

# An Efficient Algorithm for Exhaustive Template Matching based on Normalized Cross Correlation

Luigi Di Stefano, Stefano Mattoccia , Martino Mola  
DEIS-ARCES

University of Bologna

Viale Risorgimento, 2 40136 Bologna (Italy)

ldistefano@deis.unibo.it, smattoccia@deis.unibo.it, mmola@deis.unibo.it

## Abstract

*This work proposes a novel technique aimed at improving the performance of exhaustive template matching based on the Normalized Cross Correlation (NCC). An effective sufficient condition, capable of rapidly pruning those match candidates that could not provide a better cross correlation score with respect to the current best candidate, can be obtained exploiting an upper bound of the NCC function. This upper bound relies on partial evaluation of the cross-correlation and can be computed efficiently, yielding a significant reduction of operations compared to the NCC function and allows for reducing the overall number of operations required to carry out exhaustive searches. However, the bounded partial correlation (BPC) algorithm turns out to be significantly data dependent. In this paper we propose a novel algorithm that improves the overall performance of BPC thanks to the deployment of a more selective sufficient condition which allows for rendering the algorithm significantly less data dependent. Experimental results with real images and actual CPU time are reported.*

## 1. Introduction

Matching a template sub-image into a given image is an ubiquitous 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. The typical distorsion measures used in template matching algorithms are the Sum of Absolute Differences (SAD) and the Sum of Squared Differences (SSD), while Normalized

Cross Correlation (NCC) is by far the most widely used correlation measure.

Since with large-size images and/or templates the matching process can be computationally very expensive, numerous techniques aimed at speeding up the basic approach have been devised (see [6] 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 (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 (i.e. matching a sub-template first, and then the whole template only at good candidate positions). However, techniques a), b), 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 and can be trapped by local extremes resulting in wrong localisation of the template under examination.

On the other hand, in the specific case of distorsion measures two interesting techniques, called SEA (Successive Elimination Algorithm) [4], [8] and PDE (Partial Distorsion Elimination) [1], allow for notably speeding up the computation required by an exhaustive-search template matching process. 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 propose a novel technique aimed at improving the performance of exhaustive template matching algorithms based on NCC using an upper bound of the NCC function. The original technique proposed in [3] is improved using a tighter bound that renders the algorithm significantly less data dependent and improves the overall performance.

## 2. An upper bound for Normalized Cross Correlation

With Normalized Cross Correlation the template sub-image is located into the image under examination by searching for the maximum of the NCC function:

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

The numerator of (1) represents the correlation *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 initialization time, the former can be obtained very efficiently via a box-filtering scheme [5] at a cost of only 4 elementary operations (i.e. accumulation of a product term) per image point.

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

$$\beta(x, y) \geq C(x, y) = \sum_{j=1}^N \sum_{i=1}^M I(x+i, y+j) \cdot T(i, j), \quad (2)$$

by normalising  $\beta$  we obtain an upper-bound for the NCC function:

$$\frac{\beta(x, y)}{|I(x, y)|_2 \cdot |T|_2} \geq \frac{C(x, y)}{|I(x, y)|_2 \cdot |T|_2} = NCC(x, y) \quad (3)$$

Then, indicating as  $NCC_m$  the current correlation maximum, if at point  $x, y$  the elimination condition

$$\frac{\beta(x, y)}{|I(x, y)|_2 \cdot |T|_2} < NCC_m \quad (4)$$

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 (4) 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:

$$\frac{C(x, y)}{|I(x, y)|_2 \cdot |T|_2} \geq NCC_m \quad (5)$$

In order to find  $\beta(x, y)$  consider the relationship between geometric and arithmetic series known as Jensen inequality [7]:

$$\sqrt[n]{\prod_{k=1}^n a_k} \leq \frac{\sum_{k=1}^n a_k}{n}, \quad (6)$$

Applying (6) to the generic pair of homologous terms appearing in the cross-correlation product yields

$$\frac{I(x+i, y+j) + T(i, j)}{2} \geq \sqrt{I(x+i, y+j) \cdot T(i, j)} \quad (7)$$

from which we can derive by simple manipulations

$$\frac{I(x+i, y+j)^2 + T(i, j)^2}{2} \geq I(x+i, y+j) \cdot T(i, j) \quad (8)$$

Therefore, if we pose

$$\beta(x, y) = \frac{\sum_{j=1}^N \sum_{i=1}^M I(x+i, y+j)^2 + \sum_{j=1}^N \sum_{i=1}^M T(i, j)^2}{2} \quad (9)$$

from inequality (8) it follows that  $\beta(x, y)$  satisfies (2) at each image point.

The elimination condition relies on calculating  $\beta(x, y)$  instead of  $C(x, y)$ . As pointed out, this holds the potential for speeding up the matching process as long as  $\beta$  can be computed much more rapidly than  $C$ . The function defined in (9) requires evaluation of the squares of the  $L_2$  norms of the sub-image under examination,  $(|I(x, y)|_2)^2$ , and of the template,  $(|T|_2)^2$ . The latter quantity can be precomputed and the former calculated recursively running a box-filter. Moreover, one may also notice that evaluating  $\beta$  as defined in (9) does not require any extra computation since the quantity  $(|I(x, y)|_2)^2$  had to be calculated for the purpose of normalising the correlation in the new maximum condition (5).

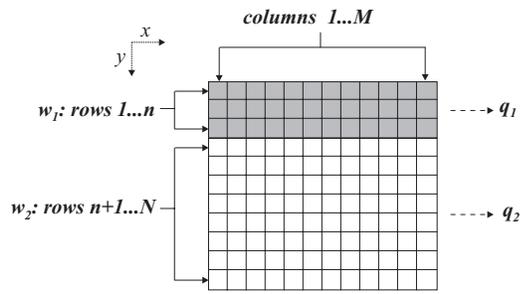
Yet, for the above described approach to yield computational advantages, the elimination condition should be also effective. Unfortunately, this is not the case for condition (4). In fact, it is straightforward to verify that the quantity

$$\frac{\beta(x, y)}{|I(x, y)|_2 \cdot |T|_2} \quad (10)$$

is lower-bounded by 1, which in turn is the upper-bound of the NCC function. Hence, the elimination condition defined via the equations (4) and (9) is never satisfied.

## 3. Bounded Partial Correlation

An effective elimination condition can be obtained by computing only a small portion of the actual correlation function, referred to as ‘‘partial correlation’’, and bounding



**Figure 1. Partitioning of the window under examination.**

the residual portion with a term that, similarly to  $\beta$ , is derived from (8).

Defining the partial correlation associated with row  $n$  as

$$P_C(x, y, M, n) = \sum_{j=1}^n \sum_{i=1}^M I(x+i, y+j) \cdot T(i, j), \quad (11)$$

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

$$\gamma(x, y, M, n) = P_C(x, y, M, n) + \tilde{\beta}(x, y, M, n), \quad (12)$$

with

$$\tilde{\beta}(x, y, M, n) = \frac{\sum_{j=n+1}^N \sum_{i=1}^M I(x+i, y+j)^2 + \sum_{j=n+1}^N \sum_{i=1}^M T(i, j)^2}{2}, \quad (13)$$

The new elimination condition result

$$\frac{\gamma(x, y, M, n)}{|I(x, y)|_2 \cdot |T|_2} < NCC_m. \quad (15)$$

Hence,  $\gamma$  combines (see Figure (1)) the evaluation of a given part of the total correlation,  $P_C$ , computed with the approximation of the residual part by means of an upper-bounding term,  $\tilde{\beta}$ , derived from (8).

For the described bound-based approach to prove useful it is mandatory to compute  $\gamma$  very efficiently. This is achieved running a distinct box-filter on each sub-window, with the filter running on  $w_1$  calculating the quantity

$$q_1(x, y, M, n) = \sum_{j=1}^n \sum_{i=1}^M I(x+i, y+j)^2 \quad (16)$$

and that running on  $w_2$  the quantity

$$q_2(x, y, M, n) = \sum_{j=n+1}^N \sum_{i=1}^M I(x+i, y+j)^2. \quad (17)$$

Then,  $q_2$  can be used to compute  $\gamma$  while  $q_1$  and  $q_2$  can be added together to obtain  $(|I(x, y)|_2)^2$ , which is needed to normalise  $\gamma$  and  $C$  in the elimination and new maximum conditions (15) and (5). It is worth pointing out that, due to the partitioning, the computational cost associated with box-filtering amounts now to 8 elementary operations per point. On the other hand, the evaluation of the partial correlation term present in  $\gamma$  does not introduce any overhead with respect to the standard algorithm since when (15) fails  $C(x, y)$  can be attained adding the residual product terms to  $P_C(x, y, M, n)$ . Eventually, the last quantity needed to obtain  $\gamma$ , i.e.  $\sum_{j=n+1}^N \sum_{i=1}^M T(i, j)^2$ , can be precomputed at initialisation time.

We call  $C_r$  (Correlation Ratio) the ratio between the number of points involved in the calculation of the partial correlation and those needed to evaluate the entire correlation function

$$C_r = \frac{n}{N} \quad (18)$$

It is clear from the definition of  $\gamma$  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  $C_r$ .

A typical parameter of template matching algorithms is a threshold on the minimum correlation value yielding a valid match. This value allows for discriminating when an instance of the template is present or not in the image under examination. The standard NCC algorithm (i.e. brute force approach) does not take any computational advantage from the selection of high thresholds. Conversely, as the correlation threshold is increased the sufficient condition of BPC gets more effective.

## 4. Multiresolution Bounded Partial Correlation

The technique outlined in the previous section, referred to as BPC (Bounded Partial Correlation), 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.  $NCC_m$  gets closer to the best score  $NCC_{max}$ ). Hence, BPC yields better speed-up if, given the scan order, the search process finds rapidly a good matching position.

An example of this behavior can be observed comparing Figure 2 and Figure 3. Figure 2 shows the NCC scores com-

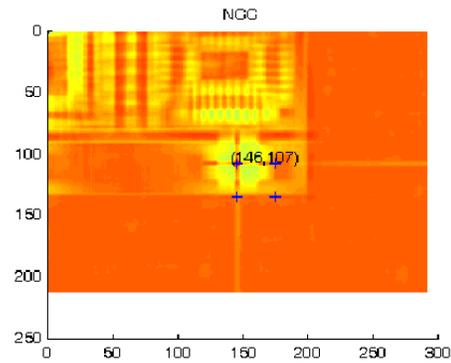
puted by the standard NCC algorithm on the image and the template *pcb1* available<sup>1</sup> at [2]. Figure 3 shows the scores of the NCC function computed by the BPC technique, highlighting in dark blue (black if the paper is printed on a BW printer) those points discarded by the elimination condition (15), on the same image *pcb1*, with a  $C_r = 30\%$  and following the usual raster scan order. Since the template is located approximately at the center of the image, the effectiveness of the BPC technique turns out to get higher starting from the middle of the search process when the template is approached.

Hence, the BPC technique can be improved and rendered less data dependent if a good score  $NCC_m$  (or better, the best score  $NCC_{max}$ ) can be quickly found. An estimated value  $\widehat{NCC}$  and location  $(\hat{x}, \hat{y})$  of the point with the highest cross correlation score ( $NCC_{max}$  at  $(x_{max}, y_{max})$ ) can be obtained at a small computational cost within a multiresolution framework. The template is first located at coarse resolution (at level  $l = k$ , in position  $(\hat{x}^{(k)}, \hat{y}^{(k)})$  with score  $\widehat{NCC}^{(k)}$ ) then the search is refined at higher resolution (i.e.  $l = 0$ ) in proximity of the best score found at lower resolution re-projected at higher resolution.

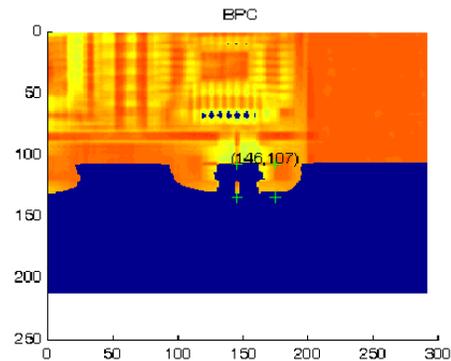
At both resolution levels the computation is carried out using the BPC technique. At higher resolution the  $\widehat{NCC}$  value (upper bounded by  $NCC_{max}$ ) is used as initial estimate of  $NCC_{max}$ . This initial estimate allows for improving the BPC technique with a more tight elimination condition (15). Since the multiresolution BPC outlined, referred from now as MBPC, relies on a initial estimate  $\widehat{NCC}$ , not necessarily equal to  $NCC_{max}$ , the coarse image and template at level  $l = k$  can be obtained with a negligible overhead simply sub-sampling the original image and template with a step  $2^k$ . In fact, MBPC provides the same solution of NCC and BPC algorithms also if the initial estimated best match obtained with the coarse to fine approach is not exact (i.e.  $(\hat{x}, \hat{y})$  does not collides with  $(x_{max}, y_{max})$ ).

Figure 4 shows at resolution level  $l = 0$  the scores of the NCC function computed with the MBPC technique proposed. The coarse resolution level was  $k = 2$ . In dark blue (black if the paper is printed on a BW printer) are highlighted those points discarded by the elimination condition (15) on the same image *pcb1* with  $C_r = 30\%$  following the usual raster scan order. Figure 4 points out that the performance of MBPC is not affected by the the position of the template within the image under examination resulting in an higher number of points rejected by the elimination condition if compared to BPC with the same  $C_r$  (see Figure 3).

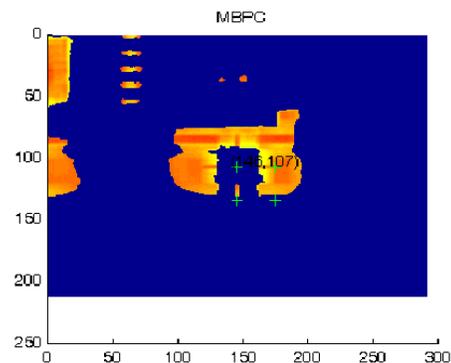
<sup>1</sup>The data set used for experimental results is not shown in the paper due to constraint on the number of pages. It can be found at [2]



**Figure 2.** NCC scores for the template and the image *pcb1* available at [2].



**Figure 3.** NCC scores for the template and the image *pcb1* available at [2]. In blue, the points rejected by BPC thanks to the elimination condition with  $C_r = 30\%$  and threshold 0.00.



**Figure 4.** NCC scores for the template and the image *pcb1* available at [2]. In blue, the points rejected by MBPC thanks to the elimination condition with  $C_r = 30\%$ , threshold 0.00 and  $k = 2$ .

## 5. Performance evaluation metric

BPC and MBPC technique belong to the class of exhaustive search algorithms. Therefore, in order to evaluate their performance we compare BPC and MBPC 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 MBPC and that required by the standard algorithm, which amounts to  $(M \cdot N + 4)$ . For example  $A_{ops} = \alpha\%$  means that the considered algorithm executes an average number of operation per point which is  $\alpha\%$  of that required by the standard algorithm.

The number of operations executed by BPC at a given point depends on the outcome of (15). If this is satisfied, the algorithm executes only the  $M \cdot n$  operations needed to evaluate the partial correlation term,  $P_C(x, y, M, n)$ . 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 8 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

$$T_{BPC} = (W \cdot H - S) \cdot M \cdot N + S \cdot M \cdot n + 8 \cdot W \cdot H \quad (19)$$

The number of operations executed by MBPC at a given point depends on the outcome of (15) at level  $l = k$  and at level  $l = 0$ . At level  $l$  if (15) is satisfied, the algorithm executes<sup>2</sup> only the  $M^{(l)} \cdot n^{(l)}$  operations needed to evaluate the partial correlation term,  $P_C^{(l)}(x^{(l)}, y^{(l)}, M^{(l)}, n^{(l)})$ . If the condition fails, the partial correlation term must be integrated with its residual part, giving a total number of operations equal to  $M^{(l)} \cdot N^{(l)}$ . In both cases we apply the double box-filtering scheme, which requires 8 additional operations per point.

At  $l$  resolution level we call  $S^{(l)}$  the number of points for which the elimination condition (15) is satisfied. Moreover, we call  $S_*^{(0)}$  the number of points for which the elimination condition is satisfied at the higher resolution level within the search area of size  $(2^{k+1} + 1) \times (2^{k+1} + 1)$  in proximity of  $((\hat{x}^{(k)} - 1) \times 2^k + 1, ((\hat{y}^{(k)} - 1) \times 2^k + 1))$ . Hence, we can express the total number of elementary operations executed by MBPC as

$$T_{MBPC} = \begin{aligned} & (W^{(0)} \cdot H^{(0)} - S^{(0)}) \cdot M^{(0)} \cdot N^{(0)} + S^{(0)} \cdot M^{(0)} \cdot n^{(0)} + \\ & (W^{(k)} \cdot H^{(k)} - S^{(k)}) \cdot M^{(k)} \cdot N^{(k)} + S^{(k)} \cdot M^{(k)} \cdot n^{(k)} + \\ & ((2^{k+1} + 1)^2 - S_*^{(0)}) \cdot M^{(0)} \cdot N^{(0)} + S_*^{(0)} \cdot M^{(0)} \cdot n^{(0)} + \\ & 8 \cdot W^{(0)} \cdot H^{(0)} + 8 \cdot W^{(k)} \cdot H^{(k)} + 8 \cdot (2^{k+1} + 1)^2 \end{aligned} \quad (20)$$

## 6. Experimental results

We present here the results, in terms of  $A_{ops}$  as well as in terms of CPU time, obtained on the data set *pcb1*, *pcb2*, *albert*, *pcb3*, *plants* and *pm* available at [2]. The sub-image used as template, also available at [2], has been extracted from a different - though similar - image. Within the entire data set we use the same  $C_r = 30\%$  for BPC and MBPC. For MBPC, within the entire data set, we use the same coarse resolution level  $k = 2$ . Table 1 reports  $A_{ops}$  for the three examined algorithms NCC, BPC and MBPC for four threshold levels. The Table shows that BPC allows for reducing the number of  $A_{ops}$  in the worse cases (i.e. *pcb1* and *pm* with threshold 0.00), up to  $\simeq 68\%$  and in the best case (i.e. image *pcb1* with threshold 0.98), up to  $\simeq 38\%$ .

As for MBPC, the Table shows that it can always halve  $A_{ops}$  independently of the threshold value. The reduction ranges from  $\simeq 50\%$  (i.e. image *albert* and *pm*) to  $\simeq 38\%$  (i.e. image *pcb1*). Table 1 shows also that MBPC yields a significant reduction of  $A_{ops}$  compared to BPC. With threshold 0.00 this reduction ranges from  $\simeq 30\%$  to  $\simeq 4\%$ , with threshold 0.98 from  $\leq 30\%$  to  $\simeq 10\%$ . Both BPC and MBPC are data dependent optimisation techniques. However, unlike BPC, MBPC turns out to be nearly unaffected by the threshold value as well as by the position of the template within the image. Both properties can be clearly appreciated analysing the data in Table 1 and observing Figures 3 and 4, which show the effectiveness of the elimination conditions of BPC and MBPC on image *pcb1* (available at [2]). As discussed, since MBPC starts the finer resolution search with a good estimate,  $\widehat{NCC}$ , of  $NCC_{max}$  it can deploy an effective elimination condition throughout the whole search process independently of the position of the instance of the template within the image.

Table 2 reports the CPU time percentages of BPC and MBPC with respect to the brute force NCC algorithm measured on a 1 GHz Pentium III processor running Linux. The three algorithms were compiled with GCC version 3.2, enabling the higher optimisation level and with target processor the Pentium III (optimisation switches: -O3 -march=pentium3 -finline-functions -fomit-frame-pointer -funroll-loops). The Table shows that the actual measurements follow quite closely the predictions yield by the defined performance evaluation metric.

<sup>2</sup>Superscript  $l$  refers to the dimensions of image, template and  $n$  scaled down according to the resolution level  $l$ .

Image	NCC	BPC	MBPC	BPC	MBPC	BPC	MBPC	BPC	MBPC
<i>pcb1</i>	100.00	68.20	38.29	45.06	38.26	39.82	38.25	38.72	38.24
<i>pcb2</i>	100.00	62.59	44.02	62.58	44.01	54.72	44.00	48.96	43.98
<i>albert</i>	100.00	53.88	49.11	53.17	49.11	51.29	49.11	50.34	49.10
<i>pcb3</i>	100.00	60.42	46.52	60.42	46.52	60.36	46.52	56.94	46.52
<i>plants</i>	100.00	59.73	44.23	49.05	44.23	45.16	44.21	44.16	44.20
<i>pm</i>	100.00	67.88	49.22	60.12	49.22	53.17	49.22	50.76	49.22
<b>Threshold</b>	—	<b>0.00</b>	<b>0.00</b>	<b>0.95</b>	<b>0.95</b>	<b>0.97</b>	<b>0.97</b>	<b>0.98</b>	<b>0.98</b>

**Table 1. Average number of elementary operations per point ( $A_{ops}$ ) for NCC, BPC and MBPC algorithms. For BPC and MBPC  $C_r = 30\%$ . For MBPC  $k = 2$ .**

Image	NCC	BPC	MBPC	BPC	MBPC	BPC	MBPC	BPC	MBPC
<i>pcb1</i>	100.00	71.88	42.92	48.63	42.92	43.32	42.92	42.21	42.92
<i>pcb2</i>	100.00	65.81	48.09	65.81	48.12	58.04	48.11	52.32	48.11
<i>albert</i>	100.00	55.36	51.08	54.64	51.08	52.75	51.07	51.78	51.07
<i>pcb3</i>	100.00	62.55	49.69	62.56	49.69	62.54	49.69	59.69	49.70
<i>plants</i>	100.00	60.02	44.62	49.31	44.62	45.41	44.62	44.42	44.62
<i>pm</i>	100.00	68.66	49.98	60.85	49.98	53.84	49.98	51.43	49.98
<b>Threshold</b>	—	<b>0.00</b>	<b>0.00</b>	<b>0.95</b>	<b>0.95</b>	<b>0.97</b>	<b>0.97</b>	<b>0.98</b>	<b>0.98</b>

**Table 2. CPU time percentages of BPC and MBPC with respect to the brute force NCC algorithm measured on a 1 GHz Pentium III processor running Linux. For BPC and MBPC  $C_r = 30\%$ . For MBPC  $k = 2$ .**

## 7. Conclusion

We have described an improvement, referred to as MBPC, to a novel template matching algorithm, i.e. BPC. The BPC technique extends to the case of the Normalized Cross-Correlation the principles of the SEA and PDE techniques. 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.

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 MBPC algorithm proposed in this paper deploys an estimated value of the best NCC score obtained using a multiresolution approach. This allows for rendering the algorithm almost independent of the threshold value as well as of the position of the template within the image. MBPC yield also a considerable reduction of the CPU time needed to carry out an exhaustive pattern matching process based on the NCC. Hence, MBPC significantly improves the overall performance of the bounded partial correlation approach

with respect to the original BPC algorithm.

## References

- [1] C. Bei and R. Gray. An improvement of the minimum distortion encoding algorithm for vector quantization. *IEEE Trans. on Communication*, 33:1132–1133, 1985.
- [2] L. Di Stefano and S. Mattoccia. *Data set used for experimental results*. World Wide Web, <http://www.vision.deis.unibo.it/~smattoccia/PatternMatching.html>, 2003.
- [3] L. Di Stefano and S. Mattoccia. Fast Template Matching using Bounded Partial Correlation. *Machine Vision and Applications*, 13(4), Feb 2003.
- [4] W. Li and E. Salari. Successive elimination algorithm for motion estimation. *IEEE Trans. on Image Processing*, 4(1):105–107, 1995.
- [5] M. Mc Donnell. Box-filtering techniques. *Computer Graphics and Image Processing*, 17:65–70, 1981.
- [6] A. Rosenfeld and A. Kak. *Digital Picture Processing*, volume 2. Academic Press, 1982.
- [7] W. Rudin. *Real and Complex Analysis*. McGraw-Hill, New York, USA, 1966.
- [8] H. Wang and R. Mersereau. Fast algorithms for the estimation of motion vectors. *IEEE Trans. on Image Processing*, 8(3):435–439, 1999.