A Template Analysis Methodology to Improve the Efficiency of Fast Matching Algorithms

Federico Tombari, Stefano Mattoccia, Luigi Di Stefano,
Fabio Regoli, and Riccardo Viti

Department of Electronics Computer Science and Systems (DEIS)
Advanced Research Center on Electronic Systems (ARCES)
University of Bologna, Italy
{federico.tombari,stefano.mattoccia,luigi.distefano}@unibo.it,
{fabio.regoli,riccardo.viti}@studio.unibo.it
www.vision.deis.unibo.it

Abstract. Several methods aimed at effectively speeding up the block matching and template matching tasks have been recently proposed. A class of these methods, referred to as exhaustive due to the fact that they optimally solve the minimization problem of the matching cost, often deploys a succession of bounding functions based on a partitioning of the template and subwindow to perform rapid and reliable detection of non-optimal candidates. In this paper we propose a study aimed at improving the efficiency of one of these methods, that is, a state-of-the-art template matching technique known as *Incremental Dissimilarity Approximations* (IDA). In particular, we outline a methodology to order the succession of bounding functions deployed by this technique based on the analysis of the template only. Experimental results prove that the proposed approach is able to achieve improved efficiency.

1 Introduction and Previous Work

Template matching and block matching are two classical image analysis problems that occur in countless vision applications. Template matching is widely deployed for tasks such as quality control, defect detection, robot navigation, face and object recognition, edge detection. On the other hand, block matching is a common approach adopted, e.g., for the purpose of motion estimation in video compression, so as to reduce temporal redundancy in video sequences.

Template matching and block matching inherently rely on a matching cost that, once minimized (or maximized, in case a similarity measure is deployed) allows locating the position of the template in the search space. Commonly used matching costs are those derived from the L_p norm:

$$\delta_p(x,y) = ||I_s(x,y) - T||_p^p = \sum_{i=1}^{M} \sum_{j=1}^{N} \left| I(x+i, y+j) - T(i,j) \right|^p \qquad (1)$$

where T is the template to be found and $I(x,y)$ is the image *subwindow* at location (x,y), both of size $M \times N$, while $|| \cdot ||_p$ denotes the L_p norm. If $p = 1$

J. Blanc-Talon et al. (Eds.): ACIVS 2009, LNCS 5807, pp. 100–108, 2009.

then $\delta_p(x, y)$ is the *Sum of Absolute Differences* (SAD), while by taking $p = 2$ it becomes the *Sum of Squared Differences* (SSD).

Since this task is usually computationally expensive, several methods have been recently proposed with the aim of rendering this minimization procedure more efficient. Within them, a class of methods is referred to as exhaustive since they optimally solve the problem by guaranteeing that the minimum they find is always the global one [2], [4], [5], [3], [7], [6], [8], [9]. One of the first exhaustive methods ever proposed is *Partial Distortion Elimination* (PDE) [9], which simply terminates the computation of the SAD function once the error becomes higher than the current minimum. Then, Li and Salari [1] proposed a method that deploys the triangular inequality to rapidly detect non-optimal candidates based on the SAD measure. Successive proposals tried to improve Li and Salari's performance by using partitioning schemes and relying on several more sophisticated bounding functions computed on corresponding parts of the template and subwindow.

Some of these techniques apply an iterative refinement of the bounding function based either on multilevel schemes [4], [7] or on the concept of partial cost [2], [9]. It is interesting to note that very few investigations have been performed so far on which parts of the template and subwindow should be analyzed first during this iterative process. As a matter of facts, the majority of these methods simply analyses the partitions in a sequential manner. In [10] an improvement of the PDE method is proposed that relies on the analysis of the image subwindow $I(x, y)$ in order to detect the regions of the subwindow where the matching cost should be computed first.

In this paper, we propose an analysis aimed at improving the performance of a state-of-the-art template matching technique known as *Incremental Dissimilarity Approximations* (IDA) [2]. In particular, we present here a strategy to optimally select on which parts of the template and subwindow the matching cost should be computed first. A relevant aspect is that the analysis is carried out on the template only, so that it can be regarded as an offline stage of the algorithm and therefore does not yield any computational overhead at runtime. The paper is structured as follows. Section 2 reviews the IDA technique, while Section 3 describes the proposed approach. Then, 4 presents experimental results aimed at assessing the improvements yielded by the proposed method.

2 The IDA Algorithm

The IDA technique is an exhaustive technique that aims at speeding up template matching based on matching costs derived from the L_p norm and relies on the *triangular inequality*:

$$||I_s(x, y) - T||_p^p \geq \big|||I_s(x, y)||_p - ||T||_p\big|^p \tag{2}$$

Hence, the right-hand term, hereinafter recalled as $\beta_p(x, y)$, is a lower-bound of $\delta_p(x, y)$. Let's now consider a partitioning of the template and the image subwindow into r disjoint regions. In order to increase the computational efficiency, IDA deploys regions made out of successive rows and characterized by equal number of rows, n (under the assumption that N is a multiple of r). Let the *partial bound* term computed between rows (ρ, θ) be defined as:

$$\beta_p(x,y)|_{\rho}^{\theta} = \left| ||I_s(x,y)||_p \big|_{\rho}^{\theta} - ||T||_p \big|_{\rho}^{\theta} \right|^p =$$
$$\left| [\sum_{i=1}^{M}\sum_{j=\rho}^{\theta} |I(x+i,y+j)|^p]^{\frac{1}{p}} - [\sum_{i=1}^{M}\sum_{j=\rho}^{\theta} |T(i,j)|^p]^{\frac{1}{p}} \right|^p \quad (3)$$

and the *partial dissimilarity* term between rows (ρ, θ) be defined as:

$$\delta_p(x,y)|_{\rho}^{\theta} = ||I_s(x,y) - T||_p^p \big|_{\rho}^{\theta} = \sum_{i=1}^{M}\sum_{j=\rho}^{\theta} |I(x+i,y+j) - T(i,j)|^p \quad (4)$$

It is possible to compute (3) on each of the r regions defined on the template and image subwindow and to sum up these terms:

$$\beta_{p,r}(x,y)|_1^N = \sum_{t=1}^{r} \beta_p(x,y)|_{(t-1)\cdot n+1}^{t\cdot n} \quad (5)$$

so as to yield a lower bound of the $\delta_p(x,y)$ function which is *tighter* than (2), i.e.:

$$\delta_p(x,y) \geq \beta_{p,r}(x,y)|_1^N \geq \beta_p(x,y) \quad (6)$$

The bounding property of function $\beta_{p,r}(x,y,n)|_1^N$ can be deployed in order to select rapidly non-optimal candidates thanks to the following sufficient condition:

$$\beta_{p,r}(x,y)|_1^N > \delta_m \quad (7)$$

where δ_m is the minimum of the distance function *found so far*. If inequality (7) holds, then the current candidate can not be the one yielding the global minimum due to the left inequality in (6); thus, the algorithm can proceed with the next candidate saving the computations needed for $\delta_p(x,y)$.

In the case that (7) is not verified, IDA determines a more efficient sufficient condition characterized by a bounding function closer to the actual value of the $\delta_p(x,y)$ function. This is done by computing the partial dissimilarity term associated with the first region and then using it to replace the corresponding partial bound term in $\beta_{p,r}(x,y)|_1^N$:

$$\gamma_{p,(r-1)}(x,y) = \delta_p(x,y)|_1^n + \beta_{p,(r-1)}(x,y)|_{n+1}^N \quad (8)$$

yielding an associated sufficient condition analogous to (7). Should this new condition be not verified either, the algorithm would continue checking other sufficient conditions characterized by increasing efficiency by substituting, at generic step i, the $i-th$ partial bound term with its corresponding partial dissimilarity term. Overall, the algorithm can test up to r increasingly tighter sufficient conditions, associated with bounding functions

$$\beta_{p,r}|_1^N, \gamma_{p,(r-1)}, \gamma_{p,(r-2)}, \cdots, \gamma_{p,1} \quad (9)$$

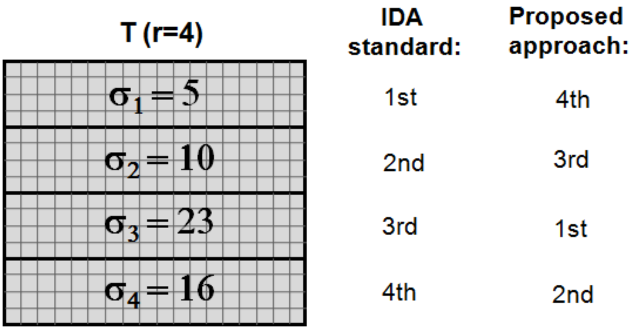
Should no one of these conditions be verified, the algorithm would complete the computation of the dissimilarity function by calculating the partial dissimilarity term associated with the last region and then compare $\delta_p(x, y)$ to δ_m. Thanks to this scheme, the IDA algorithm has proved to achieve notable speed-ups compared to state-of-the-art techniques [2].

3 The Proposed Ordering Approach

As pointed out in the previous Section, the IDA algorithm exploits a succession of increasingly tighter bounding function based on the substitution of the partial bound term with the partial dissimilarity term on the current region. This is done by following a naive *top-down* scheme that starts computing the first partial dissimilarity on the first region and sequentially follows down to the last region (*forward substitution*). The rationale beyond this work is to propose a more clever approach to order the computation of the partial dissimilarity terms on the r regions upon some criteria based on the characteristics of the data. In particular, since we don't want to increase the computational burden of the algorithm, the proposed approach should rely on the analysis of the template only, so that the computation of the best ordering of the partition could be seen as a negligible offline stage of the process, that has to be computed once for each template used in the search.

The intuitive assumption, supported by experimental analysis, is that on those areas of the template where the signal is stronger, the partial bound term tends to be less tight to the corresponding partial dissimilarity term. Hence, as a general rule the partial dissimilarity term should be computed first on those regions denoted by the highest presence of signal (i.e. non-uniform regions). In our approach, to measure the amount of signal present on the t-th region, we use a very basic indicator such as the intensity variance:

$$\sigma_t(x, y) = \frac{1}{M \cdot n} \cdot \sum_{i=1}^{M} \sum_{j=(t-1)\cdot n+1}^{t \cdot n} (T(i, j) - \mu_t(x, y))^2 \tag{10}$$



T (r=4)	IDA standard:	Proposed approach:
$\sigma_1 = 5$	1st	4th
$\sigma_2 = 10$	2nd	3rd
$\sigma_3 = 23$	3rd	1st
$\sigma_4 = 16$	4th	2nd

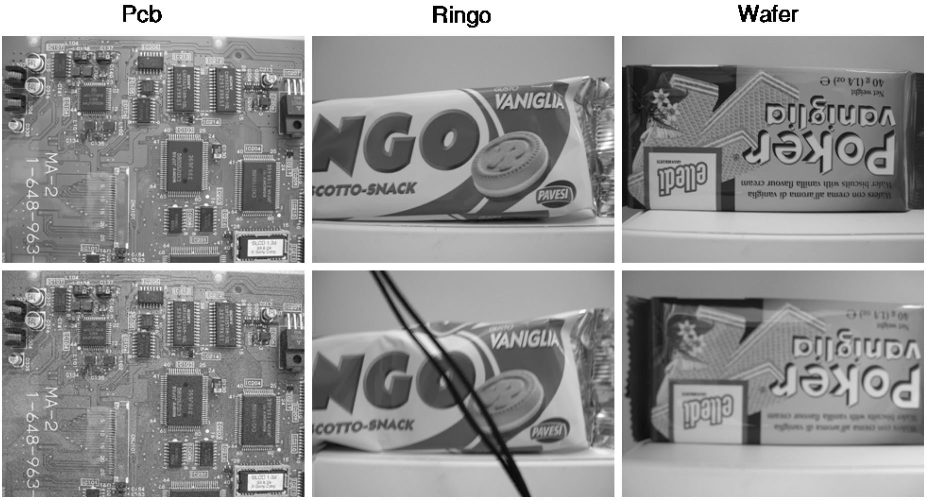
Fig. 1. Example showing the proposed ordering scheme compared to the original IDA ordering scheme on a template partitioned into 4 regions (i.e. $r = 4$.)

μ_t being the mean intensity value computed on the region:

$$\mu_t(x,y) = \frac{1}{M \cdot n} \cdot \sum_{i=1}^{M} \sum_{j=(t-1)\cdot n+1}^{t \cdot n} T(i,j) \tag{11}$$

The computation of (10) on all the r regions of the template allows to derive the following order rule: the first region on which the partial dissimilarity term is computed in case the first sufficient condition (7) does not hold is the one referring to the template region with the highest variance. Then, successive partial dissimilarity terms are computed following a decreasing variance order of the corresponding template region. The last partial dissimilarity term, that completes the computation of the dissimilarity term, is computed on the template region yielding the lowest variance. Fig. 1 shows an example that highlights the different region ordering between the original IDA algorithm and the proposed approach.

Overall, since the variance of the template regions can be computed offline, the only computational overhead introduced at runtime by the proposed scheme compared to the original IDA approach is given by the access to the index vector that stores the partition ordering for the template being used. This small overhead has been experimentally demonstrated to be negligible, as it will also evident from the results presented in the next Section. Furthermore, it is worth pointing out that the proposed scheme could be extended to all template matching and block matching methods that include a successive refinement of the bounding term, e.g. as it is the case of [4], [7], [9].



Pcb

Ringo

Wafer

Fig. 2. The dataset used for experimental results. Upper row: images where the templates are extracted from. Lower row: reference images where the templates have to be located.

4 Experimental Results

This section presents experimental results aimed at assessing the capabilities of the proposed ordering scheme to speed up the IDA technique. As shown in Fig. 2, the dataset for testing is composed by three images, where templates are extracted from (upper row in the Figure), and three reference images where the extracted templates have to be located (lower row in the Figure). All reference

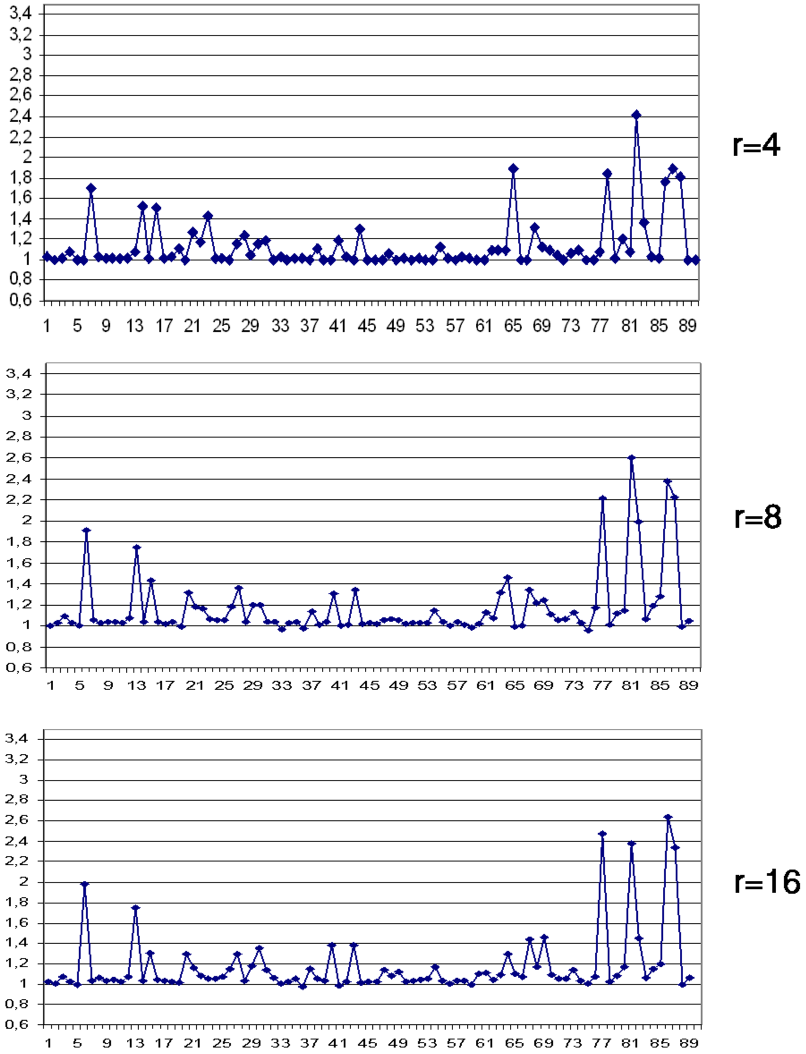


Fig. 3. Comparison between the speed-ups yielded by the proposed ordering strategy and those yielded by the conventional IDA approach (forward substitution) with different values of r. Numbers on the horizontal axis identify different template matching instances.

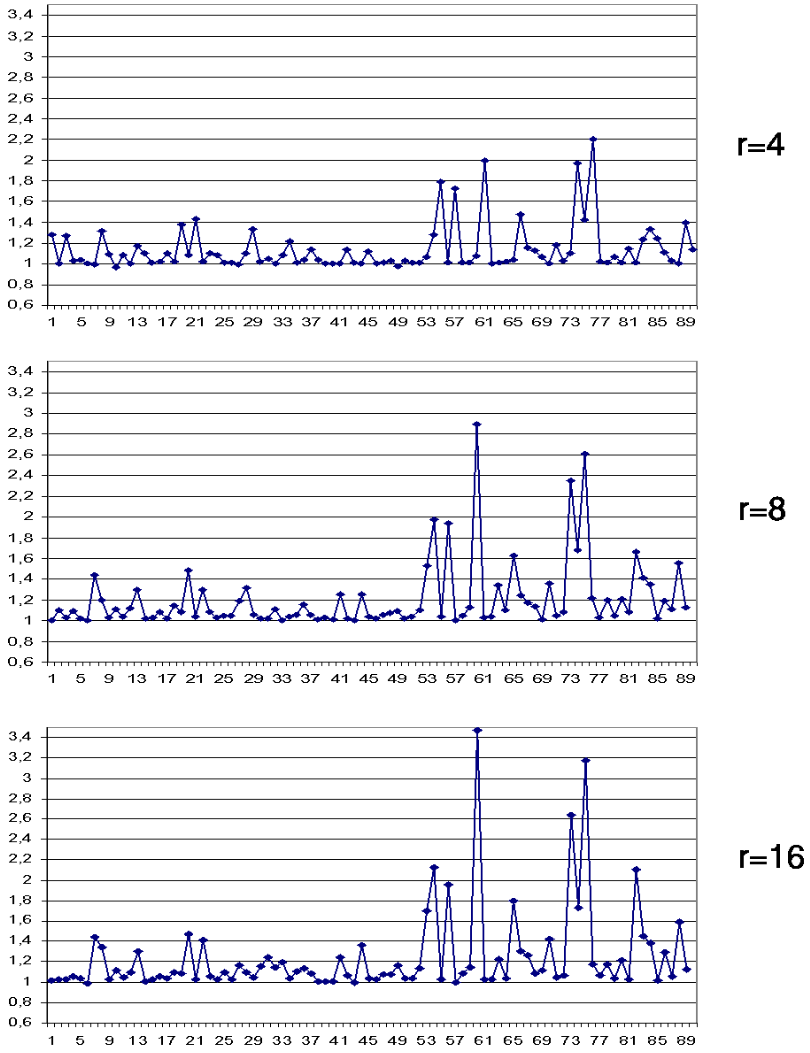


Fig. 4. Comparison between the speed-ups yielded by the proposed ordering strategy and those yielded by the conventional IDA approach (backward substitution) with different values of r. Numbers on the horizontal axis identify different template matching instances.

images are sized 640 × 480 while all extracted templates are 64 × 64. As it can be seen from the Figure, distortions between the templates and the image come from different sources such as artificial noise (*Pcb*), different view point and image defocus (*Ringo*), different view point and occlusions (*Wafer*). A total of 30 templates are randomly extracted from each image, using a fixed threshold on the variance of the intensities to reject uniform templates: hence, overall a total

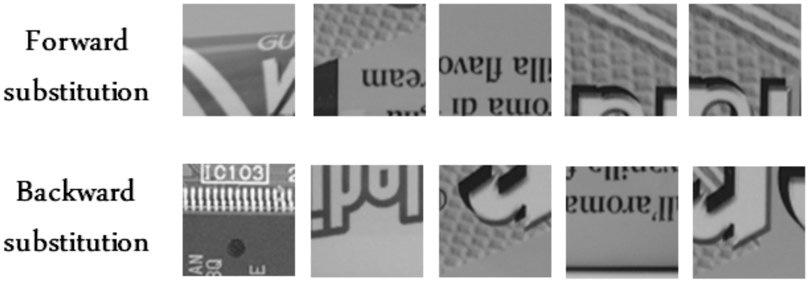


Fig. 5. Templates yielding the highest speedups between IDA with the proposed ordering scheme and the standard IDA algorithm in the forward and backward substitution cases

of 90 template matching instances are tested for each comparison. All execution times were measured on a PC with a 2.0 GHz P4 CPU. All template matching algorithms are based on the SSD measure.

Figure 3 shows the ratio between the speedups with respect to the Full Search algorithm yielded by IDA with the proposed partition ordering and those yielded by IDA with the conventional *forward* substitution (i.e. partial distances are computed starting from the first partition to the last one) for the typical values of the partition parameter r (i.e. 4, 8, 16) suggested in [2]. Analogously, Figure 4 shows the ratio between the speedups yielded by the proposed ordering strategy and those yielded by IDA deploying a *backward* substitution (from the last part to the first one) for the same values of the partition parameter r as in Fig. 3.

As it can be seen from the two Figures, the proposed ordering scheme is effective in improving the efficiency of the IDA algorithm with respect to a naive approach (i.e. forward or backward substitution). In particular, it is important to note that on the considered dataset the reported speed-ups are never significantly lower than 1 (the minimum speed-up being 0.96), while they range up to 3.4. In addition, Table 1 reports the mean speed-ups yielded by IDA deploying the proposed ordering scheme compared to the Full Search algorithm, using the same values of parameter r as in the Figures 3, 4, that demonstrates the notable efficiency of the overall technique.

To complete our experimental analysis, we also show, in Fig. 5, the 10 templates yielding the highest speedups in the results reported in Figures 3 and 4 (i.e., 5 for the forward substitution case and 5 for the backward substitution case). It is interesting to note that, in the first case, these templates show

Table 1. Speed-ups yielded by IDA deploying the proposed scheme vs. Full-Search

Dataset	r=4	r=8	r=16
Pcb	18.3	27.4	31.6
Ringo	69.0	84.8	76.9
Wafer	30.0	49.0	48.2

low-textured areas in the upper areas, with most of the signal concentrated in the middle-low part. This confirms the intuition that starting to compute the partial dissimilarity terms with the forward substitution method yields less effective sufficient conditions compared to the proposed ordering approach, thus the majority of non-optimal candidates requires the computation of more partial dissimilarity terms, that render the whole technique slower. An analogous situation occurs in the backward substitution case, where this time the low-textured regions are those in the lower part of the templates.

5 Conclusions

This paper presented an ordering method for the partitioning scheme deployed by an exhaustive template matching technique known as IDA. The experimental results presented in this paper allow us to conclude that the proposed scheme is an effective way to improve the capabilities of the IDA algorithm, since it is able to achieve notable speed-ups compared to the naive scheme embedded in the original IDA algorithm, and in the worst case it performs as well as the original IDA algorithm. We look forward to applying the proposed approach to other methods such as PDE or [7].

References

1. Li, W., Salari, E.: Successive elimination algorithm for motion estimation. IEEE Trans. on Image Processing 4(1), 105–107 (1995)
2. Tombari, F., Mattoccia, S., Di Stefano, L.: Full search-equivalent pattern matching with Incremental Dissimilarity Approximations. IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI) 31(1), 129–141 (2009)
3. Mahmood, A., Khan, S.: Early termination algorithms for correlation coefficient based block matching. In: Proc. Int. Conf. on Image Processing (ICIP 2007), vol. 2, pp. 469–472 (2007)
4. Wei, S.D., Lai, S.H.: Efficient Normalized Cross Correlation Based on Adaptive Multilevel Successive Elimination. In: Yagi, Y., Kang, S.B., Kweon, I.S., Zha, H. (eds.) ACCV 2007, Part I. LNCS, vol. 4843, pp. 638–646. Springer, Heidelberg (2007)
5. Pan, W.H., Wei, S.D., Lai, S.H.: Efficient NCC-Based Image Matching in Walsh-Hadamard Domain. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part III. LNCS, vol. 5304, pp. 468–480. Springer, Heidelberg (2008)
6. Alkhansari, M.G.: A fast globally optimal algorithm for template matching using low-resolution pruning. IEEE Trans. Image Processing 10(4), 526–533 (2001)
7. Gao, X.Q., Duanmu, C.J., Zou, C.R.: A multilevel successive elimination algorithm for block matching motion estimation. IEEE Trans. Image Processing 9(3), 501–504 (2000)
8. Hel-Or, Y., Hel-Or, H.: Real-time pattern matching using projection kernels. IEEE Trans. Pattern Analysis and Machine Intelligence 27(9), 1430–1445 (2005)
9. Bei, C.D., Gray, R.M.: An improvement of the minimum distortion encoding algorithm for vector quantization. IEEE Trans. on Communication 33, 1132–1133 (1985)
10. Montrucchio, B., Quaglia, D.: New Sorting-Based Lossless Motion Estimation Algorithms and a Partial Distortion Elimination Performance Analysis. IEEE Trans. on Circuits and Systems for Video Technology 15(2) (2005)