Fast Full-Search Equivalent Template Matching by Enhanced Bounded Correlation

Stefano Mattoccia, *Member, IEEE*, Federico Tombari, *Student Member, IEEE*, and Luigi Di Stefano

***Abstract*—We propose a novel algorithm, referred to as enhanced bounded correlation (EBC), 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 full search algorithm. 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 online 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 offline 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.**

***Index Terms*—Bounded correlation, Cauchy–Schwarz inequality, fast Fourier transform (FFT), normalized cross correlation (NCC), template matching.**

I. Introduction

THE *full search* (FS) template matching algorithm consists of 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 current image subwindow. This determines a maximum-correlation, or minimum-distortion, position that locates the template in the examined image. Functions typically employed to carry out template matching are the *sum of absolute differences* (SAD), the *sum of squared differences* (SSD), and the *normalized cross correlation* (NCC).

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, *nonexhaustive* algorithms yield computational savings by reducing the search space [1]–[3] or by decomposing the template or the image into rectangular regions and approximating each as a polynomial [4]–[7].

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, a simple approach is known as partial distortion elimination (PDE) [8] and consists of 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 of defining a rapidly computable lower bounding function of the adopted distortion measure, so as to check quickly one or more sufficient conditions to skip unmatching positions 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) [9]–[11] and Gharavi–Alkhansari's algorithm [12].

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 [13] and [14]). 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 unmatching 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 [15]. In [16], we subsequently proposed an improved BPC formulation that deploys the Cauchy–Schwarz inequality.

This paper proposes a new algorithm, referred to as *enhanced bounded correlation* (EBC), that is aimed at exhaustive NCC-based template matching and relies on the concept of bounding the NCC. The main novelty with respect to BPC algorithms [15], [16] consists of 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. A second contribution of this paper is the definition of a procedure for online estimation of all parameters of the algorithm to allow a parameter-free EBC implementation. Experiments show that EBC dramatically outperforms BPC algorithms. Moreover, EBC always runs always faster than the FFT-based approach.

The paper is organized as follows. Sections II and III introduce notation and review related work. Section IV presents the mathematical grounds of the EBC approach. Section V describes the core EBC algorithm, which derives directly from the properties outlined in the previous section. Then, Section VI

Manuscript received June 21, 2007; revised December 20, 2007. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Dimitri Van De Ville.

The authors are with the Dipartimento di Elettronica, Informatica e Sistemistica (DEIS), Advanced Research Center on Electronic Systems (ARCES), University of Bologna, 40136 Bologna, Italy (e-mail: stefano.mattoccia@unibo.it; federico.tombari@unibo.it; luigi.distefano@unibo.it).

Digital Object Identifier 10.1109/TIP.2008.919362

shows the overall EBC algorithm, which includes some important additional features with respect to the core algorithm. Section VII reports the experimental results. Finally, conclusions are drawn in Section VIII.

II. 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

$$\eta(x,y) = \frac{\psi(x,y)}{||I_c(x,y)|| \cdot ||T||} \tag{1}$$

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

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

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

$$||I_c(x,y)|| = \sqrt{\sum_{j=1}^{N}\sum_{i=1}^{M} I^2(x+i, y+j)} \tag{3}$$

and the ℓ_2-norm of the template

$$||T|| = \sqrt{\sum_{j=1}^{N}\sum_{i=1}^{M} T^2(i,j)}. \tag{4}$$

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., [17] and [18]) while $||T||$ can be computed once at initialization time.

III. Related Work

BPC techniques [15], [16] 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 $\psi(x,y)$

$$\beta(x,y) \geq \psi(x,y) \tag{5}$$

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

$$\frac{\beta(x,y)}{||I_c(x,y)|| \cdot ||T||} \geq \frac{\psi(x,y)}{||I_c(x,y)|| \cdot ||T||} = \eta(x,y). \tag{6}$$

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

$$\frac{\beta(x,y)}{||I_c(x,y)|| \cdot ||T||} < \eta_M \tag{7}$$

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, (7) 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 (7) holds, then it is necessary to compute $\eta(x,y)$ and check the condition

$$\eta(x,y) \geq \eta_M. \tag{8}$$

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 [15]. Subsequently, a tighter bound was derived in [16] by deploying the Cauchy–Schwarz inequality as follows.

Given two p-dimensional vectors $\mathbf{a}$ and $\mathbf{b}$, the Cauchy–Schwarz inequality can be written as

$$\sum_{k=1}^{p} a_k \cdot b_k \leq \sqrt{\sum_{k=1}^{p} a_k^2} \cdot \sqrt{\sum_{k=1}^{p} b_k^2}. \tag{9}$$

Applying (9) to vectors T and $I_c(x,y)$ yields

$$\beta(x,y) = ||I_c(x,y)|| \cdot ||T|| \geq \psi(x,y). \tag{10}$$

Unfortunately, plugging (10) into (7) does not yield a useful sufficient condition since (7) turns out to be always false. However, as described in [16], 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

$$\beta(x,y) = \sum_{j=1}^{n}\sum_{i=1}^{M} I(x+i, y+j) \cdot T(i,j) + \sqrt{\sum_{j=n+1}^{N}\sum_{i=1}^{M} I^2(x+i, y+j)} \cdot \sqrt{\sum_{j=n+1}^{N}\sum_{i=1}^{M} T^2(i,j)}. \tag{11}$$

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.

IV. Mathematical Framework

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

A. Lemma

Let $\mathbf{a}, \mathbf{b} \in \mathbb{R}^p$ and $S = \{1, 2, \ldots, p\}$. 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:

$$\sqrt{\sum_{k\in S_1} a_k^2}\cdot\sqrt{\sum_{k\in S_1} b_k^2}+\sqrt{\sum_{k\in S_2} a_k^2}\cdot\sqrt{\sum_{k\in S_2} b_k^2} \leq \sqrt{\sum_{k\in S} a_k^2}\cdot\sqrt{\sum_{k\in S} b_k^2}. \tag{12}$$

Proof: See Appendix I. ■

B. Property I

Let $\mathbf{a},\mathbf{b} \in \mathbb{R}^p$ and $S = \{1,2,\ldots,p\}$. Hence, $\forall r \in \{1,\ldots,p\}|$

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

then the following inequalities hold:

$$\sum_{k\in S} a_k\cdot b_k \leq \sum_{t=1}^{r}\left(\sqrt{\sum_{k\in S_t} a_k^2}\cdot\sqrt{\sum_{k\in S_t} b_k^2}\right) \leq \sqrt{\sum_{k\in S} a_k^2}\cdot\sqrt{\sum_{k\in S} b_k^2}. \tag{13}$$

Proof: The left inequality can be easily derived from the application of the Cauchy–Schwarz inequality to each subvector 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 subvector 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 $\mathbf{a}$, $\mathbf{b}$.

C. Property II

Let $\mathbf{a},\mathbf{b} \in \mathbb{R}^p$ and $S = \{1,2,\ldots,p\}$. Hence, $\forall r \in \{1,\ldots,p\}|$

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

then the following inequalities hold:

$$\sum_{k\in S} a_k\cdot b_k \leq \sum_{k\in S_i} a_k\cdot b_k + \sum_{t=1,t\neq i}^{r}\left(\sqrt{\sum_{k\in S_t} a_k^2}\cdot\sqrt{\sum_{k\in S_t} b_k^2}\right) \leq \sum_{t=1}^{r}\left(\sqrt{\sum_{k\in S_t} a_k^2}\cdot\sqrt{\sum_{k\in S_t} b_k^2}\right). \tag{14}$$

Proof: Similar to Property I, the left inequality derives from the application of the Cauchy–Schwarz inequality to each subvector 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 subvector pair defined by subset S_i. ■

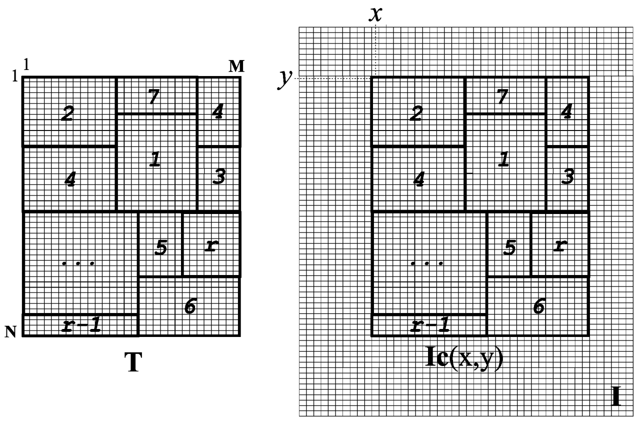

Fig. 1. Generic partitioning of template and current image subwindow.

Property II tells us that given an upper bound of ψ obtained by partitioning $\mathbf{a},\mathbf{b}$ into r subvectors as defined in Property I, a tighter upper bound can be obtained by replacing the product-of-norms term related to the subvector 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 subvector pair so as to replace the corresponding product-of-norms term.

V. Core EBC Algorithm

This section describes the core EBC algorithm, which relies on the mathematical properties presented in Section IV. 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. 1 shows a generic partitioning of such vectors into r subvectors, as required by Properties I and II. In general a subvector can consist of disjoint sets of pixels, as is the case of subvector 4 in Fig. 1.

To deploy the properties of Section IV within the bounded correlation framework outlined in Section III, 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 (7) that does not require calculation of any partial correlation term. If such a initial condition is not verified, we then apply Property II in successive steps. At each step, the product-of-norms term of a subvector pair is replaced by the corresponding cross-correlation term, to obtain a tighter bounding function and associated skipping condition (7).

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 subvectors made out of successive rows, as shown in Fig. 2. All subvectors are chosen to have the same number of rows, n, except for the last one (e.g., in Fig. 2, subvector r). Hence, with our partitioning scheme the first $r-1$ subvectors have $M\times n$ elements and the last one has $M\times(N-(r-1)\cdot n)$ elements.

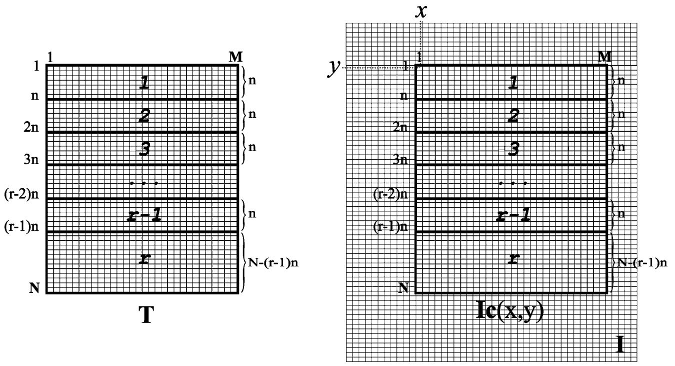

Fig. 2. Partitioning scheme adopted in our current EBC implementation.

In fact, EBC requires evaluation of the norms of all the subvectors 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 initialization 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 [17]. 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 [17], 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 subvector norms we need to run as many box-filters as the number of differently shaped rectangular windows corresponding to subvectors. Therefore, the choice of using two different shapes of subvectors 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 subvectors 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.

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

$$\beta^r(x,y)|_1^N = \sum_{t=1}^{r-1} \sqrt{\sum_{j=(t-1)\cdot n+1}^{t\cdot n} \sum_{i=1}^{M} I^2(x+i,y+j)} \cdot \sqrt{\sum_{j=(t-1)\cdot n+1}^{t\cdot n} \sum_{i=1}^{M} T^2(i,j)} + \sqrt{\sum_{j=(r-1)\cdot n+1}^{N} \sum_{i=1}^{M} I^2(x+i,y+j)} \cdot \sqrt{\sum_{j=(r-1)\cdot n+1}^{N} \sum_{i=1}^{M} T^2(i,j)}. \tag{15}$$

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

$$\frac{\beta^r(x,y)|_1^N}{||I_c(x,y)|| \cdot ||T||} < \eta_M. \tag{16}$$

The right-hand inequality of Property I guarantees the potential effectiveness of the initial sufficient condition [i.e., the left-hand term in (16) 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

$$\psi(x,y)|_\rho^\theta = \sum_{j=\rho}^{\theta} \sum_{i=1}^{M} I(x+i,y+j) \cdot T(i,j) \tag{17}$$

the next bounding function can be expressed as

$$\gamma^{r-1}(x,y) = \psi(x,y)|_1^n + \beta^{r-1}(x,y)|_{n+1}^N \tag{18}$$

$\beta^{r-1}(x,y)|_{n+1}^N$ representing a function defined as in (15) with the summation starting from $t = 2$ instead of $t = 1$, and the corresponding sufficient condition as

$$\frac{\gamma^{r-1}(x,y)}{||I_c(x,y)|| \cdot ||T||} < \eta_M. \tag{19}$$

Should (19) also 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

$$\gamma^1(x,y) = \psi(x,y)|_1^{(r-1)\cdot n} + \beta^1(x,y)|_{(r-1)\cdot n+1}^N \tag{20}$$

$$\frac{\gamma^1(x,y)}{||I_c(x,y)|| \cdot ||T||} < \eta_M. \tag{21}$$

Should the last condition not be verified, the process completes the computation of the actual cross-correlation value $\psi(x,y)$ that is used to check condition (8) 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. 3, with parameter *th* (always < 1) representing the initialization value for η_M.

As mentioned in Section I, if the initial condition (16) holds, EBC skips the current image point without calculating any partial correlation term. This is not the case for previous bounded correlation algorithms [15], [16], 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 VII (Table III), 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 (18) to (11) 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 (19) is tighter than that used in [16] and, consequently, in [15]. Moreover, if (7) is not satisfied, the algorithm in [16] carries on with

```
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 $\frac{\beta^r(x,y)\big|_1^N}{\|I_c(x,y)\| \cdot \|T\|} < \eta_M$
        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 $\eta_M$, $x_M$, $y_M$
```

Fig. 3. Pseudo-code describing the core EBC algorithm.

the computation of the remaining fraction of the correlation, while in case of failure of condition (19) 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 [15] in the framework of the initial BPC algorithm based on Jensen inequality. However, even applying the incremental method discussed in [15] to a BPC algorithm based on Cauchy–Schwarz inequality [16] 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 VII.

Finally, it can readily be inferred from (15) and (19) 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.

VI. Overall EBC Algorithm

The core EBC algorithm, as in the case of [9]–[12], [15], and [16], 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 subsampled 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 neighborhood of this candidate position. The subsampling factor $\bar{k}$ 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

$$\bar{k} = \arg\min_k \{ \frac{(W-M)\cdot(H-N)\cdot M\cdot N}{k^4} + (4\cdot k)^2 \cdot M\cdot N \} \quad (22)$$

where the first term of the function to be minimized represents the number of operations executed using the subsampled images, the second that carried out in a $4k \times 4k$ neighborhood 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 subsampling factor from (22) and the neighborhood 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 V, 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 offline 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 offline analysis of the algorithm behavior 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., $r = 2 \ldots r_{\max}$ and, given $r, n = 1 \ldots \lfloor (N-1)/(r-1) \rfloor$). 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 $(r, (n)/(N))$. For each problem instance

TABLE I
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

TABLE II
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)

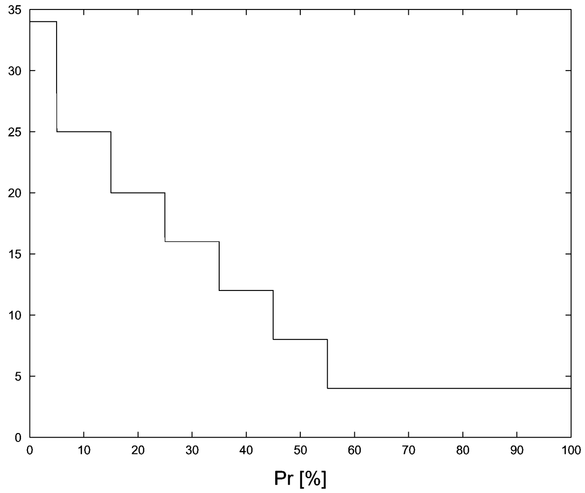

Fig. 4. Mapping function used to determine r from P_r.

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

Fig. 5. Pseudo-code for the overall EBC algorithm with online parameter estimation.

we recorded the pair $(r, (n)/(N))$ 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 seven pairs shown in Table I). 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 seven pairs).

Once the reference pairs in Table I had been determined offline, the task left to the run-time prediction step was to guess the most appropriate reference pair for 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 subsampled 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., (16)] is applied on the subsampled I, T using the reference parameter pair $r = 16, n/N = 0.06$ 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 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. 4). 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. 4 and, finally, n/N from Table I. This is denoted in the pseudo-code of Fig. 5 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 of running the core algorithm on the subsampled versions of I, T with parameters $(r = 16, n/N = 0.06)$, so that P_r can be calculated by applying the first sufficient condition. The pseudo-code for the overall EBC algorithm is shown in Fig. 5. We point out that P_FLAG allows switching between the two operating modes (with and without online parameter

TABLE III
MEASURED SPEED-UPS: EBC VERSUS 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	$\approx$ 100.0	63.1	(34,0.03)	47.6	$\approx$ 100.0	63.1
Paint 2	(24,0.04)	97.4	$\approx$ 100.0	95.2	(25,0.04)	91.7	$\approx$ 100.0	95.4
Paint 3	(14,0.06)	139.7	$\approx$ 100.0	98.9	(34,0.03)	90.3	$\approx$ 100.0	99.7
Pcb	(4,0.18)	47.0	$\approx$ 100.0	99.0	(4,0.18)	45.9	$\approx$ 100.0	99.0
Plants	(12,0.08)	80.6	$\approx$ 100.0	98.5	(8,0.13)	75.8	$\approx$ 100.0	98.1
Ringo 1	(34,0.03)	19.2	$\approx$ 100.0	70.3	(16,0.06)	14.7	$\approx$ 100.0	60.6
Ringo 2	(34,0.03)	25.6	$\approx$ 100.0	79.3	(16,0.06)	20.0	$\approx$ 100.0	72.0
Board 1	(34,0.03)	9.1	$\approx$ 100.0	29.3	(34,0.03)	9.0	$\approx$ 100.0	29.3
Board 2	(34,0.03)	10.5	$\approx$ 100.0	12.4	(34,0.03)	10.0	$\approx$ 100.0	12.4
Board 3	(16,0.05)	63.2	$\approx$ 100.0	97.4	(4,0.18)	41.0	$\approx$ 100.0	92.6
Wafer 1	(25,0.04)	25.5	$\approx$ 100.0	64.3	(20,0.05)	25.4	$\approx$ 100.0	65.6
Wafer 2	(19,0.05)	65.0	$\approx$ 100.0	93.1	(34,0.03)	59.4	$\approx$ 100.0	93.9
Wafer 3	(25,0.04)	15.1	$\approx$ 100.0	31.0	(25,0.04)	15.2	$\approx$ 100.0	31.0

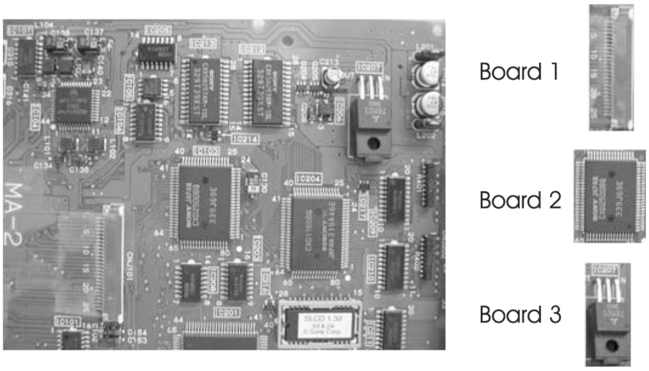

Fig. 6. *Board* image and templates (rows 8–10 of Tables II–IV).

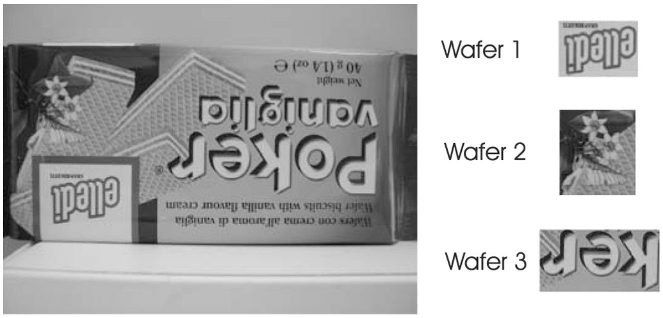

Fig. 7. *Wafer* image and templates (rows 11–13 of Tables II–IV).

prediction), and that the core EBC function, unlike that shown in Fig. 3, also has to calculate and return P_r, which is needed for the online parameter estimation stage.

VII. Experimental Results

This section compares the computational advantages of the overall EBC algorithm with respect to the other state-of-the-art exhaustive template matching algorithms, i.e., 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 [17] to compute the norms of all vectors and subvectors 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 II. 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. 6 and 7.[1]

Table III 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 online prediction stage and with parameters (r, n) chosen optimally $(\mathrm{EBC}_{\mathrm{opt}})$, that is determined by means of a thorough offline training session carried out on each instance of the dataset. Instead, columns 6–9 refer to the case where parameters have been estimated online by means of the prediction algorithm described in Section VI $(\mathrm{EBC}_{\mathrm{est}})$. In this case the time measurements for EBC include the online parameter prediction stage that, on average, requires 3.2 % of the overall execution time. The table reports in each part the parameter pair $(r, n/N)$ used by the algorithm (optimally selected in column three, estimated online in column seven), 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 (16)].

Table III 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

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

TABLE IV
MEASURED SPEED-UPS: EBC VERSUS BPC AND FFT-BASED ALGORITHMS

	EBC vs BPC [15]		EBC vs BPC [16]		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

dataset. Moreover, it also points out that the predictive approach to parameter selection described in Section VI rapidly finds a very good $(r, n/N)$ 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.[2]

By looking at both parts of the table, we can observe that EBC always runs almost one order of magnitude faster than the FS.

Comparing EBC with BPC [15], [16], 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., n/N), which places a theoretical upper bound on the maximum attainable speed-up. For example, taking $n/N = 0.3$, as proposed in [16], 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 [15], where $n/N = 0.2$ 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 (16) holds, up to the whole cross-correlation, when even condition (21) is not satisfied]. Hence, EBC 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 [16].

These theoretical considerations were experimentally confirmed by the results shown in columns 1–4 of Table IV, reporting the measured speed-ups of EBC with regard to BPC algorithms [15], [16], with the optimal choice of parameter pairs $(\mathrm{EBC_{opt}})$ and with the online estimation of parameters $(\mathrm{EBC_{est}})$. 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 [15] (i.e., $Cr1 = 0.2, Cr2 = 0.4$) and [16] (i.e., $\mathrm{Cr} = 0.3$). 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) [19]. 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 (MMX) available on state-of-the-art processors based on Intel Architecture. Columns 6 and 7 of Table IV 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 online parameter estimation, that now requires on average 3.4 % of EBC 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).

A. Experiments With Small Templates and Artificial Noise

This subsection provides further experimental results aimed at assessing the performance of EBC in case of smaller tem-

[2] The slight advantage given by $\mathrm{EBC_{est}}$ with regards to $\mathrm{EBC_{opt}}$ in the *Wafer 3* instance of Table III, 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, $\mathrm{EBC_{opt}}(P\_\mathrm{FLAG}$ switched off) uses a default parameter pair, while $\mathrm{EBC_{est}}(P\_\mathrm{FLAG}$ switched on) can already exploit the predicted parameter pair.

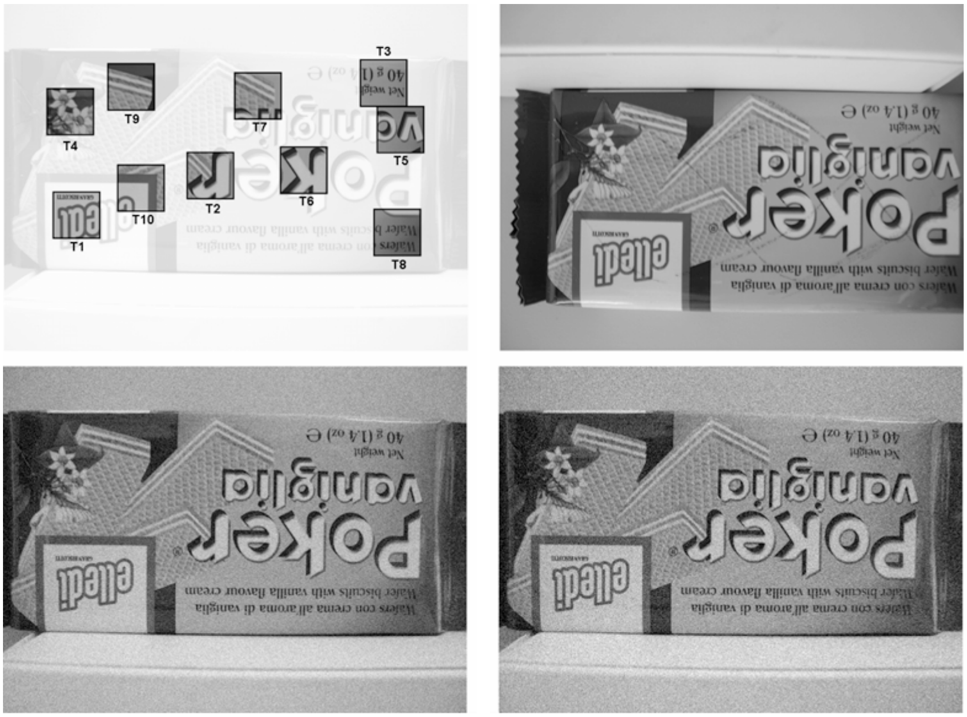

Fig. 8. Images and templates used for the experiments with small templates and artificial noise.

TABLE V
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

plate sizes and increasing artificial noise. To this purpose, ten templates of size 64 × 64 were hand-selected uniformely from an image belonging to our dataset (Fig. 8, 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 $\sigma_2 = 0.005$,[3] Fig. 8 bottom left and bottom right, respectively), and into another image taken with the same camera from a slightly different position (Fig. 8, top right). The speed-ups yielded by the overall EBC algorithm with online parameter estimation $(\mathrm{EBC}_{\mathrm{est}})$ with respect to the FS and FFT algorithms are shown in Table V. To investigate on the impact of a smaller template size, the results in the two rightmost columns of Table V can be compared directly to those reported in the last eight rows (column 7) of Tables III and IV, 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 and 3 with 6 and 7 of Table V 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 and 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 online prediction stage requires on average 2.7% of EBC overall execution time, which increases to 3.1% in the case of the EBC implementation deploying MMX-optimization.

[3] With respect to normalized pixel intensities ranging within [0, 1].

VIII. Conclusion

We have described the novel EBC algorithm for efficient and exhaustive template matching based on the NCC function, having its mathematical grounds in two properties derived from the Cauchy–Schwarz inequality. Unlike previous bounded partial correlation (BPC) algorithms, EBC can initially check a very effective "zero-correlation bound," a skipping condition based on a bounding function that does not require the computation of a partial correlation term. Moreover, the novel bounding functions devised for the EBC algorithm are much tighter and, hence, more effective than those used with previous BPC approaches. It is also peculiar for EBC to deploy the Cauchy–Schwarz inequality within a bounding strategy aimed at building a succession of functions characterized by an increasingly tighter approximation of the NCC.

Since EBC is a data-dependent algorithm, we have discussed how one major cause of data-dependency can be dealt with by an initial coarse-to-fine search aimed at finding a good initialization value for the best correlation score. As to parameter dependency, we have developed an online parameter prediction algorithm that supports an efficient parameter-free EBC formulation. The experimental results obtained on a dataset including 43 instances of template matching problems prove that EBC can provide substantial speed-ups—up to two orders of magnitude—with respect to the other exhaustive approaches such as the FS, FFT-based, and BPC algorithms. The experimental results also show that the computational benefits brought by EBC do not depend in general on image dimensions, since the EBC algorithm was faster than the other methods considered on a dataset including images and templates having very different sizes.

Appendix A
Proof of Lemma

Proof: The lemma can be proved by contradiction. Let us assume that

$$\sqrt{\sum_{k\in S_1} a_k^2}\cdot\sqrt{\sum_{k\in S_1} b_k^2}+\sqrt{\sum_{k\in S_2} a_k^2}\cdot\sqrt{\sum_{k\in S_2} b_k^2} > \sqrt{\sum_{k\in S} a_k^2}\cdot\sqrt{\sum_{k\in S} b_k^2}.$$

Since

$$\sqrt{\sum_{k\in S} a_k^2}\cdot\sqrt{\sum_{k\in S} b_k^2} = \sqrt{\left(\sum_{k\in S_1} a_k^2+\sum_{k\in S_2} a_k^2\right)\cdot\left(\sum_{k\in S_1} b_k^2+\sum_{k\in S_2} b_k^2\right)}$$

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:

$$\begin{aligned} 0 > &\sum_{k\in S_1} a_k^2\cdot\sum_{k\in S_2} b_k^2+\sum_{k\in S_2} a_k^2\cdot\sum_{k\in S_1} b_k^2 \\ &-2\cdot\sqrt{\sum_{k\in S_1} a_k^2}\cdot\sqrt{\sum_{k\in S_1} b_k^2}\cdot\sqrt{\sum_{k\in S_2} a_k^2}\cdot\sqrt{\sum_{k\in S_2} b_k^2} = \\ &=\left(\sqrt{\sum_{k\in S_1} a_k^2}\cdot\sqrt{\sum_{k\in S_2} b_k^2}-\sqrt{\sum_{k\in S_2} a_k^2}\cdot\sqrt{\sum_{k\in S_1} b_k^2}\right)^2.\end{aligned}$$

References

[1] A. Goshtasby, *2-D and 3-D Image Registration for Medical, Remote Sensing and Industrial Applications*. New York: Wiley, 2005.

[2] B. Zitova and J. Flusser, "Image registration methods: A survey," *Image Vis. Comput.*, vol. 21, no. 11, pp. 977–1000, 2003.

[3] W. Krattenthaler, K. Mayer, and M. Zeiler, "Point correlation: A reduced-cost template matching technique," in *Proc. 1st IEEE Int. Conf. Image Processing*, Austin, TX, 1994, vol. 1, pp. 208–212.

[4] K. Briechle and U. D. Hanebeck, "Template matching using fast normalized cross correlation," in *Proc. SPIE AeroSense Symp.*, Orlando, FL, 2001, vol. 4387.

[5] P. S. Heckbert, "Filtering by repeated integration," in *Proc. SIG-GRAPH*, 1986, pp. 315–321.

[6] H. Schweitzer, J. W. Bell, and F. Wu, "Very fast template matching," in *Proc. Eur. Conf. Computer Vision*, 2002, pp. 358–372.

[7] P. Simard, L. Bottou, P. Haffner, and Y. Le Cun, M. Kearns, S. Solla, and D. Cohn, Eds., "Boxlets: A fast convolution algorithm for signal processing and neural networks," *Adv. Neural Inf. Process. Syst.*, vol. 11, pp. 571–577, 1999.

[8] C. D. Bei and R. M. Gray, "An improvement of the minimum distortion encoding algorithm for vector quantization," *IEEE Trans. Commun.*, vol. COM-33, no. 10, pp. 1132–1133, Oct. 1985.

[9] W. Li and E. Salari, "Successive elimination algorithm for motion estimation," *IEEE Trans. Image Process.*, vol. 4, no. 1, pp. 105–107, Jan. 1995.

[10] H. S. Wang and R. M. Mersereau, "Fast algorithms for the estimation of motion vectors," *IEEE Trans. Image Process.*, vol. 8, no. 3, pp. 435–439, Mar. 1999.

[11] Y. Hel-Or and H. Hel-Or, "Real-time pattern matching using projection kernels," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 9, pp. 1430–1445, Sep. 2005.

[12] M. G. Alkhansari, "A fast globally optimal algorithm for template matching using low-resolution pruning," *IEEE Trans. Image Process.*, vol. 10, no. 4, pp. 526–533, Apr. 2001.

[13] J. P. Lewis, "Fast template matching," *Vis. Interface*, pp. 120–123, 1995.

[14] A. Kadyrov and M. Petrou, "The 'invaders' algorithm: Range of values modulation for accelerated correlation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 11, pp. 1882–1886, Nov. 2006.

[15] L. Di Stefano and S. Mattoccia, "Fast template matching using bounded partial correlation," *Mach. Vis. Appl.*, vol. 13, no. 4, pp. 213–221, 2003.

[16] L. Di Stefano and S. Mattoccia, "A sufficient condition based on the Cauchy–Schwarz inequality for efficient template matching," in *Proc. IEEE Int. Conf. Image Processing*, Barcelona, Spain, Sep. 2003, vol. 1, pp. 269–272.

[17] M. J. McDonnell, "Box-filtering techniques," *Comput. Graph. Image Process.*, vol. 17, pp. 65–70, 1981.

[18] F. Crow, "Summed-area tables for texture mapping," *Comput. Graph.*, vol. 18, no. 3, pp. 207–212, 1984.

[19] T. Toivonen, J. Heikkila, and O. Silven, "A new algorithm for fast full search block motion estimation based on number theoretic transforms," in *Proc. 9th Int. Workshop Systems, Signals and Image Processing*, Manchester, U.K., Nov. 2002, pp. 90–94.

Stefano Mattoccia (M'05) received the M.S. degree in electronic engineering and the Ph.D. degree in computer science engineering from the University of Bologna, Bologna, Italy, in 1997 and 2002, respectively.

He joined the Dipartimento di Elettronica Informatica e Sistemistica (DEIS) and the Advanced Research Center on Electronic Systems for Information and Communication Technologies (ARCES), University of Bologna, and is currently a Research Associate at the Faculty of Engineering, University of Bologna. His research interests include computer vision, image processing, and computer architectures. He has authored more than 30 refereed papers and two patents.

Dr. Mattoccia is a member of the Gruppo Italiano Ricercatori in Pattern Recognition (IAPR, Italy).

Federico Tombari (S'06) received the B.Eng. and M.Eng. degrees from the University of Bologna, Bologna, Italy, in 2003 and 2005, respectively, where he is currently pursuing the Ph.D. degree.

In 2006, he joined the Dipartimento di Elettronica, Informatica e Sistemistica (DEIS), University of Bologna, and the Advanced Research Center on Electronic Systems (ARCES), University of Bologna, as a Collaborator. His research interests concern computer vision and pattern recognition.

Mr. Tombari is a member of IAPR-IC.

Luigi Di Stefano graduated as an Electronic Engineer and received the Ph.D. degree in electronic engineering and computer science from the University of Bologna, Bologna, Italy, in 1989 and 1994, respectively.

In 1990, he joined the Department of Electronics, Computer Science and Systems (DEIS), University of Bologna. In 1995, he spent six months at Trinity College, Dublin, Ireland, as a Postdoctoral Fellow. He is currently an Associate Professor at DEIS. He also joined the Advanced Research Centre on Electronic Systems (ARCES), "Ercole De Castro", a research center instituted in 2001 at the University of Bologna. His research interests include computer vision, image processing, and computer architecture. He is the author of more than 70 papers and five patents.

Dr. Di Stefano is a member of the IAPR-IC.