Maschine Vision and Applications (2003) 13: 213–221

Machine Vision and
Applications

Fast template matching using bounded partial correlation

Luigi Di Stefano, Stefano Mattoccia

DEIS-ARCES, University of Bologna, Viale Risorgimento 2, 40136 Bologna, Italy (e-mail: {ldistefano,smattoccia}@deis.unibo.it)

Received: 2 November 2000 / Accepted: 25 July 2001

Abstract. This paper describes a novel, fast template-matching technique, referred to as bounded partial correlation (BPC), based on the normalised cross-correlation (NCC) function. The technique consists in checking at each search position a suitable elimination condition relying on the evaluation of an upper-bound for the NCC function. The check allows for rapidly skipping the positions that cannot provide a better degree of match with respect to the current best-matching one. The upper-bounding function incorporates partial information from the actual cross-correlation function and can be calculated very efficiently using a recursive scheme. We show also a simple improvement to the basic BPC formulation that provides additional computational benefits and renders the technique more robust with respect to the parameters choice.

Key words: Template matching– Pattern matching – Normalised cross-correlation – SEA– PDE

1 Introduction

Matching a template sub-image into a given image, generally within a relatively smaller search area, is one fundamental task occurring in countless image analysis applications. The basic template-matching algorithm consists in sliding the template over the search area and, at each position, calculating a *distorsion*, or *correlation*, measure estimating the degree of dissimilarity, or similarity, between the template and the image. Then, the minimum distorsion, or maximum correlation, position is taken to represent the instance of the template into the image under examination, with a threshold on the similarity/dissimilarity measure, regulated according to the sensitivity requirements of the application, allowing for rejection of poor matches. The typical distorsion measures used in template matching algorithms are the sum of absolute differences (SAD) and the sum of squared differences (SSD), while normalised cross-correlation (NCC) is by far the most widely used correlation measure.

Correspondence to: L. Di Stefano

Since with large-size images and templates the matching process can be computationally very expensive, numerous techniques aimed at speeding up the basic approach have been devised (see [4] for a concise review). Among general techniques (i.e. applicable with both distorsion and correlation measures), the major ones are (a) the use of multi-resolution schemes [5] (i.e. locating a coarse-resolution template into the coarse-resolution image and then refining the search at the higher resolution levels), (b) sub-sampling the image and the template, (c) two-stage matching [7] (i.e. matching a sub-template first, and then the whole template only at good candidate positions). However, techniques (a), (b) and (c) imply a non-exhaustive search process since they do not compare the full-resolution image with the full-resolution template at every search position. As a result, there is no guarantee that the algorithm finds the global distorsion minimum (or correlation maximum).

On the other hand, in the specific case of distorsion measures, two interesting techniques, called successive elimination algorithm (SEA) [2], [8] and partial distorsion elimination (PDE) [1], allow for notably speeding up the computation required by an exhaustive-search template-matching process. Besides, both can be regarded as core techniques, to be usefully integrated into the previously mentioned non-exhaustive methods. SEA relies on fast evaluation of a lower-bound for the distorsion measure: if the bounding function exceeds the current minimum, the position can be skipped without calculating the actual distorsion. PDE consists of terminating the evaluation of the distorsion measure if it exceeds the current minimum.

In this paper, we show how to extend the SEA and PDE approaches to the case of correlation. This is done by first determining an upper-bound for the NCC function and then incorporating the evaluation of this bound into a partial correlation scheme.

2 SEA and PDE for distorsion measures

The SEA by Li and Salari [2] is based on deploying a suitable lower-bound for the SAD function. Wang and Mersereau [8] have successively extended SEA so as to deal also with SSD. For brevity we describe here only the SAD formulation.

Given an image I of size $W \cdot H$ and a template T of size $M \cdot N$, the SAD at position x, y is defined as:

$$SAD\,(x,y) = \sum_{j=0}^{N-1} \sum_{i=0}^{M-1} |I\,(x+i, y+j) - T\,(i,j)| \quad (1)$$

The computation of $SAD\,(x,y)$ requires a number of operations proportional to the template area (i.e. $M \cdot N$).

Starting from (1) and by means of simple manipulations, Li and Salari derive the following relation:

$$\alpha\,(x,y) \le SAD\,(x,y) \quad (2)$$

with

$$\alpha\,(x,y) = \left| \sum_{j=0}^{N-1} \sum_{i=0}^{M-1} I\,(x+i, y+j) - \sum_{j=0}^{N-1} \sum_{i=0}^{M-1} T\,(i,j) \right|$$

The above relation states that function α is a lower-bound for the SAD distorsion measure at every position x, y. The two sums appearing in $\alpha\,(x,y)$ represent the L_1 norms of the sub-image under examination at x, y (also indicated from now on as $|I(x,y)|_1$) and of the template ($|T|_1$). The latter does not vary during the matching process, and therefore can be precomputed at initialisation time. As for $|I(x,y)|_1$, this can be calculated very efficiently [2] using a standard recursive technique, also known as box-filtering [3], which makes the calculation independent of the template area and requires only four elementary operations (i.e. accumulation of an absolute difference term) per image position.

Hence, SEA consists in calculating $|I(x,y)|_1$ at every position via the box-filtering technique and then comparing $\alpha\,(x,y)$ against the current SAD minimum:

$$\alpha\,(x,y) \ge SAD_{\min} \quad (3)$$

if (3) is verified, then the matching process can proceed with the next position without carrying out the calculation of $SAD\,(x,y)$; this is computed and compared against $SAD_{\min}$ only if the above relation is not satisfied.

Thanks to the search positions eliminated by checking (3), SEA reduces the average number of calculations required to match a template into an image. In addition, the technique does not imply any form of approximate search since the eliminated positions are guaranteed not to correspond to the global minimum.

It is worth noticing that SEA's ability to eliminate useless search positions depends on the value of the current minimum. Indeed, the algorithm owes its name to the behaviour of the elimination process, which improves its effectiveness as long as the degree of similarity between the template and current best matching sub-image increases (i.e. $SAD_{\min}$ gets smaller). As a result, SEA's performance is significantly data-dependent, with the technique turning out more effective when the matching process rapidly finds a good matching position (though not necessarily the best one). Nonetheless, the sensitivity threshold, which is a standard feature of template matching algorithms, can be used to improve SEA's performance by comparing the bounding function against the lower of the threshold or the current minimum. Thus, as long as the matching process finds sub-images relatively different from the template, the threshold itself acts as the current minimum-distorsion value. Consequently, a more selective matching process will generally run faster with SEA.

A partial distorsion is defined as a part of the entire distorsion. For example, considering (1), one may define $P_{SAD}\,(x,y,m,n)$ as the partial distorsion accumulated up to template position m, n:

$$P_{SAD}\,(x,y,m,n) = \sum_{j=0}^{n} \sum_{i=0}^{m} |I\,(x+i, y+j) - T\,(i,j)| \quad (4)$$

Since a distorsion measure increases as long as new points are examined, it is clear that if for some (m,n)

$$P_{SAD}\,(x,y,m,n) \ge SAD_{\min} \quad (5)$$

then point (x,y), can be eliminated from the search without proceeding with the calculation of the entire distorsion measure. Similar to SEA, PDE does not introduce any approximation in the search and is a data-dependent optimisation technique that is more effective when the matching process rapidly finds low-distorsion points. With PDE, one can also easily make use of the sensitivity threshold to speed up the computation. PDE has also been proposed in conjunction with SEA [8]: if relation (3) is not satisfied, the SAD calculation is carried out by means of a PDE scheme. One critical parameter associated with PDE is the granularity with which condition (5) is checked. Ideally, performing the check as soon as a new elementary contribution is added to the current distorsion would maximise the amount of skipped calculations. On the other hand, since branch instructions add a significant computational overhead, performing too many unsuccessful checks can outweigh the savings associated with skipping distorsion calculations. Hence, in a practical PDE scheme the test's granularity should be kept relatively coarse; for instance, the scheme proposed in [8] relies on checking (5) after each row.

Eventually, it is worth pointing out that both SEA and PDE can be applied to speed up the search when the template can appear at multiple positions of the image. In such a framework, the basic template-matching algorithm consists in scanning the image and taking all the image positions where the distorsion measure is below a given sensitivity threshold. Hence, deploying SEA or PDE requires just using the sensitivity threshold in place of $SAD_{\min}$ in (3) or (5). In the case of multiple instances of the template, the choice of the sensitivity threshold implies a trade-off between the capability to tolerate small differences in the appearances of the template and the need to avoid mismatches. Consequently, and akin to the single-instance case, a selective process will generally run faster with SEA and PDE.

3 An upper-bound for the NCC function

When using normalised cross-correlation, the template sub-image is located into the image under examination by searching for the maximum of the NCC function:

$$NCC\,(x,y) = \frac{\sum_{j=0}^{N-1}\sum_{i=0}^{M-1} I\,(x+i,y+j)\cdot T\,(i,j)}{\sqrt{\sum_{j=0}^{N-1}\sum_{i=0}^{M-1} I\,(x+i,y+j)^2}\cdot\sqrt{\sum_{j=0}^{N-1}\sum_{i=0}^{M-1} T\,(i,j)^2}} \quad (6)$$

In this section, similarly to the SEA approach, we try to determine a suitable function that upper-bounds the NCC and that can be computed at a significantly lower computational cost.

The numerator of (6) represents the *correlation* (or *cross-correlation*) between the template and the image, $C\,(x,y)$, and its computation turns out to be the bottleneck in the evaluation of the NCC. In fact, the two terms appearing in the denominator represent the L_2 norms of the subimage under examination $|I(x,y)|_2$ and of the template $|T|_2$. The latter can be precomputed at initialisation time, the former can be obtained very efficiently via a box-filtering scheme at a cost of only four elementary operations per image point (the elementary operation being now the accumulation of a product term).

Suppose now that a function $\beta(x,y)$ exists such that $\beta(x,y)$ is an upper-bound for $C(x,y)$, then

$$\beta(x,y) \geq C(x,y) = \sum_{j=0}^{N-1}\sum_{i=0}^{M-1} I\,(x+i,y+j)\cdot T\,(i,j) \quad (7)$$

By normalising β, we obtain an upper-bound for the NCC function as follows:

$$\frac{\beta(x,y)}{|I(x,y)|_2\cdot|T|_2} \geq \frac{C\,(x,y)}{|I(x,y)|_2\cdot|T|_2} = NCC(x,y) \quad (8)$$

Then, indicating as $NCC_{\max}$ the current correlation maximum, if at point (x,y) the elimination condition

$$\frac{\beta(x,y)}{|I(x,y)|_2\cdot|T|_2} < NCC_{\max} \quad (9)$$

is verified, the matching process can proceed with the next point without carrying out the calculation of $C(x,y)$, for the point is guaranteed not to correspond to the new correlation maximum. Conversely, if (9) does not hold, it is necessary to compute $C(x,y)$, normalise it by the product $|I(x,y)|_2\cdot|T|_2$ and check the new maximum condition:

$$\frac{C(x,y)}{|I(x,y)|_2\cdot|T|_2} \geq NCC_{\max} \quad (10)$$

In order to find $\beta(x,y)$, we consider the relationship between the geometric and arithmetic series known as Jensen's inequality [6]:

$$\frac{x_1+x_2+\ldots+x_n}{n} \geq \sqrt[n]{x_1\cdot x_2\cdot\ldots\cdot x_n} \quad (11)$$

Applying (11) to the generic pair of homologous terms appearing in the cross-correlation product yields

$$\frac{I(x+i,y+j)+T(i,j)}{2} \geq \sqrt[2]{I(x+i,y+j)\cdot T(i,j)} \quad (12)$$

from which we can derive by simple manipulations

$$\frac{I(x+i,y+j)^2+T(i,j)^2}{2} \geq I(x+i,y+j)\cdot T(i,j) \quad (13)$$

Therefore, if we pose

$$\beta(x,y) = \frac{\sum_{j=0}^{N-1}\sum_{i=0}^{M-1} I\,(x+i,y+j)^2 + \sum_{j=0}^{N-1}\sum_{i=0}^{M-1} T\,(i,j)^2}{2} \quad (14)$$

from inequality (13) it follows that $\beta(x,y)$ satisfies (7) at each image point.

4 Bounded partial correlation

The elimination condition relies on calculating $\beta(x,y)$ instead of $C(x,y)$. As pointed out, this holds potential for speeding up the matching process as long as β can be computed much more rapidly than C. The function defined in (14) requires evaluation of the squares of the L_2 norms of the sub-image under examination, $\left(|I(x,y)|_2\right)^2$, and of the template,$\left(|T|_2\right)^2$. The latter quantity can be precomputed and the former calculated recursively by running a box-filter. Moreover, one may also notice that evaluating β as defined in (14) does not require any extra computation since the quantity $\left(|I(x,y)|_2\right)^2$ had to be calculated for the purpose of normalising the correlation in the new maximum condition (i.e. (10)).

Yet, for the above described approach to yield computational advantages, the elimination condition should be also effective. Unfortunately, this is not the case for condition (9). In fact, it is straightforward to verify that the quantity

$$\frac{\beta(x,y)}{|I(x,y)|_2\cdot|T|_2} = \frac{\sum_{j=0}^{N-1}\sum_{i=0}^{M-1} I(x+i,y+j)^2+\sum_{j=0}^{N-1}\sum_{i=0}^{M-1} T(i,j)^2}{2\cdot\sqrt{\sum_{j=0}^{N-1}\sum_{i=0}^{M-1} I(x+i,y+j)^2}\cdot\sqrt{\sum_{j=0}^{N-1}\sum_{i=0}^{M-1} T(i,j)^2}} \quad (15)$$

is lower-bounded by one, which in turn is the upper-bound of the NCC function. Hence, the elimination condition defined via the equations (9) and (14) is never satisfied.

Nonetheless, we carried out a set of experiments on real images aimed at measuring and comparing β and C values. These experiments showed that β, although obviously too large to satisfy the elimination condition, turns out to be often relatively close to the actual correlation. This led us to the idea that an effective elimination condition could be attained by computing only a small portion of the actual correlation function, referred to as *partial correlation*, and bounding the residual portion with a term that, like β, is derived from (13).

We define the partial correlation associated with indexes m, n as

$$P_C\,(x,y,m,n) = \sum_{j=0}^{n}\sum_{i=0}^{m} I\,(x+i,y+j)\cdot T\,(i,j), \quad (16)$$

a new upper-bound for the correlation $C(x,y)$ as

$$\gamma(x,y,m,n) = P_C\,(x,y,m,n) + \tilde{\beta}\,(x,y,m,n)\,, \quad (17)$$

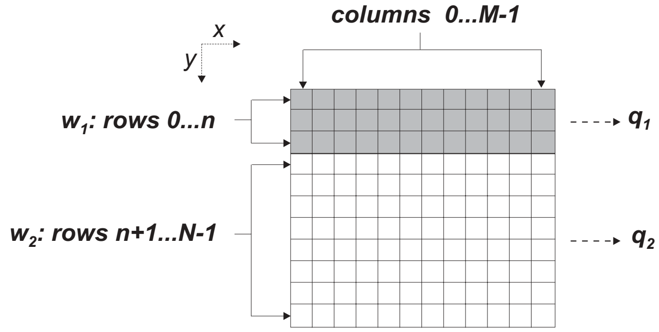

Fig. 1. Row-based partitioning of the window under examination

with

$$\tilde{\beta}(x,y,m,n) = \frac{\sum_{j=0}^{n}\sum_{i=m+1}^{M-1} I(x+i,y+j)^2+\sum_{j=0}^{n}\sum_{i=m+1}^{M-1} T(i,j)^2}{2} + \frac{\sum_{j=n+1}^{N-1}\sum_{i=0}^{M-1} I(x+i,y+j)^2+\sum_{j=n+1}^{N-1}\sum_{i=0}^{M-1} T(i,j)^2}{2}, \quad (18)$$

and a new elimination condition as

$$\frac{\gamma(x,y,m,n)}{|I(x,y)|_2 \cdot |T|_2} < NCC_{\max}. \quad (19)$$

Hence, γ combines the evaluation of a given part of the total correlation P_C with the approximation of the residual part by means of an upper-bounding term $\tilde{\beta}$ derived from (13).

For the described bound-based approach to prove useful, it is mandatory to compute γ very efficiently. This can be achieved by partitioning the window associated with the sub-image under examination into two sub-windows consisting of consecutive rows, indicated as w_1 and w_2 in Fig. 1, and associating the partial correlation term with w_1 and the upper-bounding term with w_2:

$$P_C(x,y,M-1,n) = \sum_{j=0}^{n}\sum_{i=0}^{M-1} I(x+i,y+j)\cdot T(i,j), \quad (20)$$

$$\tilde{\beta}(x,y,M-1,n) = \frac{\sum_{j=n+1}^{N-1}\sum_{i=0}^{M-1} I(x+i,y+j)^2+\sum_{j=n+1}^{N-1}\sum_{i=0}^{M-1} T(i,j)^2}{2} \quad (21)$$

In this manner it is possible to run a distinct box-filter on each sub-window, with the filter run on w_1 calculating the quantity

$$q_1(x,y,M-1,n) = \sum_{j=0}^{n}\sum_{i=0}^{M-1} I(x+i,y+j)^2 \quad (22)$$

and that run on w_2 the quantity

$$q_2(x,y,M-1,n) = \sum_{j=n+1}^{N-1}\sum_{i=0}^{M-1} I(x+i,y+j)^2. \quad (23)$$

Then, q_2 can be used to compute γ while q_1 and q_2 can be added together to obtain $\left(|I(x,y)|_2\right)^2$, which is needed to normalise γ and C in the elimination and new maximum conditions. It is worth pointing out that, due to the partitioning, the computational cost associated with box-filtering amounts now to eight elementary operations per point. On the other hand, the evaluation of the partial correlation term present in γ does not introduce any overhead with respect to the standard algorithm since when (19) fails $C(x,y)$ can be attained adding the residual product terms to $P_C(x,y,M-1,n)$. Eventually, the last quantity needed to obtain γ, i.e. $\sum_{j=n+1}^{N-1}\sum_{i=0}^{M-1} T(i,j)^2$, can be precomputed at initialisation time.

We can define a correlation ratio (C_r) as the ratio between the number of points involved in the calculation of the partial correlation and those needed to evaluate the entire correlation function such that

$$C_r = \frac{(n+1)}{N}. \quad (24)$$

It is clear from the definition of γ that, as C_r increases, the bounding function gets closer to the actual correlation and the elimination condition becomes more effective. However, the computational savings associated with skipping points satisfying the elimination condition amounts basically to the residual fraction of the correlation, which decreases with C_r. Hence, C_r turns out to be the fundamental parameter of the proposed technique, its choice implying a trade-off between two opposite requirements: it should be kept large to effectively eliminate non-maximum points and small to gain a substantial computational benefit at eliminated points. As we will show in Sect. 6, the choice of a reasonable C_r value can yield a significant speedup with respect to the standard template-matching algorithm.

The technique outlined in this section, bounded partial correlation (BPC), behaves, within the framework of correlation-based matching, analogously to SEA and PDE. In fact, it does not imply any approximation in the search and it is a data-dependent optimisation technique: once C_r has been chosen, the elimination condition becomes more effective as the correlation between the template and the current best matching sub-image improves (i.e. $NCC_{\max}$ gets higher). Hence, BPC yields better speed-up if, given the scan order, the search process find rapidly a good matching position. In addition, the sensitivity (i.e. correlation) threshold can be exploited straightforwardly to speed up the matching process by comparing the left-hand side of (19) with the higher of the current maximum and the threshold. BPC can be deployed also when the template can appear at multiple positions of the image by simply using the correlation threshold in place of $NCC_{\max}$ in (19). In both the single-instance and multiple-instance cases, BPC will generally run faster with a high correlation threshold (i.e. selective matching process).

Nonetheless, BPC notably differs from SEA and PDE. In fact, SEA uses only a bounding function and does not include evaluation of partial distorsions. PDE consists in evaluating partial distorsions without making use of any bounding function. Instead, BPC combines the use of a bounding function with the evaluation of partial information from the actual matching measure: the bounding function incorporates partial information from the actual matching measure. It is worth noticing that this approach is also different from the mixed SEA-PDE algorithm proposed in [8], which consist in first using a bound and then, if the bound-based elimination condition fails, evaluating partial distorsions.

5 Performance evaluation metric

The proposed method belongs to the class of exhaustive search algorithms. Therefore, in order to evaluate its performance gain, we compare BPC to the standard NCC-based template-matching algorithm. We will use as performance evaluation metric, referred to as A_{ops}, the ratio between the average number of elementary operations (i.e. accumulation of one product term) per point executed by BPC and that required by the standard algorithm, which amounts to $(M \cdot N + 4)$. For example $A_{\mathrm{ops}} = 0.6$ means in the considered image BPC executes an average number of operation per point that is 60% of that required by the standard algorithm.

The number of operations executed by BPC at a given point depends on the outcome of (19). If this is satisfied, the algorithm executes only the $M \cdot (n+1)$ operations needed to evaluate the partial correlation term $P_C(x, y, M-1, n)$. If the condition fails, the partial correlation term must be integrated with its residual part, giving a total number of operations equal to $M \cdot N$. In both cases, we apply the double box-filtering scheme, which requires eight additional operations per point.

Hence, if S is the number of points for which the elimination condition is satisfied, we can express the total number of elementary operations executed by BPC as

$$T_{\mathrm{ops}} = (W \cdot H - S) \cdot M \cdot N + S \cdot M \cdot (n+1) + 8 \cdot W \cdot H \quad (25)$$

Then, we define an elimination ratio (E_r) as the fraction of points at which the elimination condition is successful:

$$E_r = \frac{S}{W \cdot H} \quad (26)$$

This allows us to express A_{ops} as

$$A_{\mathrm{ops}} = \frac{[(1 - E_r) + E_r \cdot C_r] \cdot M \cdot N + 8}{M \cdot N + 4} \quad (27)$$

6 Experimental results

BPC has been tested with several images and templates. We present here the results obtained with images *pcb1*, *pcb2*, *albert*, *pcb3*, *plants* and *parking meter* (abbreviated as *pm*), shown in Figs 2, 3, 4, 5, 6 and 7. In every test the sub-image used as template, also shown in the figures, has been extracted from a different – though similar – image.

As already discussed, and similar to SEA and PDE, BPC allows attaining computational savings from the use of the sensitivity threshold and yields an increase in speedup when rapidly finding a good matching position (though not necessarily the best matching one) is highly probable. Therefore, we address two situations:

1. No information about the expected position of the template is available. Hence, we start the matching process at the upper left corner of the image and then proceed with the usual raster-scan order.
2. It is possible to predict the expected position of the template, which is actually located reasonably close to this position, i.e. within a 50×50 pixel window centred at the

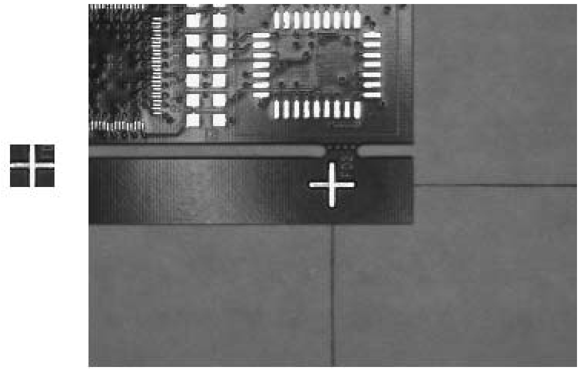

Fig. 2. *pcb1*: $W \cdot H = 320 \cdot 240$, $M \cdot N = 29 \cdot 28$, $\max(NCC) = NCC(145, 106) \simeq 0.985$

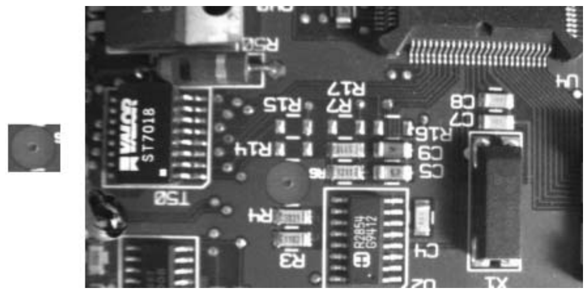

Fig. 3. *pcb2*: $W \cdot H = 380 \cdot 227$, $M \cdot N = 40 \cdot 37$, $\max(NCC) = NCC(120, 136) \simeq 0.995$

Fig. 4. *albert*: $W \cdot H = 320 \cdot 240$, $M \cdot N = 51 \cdot 58$, $\max(NCC) = NCC(198, 43) \simeq 0.995$.

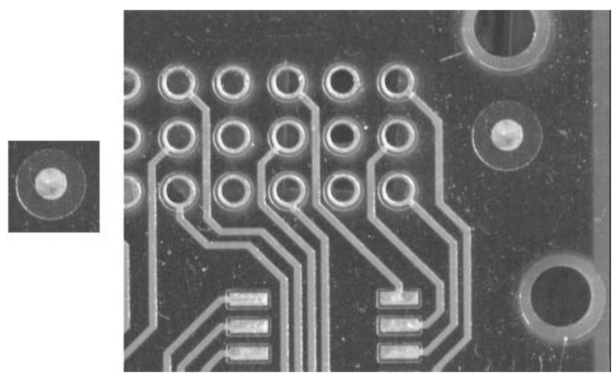

Fig. 5. *pcb3*: $W \cdot H = 384 \cdot 288$, $M \cdot N = 72 \cdot 73$, $\max(NCC) = NCC(268, 65) \simeq 0.997$

Fig. 6. *plants*: $W \cdot H = 512 \cdot 400, M \cdot N = 104 \cdot 121, \max(NCC) = NCC(333, 66) \simeq 0.986$

Fig. 7. *parking meter (pm)*: $W \cdot H = 512 \cdot 480$, $M \cdot N = 103 \cdot 137$, $\max(NCC) = NCC(150, 161) \simeq 0.989$

expected position. In this case, we first scan the 50×50 pixel window centred at the expected position and then the whole image. This kind of situation is typically found in those image registration applications in which the position of the fiducial point varies slowly between successive images (and therefore one can start the matching process in a neighbourhood of the previously found best matching position) and in tracking applications in which one can usually predict the position of the shape being tracked over successive frames.

In both situations, we provide the results obtained with correlation threshold equal to 0.95, 0.97 and 0.98.

Table 1 refers to image *pcb1* and situation 1, and reports the values found for the elimination ratio E_r and the average number of operations A_{ops} with different choices of the correlation ratio, i.e. varying C_r by a step of roughly 10%.

The table shows clearly that if C_r is chosen too low (i.e. less than roughly 10%), E_r becomes too low for BPC to provide any notable saving. However, by computing the correlation for just three more rows, i.e. augmenting C_r to approximately 20%, E_r increases dramatically and BPC yields a significant reduction of the operations. With all the three correlation thresholds the optimum C_r value lies between $\approx 20\%$ and $\approx 50\%$ and allows for reducing A_{ops} to well below 50%. For $C_r >\approx 50\%$, increasing the correlation ratio causes only a very modest increase in the elimination ratio; as a result, A_{ops} grows with C_r since the cost associated with computing additional correlation products outweighs the savings associated with skipping more points. Eventually, for each given C_r, the comparison between the results corresponding to the different thresholds shows clearly that a higher correlation threshold implies an intrinsically more effective elimination condition and thus better BPC performance.

Table 1. Results for *pcb1*, situation 1

$\mathbf{C_r}, \mathbf{C_r}[\%]$	$\mathbf{E_r}[\%]$	$\mathbf{A_{ops}}[\%]$	$\mathbf{E_r}[\%]$	$\mathbf{A_{ops}}[\%]$	$\mathbf{E_r}[\%]$	$\mathbf{A_{ops}}[\%]$
(2/28), ≈ 10	1.05	$\approx$ 100.0	2.56	98.12	4.61	96.23
(5/28), ≈ 20	50.43	59.27	74.42	39.67	79.97	35.13
(8/28), ≈ 30	80.11	43.55	87.71	38.15	89.30	37.02
(11/28), ≈ 40	90.20	45.99	92.14	44.82	92.76	44.45
(14/28), $= 50$	97.93	51.78	98.43	51.52	98.57	51.45
(16/28), ≈ 60	99.99	57.85	$\approx$ 100.0	57.84	$\approx$ 100.0	57.84
(19/28), ≈ 70	$\approx$ 100.0	68.51	$\approx$ 100.0	68.51	$\approx$ 100.0	68.51
Threshold	0.95	0.95	0.97	0.97	0.98	0.98

Table 2. Results for *pcb1*, situation 2

$\mathbf{C_r}, \mathbf{C_r}[\%]$	$\mathbf{E_r}[\%]$	$\mathbf{A_{ops}}[\%]$	$\mathbf{E_r}[\%]$	$\mathbf{A_{ops}}[\%]$	$\mathbf{E_r}[\%]$	$\mathbf{A_{ops}}[\%]$
(2/28), ≈ 10	6.49	94.49	6.52	94.47	6.59	94.40
(5/28), ≈ 20	81.61	33.78	81.82	33.61	82.02	33.44
(8/28), ≈ 30	89.53	36.85	89.71	36.73	89.77	36.68
(11/28), ≈ 40	92.59	44.55	92.82	44.41	93.00	44.30
(14/28), $= 50$	98.40	51.53	98.59	51.44	98.63	51.42
(16/28), ≈ 60	$\approx$ 100.0	57.85	$\approx$ 100.0	57.85	$\approx$ 100.0	57.85
(19/28), ≈ 70	$\approx$ 100.0	68.51	$\approx$ 100.0	68.51	$\approx$ 100.0	68.51
Threshold	0.95	0.95	0.97	0.97	0.98	0.98

By comparing Table 1 and Table 2, we can appreciate the computational benefit associated with finding a good matching position at an early stage of the search. Considering again the interval $[\approx 20\%, \approx 50\%]$, for each pair ($C_r$, threshold value) the elimination ratio is higher and the number of operations smaller. It is interesting to note how, as the threshold becomes higher, the impact of the initial matching position becomes less notable: as already discussed, from the computational point of view the threshold itself acts as the initial correlation score, a high threshold behaving similarly to a good initial matching position.

Our experiments show that BPC's behaviour with image *pcb1* is rather typical: if C_r is less than 10%, the amount of eliminated points is too low while for C_r greater than 50% $A_{\text{ops}}(C_r)$ is an increasing function. Our experiments suggest also that a rather conservative choice, such as $C_r = 40\%$, can provide in most cases a satisfactory trade-off between the opposite requirements associated with setting the correlation ratio. Table 3 and 4 show the results obtained on the other images considered in this section using $C_r = 40\%$.

Table 3. Results with C_r = *40%*, situation 1

Image	$\mathbf{E_r}$[%]	$\mathbf{A_{ops}}$[%]	$\mathbf{E_r}$[%]	$\mathbf{A_{ops}}$[%]	$\mathbf{E_r}$[%]	$\mathbf{A_{ops}}$[%]
pcb2	70.32	56.68	79.02	51.28	83.63	48.43
albert	82.43	50.49	84.78	49.07	86.18	48.23
pcb3	68.69	58.70	78.20	52.98	87.02	47.67
plants	91.42	44.90	92.45	44.28	92.60	44.18
pm	85.38	48.32	92.29	44.13	94.80	42.61
Threshold	0.95	0.95	0.97	0.97	0.98	0.98

Table 4. Results with C_r = *40%*, situation 2

Image	$\mathbf{E_r}$[%]	$\mathbf{A_{ops}}$[%]	$\mathbf{E_r}$[%]	$\mathbf{A_{ops}}$[%]	$\mathbf{E_r}$[%]	$\mathbf{A_{ops}}$[%]
pcb2	87.10	46.27	87.39	46.09	87.89	45.78
albert	88.07	47.09	88.29	46.96	88.32	46.94
pcb3	96.70	41.84	97.01	41.65	97.33	41.46
plants	92.63	44.17	92.65	44.15	92.66	44.15
pm	96.47	41.60	96.57	41.54	96.63	41.50
Threshold	0.95	0.95	0.97	0.97	0.98	0.98

7 Improving the BPC technique

The basic BPC technique presented so far relies on checking the elimination condition only once at each image point. As discussed previously, this calls for a conservative choice of the correlation ratio (for which we propose $C_r = 40\%$). However, this choice limits the maximum speedup attainable by BPC since A_{ops} cannot be lower than C_r.

The natural approach to improve BPC consists in trying to use at each point the correlation ratio which is actually needed to judge that point. Points that can be eliminated on the basis of a small partial correlation term would then be skipped sooner than with the basic BPC, while points harder to assess would require the computation of more correlation products.

This means, in practice, evaluating the partial correlation, P_C, the corresponding function $\tilde{\beta}$ and the elimination condition with a fine granularity. The process can be described as follows. Given the point under examination, firstly a small group of initial rows is considered so as to compute a partial correlation term P_{C_0} together with the corresponding $\tilde{\beta}$ term $\tilde{\beta}_0$. These are added together to obtain γ_0 and check the elimination condition. If the condition is satisfied, the point is skipped, otherwise its analysis proceeds by taking into account a successive, small group of rows. This group is used to calculate an additional fraction of the correlation δP_{C_1} and the corresponding term $\delta\tilde{\beta}_1$. The former is added to P_{C0} to attain the current partial correlation P_{C_1}, the latter subtracted from $\tilde{\beta}_0$ to attain the current value of function $\tilde{\beta}$, $\tilde{\beta}_1$. Then, evaluation of $\gamma_1 = P_{C_1} + \tilde{\beta}_1$ allows for a new check of the elimination condition, which is now based on a tighter bound ($\gamma_0 \geq \gamma_1$). The process is iterated by considering successively small groups of rows, until the point is skipped at some stage or it ends up in computing the whole cross-correlation.

Although in principle this approach is feasible, the actual granularity with which the process is carried out strongly impacts its effectiveness. In fact, the higher the frequency with which the elimination condition is checked, the higher the number of branches that need to be inserted into the algorithm's code. This overhead may neutralise the savings brought by skipping points sooner than with the basic BPC. Moreover, the evaluation of the bounding function itself places an overhead growing with the granularity: if the process consists of k steps (i.e the elimination condition can be checked as many times as k) then it requires running $(k+1)$ distinct box-filters, with an overhead of $4(k+1)$ elementary operations per point. Besides that, the higher the number of box-filters, the larger the amount of data storage required by the algorithm; yet, with state-of-the-art processors, storage requirements should be minimised so as to enable efficient cache memory usage.

Therefore, we investigate here only the simplest modification to BPC's basic structure (from now on also abbreviated as BPC_2), which is based on two checks of the elimination condition (i.e. $k = 2$). This implies defining two correlation ratios

$$C_{r_1} = \frac{(n_1+1)}{N}, \qquad C_{r_2} = \frac{(n_2+1)}{N} \tag{28}$$

resulting from the partitioning of the sub-image under examination into the three sub-windows

$$\begin{aligned} &w_1 : \text{rows}\, 0 \ldots n_1, \quad w_2 : \text{rows}\, n_1+1 \ldots n_2, \\ &w_3 : \text{rows}\, n_2+1 \ldots N-1 \end{aligned} \tag{29}$$

and running three box-filters.

With this approach, the overhead associated with branches is still kept extremely low, as it is the case for the storage requirements associated with the box-filters. This allows us to rely on the usual hardware-independent metric A_{ops}, in order to assess the performance of the modified, two-check BPC technique.

Let's suppose that the two correlation ratios C_{r_1} and C_{r_2} have been chosen and that S_1 and S_2 are the number of points which would be eliminated by the basic BPC technique using, respectively, $C_r = C_{r_1}$ and $C_r = C_{r_2}$. Then, the total number of elementary operations executed by BPC_2, including the overhead of the triple box-filtering scheme, can be expressed as:

$$\begin{aligned} T_{\mathrm{ops}_2} = (W \cdot H - S_2) \cdot M \cdot N + (S_2 - S_1) \cdot M \cdot (n_2+1) \\ + S_1 \cdot M \cdot (n_1+1) + 12 \cdot W \cdot H. \end{aligned} \tag{30}$$

By introducing the elimination ratios

$$E_{r_1} = \frac{S_1}{W \cdot H}, \qquad E_{r_2} = \frac{S_2}{W \cdot H} \tag{31}$$

we can express the ratio to the number of operations executed by the standard pattern-matching algorithm as

$$A_{\mathrm{ops}_2} = \frac{[(1-E_{r_2}) + (E_{r_2} - E_{r_1}) \cdot C_{r_2} + E_{r_1} \cdot C_{r_1}] \cdot M \cdot N + 12}{M \cdot N + 4}. \tag{32}$$

A practical BPC_2 implementation requires choosing C_{r_1} and C_{r_2}, and thus dealing again with the trade-off related to the choice of correlation ratios. However, BPC_2 provides two degrees of freedom. Thus, the first check can be carried out relatively early, so as to achieve a significant saving at points that can be eliminated quickly, while the second one can be aimed at substantially increasing the elimination ratio. Based on these considerations and on the analysis of our experimental results, we propose $C_{r_1} \approx 20\%$ and $C_{r_2} \approx 40\%$ as default choice for BPC_2's correlation ratios.

Table 5. Results using BPC_2, with $C_{r_1} \approx 20\%$ and $C_{r_2} \approx 40\%$

Image	$\mathbf{A_{ops_2}}[\%]$	$\mathbf{A_{ops_2}}[\%]$	$\mathbf{A_{ops_2}}[\%]$
pcb1 (1)	35.73	29.44	27.88
pcb1 (2)	27.64	27.45	27.30
pcb2 (1)	51.74	44.99	40.28
pcb2 (2)	34.92	34.71	34.38
albert (1)	39.24	37.28	36.17
albert (2)	35.19	34.76	34.66
pcb3 (1)	50.82	45.10	39.74
pcb3 (2)	31.22	31.03	30.83
plants (1)	35.48	32.77	31.56
plants (2)	31.39	31.23	31.19
pm (1)	45.10	37.65	34.65
pm (2)	32.64	32.55	32.47
Threshold	0.95	0.97	0.98

Table 5 reports BPC_2's results for the images considered throughout the paper and with the proposed choice of correlation ratios. By comparing these results with those shown in the previous section, we can observe that BPC_2 yields notable performance improvements with respect to BPC.

Tables 1 and 2 indicate that with BPC, when the default, conservative choice of C_r differs from the ideal optimum one there can be a significant difference between the actual performance and the potential best performance of the technique. For instance, with *pcb1* this difference varies from $\cong 2\%$ to $\cong 11\%$. We have collected the results attained with BPC_2 using, with every image of the data set and in both situations, the optimum pair of correlation ratios (chosen among those considered throughout the paper). In most cases the optimum pair differs from the default one; yet now the difference between the performance attained with the proposed choice and the ideal best performance varies within a smaller range, i.e. from $\cong 0.6\%$ to $\cong 7\%$; for instance, in the case of *pcb1* the difference ranges from $\cong 0.6\%$ to $\cong 1.7\%$. This analysis provides the indication that BPC_2 is intrinsically more robust than BPC with respect to the possible mismatch between the proposed choice of the parameters and the unknown optimum one.

8 Conclusion

We have described a novel template-matching algorithm, called BPC, which extends to NCC the principles of the SEA and PDE techniques, which had been proposed so far only for matching algorithms relying on distorsion measures (i.e. SAD, SSD).

BPC exploits a suitable bound for the NCC function to establish an elimination condition aimed at discarding rapidly the search positions that are guaranteed not to provide a better degree of match with respect to the current best-matching one. Evaluation of the elimination condition requires the calculation of a given portion of the cross-correlation function, which is defined through the algorithm's basic parameter, namely the correlation ratio (C_r).

Since BPC is a data-dependent optimisation technique – as it is the case of SEA and PDE from which BPC derives – it is not possible to assess its computational benefit in an absolute manner. Indeed, this depends on the image, the template, the position of the template within the image, the correlation threshold, as well as on whether or not one may deploy information concerning the expected matching position. In such a context, we have provided experimental results showing that, by choosing C_r properly, BPC can significantly reduce the number of calculations with respect to the standard NCC-based template-matching algorithm. With the images and in the situations considered throughout the paper, running BPC with the suggested choice of the correlation ratio allows for nearly halving the number of calculations.

BPC basic formulation can be extended so as to define a sequence of elimination conditions based on increasing correlation ratios. We have investigated the simplest algorithm based on this idea, called BPC_2, which consists of the introduction of a second elimination condition. Our experimental results show that, compared to BPC, BPC_2 yields notable computational benefits and is intrinsically more robust with respect to the possible mismatch between the suggested parameters choice and the unknown optimum one.

Acknowledgements. We wish to thank CORECO Inc. for the use of image *pcb1* and the Vision and Autonomous Systems Center (VASC) at Carnegie Mellon University for the use of images *plants* and *parking meter*.

References

1. Bei CD, Gray RM (1985) An improvement of the minimum distorsion encoding algorithm for vector quantization. IEEE Trans Commun 33: 1132–1133
2. Li W, Salari E (1995) Successive elimination algorithm for motion estimation. IEEE Trans Image Process 4(1): 105–107
3. McDonnell MJ (1981) Box-filtering techniques. Comput Graph Image Process 17: 65–70
4. Rosenfeld A, Kak AC (1982) Digital picture processing, vol 2. Academic Press, New York, pp 48–49
5. Rosenfeld A, Vanderburg GJ (1977) Coarse-fine template matching. IEEE Trans Syst Man Cyb 7: 104–107
6. Rudin W (1966) Real and complex analysis. McGraw-Hill, New York
7. Vanderburg GJ, Rosenfeld A (1977) Two-stage template matching. IEEE Trans Comput 26: 384–393
8. Wang HS, Mersereau RM (1999) Fast algorithms for the estimation of motion vectors. IEEE Trans Image Process 8(3): 435–439

Stefano Mattoccia received his Laurea (Msc) degree in Electronic Engineering from the University of Bologna in 1997. In 1998 he joined the Department of Electronics, Computer Science and Systems (DEIS) of the University of Bologna, where he is currently pursuing his Ph.D. degree in electronic engineering and computer science. His main research interests are real-time stereo vision, computer vision and pattern recognition.

Luigi Di Stefano graduated at the University of Bologna as an electronic engineer in 1989. In 1990 he joined the Department of Electronics, Computer Science and Systems (DEIS) of the University of Bologna where he received a Ph.D. in electronic engineering and computer science in 1994. In 1995 he spent six months at Trinity College Dublin as a postdoctoral fellow. He is currently an associate professor at DEIS. He has also joined ARCES (Advanced Research Centre on Electronic Systems “Ercole De Castro”), a research centre instituted in 2001 at the University of Bologna. His research interests include image processing, machine vision and computer architecture.