, B_R, where photometric distortions with respect to the current frame F have been removed. In Fig. 4.20 an example is presented, which shows the application of the background registration to a frame. In particular, the histogram of the current frame F (left) is used as a model to tonally register the histogram of the 2D background model, B_{2d} (center). The histogram and the image of the registered background B_R, obtained as output of this stage, are shown on the right side of the Figure.

2D Output Once the background is tonally aligned to the current frame, a simple pixel wise frame difference can highlight structural changes robustly with respect to possible brightness distortions. Hence, C_{2D} is generated by subtracting B_R from F. The main strength of this approach is robustness against sudden illumination changes as well its accuracy in the foreground segmentation stage. Nevertheless, the shadow and camouflage issues are not properly dealt with.

3D Output Conversely to 2D Output, this approach relies more on 3D information. In particular, all pixels whose disparity value was reliably determined on M_{3D} as unchanged ($\mathbf{b_1}$, light blue color) are set as unchanged on C_{3D}. Similarly, all pixels whose disparity value was reliably determined on M_{3D} as changed ($\mathbf{b_2}$, pink color) are set as changed on C_{3D}. For what means pixels whose disparity can not be determined reliably, all pixels that *might* denote a structural change (i.e. from NM to a valid disparity or vice versa, that is $\mathbf{b_6}$ and $\mathbf{b_7}$ on M_{3D}) are set on C_{3D} as the correspondent value on C_{2D}. Finally, all remaining pixels ($\mathbf{b_3}$, $\mathbf{b_4}$, $\mathbf{b_5}$ on M_{3D}) for whom nothing can be said are set as unchanged on M_{3D} [3]. This solution represents a robust approach toward shadows, camouflage and sudden illumination changes, but foreground segmentation is less accurate compared to the other approach due to the depth borders inaccuracy brought in by the stereo matching process.

4.3.2 Experimental results

In this section we show some qualitative experimental results obtained on three different sequences [4], acquired with a Videre Design stereo camera, referred to as Indoor, Outdoor and Office. In the first sequence, which is indoor, photometric distortions are induced by real illumination changes. The second sequence, which is outdoor, is affected by illumination changes as well as by the strong presence of shadows and camouflage problems. The Office sequence, which is indoor, shows the strong photometric distortions induced by switching lights on and off. Fig. 4.21 shows the two outputs C_{3D} and C_{2D} on a frame of the Indoor and Outdoor sequences using the disparity maps computed by the SMP algorithm. No morphology operator was used at any stage of the algorithm in order to obtain these results, which demonstrate that our approach is in general robust to photometric distortions. Moreover, it can be noted that 2D Output generally retrieves more accurately foreground borders (both sequences), and that 3D Output suffers much less of shadows (outdoor sequence) and camouflage (both sequences). Finally, top right frame in Fig. 4.21, which shows the output of a classical background difference algorithm, demonstrates the strong entity of the photometric distortions, and how less accurate is the segmentation compared to the proposed approaches. Figures 4.22 and 4.23 show the results provided by the change detection strategies proposed in this paper on the more challenging Office sequence using, respectively, the SMP and Variable Windows algorithms. In both figures we show 9 out of 195 frames. Similarly to the Indoor and Outdoor sequences no morphology operator was used at any stage. The Office sequence presents dramatic artificially induced illu-

[3] Another solution is to have these pixels represent regions in the final change mask where detection can not be performed.

[4] Sequences available at: `www.vision.deis.unibo.it/smatt`

mination changes clearly observable comparing frames ♯15, ♯85 and ♯185 in Figures 4.22 and 4.23. It is worth observing that under these difficult conditions the disparity maps generated by the two stereo matching algorithms are significantly noisy. Nevertheless, as shown in the two rightmost columns of Figure 4.22, although the shapes of the objects are not accurately retrieved, the proposed strategies provide a robust detection. Moreover, similarly to the previous sequences, C_{3D} output provides less accurate detection of borders compared to C_{2D} output but it the seems less affected by shadows (on the wall and on the table). Similar considerations apply by observing the results shown in Figure 4.23. However, in this case, the WTA strategy adopted by the Variable Windows algorithm results in even more noisy disparity maps and consequently in more noisy results provided by C_{2D} and C_{3D} outputs. As for Variable Windows, these results do not highlight a better accuracy in recovering the object borders in C_{3D} compared to SMP. We think that this is due to the WTA strategy adopted in our current implementation of the algorithm and that a higher accuracy may be obtained by enforcing into the algorithm constraints such as uniqueness, distinctiveness and sharpness or left-right consistency.

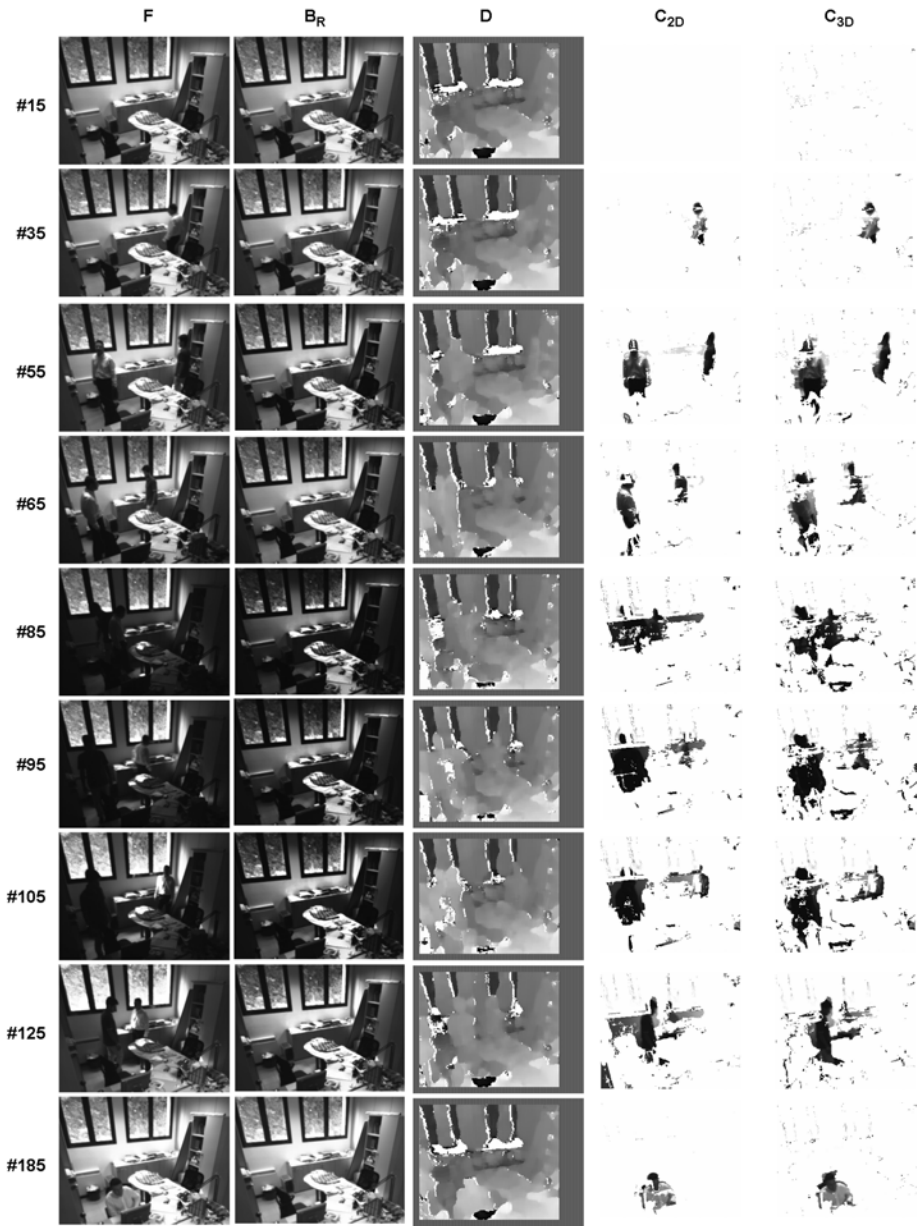


Figure 4.22: Experimental results on 9 frames of the Office stereo sequence using the disparity map provided by the SMP algorithm [34]. (First column) - Reference image F of the stereo pair. (Second column) Background model B_{2D} registered according to the specification given by the histogram of the frame F. (Third column) - Disparity map D computed by the SMP algorithm. (Fourth column) - Change mask C_{2D} provided by the proposed *2D Output* approach. (Fifth column) - Change mask C_{3D} provided by the proposed *3D Output* approach.

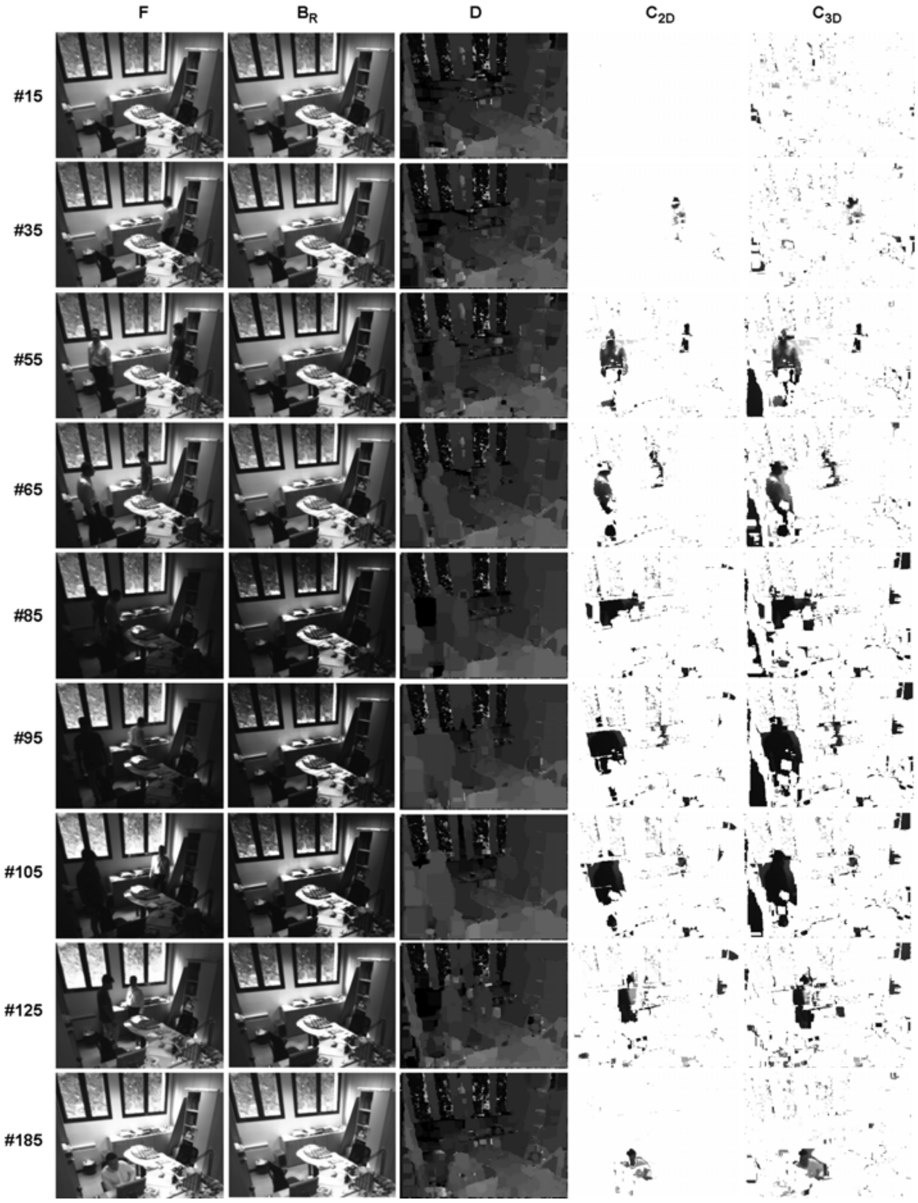


Figure 4.23: Experimental results on 9 frames of the Office stereo sequence using the disparity map provided by the Variable Windows [130] algorithm. (First column) - Reference image F of the stereo pair. (Second column) Background model B_{2D} registered according to the specification given by the histogram of the frame F. (Third column) - Disparity map D computed by the Variable Windows algorithm. (Fourth column) - Change mask C_{2D} provided by the proposed *2D Output* approach. (Fifth column) - Change mask C_{3D} provided by the proposed *3D Output* approach.

Chapter 5

Robust visual correspondence

5.1 Introduction

The visual correspondence task can be extremely challenging in presence of disturbance factors which typically affect images. A common source of disturbances can be related to photometric distortions between the images under comparison. These can be ascribed to the camera sensors employed in the image acquisition process (due to dynamic variations of camera parameters such as auto-exposure and auto-gain, or to the use of different cameras), or can be induced by external factors such as changes of the amount of light emitted by the sources or viewing of non-lambertian surfaces at different angles.

All of these factors tend to produce brightness changes in corresponding pixels that can not be neglected in real applications implying visual correspondence between images acquired from different spatial points (e.g. stereo vision) and/or different time instants (e.g. pattern matching, change detection). In addition to photometric distortions, differences between corresponding pixels can also be due to the noise introduced by camera sensors. Finally, the acquisition of images from different spatial points or different time instants can also induce occlusions. Evaluation assessments have also been proposed which compared visual correspondence approaches for tasks such as stereo correspondence [19], image registration [149] and image motion [48].

5.2 Literature review

Let I_r, I_t be respectively the reference image patch vector and the target image patch vector, to be matched together. Traditional matching measures can be subdivided into either correlation-based or distance-based. Between the correlation-based the most commonly adopted are the *Normalized Cross-Correlation* (NCC) and the *Zero-mean Normalized Cross-Correlation* (ZNCC):

$$NCC(I_r, I_t) = \frac{I_r \circ I_t}{||Ir|| \cdot ||I_t||} \tag{5.1}$$

$$ZNCC(I_r, I_t) = \frac{(I_r - \overline{I_r}) \circ (I_t - \overline{I_t})}{||Ir - \overline{I_r}|| \cdot ||I_t - \overline{I_t}||} \tag{5.2}$$

with $\circ$ being the dot product, $|| \cdot ||_p$ the L_p norm, $\overline{\cdot}$ the mean value over the patch. Thanks to normalization with regards to the magnitude of the vectors and to the mean intensity value of the image patch, NCC and ZNCC are invariant, respectively, to linear and affine transformation between I_r and I_t.

On the other side, commonly used dissimilarity measures are those derived from the L_p-distance between I_r and I_t. Between this class, the most popular ones are the *Sum of Absolute Differences* (SAD) and the *Sum of Squared Differences* (SSD):

$$SAD(I_r, I_t) = |I_r - I_t| \tag{5.3}$$

$$SSD(I_r, I_t) = ||I_r - I_t||^2 \tag{5.4}$$

These two measures showed experimentally good robustness towards noise [5,89]. While all these measures are usually computed directly on the pixel intensities of the images, in [89] it was shown that by computing these measures on the gradient norm of each pixel a higher robustness is attained, i.e. for what concerns insensitivity to illumination changes the SSD and the NCC applied on gradient norms (referred to here respectively as G-SSD and G-NCC) showed to perform well. In particular, if we denote with $G_r(i, j)$ the gradient of I_r at pixel (i, j):

$$G_r(i, j) = \left[\frac{\partial I_r(i, j)}{\partial i}, \frac{\partial I_r(i, j)}{\partial j}\right]^T = \left[G_i^r(i, j), G_j^r(i, j)\right]^T \tag{5.5}$$

and similarly with $G_t(i, j)$ the gradient of I_t at pixel (i, j):

$$G_t(i, j) = \left[\frac{\partial I_t(i, j)}{\partial i}, \frac{\partial I_t(i, j)}{\partial j}\right]^T = \left[G_i^t(i, j), G_j^t(i, j)\right]^T \tag{5.6}$$

the gradient norm, or magnitude, in both cases is defined as:

$$||G_r(i, j)|| = \sqrt{G_i^r(i, j)^2 + G_j^r(i, j)^2} \tag{5.7}$$

$$||G_t(i, j)|| = \sqrt{G_i^t(i, j)^2 + G_j^t(i, j)^2} \tag{5.8}$$

Hence the G-NCC function can be defined as:

$$G - NCC(x,y) = \frac{\sum\limits_{(i,j)\in I_r} ||G_r(i,j)|| \cdot ||G_t(i,j)||}{\sqrt{\sum\limits_{(i,j)\in I_r} ||G_r(i,j)||^2} \cdot \sqrt{\sum\limits_{(i,j)\in I_t} ||G_t(i,j)||^2}} \tag{5.9}$$

and the G-SSD function as:

$$G - SSD(x,y) = \sum_{(i,j)\in I_r} (||G_r\,(i,j)\,|| - ||G_t\,(i,j)\,||)^2 \tag{5.10}$$

In addition to these measures, many alternatives have been proposed in literature with the specific aim of deploying robust image matching. The *Gradient Correlation* (GC) measure, proposed in [23] and derived from a measure originally introduced in [111], is based on two terms, referred to as distinctiveness (D) and confidence (C), both computed from intensity gradients:

$$D(x,y) = \sum_{(i,j)\in I_r} ||G_r(i,j) - G_t(i,j)|| \tag{5.11}$$

$$C(x,y) = \sum_{(i,j)\in I_r} (||G_r(i,j)|| + ||G_t(i,j)||) \tag{5.12}$$

The GC measure is then defined as:

$$GC(x,y) = \frac{D(x,y)}{C(x,y)} \tag{5.13}$$

Its minimum value is 0, indicating the pattern is identical to the image subwindow. For any other positive value, the greater the value, the higher the dissimilarity between the two vectors. In order to compute the partial derivatives, [23] proposes to use either the Sobel operator or the Shen-Castan ISEF filter [117].

The *Orientation Correlation* (OC) measure (Fitch et al., 2002) is based on the correlation of the orientation of the intensity gradient. In particular, for each gradient $G_r(i,j)$ a complex number representing the orientation of the gradient vector is defined as:

$$O_r(i,j) = sgn(G^r_i(i,j) + \iota\; G^t_j(i,j)) \tag{5.14}$$

with ι denoting the imaginary unit and where:

$$sgn(x) = \begin{cases} 0 & if|x| = 0 \\ \frac{x}{|x|} & otherwise \end{cases} \tag{5.15}$$

Analogously, a complex number representing the orientation of the image subwindow gradient vector $G_t(i,j)$ is defined as:

$$O_t(i,j) = sgn(G_i^t(i,j) + \iota\, G_j^t(i,j)) \tag{5.16}$$

As proposed in [41], the partial derivatives for the gradient computation are calculated by approximating them with the *central differences*. Hence, the OC measure between I_r and I_t is defined as the real part of the correlation between all gradient orientations belonging to I_r and I_t:

$$OC(x,y) = Re\{\sum_{(i,j)\in I_r} O_r(i,j) \cdot O_t^*(i,j)\} \tag{5.17}$$

with $*$ indicating the conjugate of the complex vector. [41] proposes to compute the correlation operation in the frequency domain by means of the FFT by exploiting the correlation theorem in order to achieve computational efficiency.

Another class of measures concerns the so-called order-consistency or order-preservation hypothesis, that is the assumption that the considered distortions do not violate the ordering between the intensities of neighbouring pixels. This assumption includes a more general class of transformations compared to the linear or affine case. These measures are called ordinal and a typical example of this class is represented by the Rank transform. As for this measure, both Ir and It are transformed into two novel images where each pixel stores the number of points in the patch whose intensity is less than that of the central point of the patch:

$$R_r = |\{(u,v) \in I_r | I_r(u,v) < I_r(i,j)\}|_c \tag{5.18}$$

$$R_t(i,j) = |\{(u,v) \in I_t | I_t(u,v) < I_t(i,j)\}|_c \tag{5.19}$$

where $|\cdot|_c$ represents the cardinal operator. Once the two transforms are computed, a matching measure is deployed to compare R_r and R_t, e.g. [144] proposes to use the SAD.

Other examples of ordinal measures are the Census transform [144], and the measure proposed in [11]. Further approaches of robust visual correspondence measures specifically conceived for change detection are [95, 99, 137].

Finally, other robust approaches have been proposed in [66, 74, 98, 116, 127].

5.3 A novel measure for robust visual correspondence

This section describes a novel approach, referred to here as Matching Function (MF), which is implicitly based on the ordering assumption. In particular, MF aims at quantifying how well the order is preserved between corresponding pairs of neighbouring

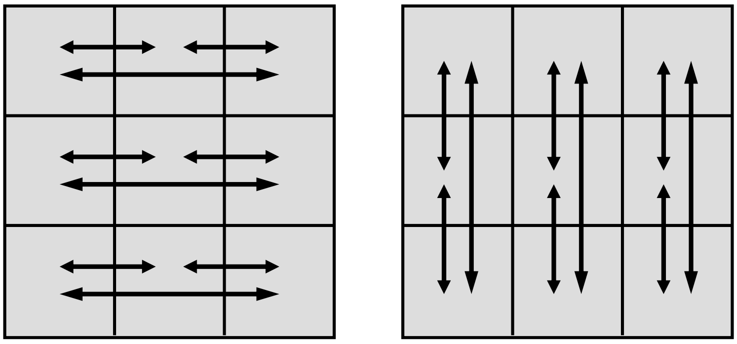


Figure 5.1: Considered subset of horizontal and vertical pairs of neighbouring pixels in a 3×3 patch.

pixels in the two images. A simple and effective approach for evaluating the order-consistency is to evaluate the difference between the intensities of pairs of neighbouring pixels. As an example, lets consider I_r as a 3×3 patch. In order to evaluate the order preservation between neighbouring elements within this window, many pairs (e.g. 72) should be considered, as each of the 9 pixels has to be put in correspondence with each other. In order to simplify the problem, we propose to consider only a subset of the whole neighbouring pairs set by evaluating only horizontal and vertical neighbouring pixels. Hence, the considered pairs are reduced to 18, as shown in Fig. 5.1.

In particular, in order to quantify how well the ordering is preserved between the two image patches I_r and I_t we propose to correlate the differences between the considered corresponding pairs within the 33 window. If the ordering is preserved for a given pair, the result of the pointwise correlation is a positive coefficient regardless of the sign of the intensity difference, which tends to increase the correlation score associated with the 3×3 window. Conversely, if the order is not preserved the correlation coefficient is negative, and the correlation score is decreased. Moreover, since horizontal and vertical differences may be thought as the discrete approximation of the horizontal and vertical derivatives of the image, the proposed measure can also be interpreted as the cross-correlation between two vectors made out of derivatives computed within the two 3×3 patches.

In the general case of two $M \times N$ patches, the considered pairs of pixels in each set include all pixels at distance 1 and 2 along horizontal and vertical directions. In order to compute this set, we define a vector of pixel differences computed at a point (i, j) on I_r:

$$\delta^{\mathbf{r}}_{\mathbf{1,2}}(i,j) = \begin{bmatrix} I_r(i-1,j) - I_r(i,j) \\ I_r(i,j-1) - I_r(i,j) \\ I_r(i-1,j) - I_r(i+1,j) \\ I_r(i,j-1) - I_r(i,j+1) \end{bmatrix} \tag{5.20}$$

and, similarly, at a point (i, j) on I_t:

$$\delta^{\mathbf{t}}_{\mathbf{1,2}}(i,j) = \begin{bmatrix} I_t(i-1,j) - I_t(i,j) \\ I_t(i,j-1) - I_t(i,j) \\ I_t(i-1,j) - I_t(i+1,j) \\ I_t(i,j-1) - I_t(i,j+1) \end{bmatrix} \tag{5.21}$$

Hence, the MF function consists in correlating these two vectors for each point of I_r, I_t, and in normalizing the correlation with the L_2 norm of the vectors themselves:

$$MF_{1,2}(x,y) = \frac{\sum_{(i,j)\in I_r} \delta^{\mathbf{r}}_{\mathbf{1,2}}(i,j) \circ \delta^{\mathbf{t}}_{\mathbf{1,2}}(i,j)}{\sqrt{\sum_{(i,j)\in I_r} \delta^{\mathbf{r}}_{\mathbf{1,2}}(i,j) \circ \delta^{\mathbf{r}}_{\mathbf{1,2}}(i,j)} \cdot \sqrt{\sum_{(i,j)\in I_t} \delta^{\mathbf{t}}_{\mathbf{1,2}}(i,j) \circ \delta^{\mathbf{I^t}_{1,2}}(\mathbf{i},\mathbf{j})}} \tag{5.22}$$

It is worth noticing that the normalization allows the measure to range between $[-1, 1]$. It is a peculiarity of this method that, because of the correlation between differences of pixel pairs, intensity edges tend to determine higher correlation coefficients (in magnitude) with respect to low-textured regions. Thus, this can be seen as if the measure mostly relies on the patch edges. For this reason, MF can be usefully employed also in presence of high levels of noise, as this disturbance factor can typically violate the ordering constraint on low-textured regions, but seldom along intensity edges. Similar considerations can be made in presence of partially occluded patches.

The set of pixel pairs in 5.20, 5.21 can be seen as made out of two subsets: the set of horizontal and vertical lateral derivatives (i.e. all pixels at distance 1 one to another along horizontal and vertical directions), and the set of horizontal and vertical central derivatives (i.e. all pixels at distance 2 one to another along same directions). Theoretically, the former should benefit of the higher correlation given by adjacent pixels, while the latter should be less influenced by quantization (sampling) noise that is introduced by the camera sensor. We will refer to two additional measures of the MF class applied on each of these two subsets as, respectively, MF_1 and MF_2. For these last two cases, we define the vector of pixel differences at distance 1 pixel:

$$\delta^{\mathbf{r}}_{\mathbf{1}}(i,j) = \begin{bmatrix} I_r(i-1,j) - I_r(i,j) \\ I_r(i,j-1) - I_r(i,j) \end{bmatrix} \tag{5.23}$$

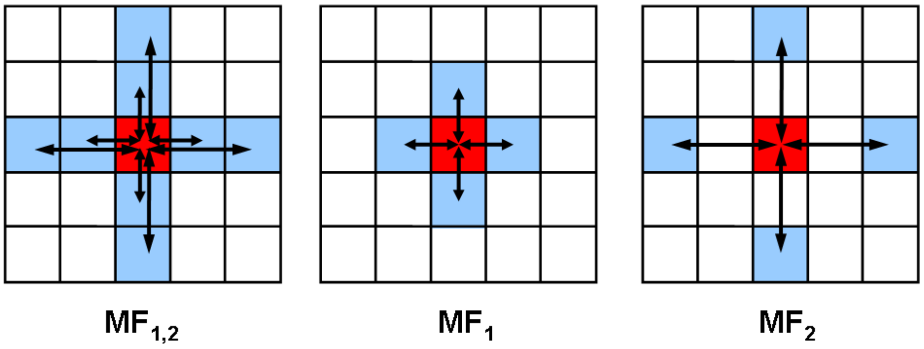


Figure 5.2: The 3 considered sets of neighbouring pixel pairs.

$$\delta_{\mathbf{1}}^{\mathbf{t}}(i,j) = \begin{bmatrix} I_t(i-1,j) - I_t(i,j) \\ I_t(i,j-1) - I_t(i,j) \end{bmatrix} \tag{5.24}$$

and the pixel differences relative to the case of distance 2:

$$\delta_{\mathbf{2}}^{\mathbf{r}}(i,j) = \begin{bmatrix} I_r(i-1,j) - I_r(i+1,j) \\ I_r(i,j-1) - I_r(i,j+1) \end{bmatrix} \tag{5.25}$$

$$\delta_{\mathbf{2}}^{\mathbf{t}}(i,j) = \begin{bmatrix} I_t(i-1,j) - I_t(i+1,j) \\ I_t(i,j-1) - I_t(i,j+1) \end{bmatrix} \tag{5.26}$$

the MF_1 and MF_2 measures are defined respectively as:

$$MF_1(x,y) = \frac{\sum\limits_{(i,j)\in I_r} \delta_{\mathbf{1}}^{\mathbf{r}}(i,j) \circ \delta_{\mathbf{1}}^{\mathbf{t}}(\mathbf{i},\mathbf{j})}{\sqrt{\sum\limits_{(i,j)\in I_r} \delta_{\mathbf{1}}^{\mathbf{r}}(i,j) \circ \delta_{\mathbf{1}}^{\mathbf{r}}(i,j)} \cdot \sqrt{\sum\limits_{(i,j)\in I_t} \delta_{\mathbf{1}}^{\mathbf{t}}(i,j) \circ \delta_{\mathbf{1}}^{\mathbf{t}}(i,j)}} \tag{5.27}$$

$$MF_2(x,y) = \frac{\sum\limits_{(i,j)\in I_r} \delta_{\mathbf{2}}^{\mathbf{r}}(i,j) \circ \delta_{\mathbf{2}}^{\mathbf{t}}(i,j)}{\sqrt{\sum\limits_{(i,j)\in I_r} \delta_{\mathbf{2}}^{\mathbf{r}}(i,j) \circ \delta_{\mathbf{2}}^{\mathbf{r}}(i,j)} \cdot \sqrt{\sum\limits_{(i,j)\in I_t} \delta_{\mathbf{2}}^{\mathbf{t}}(i,j) \circ \delta_{\mathbf{2}}^{\mathbf{t}}(i,j)}} \tag{5.28}$$

A graphical representation of the 3 different pixel pair sets used by $MF_{1,2}$, MF_1 and MF_2 is shown in Fig. 5.2.

5.4 Application to template matching

This section shows the application of the class of measures referred to as MF in a typical template matching scenario. As already discussed in Section 2.4, template matching aims at finding the most similar instances of a given pattern, P, within an image. In particular, in this section MF measures are compared against traditional general purpose approaches as well as against proposals specifically conceived to achieve robustness. One goal of the proposed comparison is to determine which measure is more suitable to deal with the aforementioned disturbance factors represented by photometric distortions, noise and occlusions.

More precisely, in the comparison with MF we will consider the following matching measures: GC [23], OC [41], G-NCC, G-SSD [89]. Considered traditional measures are NCC, ZNCC and SSD. All the considered measures are tested on 3 datasets which represent a challenging framework for what regards the considered distortions. These datasets, which are publicly available [1], are characterized by a significant presence of the disturbance factors discussed previously, and are now briefly described.

Guitar In this dataset, 7 patterns were extracted from a picture which was taken with a good camera sensor (3 MegaPixels) and under good illumination conditions given by a lamp and some weak natural light. All these patterns have to be sought in 10 images which were taken with a cheaper and more noisy sensor (1.3 MegaPixels, mobile phone camera). Illumination changes were introduced in the images by means of variations of the rheostat of the lamp illuminating the scene (G1-G4), by using a torch light instead of the lamp (G5-G6), by using the camera flash instead of the lamp (G7- G8), by using the camera flash together with the lamp (G9), by switching off the lamp (G10). Furthermore, additional distortions were introduced by slightly changing the camera position at each pose and by the JPEG compression. The *Guitar* dataset is shown in Fig. 5.3, 5.4.

Mere Poulard - Illumination Changes In dataset *Mere Poulard - Illumination Changes* (MP-IC), the picture from which the pattern was extracted was taken under good illumination conditions given by neon lights by means of a 1.3 MegaPixels mobile phone camera sensor. This pattern is then searched within 12 images which were taken either with the same camera (prefixed by GC) or with a cheaper, 0.3 VGA camera sensor (prefixed by BC). Distortions are due to slight changes in the camera point of view and by different illumination conditions such as: neon lights switched off and use of a very high exposure time (BC - N1, BC - N2, GC - N), neon lights switched off (BC - NL, GC-NL), presence of structured light given by a lamp light partially occluded

[1] available at `www.vision.deis.unibo.it/pm-eval.asp`

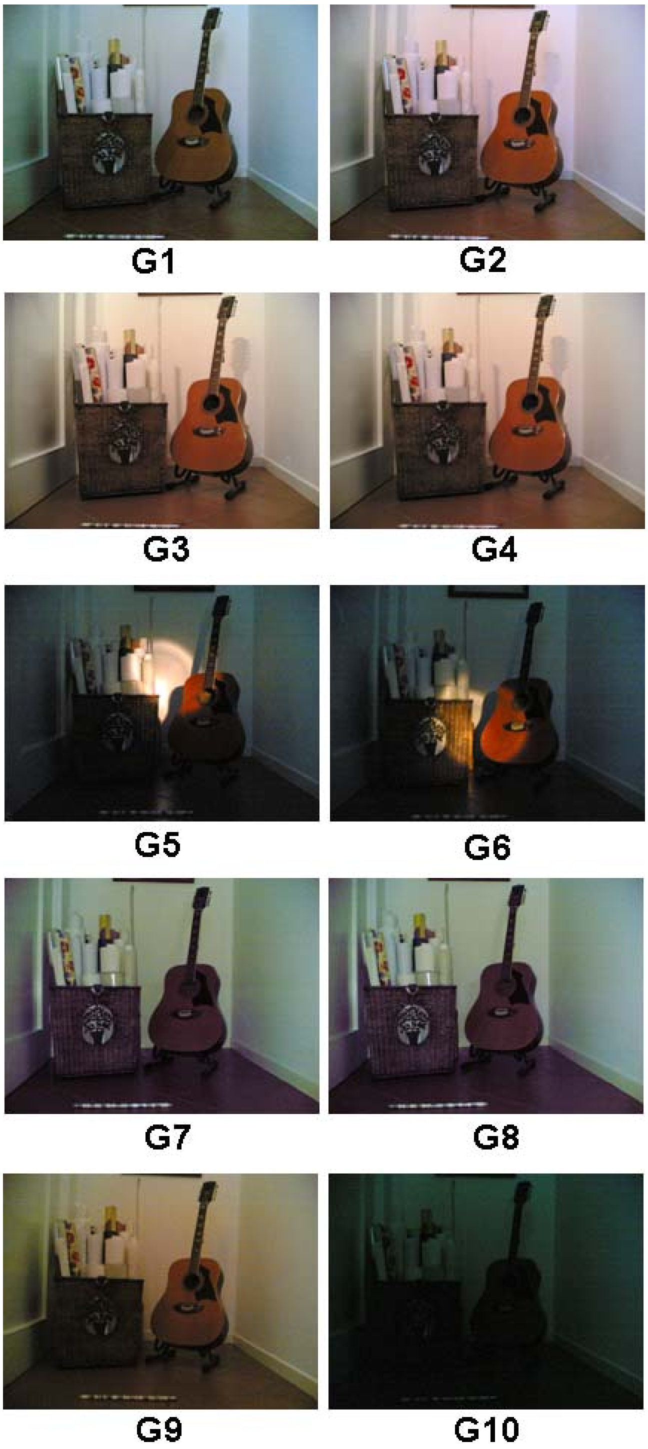


Figure 5.3: Images of the *Guitar* dataset.

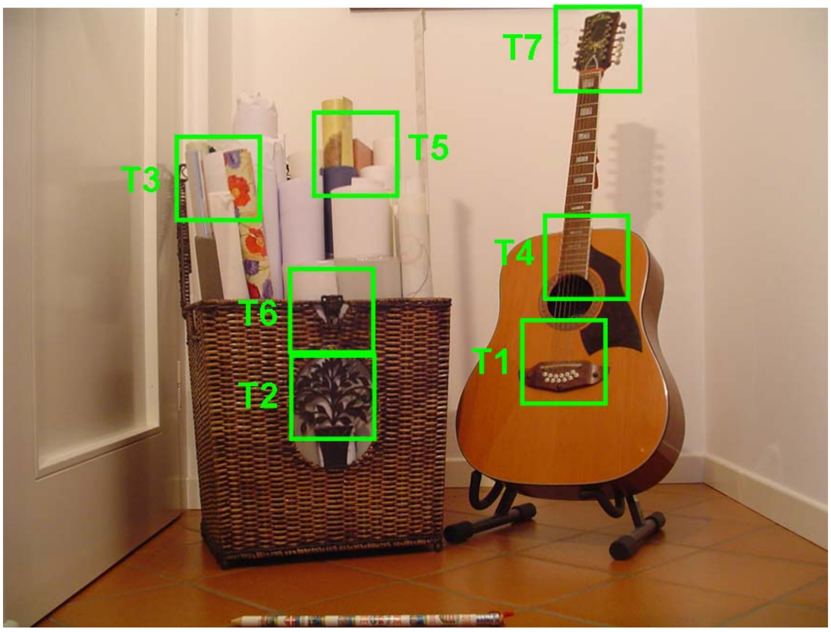


Figure 5.4: Patterns of the *Guitar* dataset.

by various obstacles (BC-ST1, ···, BC-ST5), neon lights switched off and use of the camera flash (GC-FL), neon lights switched off, use of the camera flash and of a very long exposure time (GC-NFL). Also in this case, images are JPEG compressed. The *MP-IC* dataset is shown in Fig. 5.5.

Mere Poulard - Occlusions In the dataset *Mere Poulard - Occlusions* (MP-Occl) the pattern is the same as in dataset MP-IC, which now has to be found in 8 images taken with a 0.3 VGA camera sensor. In this case, partial occlusion of the pattern is the most evident disturbance factor. Occlusions are generated by a person standing in front of the camera (OP1, ···, OP4), and by a book which increasingly covers part of the pattern (OB1, ···, OB4). Distortions due to illumination changes, camera pose variations, JPEG compression are also present. The *MP-Occl* dataset is shown in Fig. 5.6.

The number of template matching instances is thus 70 for the Guitar dataset, 12 for the MP-IC dataset and 8 for the MP-Occl dataset, for a total of 90 instances overall. The result of a template matching process is considered erroneous when the coordinates of the best matching subwindow found by a certain measure are further than 5 pixel from the correct ones.

Figures 5.7, 5.8 report the matching errors yielded by the considered measures respectively on each of the 3 datasets and overall. As it can be seen, approaches specif-

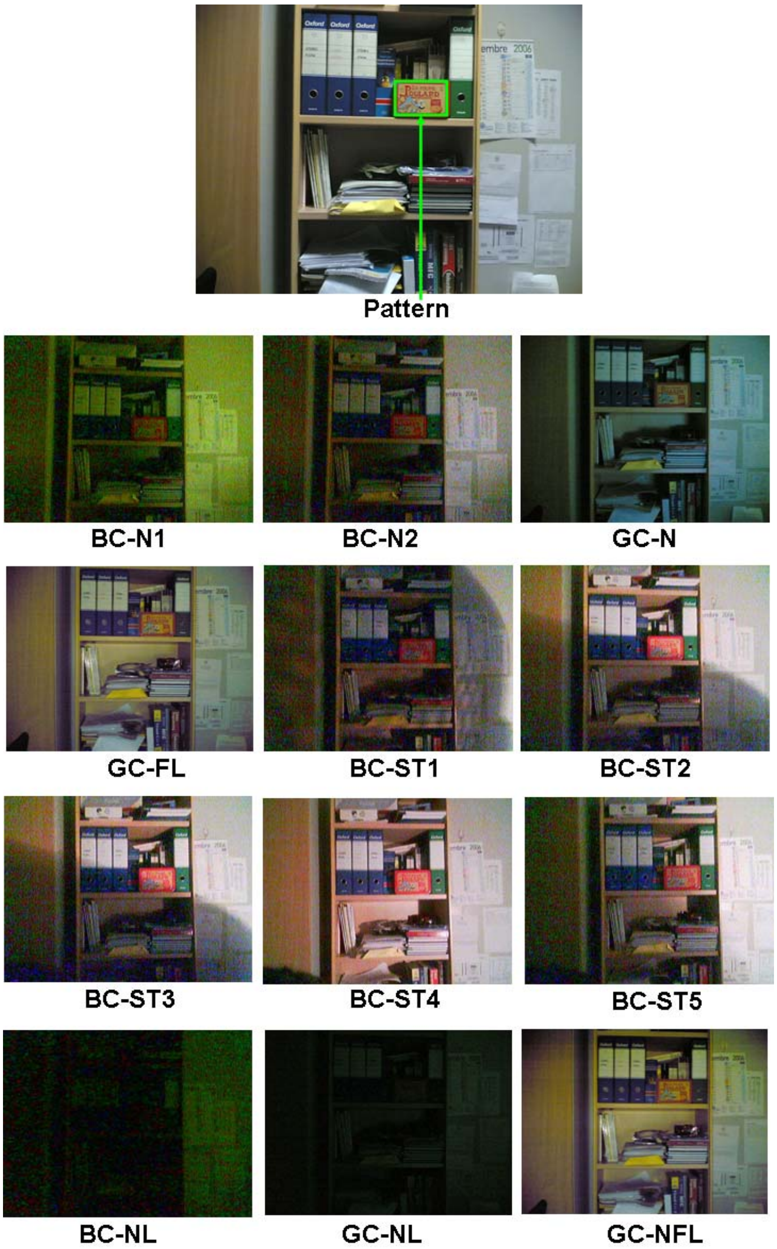
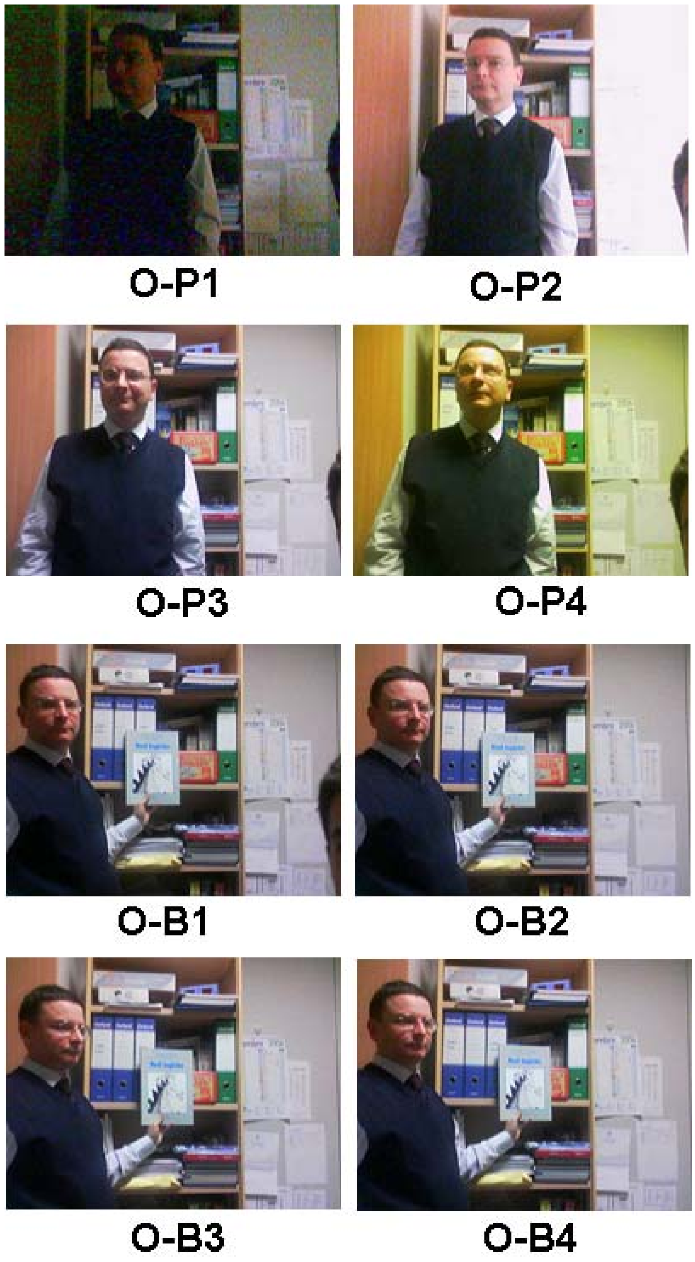


Figure 5.5: *MP-IC* dataset.

ically conceived to achieve robustness generally outperform classical measures, apart from the ZNCC which performs badly in presence of occlusions but shows good robustness in handling strong photometric distortions. The two measures which yield the best performance are MF and GC, with a number of total errors respectively equal to 6 and 8. In particular, MF performs better on datasets characterized by strong photomet-



O-P1

O-P2

O-P3

O-P4

O-B1

O-B2

O-B3

O-B4

Figure 5.6: *MP-Occl* dataset.

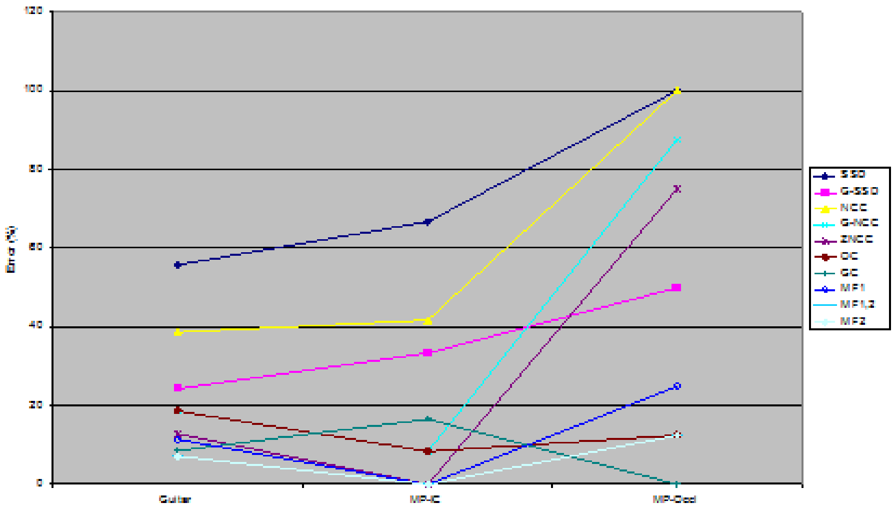


Figure 5.7: Results of the comparison on the 3 datasets.

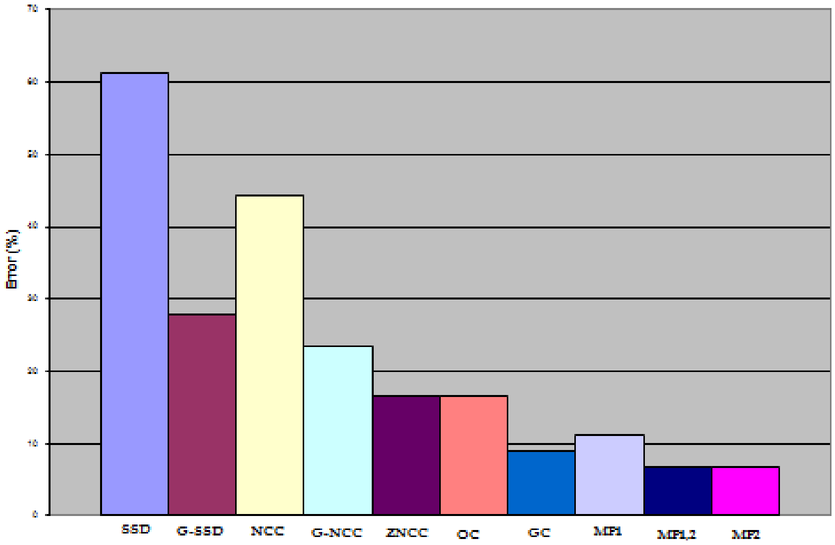


Figure 5.8: Overall results of the template matching evaluation.

ric distortions, conversely GC seems to perform better in presence of occlusions.

For what regards the 3 MF measures themselves, it seems clear that the use of differences relative to adjacent pixels suffers of the sampling noise introduced by the camera sensor, hence they appear less reliable compared to differences computed on a distance equal to 2. Moreover, as a consequence of the fact that $MF_{1,2}$ and MF_2 yield the same results on all datasets, MF_2 seems the more appropriate measure of the class since it requires only 2 correlation terms instead of the 4 needed by $MF_{1,2}$. Finally, for what regards traditional approaches, it is interesting to note that the application of NCC and SSD on the gradient norms rather than on the pixel intensities allows for a significantly higher robustness throughout all the considered datasets.

5.5 Application to Change detection

In this section we present the application of the proposed MF measures to the change detection task. Change detection aims at detecting structural changes occurring in time in a scene by analyzing a sequence of frames. This is a key task in most advanced video-surveillance applications, for the mask highlighting changed pixels (change mask) typically represents the input data to higher level vision algorithms. This is the case of traditional single view as well as more recent and advanced multiple-views systems. The most common change detection approach is referred to as background subtraction: given the current frame, F, and a model of the background of the scene, B, the change mask is obtained by comparing F and B. This approach assumes that the background model is available or can be obtained by processing a short sequence of frames at initialization time (e.g. as shown in [55]). A wide variety of change detection algorithms has been proposed in literature, so as to address issues such as illumination changes, camouflage and vacillating background. A recent survey providing good coverage of this research area is given by [106].

A major issue for most practical change detection applications is represented by sudden illumination changes occurring in the scene. Properly dealing with such a problem is a challenging task for change detection algorithms since the resulting photometric variations can be easily misinterpreted as structural changes, leading to many false positives in the change mask. Algorithms [38, 99, 137] that specifically aim at detecting changes robustly with respect to sudden illumination variations typically take the decision of voting a pixel as changed or unchanged based on a given spatial support (e.g. a 3×3 or larger window centered at the pixel under evaluation). Typically such algorithms rely on a parametric (e.g. linear [38, 99]) or non parametric (e.g. order preserving [137]) model for the false image changes due to sudden illumination variations. However, it is well known that such algorithms suffer from an aperture problem, i.e. they cannot detect as changed the pixels belonging to untextured foreground regions. As a result, they typically enable to detect the borders of foreground objects but not accurately their interior parts. Moreover, the use of a spatial support rather than pointwise background subtraction implies inaccuracy as regards localization of the borders of the detected foreground objects. Coarse-to-fine approaches such as [10] can alleviate these

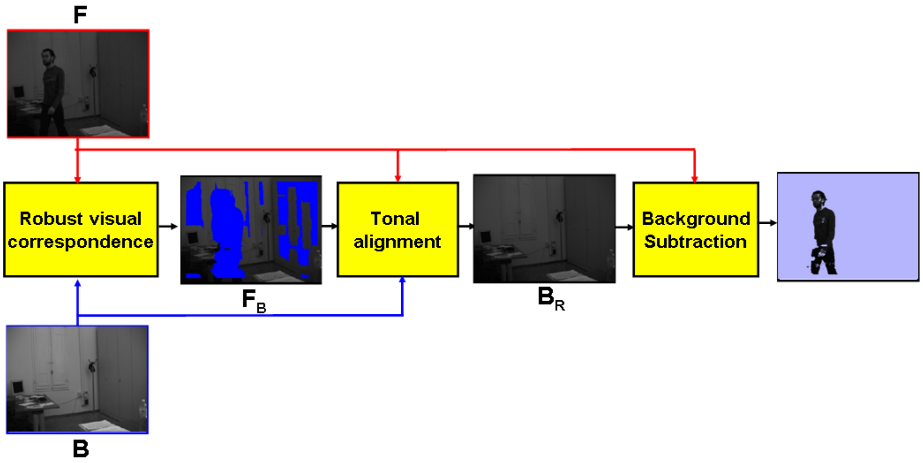


Figure 5.9: Flow diagram of the proposed change detection algorithm.

problems.

This section presents a novel approach aimed at obtaining robust and accurate foreground segmentation under sudden illumination variations. In particular, as depicted in Fig. 5.9, the proposed approach consists of three processing stages. In the first stage, the MF measure is used to extract a subset of pixels in the current frame that can be marked as background with a high confidence level. Once such a subset, referred to as F_B, is obtained, it can be usefully employed to remove the photometric distortion between F and B. To this purpose, in the second stage the algorithm computes the transformation that aligns tonally the current frame, F, to the background image, B, using as support subset F_B. In the third stage, the final change mask is achieved by a pixelwise subtraction between F and the tonally registered background image, B_R. In the following we provide more details on these three processing stages.

Robust visual correspondence In order to get F_B we match the points in the background image to the current frame. To achieve robustness with respect to outliers and noise, a block-based approach is used: that is, for each pair of correspondent points in B and F, a $M \times M$ surrounding block is considered, and the MF measure is computed between the two blocks. Points having a score higher than a given threshold are included into F_B . To explain the usefulness of the MF measure, lets discuss Fig. 5.10 where, for the sake of simplicity, we consider only two kind of regions, i.e. uniform and highly-textured. When dealing with a uniform region in both F and B (case *a* in Fig. 5.10), photometric differences between F and B can occur due to either variations of the illumination conditions of the background scene as well as to structural changes induced

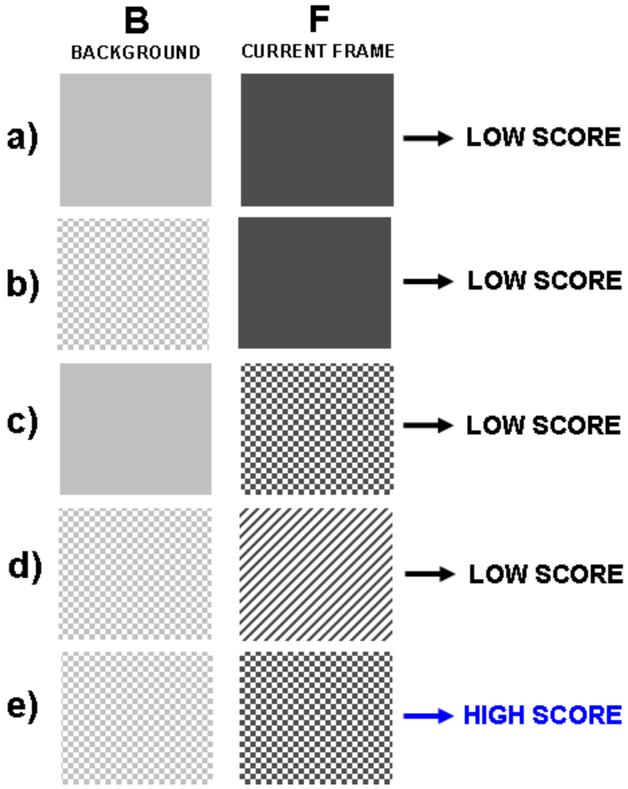


Figure 5.10: Reasoning concerning the robust visual correspondence stage.

by a uniform foreground object. Thus, in this case the required matching measure should yield a low score, for nothing can be said reliably on whether the point belongs to the background or not. As for cases *b,c,d*, it is easy to observe that the matching score should be low too, since theres evidence of the presence of a foreground object. Finally, when the background is highly textured and the texture pattern does not change in spite of possible photometric changes (case *e*), it is reasonable to flag the point as background with a high confidence level. Hence, in case e we should get a high score from the required matching measure. Based on the above considerations, we adopt the MF measure which, as previously mentioned, matches corresponding blocks of two images by implicitly checking an ordering constraint. Since photometric variations tend not to violate the ordering of intensities in a neighbourhood of pixels, MF allows handling sudden and strong illumination variations between the background scene and the current frame. As previously discussed, MF tries to match the high contrast regions (i.e. the intensity edges) of the two blocks under comparison, since only high intensity differences can provide high contributions to the correlation score. Hence, MF behaves exactly as pointed out in Fig. 5.10. In fact, only two highly textured and highly corre-

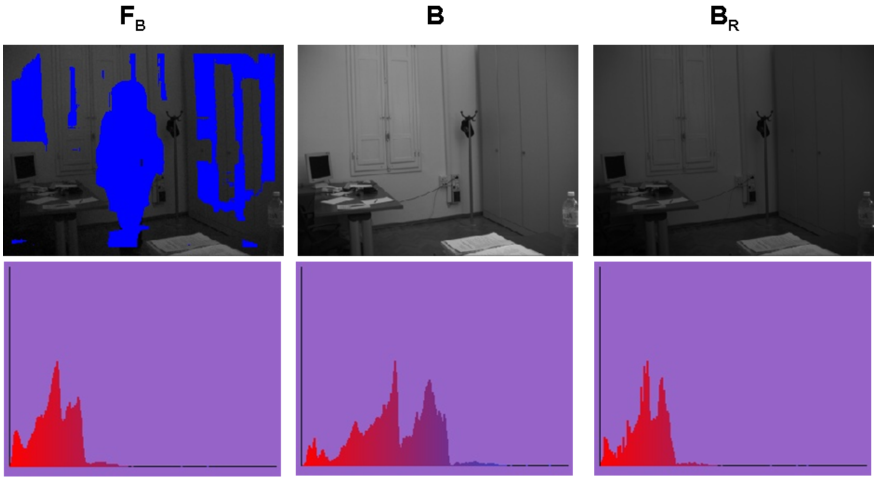


Figure 5.11: The tonal alignment stage registers, on the basis of the histogram specified by F_B (left), the histogram of the background B (center), obtaining the tonally registered background B_R (right)

lated patterns can provide a high matching score (case *e*), while the presence of at least one untextured region (cases *a,b,c*) or of two textured but uncorrelated patterns (case *d*) yields a low score.

Tonal alignment At this point of the algorithm, F_B represents a subset of F denoting pixels that reliably belong to the current background. Hence, B is tonally aligned to F by applying the histogram specification method [52]. In the evaluation of the IMF (*Intensity Mapping Function*) that aligns B to F only the set of corresponding points that belong to the mask F_B is taken into account. By applying the IMF obtained from the histogram specification method to B we get a novel background, B_R, where the photometric distortions have been removed. An example is shown in Fig. 5.11.

Background subtraction Finally, a simple pixelwise difference between B_R and F highlights structural changes by correctly extracting foreground regions. It is worth pointing out that since background subtraction is carried out pixelwise, it is not affected by the aperture problem and allows for accurately detecting the borders as well as interior parts of foreground objects. Obviously, false negatives can still be found due to the possible camouflage between the tonally registered background and the foreground objects.

5.5.1 Computational requirements

The bottleneck of the proposed algorithm can be identified in the computation of the MF funcion. In fact, denoting as W and H respectively the width and height of images, the computation of the numerator requires theoretically $2M^2WH$ differences, multiplications and summations, and similarly the two norms at denominator require each $2M^2WH$ differences, multiplications and summations plus WH square roots.

Nevertheless, all differences concerning the background model B can be computed once for all at initialization, while all differences concerning the current frame can be computed once for all at each new frame, accounting for a total of $2WH$ differences. Furthermore, since the matching metric has to be applied on neighboring blocks in a ”sliding window” fashion, well-known incremental approaches such as [91] allow for further shrinking the total number operations required for the computation of N. In particular, this can be done by computing the product of corresponding pixel differences once for all at each new frame (accounting for $2WH$ multiplications), then running two Box-Filter instances to compute the final accumulation term, which accounts for $8WH$ summations overall. Similar deductions apply to the computation of the two denominator terms, D_F and D_B. The latter can be computed once for all at initialization, while, by means of a strategy analogous to that used to compute N, the former requires only $2WH$ additional multiplications, $8WH$ summations and WH square roots overall at each new frame.

Thanks to these optimizations, our implementation of the proposed algorithm easily deals with the real-time requirements of many change-detection applications (e.g. video-surveillance), with an average frame rate of 15 fps on a 320×240 frame size.

5.5.2 Experimental Results

We now provide some experimental results dealing with the proposed approach. In particular, our algorithm has been tested with real as well as with a synthetic benchmark sequence: real sequences are affected by sudden and strong brightness variations due to illumination changes, while the synthetic one[2] by artificial brightness changes.

First of all, we show some qualitative results. In Fig. 5.12 some screenshots of the change masks obtained by the proposed algorithm on a real sequence (above) and a synthetic one (below) are presented, which clearly prove that the proposed approach is able to accurately segment foreground objects in presence of heavy photometric changes. It is worth pointing out that no morphology operator was used at any stage of the algorithm in order to obtain these results: nevertheless, uniform regions of the foreground are correctly segmented and no false positives arise on low textured regions

[2] available at: `http://muscle.prip.tuwien.ac.at/data_here.php`

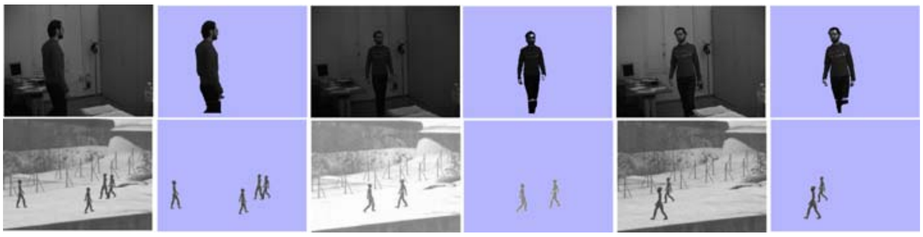


Figure 5.12: Change masks yielded by the proposed algorithm in two sequences affected by sudden brightness variations

of the background.

Furthermore, we show some results dealing with a quantitative comparison between our approach and other proposals. In particular, as representative of change detection algorithms that model false image changes according to a linear relation we consider the NCC between pixel intensities. As for algorithms relying on checking the order preservation of intensities we consider the Rank transform [144]. We also consider as baseline for comparison the *basic pixelwise background subtraction* approach (BBS).

For a fair comparison, we used the same block side for each algorithm (i.e. equal to 7). Then, for what regards the other parameters of each algorithm (in particular, the threshold for the final change mask), in order to determine the best parameter set of each algorithm we selected as a measure of comparison the *Precision*, i.e. the ratio between the true positives (TP) and the sum between true positives and false positives (FP):

$$Precision = \frac{TP}{TP + FP} \tag{5.29}$$

and the *Recall*, i.e. the ratio between the true positives and the sum between true positives and false negatives (FN):

$$Recall = \frac{TP}{TP + FN} \tag{5.30}$$

In order to obtain experimental results, we started from the observation that most change detection algorithms, especially for video-surveillance applications, require to have a minimum guaranteed value of Recall. Hence, for different thresholds of minimum Recall (i.e. 70%, 80%, 90%), we selected for each algorithm the optimal parameter set maximizing the Precision value. Such results are shown in Tab. 5.1. It is worth pointing out that we fixed the maximum constraint value of Recall to 90%, since with higher values all algorithms would provide Precision values lower than 50%, which would result in very poor change masks (the number of false positives being higher

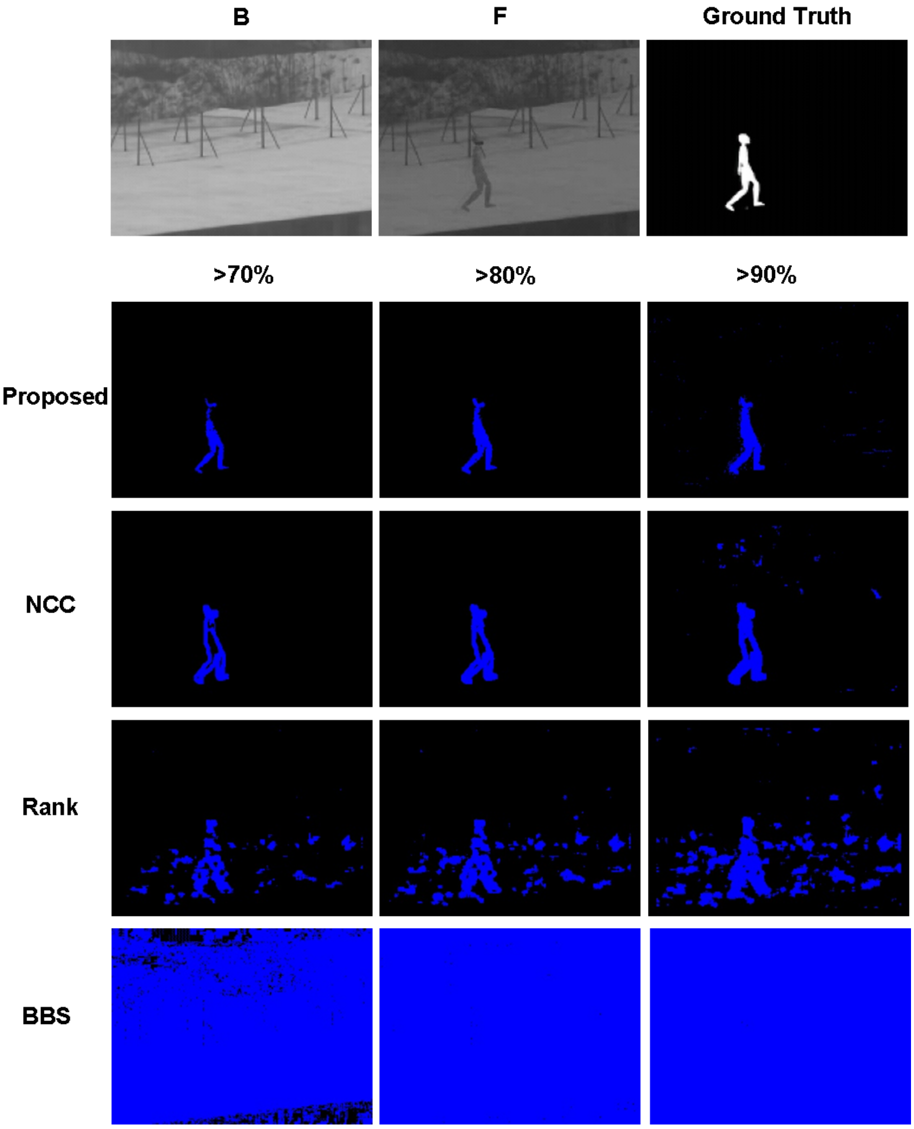


Figure 5.13: Comparison of outputs yielded by the evaluated algorithms on the same sequence and with the same constraint values on Recall used for results in Tab. 5.1. First row, from left to right: background model B, current frame F, Ground Truth.

than the number of true positives). Moreover, it is worth noting that also for these results no post processing was added to the output of the evaluated algorithms, similarly no morphology operator was used at any stage of the evaluated algorithms.

From the Table it is easy to infer that the proposed algorithm is the most robust and accurate between the evaluated ones, since it always outperforms the other approaches in terms of Precision for all different constraint values of Recall. In addition, Fig.

Table 5.1: Best values of Precision yielded by the evaluated algorithms with different constraint values on Recall.

	> 70%	> 80%	> 90%
Proposed	87.3	81.7	52.2
NCC	59.6	57.2	43.0
RANK	24.5	18.8	13.1
BBS	2.2	1.9	1.7

5.13 shows, for a single frame of the evaluated testing sequence, the outputs of the various algorithms at the different constraint values of Recall. In addition, in the first row of the Figure the background model as well as the current frame together with the correspondent ground truth frame are shown. These results qualitatively confirm the trend shown in Tab. 5.1, proving that our approach provides overall the most accurate results.

5.6 Application to video-surveillance: a case study

In this section a case study is presented where access to a high security gate has to be monitored to assess for the presence/absence of people as well as to ensure that only one person occupies the gate at a given time (anti-tailgating).

Monitoring access to interlocks and secured entrance areas, such as revolving doors, is a very important task in many security applications. The aim of the task is twofold: first of all, detecting the presence or absence of people in the monitored interlock, secondly allowing only one person at a time to be present in the interlock (*singularization*). Singularization is needed to avoid two (or more) people simultaneously crossing the gate (*piggybacking*) or an unauthorized person crossing the interlock other than the authorized one (*tailgating*). Most solutions deploy sensors such as weight controllers, light barriers and ultrasonic devices [113, 114] which are mounted inside the interlock walls and floor. However, these systems are generally expensive and not accurate enough. Furthermore, they are often unpractical for maintenance.

Therefore, the use of simple and cost effective video-based approaches is gaining increasing attention in the security industry. These approaches usually rely on background subtraction to detect the presence and estimate the number of people in the interlock. However, it is difficult to perform reliable and accurate background subtraction in real environments, which are characterized by unstructured backgrounds (typically, the floor of the gate), sudden illumination changes, presence of shadows. Commercial systems, such as *Smart Airlock Control System* (SMACS) by Fastcom Technology, deal with this issue by heavily structuring the working environment, i.e. by deploying markers on both the interlock floor (e.g. by means of a chessboard patterned carpet) and walls (e.g. by means of patterned stripes), and by requiring a specially aligned illumination to avoid shadows (i.e. by means of fluorescent tubes disposed in a rectangular arrangement). Another difficulty arises from the size of the interlock, that can often be very small. This constrains the positioning of cameras and generally forces the monitoring system to work with very small field of views, which often do not allow the person to be seen entirely into one single view.

The proposed approach exploits two views to accurately and robustly perform presence detection and singularization in small interlocks. The only modification to the environment required by the system consists in installation of a tiny stripe of reflective material around the borders of the interlock floor. This is not needed for presence detection, but helps improving the feature extraction process required for singularization.

Our approach is based on two main stages. On one side, the use of a novel background subtraction approach, which deploys the MF measure as well as different background models, allows to accurately segment the foreground from the background even in presence of shadows and strong and sudden illumination changes. On the other side, the deployment of distinctive features extracted by comparing the two views allows our system to work with very small field of views.

5.6.1 System setup and preliminary work

Since the specifications and the design of our method are closely related to its practical application, we will refer hereinafter to a specific model of interlock, which was used for the study and performance evaluation of our approach. Nonetheless, the developed methodology can be easily generalized to different kind of interlocks. The considered interlock is a two-door cylindric revolving door, typically used for security accesses such as bank entrance areas. As shown in Fig. 5.14, the entrance and exit doors of the interlock are placed at a right angle and two cameras are fixed to the ceiling of the revolving door.

The gate setup is particularly challenging for video-based monitoring. The small height of the interlock (approximately 2 meters) typically enables each camera to get only a partial view of the gate occupant. Moreover, the interlock floor is characterized by a black anti-slippery material. The presence of small knobs in this material determines a natural light pattern that, unfortunately, changes notably its appearance under different illumination conditions, as depicted in Fig. 5.15. Besides, the presence of bullet-proof glass all around the interlock typically creates serious light artifacts on the floor and walls, due to the presence of different illumination conditions inside and

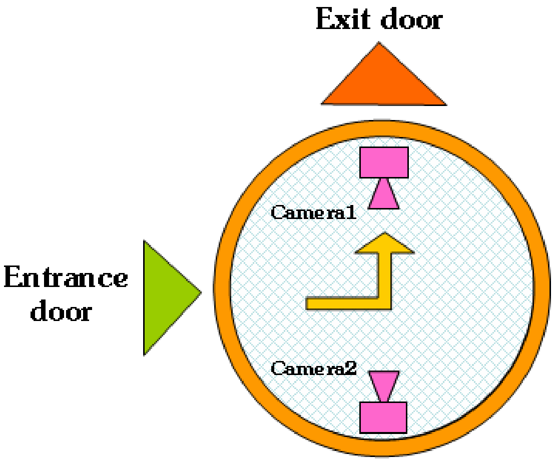


Figure 5.14: Outlook of the interlock used to test the approach

outside the building. The presence of illuminators inside the interlock and light sources outside generally produces strong and unpredictable shadows when people occupy the interlock. Illumination changes and movements of people and vehicles outside the interlock also account for additional photometric distortions. Finally, further artifacts comes up in time due to dirtying of the walking surface. Some examples showing different appearances of the gate floor when the revolving door is empty are shown in Fig. 5.16.

Preliminary work, just sketched here for the sake of brevity, was carried out to simulate a real system and verify the feasibility of a possible solution based on a feature extraction and classification approach. To reach this aim, a 3D graphic model of the interlock and the occupants was developed (see Fig. 5.17). Then, a study concerning the selection of the features to be used by the system was carried out by rendering the two camera views under many working conditions. In particular, the simulations considered the number of occupants as well as their position, orientation, height and size. Promising results of these simulations pushed forward for a thorough study addressing the real working conditions.

Fig. 5.18 outlines the proposed approach, which is aimed at both presence detection and singularization. As it can be seen, both processes exploit a common stage concerning segmentation of the floor area visible by both cameras. Then, to carry out presence detection the output of the floor segmentation stage is analysed to decide whether the interlock is currently empty or one or more occupants are present. Differently, singularization exploits also the segmentation of the floor border visible by cameras, so as to extract additional features that allow to discriminate whether the number of occupants is higher or equal than 1. The use of two cameras is mainly motivated by the fact that

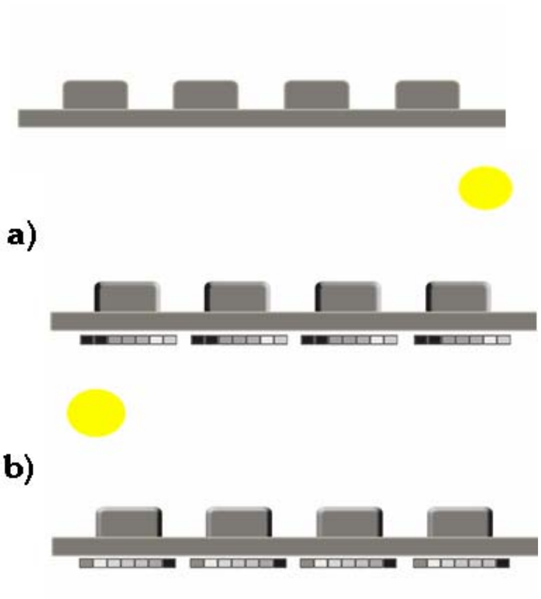


Figure 5.15: Effect of different illumination conditions (a,b) on the appearance of the gate floor

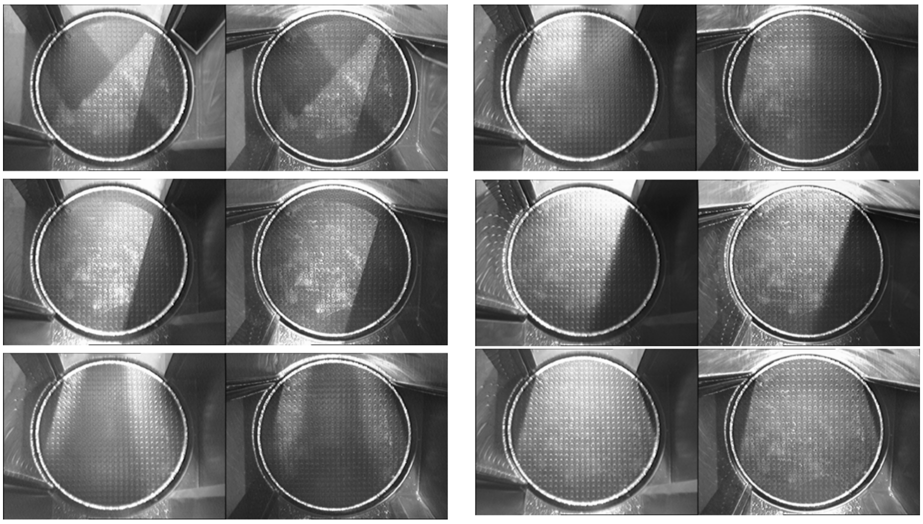


Figure 5.16: Typical photometric distortions and artifacts affecting the appearance of the gate floor.

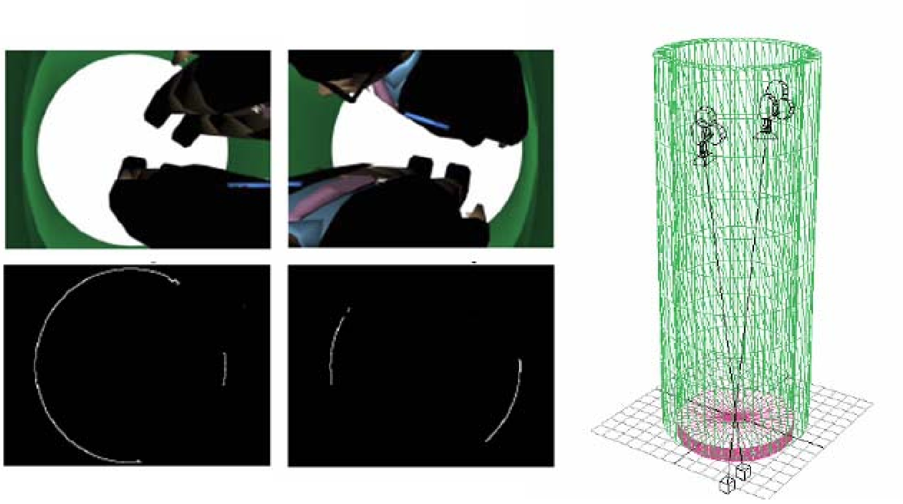


Figure 5.17: Simulation of the working conditions

features are computed based on the output of the segmentation processes associated with the two views.

5.6.2 Segmentation processes

Segmentation of the visible floor area A background subtraction approach is deployed in order to segment in each frame of both views the visible area of the interlock floor. Yet, we have to deal with all the disturbance factors briefly described in previous section: presence of sudden local illumination changes and shadows, mutable aspect of the background depending on the position of the light sources. Under these circumstances, most background subtraction algorithm relying on image intensities are prone to fail. Hence, we deploy the MF measure. In our experiments the patch side r deployed by MF was tuned to 11. Let B_{c1} and F_{c1} be respectively the background model and the current frame for camera 1. To perform background subtraction on the current frame, $MF_{B_{c1},F_{c1}}(x,y)$ is computed at each point (x,y) that may belong to the interlock floor, then a threshold is used to discriminate between floor points (high score) and non-floor points (low score). A pre-computed binary mask is deployed to eliminate the points of the images lying outside the floor area.

Preliminary results dealing with the use of MF showed that this was able to determine a rough estimation of the visible floor area and was robust to strong photometric distortions and artifacts. Anyway, due to the mutable aspect of the floor (as shown in Fig. 5.15), accuracy along the segmented borders was sometimes not precise enough. Hence, we devised an improvement to the basic approach. In particular, several back-

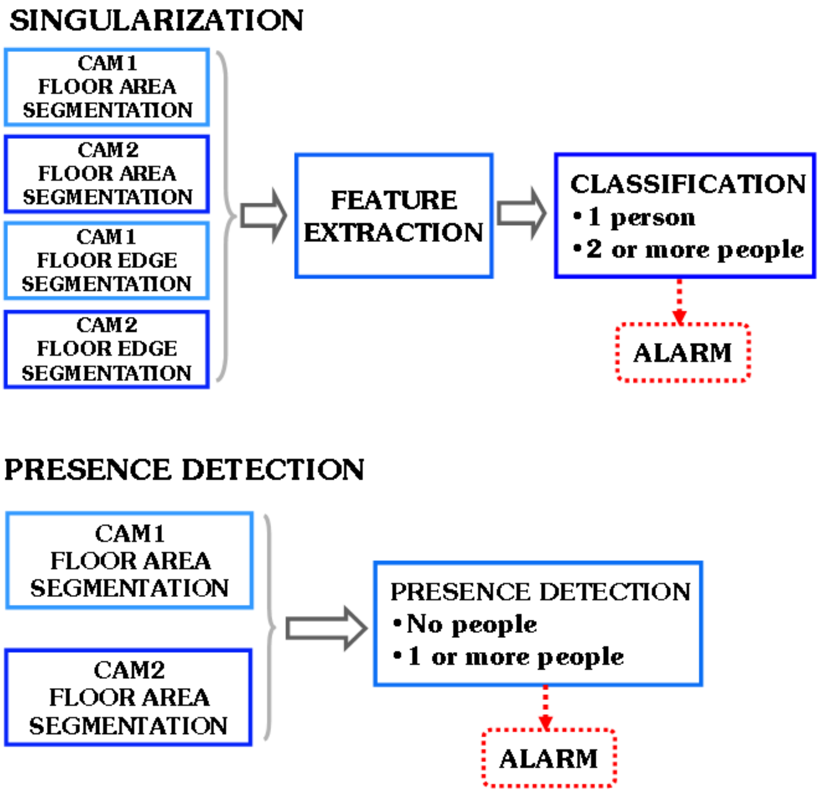

SINGULARIZATION
CAM1 FLOOR AREA SEGMENTATION
CAM2 FLOOR AREA SEGMENTATION
CAM1 FLOOR EDGE SEGMENTATION
CAM2 FLOOR EDGE SEGMENTATION
FEATURE EXTRACTION
CLASSIFICATION
• 1 person
• 2 or more people
ALARM
PRESENCE DETECTION
CAM1 FLOOR AREA SEGMENTATION
CAM2 FLOOR AREA SEGMENTATION
PRESENCE DETECTION
• No people
• 1 or more people
ALARM

Figure 5.18: Flow diagrams of singularization and presence detection algorithms

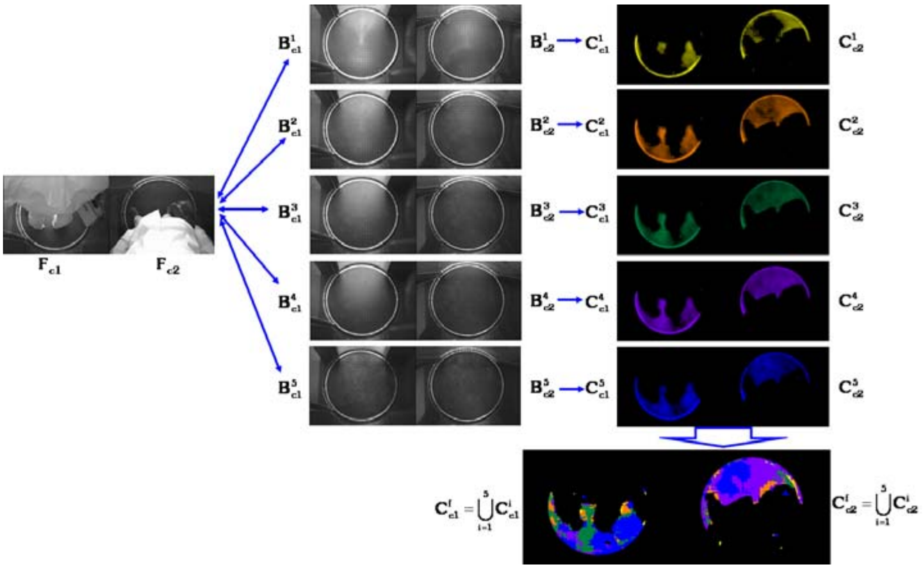

F_{c1}
F_{c2}
B^1_{c1}
B^2_{c1}
B^3_{c1}
B^4_{c1}
B^5_{c1}
$B^1_{c2} \rightarrow C^1_{c1}$
$B^2_{c2} \rightarrow C^2_{c1}$
$B^3_{c2} \rightarrow C^3_{c1}$
$B^4_{c2} \rightarrow C^4_{c1}$
$B^5_{c2} \rightarrow C^5_{c1}$
C^1_{c2}
C^2_{c2}
C^3_{c2}
C^4_{c2}
C^5_{c2}
$C^f_{c1} = \bigcup_{i=1}^{5} C^i_{c1}$
$C^f_{c2} = \bigcup_{i=1}^{5} C^i_{c2}$

Figure 5.19: Segmentation of the floor area by means of different background models.

ground models are computed rather than just one, each one encoding the appearance of the gate floor under different static illumination conditions. To this purpose, 5 different background models for each view are built, e.g. for camera 1 $B^1_{c1}, \cdots, B^5_{c1}$, with illumination conditions varying from very dark (B^1_{c1}) to very bright (B^5_{c1}). Each background model is obtained by varying the rheostat of the illuminator present inside the interlock. Under each illumination condition the actual background model is attained by averaging the values assumed by each point along an initialization sequence of 30 frames, during which the interlock is empty. This can be regarded as an initialization stage of our system. Then, at run-time, for each background model B^i_{c1}, a corresponding floor mask C^i_{c1} is computed by thresholding the MF score between F_{c1} and B^i_{c1}:

$$C^i_{c1}(x, y) = TH\left(MF_{B^i_{c1},F_{c1}}(x, y)\right) \tag{5.31}$$

Throughout all of our experiments, the threshold was set to 0.215. The outcome of this operation on each background model votes for the similarity between the current frame and the background. To classify a point as floor, the portion of the current frame must match at least one background model, i.e. the matching score for at least one background model must be above threshold. Hence, the final floor mask is obtained as the union between each floor mask:

$$C^f_{c1}(x, y) = \cup^5_{i=1} C^i_{c1} \tag{5.32}$$

Given the 5 background models $B^1_{c2}, \cdots, B^5_{c2}$ and the current frame F_{c2}, the same processing stages provide the final floor mask for the second view, C^f_{c2}. An example of how the approach works can be seen in Fig. 5.19. Here, the two current frames F_{c1}, F_{c2} (left column) are matched with each of the corresponding 5 background models (central column), yielding the floor masks shown in the right column. In the final floor masks, C^f_{c1}, C^f_{c2}, the contributions of the 5 floor masks are depicted with different colours, so as to point out how the deployment of different background models allows to increase the accuracy of this process.

A further step based on area-closing and area-opening morphology is applied to improve the output of the background subtraction stage by reducing false positives and false negatives from the final floor mask. An example is shown in Fig. 5.20: red and green circles (above) indicate the connected components filtered out from the floor masks, while the final binary output for both views is shown below.

To conclude this part, we show some experimental results regarding the floor area segmentation process. Footage has been acquired concerning scenes with different numbers of people inside the gate (0, 1, and 2) and notably varied illumination conditions obtained by the use of light sources placed outside the revolving door (as in Fig. 5.16). In particular, the reported segmentation results refer to 546 frames taken

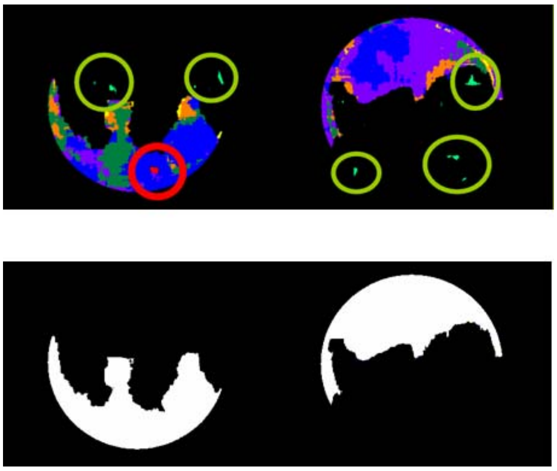


Figure 5.20: Morphological post-processing of the final floor mask.

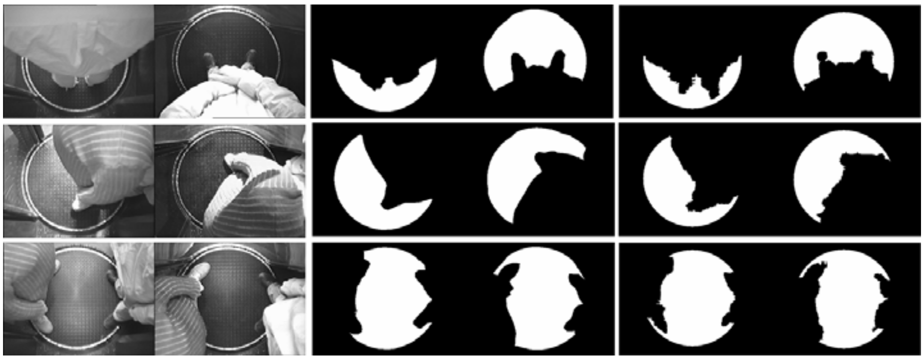


Figure 5.21: Qualitative comparison between ground-truth (centre) and output of the floor area segmentation algorithm (right).

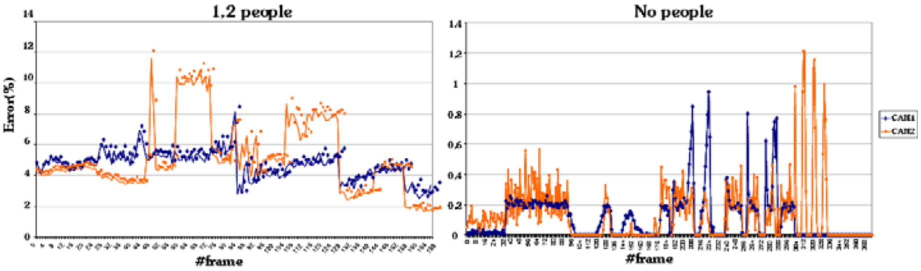


Figure 5.22: Quantitative assessment of the errors yielded by the floor area segmentation algorithm.

from 18 real sequences for each view. Of these frames, for those concerning 1 or 2 people (170 frames taken from 12 sequences) the ground-truth was obtained by hand-segmentation of the visible floor area and border. Fig. 5.21 shows samples of frames with 1 and 2 people occupying the interlock and allows for qualitative comparison between the ground-truth (centre) and the output obtained by the proposed floor segmentation approach. Besides, Fig. 5.22 reports the segmentation error (sum of false positives and false negatives normalized by number of points within the floor area) along the footage. These results demonstrate that the proposed approach provide significant accuracy, with a very small mean error around 5% when the interlock is occupied, and a negligible error when the interlock is empty.

Segmentation of the visible floor border In order to estimate the percentage of visible floor border, as illustrated in Fig. 5.23, an approach similar to the floor area segmentation algorithm is adopted. A binary mask is used to filter out the output of the background subtraction algorithm for those pixels not belonging to the border area (Fig. 5.23, b)). Even though the algorithm worked rather well by exploiting the natural edge of the floor border, we tested extensively our approach by applying a reflective tape on the floor border. In fact, this solution is minimally invasive and allows for a significant increase in the robustness of the border segmentation stage under strongly varying illumination conditions. This reflective tape is the only synthetic fiducial used in the whole system.

For what regards quantitative results, Fig. 5.24 reports the segmentation error computed along the footage by comparing the hand-segmented ground-truth with the output of the proposed floor border segmentation algorithm. When the interlock is occupied (1 or 2 people) the segmentation error is typically below 10%, while in case the interlock is empty the error is always negligible (never higher than 0.3%).

5.6.3 Presence detection

The algorithm enabling presence detection aims at raising an alarm when one or more occupants are present within the monitored interlock. This allows, for instance, to avoid performing singularization when the gate is empty, or to alert security personnel when the gate is occupied. The idea is to deploy the segmentation of the visible floor of the gate from the two views to estimate whether the interlock is empty or not. In particular, denoting as A_{c1} and A_{c2} the estimated percentage of visible floor area respectively in view 1 and 2, the presence detection rule is defined as follows:

$$presence = \begin{cases} true & (A_{c1} < Th_p) or (A_{c2} < Th_p) \\ false & otherwise \end{cases} \tag{5.33}$$

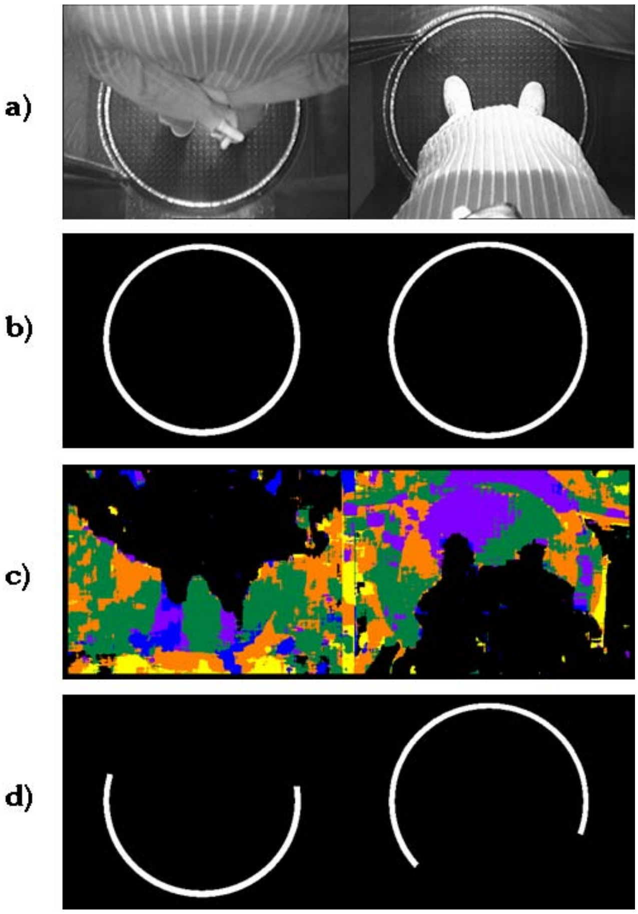


Figure 5.23: Floor border segmentation: a) input frames, b) binary mask, c) background subtraction results, d) final segmentation

In other words, presence is detected when the floor is not almost completely visible from at least one of the two cameras. Typically, Th_p is set to 80%. Obviously, the reliability of the presence detection algorithm heavily depends on the outcome of the floor area segmentation stage described in subsection 5.6.2. Some experimental results are shown Fig. 5.25, where the estimated percentage of visible floor area in the two views is plotted along the different frames of a video sequence. In this sequence, a person enters the revolving door, then the entrance door is automatically closed. The most important events are chronologically marked along the frames, that is: the person's first foot is visible, his second foot is visible, the person is completely inside the interlock, the entrance door starts to close, the entrance door is completely closed. In this sequence, the ground-truth is available only for certain frames and is indicated by blue dots. As it can be seen by comparison with the available groun-truth, the accu-

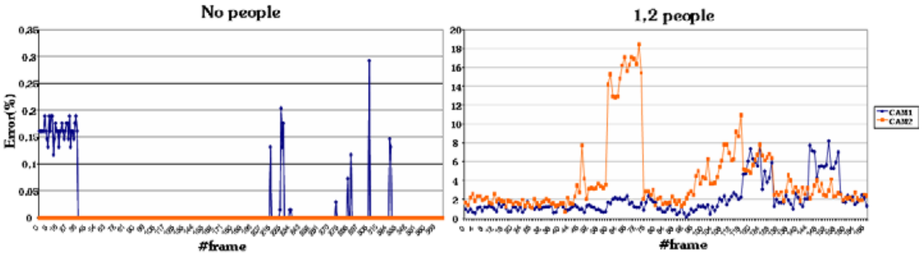


Figure 5.24: Error concerning the floor border segmentation stage in the two views compared to ground-truth.

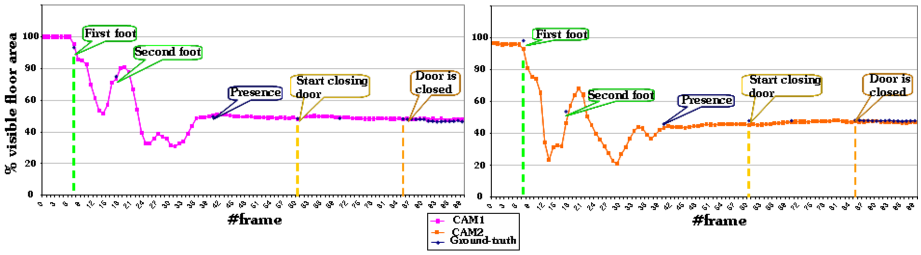


Figure 5.25: Experimental results comparing the percentage of visible floor area in the two views with ground-truth along a test sequence.

racy of the segmented floor area is very high along the frames of the sequence and for both views, despite the notable photometric distortions, shadows and artifacts appearing during the entrance of the person and the entrance door closing time. Furthermore, the use of the presence detection rule in (5.33) allows to reliably detect that the gate is not empty just after the first foot of the person is placed into the interlock.

5.6.4 Singularization by means of feature extraction and classification

In order to perform fast and reliable singularization we propose an approach based on feature extraction and classification. The key point for such an approach is to rely on a small number of distinctive features: the small number allows for a reduced computational cost, while the distinctiveness is necessary for reliability of singularization.

A thorough study concerning the possible features to be adopted in the singularization stage lead us to the determination of two simple features to be computed once the two views have been registered. The first feature, referred to as *Area OR*, is given by the percentage of the floor area which is visible in at least one of the views. This is motivated by the fact that the presence of a single person which stands around the

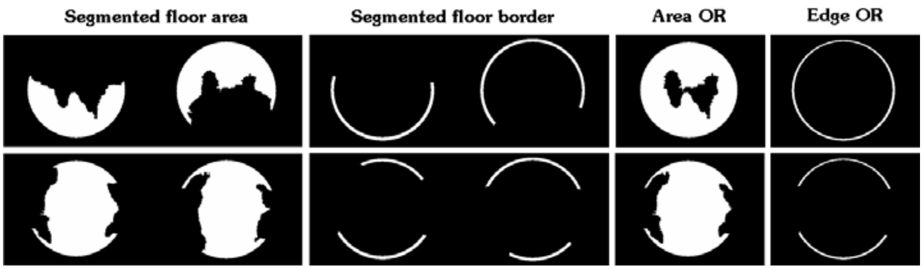


Figure 5.26: Extracted features ”Area OR” and ”Edge OR” in case of one person (above) and two people (below).

center of the interlock allows a large portion of floor area being seen by at least one of the two cameras (ideally, the whole area except for the portion of the person touching the floor, i.e. his feet). Conversely, the presence of two or more people would force a notable part of the floor to be not visible in both views. Similarly, the second feature is given by the percentage of the floor border which is visible in at least one of the views (referred to as *Edge OR*). This is also motivated by the fact that the presence of a single person standing around the center of the interlock generally allows all floor border points to be seen in at least one of the two views, which is not the case when two or more people are present. An example of how the Area OR and Edge OR features appear in two cases (one person and two people) is shown in Fig. 5.26.

Then, the output of the feature extraction step is provided to a trained classifier which discriminates between the two cases: ”one person” or ”two or more people”. The two features can be rapidly computed since the homographies between the two planar views with respect to the floor are fixed and precomputed at startup. During operating mode, singularization is applied only when the two doors of the interlock are closed. This reduces false alarms due to the single user’s movements while entering the revolving door. If singularization detects piggybacking, then an alarm is raised, the exit door stays closed and the entrance door is opened once again to allow the occupants to leave the interlock.

Experimental results are shown here by means of iterated 2-fold cross-validation. In particular, the frames of the available footage sequences concerning the presence of 1 or 2 people (overall 170 frames taken from 12 real sequences) were randomly subdivided into two groups, A and B. As shown in Tab. 5.2, this subdivision has been done 10 times by means of different shuffles. For each case, two SVM classifiers [22] are trained on the two sets, then each set is used as testing sequence for the other. An example of two classifiers obtained with this kind of evaluation is depicted in Fig. 5.27, which shows that training has been done toward generalization and to avoid data

♯	% E_A	% M_A	% F_A	% E_B	% M_B	% F_B
1	2	0	2	0	0	0
2	0	0	0	0	0	0
3	0	0	0	2	0	2
4	2	0	2	0	0	0
5	0	0	0	0	0	0
6	0	0	0	0	0	0
7	0	0	0	3	0	3
8	0	0	0	3	0	3
9	0	0	0	0	0	0
10	0	0	0	0	0	0

Table 5.2: Cross-validation of SVM-based singularization: percentages of error, missed detection, false alarm yielded for different subdivision of data into two subsets (A and B).

over-fitting. Tab. 5.2 shows the percentages of, respectively, total errors (E), missed detection (M, i.e. 'two or more people' being recognized as 'one person') and false alarms (F, i.e. 'one person' being recognized as 'two or more people') for both sets A and B along the various subdivisions. It can be noted that the total error percentages are extremely low in all cases, being often 0% and anyway never higher than 3%. This demonstrates that the two classes are distinguishable by means of the chosen features. Furthermore, in all cases the errors are represented by false alarms only, with no missed detection being actually yielded by our system: this is very important for practical applications, since the most important aspect in anti-piggibacking and anti-tailgating is to avoid misclassifying two or more people as one, while a few false alarms can be usually easily dealt with.

It is worth pointing out that the proposed method including all processing stages is very efficient: in our implementation, which does not contemplate any hardware or SIMD optimization, response time for each new frame is 140 ms (i.e. $\approx$ 7 fps) on a 2.14 GHz Intel Core Duo.

5.7 Acknowledgments

The research work presented in Sec. 5.6 has been supported by Plexa (`www.plexa.com`). We wish to thank them for the courtesy of granting the publishing of the results presented therein.

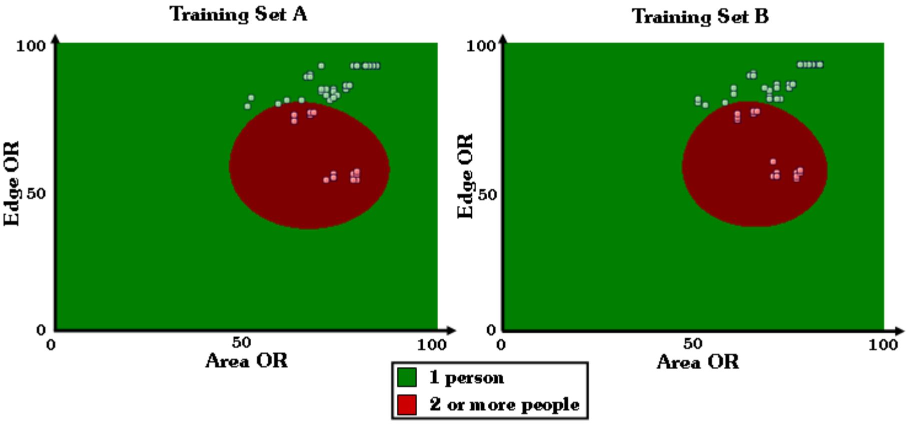


Figure 5.27: An example of SVM classification of two sets A, B.

Chapter 6

Conclusions

This dissertation has presented the research activity concerning visual correspondence carried out during the Ph.D. course. In particular, three main problems related to visual correspondence have been investigated: fast pattern matching, stereo correspondence, robust image matching.

With regards to the first topic, novel methods for fast and exhaustive pattern matching have been proposed. These methods rely on L_p norm-based dissimilarity functions, as well as correlation-based similarity functions. Thanks to the derivation of successions of increasingly tighter bounding functions, it has been shown that it is possible to safely detect mismatching candidates at a small computational cost. The consequence is that it is possible to notably speed-up pattern matching (up to two orders of magnitude) without deteriorating the optimality of the search, i.e. the candidates found are the same as those detected by a Full-Search investigation (*exhaustive* search). Measures for which the proposed techniques have shown to yield computational savings are the SAD and the SSD, for what concerns those measures derived from the L_p norm, and the NCC and ZNCC, as for correlation-based measures.

Interesting future work may be carried out to further develop the ideas at the basis of the proposed techniques. For example, the derivation of specific criteria to perform a more advanced partitioning of template and image subwindow, as well as the use of different bounding functions. It is also worth noting that recently novel research work [87, 88, 102, 134] has been proposed in literature that demonstrates the timeliness of this research topic.

As far as stereo correspondence is concerned, two novel aggregation strategies have been proposed. The first strategy concerns a very accurate approach to carry out variable support-based stereo correspondence based on segmentation information, and it turned out to be state-of-the-art in terms of accuracy between local stereo algorithms on the standard evaluation framework for the stereo community [92]. The latter approach deploys incremental techniques and segmentation to derive an very efficient, yet accurate, stereo aggregation method. This approach turned out to be the best performing one between state-of-the-art local algorithms in terms of cost-performance trade-off within the performance evaluation carried out in Section 3.5. In addition, a novel global method has been proposed, which is based on a joint exploitation of a local aggregation cost based on a variable support and a global optimization framework based on Scanline Optimization. The proposed method resulted being state-of-the-art in terms of accuracy between Scanline Optimization and Dynamic Programming-based stereo algorithms on the stereo evaluation framework [92].

In addition, the proposed taxonomy and performance evaluation introduced a new approach that extends the standard reference methodology for performance evaluation of stereo correspondence algorithms [92] by jointly evaluating accuracy and computational cost.

The research carried out on stereo correspondence opened up the way to novel aggregation strategies based on variable support, and paved the way for interesting future work. For instance, the proposed aggregation strategies might be deployed within a global framework. In addition, the proposed method might benefit of optimization strategies (e.g. implementation over embedded hardware (FPGA, ASICs, ..) or over GPUs, use of SIMD-based optimization) in order to notably decrease the computational burden and achieve near real-time or real-time performances. Finally, it is worth pointing out that the methodology deployed to evaluate the performance of stereo aggregation strategies based on variable support might be extended to stereo correspondence algorithms in general.

Research activity concerning specific applications of stereo vision has also been presented. In particular, a study concerning accurate 3D reconstruction based on space-time stereo and projected texture has been proposed, that yielded a novel local stereo algorithm as well as a novel approach to perform space-time stereo under dynamic scenes. Moreover, a multi-view algorithm for detection of vandal events such as graffiti has been proposed. This approach allowed to overcome the main limitations of single-view state-of-the-art techniques, in particular in terms of a notable reduction of false positives. Finally, a novel algorithm for change detection based on a stereo camera has been proposed, that exploits both appearance and range information to increase the robustness of the system against camouflage, shadows and sudden illumination changes.

Finally, as for the third visual correspondence topic investigated, a novel class of measures for robust visual correspondence under disturbance factors such as photometric distortions, noise and occlusions has been proposed. The proposed approach is based on the order preservation hypothesis, and aims at measuring how well the ordering constraint between neighboring pixels is preserved. The novel measures have demonstrated to be state-of-the-art in a pattern matching context, where also a review of the state of the art and a performance evaluation on a novel dataset has been proposed. Furthermore, the novel measures were deployed also for the task of change detection for video surveillance. In particular, a novel algorithm for background subtraction robust to sudden illumination changes has been proposed, as well as a case study dealing with singularization and access monitoring in interlocks.

6.1 Summary of contributions

We report here the list of the peer-reviewed international publications that deal with the material presented in this dissertation. The presented list of publications is subdivided into sections which refer to the corresponding chapters of the dissertation.

Chapter 1

1. F. Tombari, S. Mattoccia, L. Di Stefano, ”Full search-equivalent pattern matching with Incremental Dissimilarity Approximations”, *IEEE Transactions on Pattern Analysis and Machine Intelligence* (PAMI), 31(1), pp 129-141, January 2009

2. S. Mattoccia, F. Tombari, L. Di Stefano, ”Fast full-search equivalent template matching by Enhanced Bounded Correlation”, *IEEE Transactions on Image Processing* (TIP),17(4), pp 528-538, April 2008

3. S. Mattoccia, F. Tombari, L. Di Stefano, M. Pignoloni, Efficient and optimal block matching for motion estimation”, *14th IAPR International Conference on Image Analysis and Processing* (ICIAP 2007), September 10-13, 2007, Modena, Italy

4. F. Tombari, S. Mattoccia, L. Di Stefano, Template Matching based on the L_p norm using sufficient conditions with incremental approximations”, *IEEE International Conference on Advanced Video and Signal Based Surveillance* (AVSS 2006), November 22-24, 2006, Sydney, Australia

5. S. Mattoccia, F. Tombari, L. Di Stefano, Reliable rejection of mismatching candidates for efficient ZNCC template matching”, *IEEE International Conference on Image Processing* (ICIP 2008), October 12-15, 2008, San Diego, California, USA

Chapter 2

6. F. Tombari, S. Mattoccia, L. Di Stefano, E. Addimanda, Near real-time stereo based on effective cost aggregation", *International Conference on Pattern Recognition* (ICPR 2008), December 8-11, 2008, Tampa, Florida, USA

7. F. Tombari, S. Mattoccia, L. Di Stefano, E. Addimanda, Classification and evaluation of cost aggregation methods for stereo correspondence", *IEEE International Conference on Computer Vision and Pattern Recognition* (CVPR 2008), June 24-26, 2008, Anchorage, Alaska

8. F. Tombari, S. Mattoccia, L. Di Stefano, Segmentation-based adaptive support for accurate stereo correspondence", *IEEE Pacific-Rim Symposium on Image and Video Technology* (PSIVT 2007), December 17-19, 2007, Santiago, Chile

9. S. Mattoccia, F. Tombari, L. Di Stefano, Stereo vision enabling precise border localization within a scanline optimization framework", *8th Asian Conference on Computer Vision* (ACCV 2007), November 18-22, 2007, Tokyo, Japan

Chapter 3

10. F. Tombari, L. Di Stefano, S. Mattoccia, A. Zanetti, Graffiti detection using a Time-Of-Flight camera", *Advanced Concepts for Intelligent Vision Systems* (ACIVS 2008), October 20-24, 2008, Juan-les-Pins, France

11. L. Di Stefano, F. Tombari, A. Lanza, S. Mattoccia, S. Monti, "Graffiti detection using two views", *ECCV 8th International Workshop on Visual Surveillance* (VS 2008), October 17, 2008, Marseille, France

12. F. Tombari, S. Mattoccia, L. Di Stefano, F. Tonelli, Detecting motion by means of 2D and 3D information", *ACCV'07 Workshop on Multi-dimensional and Multi-view Image Processing* (ACCV WS 2007), November 19, 2007, Tokyo, Japan

Chapter 4

13. M. Magno, F. Tombari, D. Brunelli, L. Di Stefano, L. Benini, "Multi-modal video surveillance aided by pyroelectric infrared sensors", *ECCV Workshop on Multi-camera and Multi-modal Sensor Fusion, Algorithms and Applications* (M2SFA2), October 18, 2008, Marseille, France

14. L. Di Stefano, F. Tombari, S. Mattoccia, M. Balasso, Multi-view access monitoring and singularization in interlocks", *IEEE International Conference on Advanced Video and Signal Based Surveillance* (AVSS 2008),September 1-3, 2008, Santa Fe, New Mexico, USA

15. P. Azzari, L. Di Stefano, F. Tombari, S. Mattoccia, "Markerless augmented reality using image mosaics", *International Conference on Image and Signal Processing* (ICISP 2008), July 1-3, 2008, Cherbourg-Octeville, Normandy, France

16. F. Tombari, L. Di Stefano, S. Mattoccia, A. Galanti, Performance evaluation of robust matching measures", *3rd International Conference on Computer Vision Theory and Applications* (VISAPP 2008), January 22-25, 2008, Funchal, Madeira Island, Portugal

17. L. Di Stefano, F. Tombari, S. Mattoccia, E. De Lisi, Robust and accurate change detection under sudden illumination variations", *ACCV'07 Workshop on Multidimensional and Multi-view Image Processing* (ACCV WS 2007), November 19, 2007, Tokyo, Japan

18. F. Tombari, L. Di Stefano, S. Mattoccia, A robust measure for visual correspondence", *14th IAPR International Conference on Image Analysis and Processing* (ICIAP 2007), September 10-13, 2007, Modena, Italy

19. F. Tombari, L. Di Stefano, S. Mattoccia, chapter "Robust visual correspondence: theory and applications" in book: "Stereo vision", editor: Asim Bhatti, November 2008, publisher: I-Tech Education and Publishing Kirchengasse 43/3, A-1070, Vienna, Austria

Appendix A

On the generalization of the IDA technique

We investigate here the possibility of generalizing the IDA approach, presented in Section 2.2, to an arbitrary metric. According to the notation adopted, we indicate the arbitrary metric used to evaluate dissimilarities as $d\left(X, Y_j\right)$ and the corresponding *partial* distances induced by P as $d\left(X, Y_j\right)_{S_t}$. Though the triangular inequality can still be applied to subvectors

$$d\left(X, Y_j\right)_{S_t} \geq \left|d\left(X, 0\right)_{S_t} - d\left(Y_j, 0\right)_{S_t}\right|, \quad t = 1, \dots r \tag{A.1}$$

summation of both members now yields

$$\sum_{t=1}^{r} d\left(X, Y_j\right)_{S_t} \geq \sum_{t=1}^{r} \left|d\left(X, 0\right)_{S_t} - d\left(Y_j, 0\right)_{S_t}\right| \tag{A.2}$$

Therefore, a sufficient condition for the right-hand side of (A.2) to be a lower-bound of $d\left(X, Y_j\right)$ is

$$d\left(X, Y_j\right) \geq \sum_{t=1}^{r} d\left(X, Y_j\right)_{S_t} \tag{A.3}$$

Unfortunately, the above inequality does not hold for an arbitrary metric (e.g. for the L_p-distance when $p > 1$).

Interestingly, though perhaps of rather limited practical relevance, it is possible to define at least one class of metrics that allows for the generalization of our method. Let each of $d_t\left(\cdot\right), \ t = 1, \dots r$ and $\tilde{d}\left(\cdot\right)$ be a metric, we define

$$\bar{d}\left(X, Y_j\right) = \left(\sum_{t=1}^{r} d_t\left(X, Y_j\right)_{S_t}\right) + \tilde{d}\left(X, Y_j\right) \tag{A.4}$$

with distances between subvectors denoted according to the usual notation. It is straightforward to prove that function $\overline{d}\left(X, Y_j\right)$ is a metric and that (A.4) defines a class of distances satisfying sufficient condition (A.3), with

$$\overline{d_0}\left(X, Y_j\right) = \sum_{t=1}^{r} d_t\left(X, Y_j\right)_{S_t} \tag{A.5}$$

being the smallest of such distances.

Appendix B

Proof of *Lemma* 2.4.4

Proof. The lemma can be proved by contradiction. Let us assume that:

$$\sqrt{\sum_{k\in S_1} a_k^2}\cdot\sqrt{\sum_{k\in S_1} b_k^2}+\sqrt{\sum_{k\in S_2} a_k^2}\cdot\sqrt{\sum_{k\in S_2} b_k^2}>\sqrt{\sum_{k\in S} a_k^2}\cdot\sqrt{\sum_{k\in S} b_k^2} \tag{B.1}$$

Since

$$\sqrt{\sum_{k\in S} a_k^2}\cdot\sqrt{\sum_{k\in S} b_k^2}=\sqrt{(\sum_{k\in S_1} a_k^2+\sum_{k\in S_2} a_k^2)\cdot(\sum_{k\in S_1} b_k^2+\sum_{k\in S_2} b_k^2)} \tag{B.2}$$

the assumption can be rewritten as:

$$\left(\sqrt{\sum_{k\in S_1} a_k^2}\cdot\sqrt{\sum_{k\in S_1} b_k^2}+\sqrt{\sum_{k\in S_2} a_k^2}\cdot\sqrt{\sum_{k\in S_2} b_k^2}\right)^2>\left(\sum_{k\in S_1} a_k^2+\sum_{k\in S_2} a_k^2\right)\cdot\left(\sum_{k\in S_1} b_k^2+\sum_{k\in S_2} b_k^2\right)$$

Then by algebraic manipulation the following absurd result is attained:

$$\begin{aligned}0 &> \sum_{k\in S_1} a_k^2\cdot\sum_{k\in S_2} b_k^2+\sum_{k\in S_2} a_k^2\cdot\sum_{k\in S_1} b_k^2-\\ &-2\cdot\sqrt{\sum_{k\in S_1} a_k^2}\cdot\sqrt{\sum_{k\in S_1} b_k^2}\cdot\sqrt{\sum_{k\in S_2} a_k^2}\cdot\sqrt{\sum_{k\in S_2} b_k^2}=\\ &=\left(\sqrt{\sum_{k\in S_1} a_k^2}\cdot\sqrt{\sum_{k\in S_2} b_k^2}-\sqrt{\sum_{k\in S_2} a_k^2}\cdot\sqrt{\sum_{k\in S_1} b_k^2}\right)^2\end{aligned}$$

□

Bibliography

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

[2] Dean Anderson, Herman Herman, and Alonzo Kelly. Experimental characterization of commercial flash ladar devices. In *Int. Conf. of Sensing and Technology*, November 2005.

[3] D. Angiati, G. Gera, S. Piva, and C. Regazzoni. A novel method for graffiti detection using change detection algorithm. In *Proc. Int. Conf. Advanced Video and Signal-based Surveillance (AVSS'05)*, pages 242–246, 2005.

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

[5] P Aschwanden and W Guggenbuhl. Experimental results from a comparative study on correlation-type registration algorithms. In W Forstner and St. Ruwiedel, editors, *Robust computer vision*, pages 268–289. Wichmann, 1992.

[6] *http://www.axiumtech.net*. Axium Technologies.

[7] D.I. Barnea and H.F. Silverman. A class of algorithms for digital image registration. *IEEE Trans. on Computers*, C-21(2):179–186, 1972.

[8] *http://www.broadbanddiscovery.com*. Broadband Discovery Systems Inc.

[9] C.D. Bei and R.M. Gray. An improvement of the minimum distorsion encoding algorithm for vector quantization. *IEEE Trans. on Communication*, 33:1132–1133, 1985.

[10] A. Bevilacqua, L. Di Stefano, and A. Lanza. Coarse-to-fine strategy for robust and effcient change detectors. In *Proc. AVSS*, 2005.

[11] D N Bhat and S K Nayar. Ordinal measures for image correspondence. *IEEE Trans. Pattern Recognition and Machine Intelligence*, 20(4):415–423, April 1998.

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

[13] M. Bleyer and M. Gelautz. A layered stereo matching algorithm using image segmentation and global visibility constraints. *Jour. Photogrammetry and Remote Sensing*, 59:128–150, 2005.

[14] A.F. Bobick and S.S. Intille. Large occlusion stereo. *Int. Journal Computer Vision*, 33(3):181–200, 1999.

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

[16] K. Briechle and U.D. Hanebeck. Template matching using fast normalized cross correlation. In *Proc. of SPIE AeroSense Symposium*, volume 4387, Orlando, Florida, USA, 2001.

[17] M. Z. Brown, D. Burschka, and G. D. Hager. Advances in computational stereo. *IEEE Trans. Pattern Anal. Mach. Intell.*, 25(8):993–1008, 2003.

[18] M. Brunig and W. Niehsen. Fast full-search block matching. *IEEE Transactions on Circuits and Systems for Video Technology*, 11(2):241–247, 2001.

[19] S Chambon and A Crouzil. Dense matching using correlation: new measures that are robust near occlusions. In *Proc. British Machine Vision Conference (BMVC 2003)*, volume 1, pages 143–152, 2003.

[20] S. Chan, Y. Wong, and J. Daniel. Dense stereo correspondence based on recursive adaptive size multi-windowing. In *Proc. Image and Vision Computing New Zealand (IVCNZ'03)*, volume 1, pages 256–260, 2003.

[21] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Trans. PAMI*, 24:603–619, 2002.

[22] C Cortes and V Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

[23] A Crouzil, L Massip-Pailhes, and S Castan. A new correlation criterion based on gradient fields similarity. In *Proc. Int. Conf. Pattern Recognition (ICPR)*, pages 632–636, 1996.

[24] F. Crow. Summed-area tables for texture mapping. *Computer Graphics*, 18(3):207–212, 1984.

[25] Brian Curless and Marc Levoy. Better optical triangulation through spacetime analysis. In *ICCV*, 1995.

[26] `http://sourceforge.net/projects/opencvlibrary`.

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

[28] J. Davis, D. Nehab, R. Ramamoorthi, and S. Rusinkiewicz. Spacetime stereo: a unifying framework dor depth from triangulation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(2), February 2005.

[29] `http://people.csail.mit.edu/torralba/images`.

[30] `www.data-compression.info/Corpora/LukasCorpus`.

[31] `http://zulu.ssc.nasa.gov/mrsid`.

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

[33] 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.

[34] L. Di Stefano, M. Marchionni, and S. Mattoccia. A fast area-based stereo matching algorithm. *Image and Vision Computing*, 22(12):983–1005, 2004.

[35] L Di Stefano and S Mattoccia. Fast Template Matching using Bounded Partial Correlation. *Machine Vision and Applications*, 2002.

[36] L Di Stefano and S Mattoccia. A sufficient condition based on the cauchy-schwarz inequality for efficient template matching. In *Proc IEEE Int. Conf. on Image Processing (ICIP 2003)*, Barcelona, Spain, September 2003.

[37] L Di Stefano, S Mattoccia, and F Tombari. Zncc-based template matching using bounded partial correlation. *Pattern Recognition Letters*, 26(14):2129–2134, 2005.

[38] E. Durucan and T. Ebrahimi. Change detection and background extraction by linear algebra. *Proc. IEEE*, 89(10):1368–1381, 2001.

[39] C. Eveland, K. Konolige, and R.C. Bolles. Background modeling for segmentation of video-rate stereo sequences. In *1998 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'98)*, pages 266–271, 1998.

[40] P. Felzenszwalb and D. Huttenlocher. Efficient belief propagation for early vision. In *Proc. Int. Conf. on Computer Vision and Pattern Recognition (CVPR'04)*, 2004.

[41] A J Fitch, A Kadyrov, W J Christmas, and Kittler J. Orientation correlation. In P.L. Rosin and D. Marshall, editors, *British Machine Vision Conference*, volume 1, pages 133–142, 2002.

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

[43] X.Q. Gao, C.J. Duanmu, and C.R. Zou. A multilevel successive elimination algorithm for block matching motion estimation. *IEEE Trans. Image Processing*, 9(3):501–504, 2000.

[44] D. Geiger, B. Ladendorf, and A. Yuille. Occlusions and binocular stereo. *Int. Journ. of Computer Vision*, 14(3):211–226, 1995.

[45] M. Gerrits and P. Bekaert. Local stereo matching with segmentation-based outlier rejection. In *Proc. Canadian Conf. on Computer and Robot Vision (CRV 2006)*, pages 66–66, 2006.

[46] M. Gharavi-Alkhansari. A fast globally optimal algorithm for template matching using low-resolution pruning. *IEEE Trans. Image Processing*, 10(4):526–533, 2001.

[47] M. Ghazal, C. Vazquez, and A. Amer. Real-time automatic detection of vandalism behavior in video sequences. In *Proc. IEEE Int. Conf. on Systems, Man and Cybernetics (ISIC 2007)*, pages 1056–1060, 2007.

[48] S Giachetti. Matching techniques to compute image motion. *Image and Vision Computing*, 18:247260, 2000.

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

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

[51] M. Gong and Y.H. Yang. Near real-time reliable stereo matching using programmable graphics hardware. In *Proc. Conf. on Computer Vision and Pattern Recognition (CVPR 2005)*, volume 1, pages 924–931, 2005.

[52] R. Gonzalez and R. Woods. *Digital Image Processing*. Prentice Hall, 2nd edition, 2002.

[53] G. Gordon, M. Darrel, M. Harville, and J. Woodfill. Background estimation and removal based on range and color. In *1999 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'99)*, 1999.

[54] A. Goshtasby. *2-D and 3-D image registration for medical, remote sensing and industrial applications*. Wiley, 2005.

[55] D. Gutchess and al. A background model initialization algorithm for video surveillance. In *Proc. ICCV*, volume 1, pages 733–740, 2001.

[56] P.S. Heckbert. Filtering by repeated integration. In *Proc. of SIGGRAPH*, pages 315–321, 1986.

[57] Y. Hel-Or and H. Hel-Or. Real-time pattern matching using projection kernels. *IEEE Tran. Pattern Analysis and Machine Intelligence*, 27(9):1430–1445, 2005.

[58] H. Hirschmuller. 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.

[59] H Hirschmuller. Stereo vision in structured environments by consistent semi-global matching. In *Proc. Conf. on Computer Vision and Pattern recognition (CVPR 2006)*, volume 2, pages 2386–2393, 2006.

[60] H. Hirschmuller, P. Innocent, and J. Garibaldi. Real-time correlation-based stereo vision with reduced border errors. *Int. Journ. of Computer Vision*, 47:1–3, 2002.

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

[62] L. Hong and G. Chen. Segment-based stereo matching using graph cuts. In *Proc. CVPR*, volume 1, page 7481, 2004.

[63] Y. W. Huang, C. Y. Chen, C. H. Tsai, C. F. Shen, and L. G. Chen. Survey on block matching motion estimation algorithms and architectures with new results. *The Journal of VLSI Signal Processing*, 42(3):297–320, 2006.

[64] Y. A. Ivanov, A. F. Bobick, and J. Liu. Fast lighting independent background subtraction. *International Journal of Computer Vision*, 37(2):199–207, June 2000.

[65] A. Kadyrov and M. Petrou. The 'invaders' algorithm: Range of values modulation for accelerated correlation. *IEEE Tran. Pattern Analysis and Machine Intelligence*, 28(11):1882–1886, 2006.

[66] S Kaneko, Y Satoh, and S Igarashi. Using selective correlation coefficient for robust image registration. *Journ. Pattern Recognition*, 36(5):1165–1173, May 2003.

[67] S.B. Kang, R. Szeliski, and J. Chai. Handling occlusions in dense multi-view stereo. In *Proc. Conf. on Computer Vision and Pattern Recognition (CVPR 2001)*, pages 103–110, 2001.

[68] S. M. Khan and M. Shah. A multiview approach to tracking people in crowded scenes using a planar homography constraint. In *Proc. European Conference on Computer Vision (ECCV'06)*, volume 4, pages 133–146, May 2006.

[69] J.C. Kim, K.M. Lee, B.T. Choi, and S.U. Lee. A dense stereo matching using two-pass dynamic programming with generalized ground control points. In *Proc. Conf. on Computer Vision and Pattern Recognition (CVPR 2005)*, pages 1075–1082, 2005.

[70] A. Klaus, M. Sormann, and K. Karner. Segment-based stereo matching using belief propagation and a self-adapting dissimilarity measure. In *Proc. Int. Conf. on Pattern Recognition (ICPR 2006)*, volume 3, pages 15–18, 2006.

[71] V. Kolmogorov and R. Zabih. Computing visual correspondence with occlusions via graph cuts. In *Proc. Int. Conf. Computer Vision (ICCV 2001)*, volume 2, pages 508–515, 2001.

[72] Kurt Konolige. Small vision systems: hardware and implementation. In *Eighth International Symposium on Robotics Research*, pages 111–116, 1997.

[73] W. Krattenthaler, K.J. Mayer, and M. Zeiler. Point correlation: a reduced-cost template matching technique. In *Proc. of the 1st IEEE International Conference on Image Processing*, volume 1, pages 208–212, Austin, Texas, USA, 1994.

[74] S H Lai. Robust image matching under partial occlusion and spatially varying illumination change. *Computer Vision and Image Understanding*, 78:84–98, 2000.

[75] A Lanza and L Di Stefano. Detecting changes in grey level sequences by *ML* isotonic regression. In *Proc. IEEE Int. Conf. on Video and Signal Based Surveillance (AVSS)*, page 4, 2006.

[76] A. Lanza, L. Di Stefano, J. Berclaz, F. Fleuret, and P. Fua. Robust multi-view change detection. In *Proc. British Machine Vision Conference (BMVC'07)*, September 2007.

[77] C.H. Lee and L.H. Chen. A fast motion estimation algorithm based on the block sum pyramid. *IEEE Trans. Image Processing*, 6(11):1587–1591, 1997.

[78] C. Lei, J. Selzer, and Y.H. Yang. Region-tree based stereo using dynamic programming optimization. In *Proc. Conf. on Computer Vision and Pattern Recognition (CVPR 2006)*, volume 2, pages 2378 – 2385, 2006.

[79] G.H. Lerg, A.J. Devine, D.L. Roberts, and R.E. Johnson. Graffiti detection system and method of using the same. US Patent 6600417, July 2003.

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

[81] J.P. Lewis. Fast template matching. In *Proc. Conf. on Vision Interface*, pages 120–123, Quebec, Canada, May 1995.

[82] Reoxiang Li, Bing Zeng, and M. L. Liou. A new three-step search algorithm for block motion estimation. *IEEE Transactions on Circuits and Systems for Video Technology*, 4(4):438–442, 1994.

[83] W. Li and E. Salari. Successive elimination algorithm for motion estimation. *IEEE Trans. on Image Processing*, 4(1):105–107, 1995.

[84] S. N. Lim, A. Mittal, L. S. Davis, and N. Paragios. Fast illumination-invariant background subtraction using two-views: Error analysis, sensor placement and applications. In *Proc. IEEE Int. Conf. Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 1071–1078, June 2005.

[85] D.G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[86] Gelautz M. Bleyer, M. Simple but effective tree structures for dynamic programming-based stereo matching. In *Proc. Int. Conf. on Computer Vision Theory and Applications (VISAPP)*, volume 2, 2008.

[87] A. Mahmood and S Khan. Early termination algorithms for correlation coefficient based block matching. In *Proc. Int. Conf. on Image Processing (ICIP 07)*, volume 2, pages 469–472, 2007.

[88] A. Mahmood and S. Khan. Exploiting inter-frame correlation for fast video to reference image alignment. In *Proc. Asian Conference on Computer Vision (ACCV 07)*, pages 647–656, 2007.

[89] J Martin and J L Crowley. Experimental comparison of correlation techniques. In *Proc. Int. Conf. on Intelligent Autonomous Systems*, volume 4, pages 86–93, 1995.

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

[91] M. Mc Donnel. Box-filtering techniques. *Computer Graphics and Image Processing*, 17:65–70, 1981.

[92] `www.vision.middlebury.edu/stereo`.

[93] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. In *Proc. Int. Conf. on Computer Vision and Pattern Recognition (CVPR 03)*, pages 257–263, 2003.

[94] K. Mikolajczyk and C. Schmid. Scale and affine invariant interest point detectors. *International Journal of Computer Vision*, 60(1):63–86, 2004.

[95] A Mittal and V Ramesh. An intensity-augmented ordinal measure for visual correspondence. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 849–856, 2006.

[96] A. Moradi, R. Dianat, S. Kasaei, and M.T.M. Shalmani. Enhanced cross-diamond-hexagonal search algorithms for fast block motion estimations. In *IEEE Conference on Advanced Video and Signal Based Surveillance*, pages 558–563, Como, Italy, September 2005.

[97] H. K. Nishihara. Prism: A practical real-time imaging stereo matcher. Technical report, Cambridge, MA, USA, 1984.

[98] F Odone, E Trucco, and A Verri. General purpose matching of grey level arbitrary images. In C Arcelli, L Cordella, and G Sanniti di Baja, editors, *4th Int. Workshop on Visual Form, LNCS*, pages 573–582. Springer-Verlag, 2001.

[99] N. Ohta. A statistical approach to background subtraction for surveillance systems. In *Proc. Int. Conf. on Computer Vision (ICCV01)*, volume 2, page 481486, 2001.

[100] M. Okutomi and T. Kanade. A locally adaptive window for signal matching. *Int. Journal of Computer Vision*, 7(2):143–162, 1992.

[101] M. Okutomi, Y. Katayama, and S. Oka. A simple stereo algorithm to recover precise object boundaries and smooth surfaces. *Int. Journ. of Computer Vision*, 47(1-3):261–273, 2002.

[102] W.H. Pan and S.D. Wei. Efficient ncc-based image matching in walsh-hadamard domain. In *Proc. 10th European Conference on Computer Vision (ECCV 08)*, pages 468–480, 2008.

[103] Z. Pan, K. Kotani, and T. Ohmi. Fast encoding method for vector quantization using modified l_2-norm pyramid. *IEEE Signal Processing Letters*, 12(9):609–612, 2005.

[104] `www.faculty.idc.ac.il/toky/Software/software.htm`.

[105] R.B. Potts. Some generalized order-disorder transitions. In *Proc. Cambridge Philosophical Society*, volume 48, pages 106–109, 1995.

[106] R.J. Radke, S. Andra, O. Al-kofahi, and B. Roysam. Image change detection algorithms: a systematic survey. *IEEE Trans. Image Processing*, 14(3):294–307, 2005.

[107] A Rosenfeld and G.J. Vanderburg. Coarse-fine template matching. *IEEE Trans. on Sys., Man and Cyb.*, 7:104–107, 1977.

[108] H. L. Royden. *Real analysis*, page 118. Prentice Hall, 3rd edition, 1988.

[109] C. Sacchi, C. Regazzoni, and G. Vernazza. A neural network-based image processing system for detection of vandal acts in unmanned railway environments. In *Proc. Int. Conf. Image Analysis and Processing (ICIAP'01)*, pages 529–534, 2001.

[110] J. Salvi, J. Pages, and J. Batlle. Pattern docification strategies in structured light systems. *Pattern Recognition*, 37(4), 2004.

[111] D Scharstein. Matching images by comparing their gradient fields. In *Proc. Int. Conf. Pattern Recognition (ICPR)*, volume 1, pages 572–575, 1994.

[112] 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.

[113] M Schwarz. Revolving door with metal detection security. US Patent 946867, June 2004.

[114] M Schwarz and R Mayer. Anti-piggybacking: sensor system for security door to detect two individuals in one compartment. US Patent 5201906, April 1993.

[115] H. Schweitzer, J.W. Bell, and F. Wu. Very fast template matching. In *Proc. European Conference on Computer Vision*, pages 358–372, 2002.

[116] P Seitz. Using local orientational information as image primitive for robust object recognition. In *Proc. SPIE, Visual Communication and Image Processing IV*, volume 1199, pages 1630–1639, 1989.

[117] J Shen and S Castan. An optimal linear operator for step edge detection. *Graphical Models and Image Processing (CVGIP)*, 54(2):112–133, 1992.

[118] P. Simard, L. Bottou, P. Haffner, and Y. Le Cun. Boxlets: a fast convolution algorithm for signal processing and neural networks. *Advances in Neural Information Processing Systems, Kearns, M., Solla, S., Cohn, D. (Eds.)*, 11:571–577, 1999.

[119] Changming Sun. Moving average algorithms for diamond, hexagon, and general polygonal shaped window operations. *Pattern Recognition Letters*, 27(6):556–566, 2006.

[120] J. Sun, Y. Li, S.B. Kang, and H.Y. Shum. Symmetric stereo matching for occlusion handling. In *Proc. CVPR*, volume 2, pages 399–406, 2005.

[121] J. Sun, H. Y. Shum, and N.N. Zheng. Stereo matching using belief propagation. *IEEE Trans. PAMI*, 25(7):787–800, 2003.

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

[123] T. Toivonen, J. Heikkila, and O. Silven. A new algorithm for fast full search block motion estimation based on number theoretic transforms. In *Proc. of the 9th International Workshop on Systems, Signals and Image Processing*, pages 90–94, Manchester, U.K., November 2002.

[124] F. Tombari, S. Mattoccia, and L. Di Stefano. Segmentation-based adaptive support for accurate stereo correspondence. In *Proc. IEEE Pacific-Rim Symposium on Image and Video Technology (PSIVT'07)*, 2007.

[125] *http://www.traptec.com*. Traptec Inc.

[126] E. Trucco and A. Verri. *Introductory Techniques for 3-D Computer Vision*. Prentice Hall, 1998.

[127] F Ullah, S Kaneko, and S Igarashi. Orientation code matching for robust object search. *IEICE Trans. Information and Systems*, E-84-D(8):999–1006, 2001.

[128] G.J. Vanderburg and A Rosenfeld. Two-stage template matching. *IEEE Trans. on Image Processing*, 26:384–393, 1977.

[129] O. Veksler. Stereo matching by compact windows via minimum ratio cycle. In *Proc. Int. Conf. on Computer Vision (ICCV'01)*, volume 1, pages 540–547, 2001.

[130] O. Veksler. Fast variable window for stereo correspondence using integral images. In *Proc. Conf. on Computer Vision and Pattern Recognition (CVPR 2003)*, pages 556–561, 2003.

[131] P. Viola and M. J. Jones. Robust real-time face detection. *Int. Journal of Computer Vision*, 57(2):137–154, 2004.

[132] H.S. Wang and R.M. Mersereau. Fast algorithms for the estimation of motion vectors. *IEEE Trans. on Image Processing*, 8(3):435–439, 1999.

[133] Liang Wang, Miao Liao, Minglun Gong, Ruigang Yang, and David Nister. High-quality real-time stereo using adaptive cost aggregation and dynamic programming. In *Proc. 3rd Int. Symposium 3D Data Processing, Visualization and Transmission (3DPVT'06)*, pages 798–805, 2006.

[134] S.D. Wei and S.H. Lai. Efficient normalized cross correlation based on adaptive multilevel successive elimination. In *Proc. Asian Conference on Computer Vision (ACCV 07)*, 2007.

[135] Y. Wei and L. Quan. Region-based progressive stereo matching. In *Proc. CVPR*, volume 1, page 106113, 2004.

[136] O. Williams, M. Isard, and J. MacCormick. Estimating disparity and occlusions in stereo video sequences. In *Proc. CVPR*, 2005.

[137] B Xie, V Ramesh, and T Boult. Sudden illumination change detection using order consistency. *Image and Vision Computing*, 22(2):117–125, 2004.

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

[139] Q. et al. Yang. Stereo matching with color-weighted correlation, hierachical belief propagation and occlusion handling. In *Proc. Conf. on Computer Vision and Pattern Recognition (CVPR 2006)*, volume 2, pages 2347 – 2354, 2006.

[140] Q.X. Yang, L. Wang, R.G. Yang, S. Wang, M. Liao, and D. Nister. Real-time global stereo matching using hierarchical belief propagation. In *Proc. British Machine Vision Conference*, 2006.

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

[142] K.J. Yoon and I.S. Kweon. Stereo matching with symmetric cost functions. In *Proc. Conf. on Computer Vision and Pattern Recognition (CVPR 2006)*, volume 2, pages 2371 – 2377, 2006.

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

[144] R Zabih and J Woodfill. Non-parametric local transforms for computing visual correspondence. In *Proc. European Conf. Computer Vision*, pages 151–158, 1994.

[145] L. Zhang, B. Curless, and S.M. Seitz. Spacetime stereo: shape recovery for dynamic scenes. In *Proc. CVPR*, 2003.

[146] J. Zhao and J. Katupitiya. A fast stereo vision algorithm with improved performance at object borders. In *Proc. Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 5209–5214, 2006.

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

[148] C.L. Zitnick and S.B. Kand. Stereo for image-based rendering using image over-segmentation. *IJCV*, 75(1):49–65, 2007.

[149] B. Zitová and J. Flusser. Image registration methods: a survey. *Image and Vision Computing*, 21(11):977–1000, 2003.