Ordering Pixels for Fast ZNCC Template Matching

Yiping Shen, Shuxiao Li, Yi Song, Hongxing Chang
Institute of Automation Chinese Academy of Sciences
Beijing, China
Emails: {yiping.shen, shuxiao.li, yi.song, hongxing.chang}@ia.ac.cn

***Abstract*—Basic-mode partial correlation elimination (BMP) has received popularity in recent years for fast template matching based on Zero-mean Normalized Cross Correlation (ZNCC). BMP is a full search equivalent template matching method, which examines pixel by pixel, and eliminates the rest of the computation as soon as it finds out that the current location is not likely to be the best match. In this paper, we reveal that some pixels have more contributions for matching than others. Based on this observation, an optimal ordering method is proposed to examine the pixels to accelerate the computation of BMP, called Ordering BMP (OBMP) for short. To further accelerate OBMP, we present a near-optimal ordering method which is solely based on the template image, thus can be generated offline. Extensive experiments show that the optimal or the near-optimal OBMP is on average 2.36 times faster than BMP for a template with 50×50 pixels.**

Keywords—template matching; ZNCC; BMP; OBMP

I. INTRODUCTION

Template matching, also called pattern matching, aims to search every equal-size sub-window (referred as candidate or search location in this paper) in a larger image (referred as the search image) to locate the best matching which maximizes a score function of measuring the similarity between the template and the candidate. Zero-mean normalized cross correlation (ZNCC) [1] is widely employed as a score function to measure the degree of similarity between the template and the candidate due to its robustness to linear illumination variations. Although there are other measures of score function, such as the sum of absolute differences (SAD) [2], the sum of squared differences (SSD) [2], image hamming distance family (IHDF) [3] and asymmetric correlation (ASC) [4], we preliminarily focus on ZNCC in this paper.

Unfortunately, ZNCC encounters criticisms for its high computational complexity. Researchers have investigated different methods to accelerate the computation. The earliest work dates back to [1], in which the author proposed to use the integral image [5] and fast Fourier transform (FFT) for fast template matching. Nevertheless, the computational advantage of FFT decreases with the size of the template, rendering it inefficient for small size template.

On the other hand, considering that the search space of template matching is incredible huge, research has been dedicated to algorithms that eliminate or early prune the computation of the score function at mismatching candidates. These algorithms include complete elimination algorithms, which skip a search location based on a bounding computation, e.g. [6], and partial elimination algorithms, which prune the computation as soon as it finds out that this location is a mismatching candidate, e.g.[7]. As an example, Stefano and Mattoccia [8] presented to utilize Jensen's inequality [9] to derive an upper-bounding function for normalized cross correlation (NCC) template matching, which incorporated partial information from the actual cross correlation function and can be calculated very efficiently. Later, they generalized and extended this concept of upper-bounding function to ZNCC and derived two upper-bounding functions based on the Cauchy-Schwarz inequality [6]. This algorithm is referred as zero-mean enhanced bounded correlation (ZEBC) hereafter.

Afterwards, Mahmood and Khan [10] put forward an elimination algorithm by exploiting transitivity of correlation and obtained an order of magnitude faster than ZEBC. However, its performance depended heavily upon the magnitude of the autocorrelation. In more recent studies they presented two partial correlation elimination (PCE) algorithms based on a monotonic formulation of ZNCC, namely basic-mode PCE (BMP) and extended-mode PCE (EMP) [7]. The former was efficient on small templates while the latter was efficient on medium and large templates. In [10], PCE was demonstrated to outperform other existing fast full search equivalent methods, such as ZEBC.

BMP examined pixel by pixel, and the partial similarity monotonically decreased from 1.0 to the exact value. Elimination took place as soon as the partial similarity went beneath a given maximum. The maximum can be obtained in a previous examination or initially defined. The less the number of pixels examined, the faster the algorithm. We reveal that some pixels in the template have more contributions for matching than others. If these pixels are examined first, chance stands that BMP can eliminate more mismatching candidates after dealing with a small set of pixels, thus further speeds up the computation.

In this paper, an optimal ordering method is proposed to examine the pixels to accelerate the computation of BMP, called Ordering BMP (OBMP) for short. To further accelerate OBMP, we present a near-optimal ordering method which is solely based on the template image, thus can be generated offline. Experiments show that the propose method is efficient.

II. BASIC-MODEL PCE ALGORITHM

In this section, ZNCC is briefly introduced, followed by the description of BMP and the amount of computations required by BMP.

A. ZNCC Template Matching

Given a template T of size $m\times n$ pixels and a search image I of size $p\times q$ pixels, where $m \le p$ and $n \le q$, the ZNCC value between the template T and the search image I at location (x,y) can be written as:

978-1-4799-2827-9/13/$31.00

$$ZNCC(x,y) = \frac{\sum_{j=1}^{m}\sum_{i=1}^{n}[I(x+i,y+j)-\mu(x,y)]\cdot[T(i,j)-\mu(T)]}{\sqrt{\sum_{j=1}^{m}\sum_{i=1}^{n}[I(x+i,y+j)-\mu(x,y)]^2}\cdot\sqrt{\sum_{j=1}^{m}\sum_{i=1}^{n}[T(i,j)-\mu(T)]^2}} \quad (1)$$

where $\mu(T)$ and $\mu(x,y)$ are the means of the template and the candidate at (x,y), respectively.

B. Basic-Mode Partial Correlation Elimination Algorithm

To early prune the computation at a mismatching candidate, the score function must exhibit a monotonic behavior, that is, as the number of processed pixels increases, the value of the score function decreases. Rewrite (1) as a dot product of two normalized vectors:

$$ZNCC(x,y) = \sum_{j=1}^{m}\sum_{i=1}^{n}\{[\delta_{x,y}(i,j)/\sigma_{x,y}]\cdot[\delta_T(i,j)/\sigma_T]\} \quad (2)$$

where $\delta_{x,y}(i,j) = I(x+i, y+j)-\mu(x,y)$, $\delta_T(i,j) = T(i,j)-\mu(T)$, $\sigma_{x,y}$ and σ_T denote the denominator terms in (1). As we can see, (2) shows no monotonic property because the addend may be positive, negative or zero depending on the sign of the product of $\delta_{x,y}(i, j)$ and $\delta_T(i, j)$ ($\sigma_{x,y}$ and σ_T are always positive). Observing that the norm of each vector in (2) is equal to unity, we have:

$$\sum_{j=1}^{m}\sum_{i=1}^{n}[\delta_T(i,j)/\sigma_T]^2 + \sum_{j=1}^{m}\sum_{i=1}^{n}[\delta_{x,y}(i,j)/\sigma_{x,y}]^2 = 2 \quad (3)$$

Combining (2) and (3) and rearranging, a monotonic decreasing form of score function can be derived by:

$$ZNCC(x,y) = 1 - \sum_{j=1}^{m}\sum_{i=1}^{n}(\delta_{x,y}(i,j)/\sigma_{x,y} - \delta_T(i,j)/\sigma_T)^2/2 \quad (4)$$

Let $\xi_{x,y}(u,v)$ denote the partial similarity at location (x,y) after processing u rows and v columns:

$$\xi_{x,y}(u,v) = 1 - \sum_{j=1}^{u}\sum_{i=1}^{v}(\delta_{x,y}(i,j)/\sigma_{x,y} - \delta_T(i,j)/\sigma_T)^2/2 \quad (5)$$

Using (5), the partial similarity decreases from 1.0 to the exact value $ZNCC(x,y)$ as consecutive pixels are processed, which enables pruning the computation as soon as it finds out that this location is a mismatch. For example, if $\xi_{x,y}(u_0,v_0)$ is below the current known maximum ρ_{th}, the exact value $ZNCC(x,y)$ is definitely below ρ_{th}, and further computation at this location is unnecessary thus can be pruned. The comparison of $\xi_{x,y}(u_0,v_0)$ with ρ_{th} is called an elimination test. BMP does not take an elimination test after processing each pixel, instead it performs the test after processing some initial number of pixels. The number of processed pixels in each elimination test can be set according to the intersections of the average growth curves of the score function and ρ_{th}, refer to [7] for more details.

C. The Amount of Computations for BMP

Assuming that τ elimination tests are performed, the number of processed pixels in the ith test is α_i, and β_i candidates are skipped after the ith test, i=1, 2, ⋯, τ. According to [7], the amount of computations is:

$$\psi = \sum_{k=1}^{\tau}\beta_k(17\sum_{j=1}^{k}\alpha_j + 14) \quad (6)$$

where $\sum_{k=1}^{\tau}\beta_k = m\cdot n$. Mahmood and Khan [7] presented an efficient testing scheme to determine τ and α_i, and processed pixels in a consecutive scanning ordering (i.e. from left to right and from top to bottom). We do not explore the performance of the testing scheme. Therefore, we set $\alpha_i = mn/\tau$, i=1, 2, …, τ in this paper. BMP obtains its computational efficiency by skipping as many candidates as possible in the first few tests (i.e., large β_i for a small i). In the next section, we show that there is an optimal ordering to examine the pixels, using which we can obtain a large β_i for a small i (e.g., i=1), thus leading to a smaller amount of computations.

III. AN OPTIMAL/NEAR-OPTIMAL ORDERING FOR EXAMINING PIXELS

In this section, we first describe what is the optimal ordering and how to reduce the computation by using the optimal ordering. Then, we introduce how to quickly generate a near-optimal ordering for fast ZNCC template matching.

A. Reducing the Computations by Using an Optimal Ordering

Denote the addend in (5) by:

$$sd_{x,y}(i,j) = (\delta_{x,y}(i,j)/\sigma_{x,y} - \delta_T(i,j)/\sigma_T)^2 \quad (7)$$

and rank the squared difference of pixel (i,j) denoted by $sd_{x,y}(i,j)$ in a descending way as:

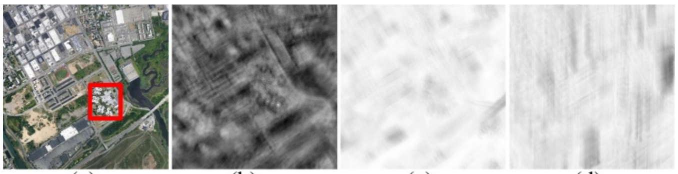
$$\begin{aligned}&sd_{x,y}(i1,j1) \ge sd_{x,y}(i2,j2) \ge \cdots \ge sd_{x,y}(ik,jk) \ge \cdots \ge sd_{x,y}(iL,jL),\\ &ik \in [1,m], \quad jk \in [1,n], \quad L = mn\end{aligned} \quad (8)$$

Combining (7) and (8), (5) can be rearranged as:

$$\xi_{x,y}(t) = 1 - \sum_{k=1}^{t} sd_{x,y}(ik,jk)/2, \quad 0 \le t \le L \quad (9)$$

For a mismatching candidate, the partial similarity definitely decreases faster below ρ_{th} by using (9) than (5). However, it is impractical to rank $sd_{x,y}(i,j)$ for each candidate. Fortunately, we observe that some pixels contribute large differences at most mismatching candidates and provide small differences at promising candidates (i.e. candidates with high similarities). We call these pixels "discriminating pixels". Since majority candidates are dissimilar with the template, whose similarities are quite small, a small set of discriminating pixels may provide enough information to eliminate a large amount of these candidates. If we sort the pixels according to their average squared differences in all candidates in a descending way, we have an ordering to process these pixels and name it "the optimal ordering". This "optimal" is statistical, meaning that the optimal ordering may not strictly optimal for each candidate; in contrast, it is "optimal" for consideration of all candidates. In this way pixels with larger average squared differences are encouraged to be dealt with first, which leads to the fastest decreasing of the partial similarity at mismatching candidates. Comparing with the consecutive scanning order, the optimal ordering reduce the amount of computations by obtaining a larger β_i for a small i .

An example is illustrated in Fig. 1. When searching for a template of size 87×94 in an image of size 480×480, after processing the same amount of pixels (here, 100), the partial similarity is quite different in three cases: (1) the first 100 pixels in the optimal ordering, (2) the last 100 pixels in the optimal ordering, (3) the first 100 pixel in the consecutive scanning ordering. On average, the partial similarities at most mismatching candidates are smallest (i.e. darkest) in case (1), followed by those in case (3), and then by those in case (2).



(a) (b) (c) (d)

Fig. 1. Comparison of the partial similarity. (a) The search image (480×480 in pixels) and the template (87×94 in pixels) in the red rectangle. The partial similarities after processing the first 100 pixels in the optimal ordering (b), the last 100 pixels in the optimal ordering (c), the first 100 pixel in the consecutive scanning ordering (d). The partial similarities at mismatching candidates in (b) are smallest (i.e. darkest). Using a threshold ρ_{th}=0.9, 28.5% candidates are eliminated in (b), while 0.5% are eliminated in (d).

Using a threshold ρ_{th}=0.9, 28.5% candidates are eliminated in Fig. 1(b), while 0.5% are eliminated in Fig. 1(d) after processing 100 pixels.

To further explain the effectiveness of the optimal ordering, we divide the range of ZNCC [-1, 1] into 11 bins equally and partition the candidates of the search image in Fig. 1(a) into 11 groups according to their exact ZNCC values. The average growth curves of each group using a consecutive scanning ordering and the optimal ordering are depicted in Fig. 2. As we can see, the slopes of the average growth curves using the optimal ordering are steeper at lower percentage of processed pixels, while those of the average growth curves using the consecutive ordering are random depending on the data. More pixels are needed to be processed to prune the same number of candidates using the consecutive ordering. Take an example, to prune the candidates in Bin 9 (the blue curves in Fig. 2), the percentages of processed pixels for the consecutive ordering and the optimal ordering are 20% and 7%, respectively. In other words, after processing the same number of pixels, more candidates are skipped by using the optimal ordering.

B. Quickly Generating a Near-Optimal Ordering

In the previous subsection, we describe what is the optimal ordering and how it helps to reduce the computations. However, in real-world applications of template matching, it is impossible to obtain the optimal ordering since the search image is unknown beforehand. In the following, we will introduce an algorithm that quickly generates a near-optimal ordering, which is solely based on the template image.

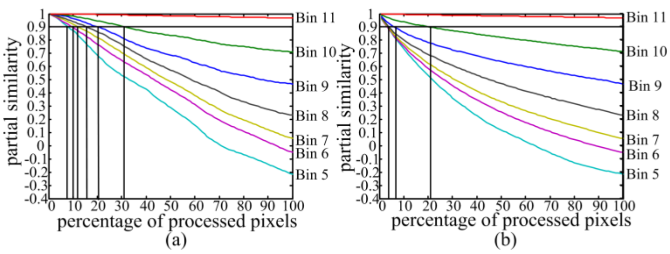


Fig. 2. The range of ZNCC [-1, 1] is divided into 11 bins equally. The average growth curves of partial similarity in each bin are plotted (a) using the consecutive ordering, (b) using the optimal ordering. The intersection of ρ_{th} and each average growth curve indicates the percentage of processed pixels when the partial similarity goes beneath ρ_{th}. For ρ_{th}=0.9, to eliminate candidates with its ZNCC in Bin 5, 6, 7, 8, 9, 10, on average 8%, 10%, 13%, 16%, 20%, 32% of pixels are needed to be processed by using a consecutive scanning ordering, while the percentage of processed pixels are 4%, 4%, 4%, 4%, 7%, 21% on average by using the optimal ordering.

For a pixel (i,j) in the template, its normalized gray is $z(i,j) = [T(i,j)-\mu(T)]/\sigma_T$. Assuming that the normalized gray of its corresponding pixel in the candidate is a random variable with its probability density function $p(x)$, then the average of squared differences (ASD) for (i,j) can be given by:

$$ASD(z) = \int_x (z-x)^2 p(x)dx \tag{10}$$

Note that $p(x)$ can be evaluated empirically using a representative sample. Take an example, we assume that case 1: x obeys the uniform distribution $x \sim U(a,b)$, a<b, (i.e. $p(x) = 1/(b-a)$), case 2: x obeys the Gaussian distribution $x \sim N(\mu,\sigma^2)$. Then the integral result of (10) can be obtained by:

$$ASD(z) = \begin{cases} z^2 - (a+b)z + (a^2+ab+b^2)/3, & x \sim U(a,b) \\ z^2 - 2\mu z + \sigma^2 + \mu^2, & x \sim N(\mu,\sigma^2) \end{cases} \tag{11}$$

$ASD(z)$ reaches its minimum at $z0=E(x)$, where $E(x)$ is the mean of x, and monotonically increases with $|z-z0|$ increasing, where $|.|$ denotes the absolute value. For a normalized candidate, we have $z0=0$. Therefore, a conclusion can be inferred that pixels which have larger $|z(i,j)|$ always show a larger ASD. Since $z(i,j) = [T(i,j)-\mu(T)]/\sigma_T$, the conclusion can be expressed in another way that pixels that own a larger $|T(i,j)-\mu(T)|$ usually indicate a larger ASD.

The histogram of the normalized template (see in Fig.1(a)) is depicted in Fig. 3(a). We considered the first 1200 pixels in the optimal ordering, and partitioned them into 12 parts based on their ordering, each part including 100 pixels, then draw the histogram of the normalized gray in each part as illustrated in Fig. 3(b). The pixels in the front of the optimal ordering show a large absolute normalized gray, which convinces our conclusion mentioned previously.

In all, a near-optimal ordering can be generated in this way: first the mean of the template $\mu(T)$ is computed, then for each pixel in the template, its absolute difference to $\mu(T)$, i.e. $|T(i,j)-\mu(T)|$ is obtained, which is sorted descendingly to yield the near-optimal ordering.

(a)

(b)

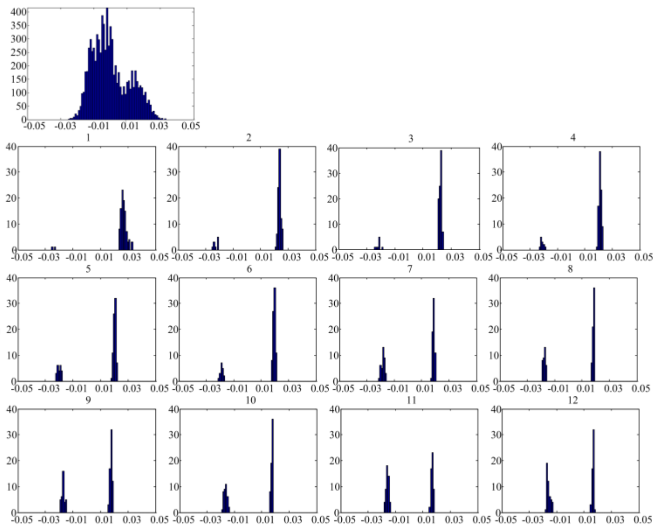


Fig. 3. (a) The histogram of the normalized template in Fig. 1(a). (b) The histograms of the first 100 pixels, second 100 pixels, …, twelve 100 pixels in the optimal ordering. The first 1200 pixels in the optimal ordering show a large absolute normalized gray, implying that pixels having a large absolute normalize gray may have a strong discriminative power.

IV. EXPERIMENTS

Experiments are performed to evaluate the effectiveness of the proposed method. All algorithms are implemented in C and run on an Intel Core2 CPU 2.00 GHz/2G RAM computer. We use the term "speed-up" as [10] to measure the performance of the proposed method. For example, the speed-up of A over B is the ratio of the runtime required by B with respect to the runtime required by A. OBMP, NOBMP, CBMP denote BMP using the optimal ordering, the near-optimal ordering, and the consecutive scanning ordering, respectively, hereafter. Note that the performance of these three methods are the same, and are all full search equivalent methods.

A typical example is shown in Fig. 1(a), in which a template of size 87×94 was searched in an image of size 480×480. We compared the performances of OBMP, NOBMP and CBMP. Results are showed in Fig. 4. When varying τ from 10 to 100, the runtime required these three algorithms decreased as illustrated in Fig. 4(a). The runtime of NOBMP was very close to that of OBMP. One of the reasons is that the elimination test takes place after processing a number of pixels, making the average decreasing of the partial similarity for NOBMP closing to OBMP. In comparison, the runtime of CBMP was more than that of OBMP or NOBMP. The speep-ups of NOBMP over CBMP increased from 1.3 at τ=10 to 3.1 at τ=100. When ρ_{th} increased, the runtime of these three algorithms decreased, and the speed-ups of NOBMP over CBMP increased from 1.4 to 3.4 as showed in Fig. 4(b). When τ=100, ρ_{th}=0.90, the percentage of eliminated candidates after each eliminate test are showed in Fig. 4(c). NOBMP can reject 99.7% mismatching locations after processing only 5% pixels, while CBMP needed to process 19% pixels to reject 99.7% mismatching locations. In all, the proposed method can reject more mismatching locations after processing the same number of pixels, thus leading to a more efficient BMP.

To further demonstrate the effectiveness of our method, we collected 6 images with size 600×1024 pixels from Quickbird as illustrated in Fig. 5. Five different template sizes {10×10, 20×20, 30×30, 40×40, 50×50} were used, and 180 templates for each size. We set ρ_{th} = 0.90, and τ equaled to the row of the template, meaning that the elimination test occurred at the end of each row for CBMP. The speed-ups over CBMP are showed in Tab. 1. The speed-ups of OBMP and NOBMP over CBMP increased with the size of the template, which demonstrated the effectiveness of the ordering, especially for larger template size. Although the speed-up varied, the average speed-ups of NOBMP for these five templates achieved 1.104, 1.331, 1.674, 2.095, 2.360, respectively.

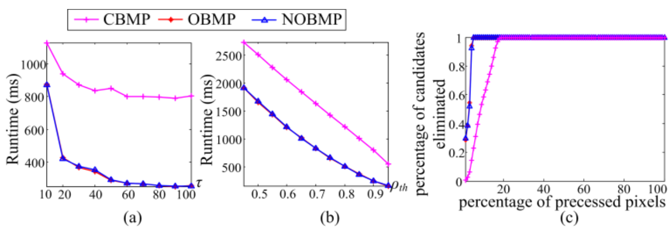


Fig. 4. (a) The runtime vs. the number of elimination tests τ. (b) The runtime vs. the threshold ρ_{th}. (c) The percentage of eliminated candidates vs. the percentage of processed pixels when ρ_{th} =0.90, τ=100.

Fig. 5. Images from Quickbird.

TABLE I. THE SPEED-UPS OVER CBMP

Speed-up	10×10	20×20	30×30	40×40	50×50
Average (OBMP)	1.108	1.331	1.679	2.094	2.359
Average (NOBMP)	1.104	1.331	1.674	2.095	2.360
Maximum (OBMP)	1.650	2.450	2.648	5.530	7.447
Maximum (NOBMP)	1.581	2.515	2.629	5.492	7.440

V. CONCLUSION

In this paper, an optimal ordering to examine pixels was proposed to accelerate the computation of BMP. We also presented a near-optimal ordering method which is solely based on the template image, thus can be generated offline. Experiments showed that the proposed OBMP and NOBMP obtained a faster speed than CBMP, and the speed-ups of OBMP and NOBMP over CBMP increased from 1.1 times for size of 10×10 to 2.36 times for size of 50×50 in a search image of size 600×1024 pixels.

However, BMP is less efficient than EMP on medium and large templates. Since the implementations of EMP are processed in rows, efforts will be given to ordering rows to accelerate the computation of EMP in the future.

ACKNOWLEDGMENT

This work is supported by National Natural Science Foundation of China (NSFC) under grants no. 61005028 and no.61175032.

REFERENCES

[1] J. P. Lewis, "Fast normalized cross-correlation, Vision Interface", 10(1), pp. 120-123, 1995

[2] W., Ouyang, F., Tombari, S., Mattoccia, L.Di. Stefano, W., Cham, "Performance evaluation of full search equivalent pattern matching algorithms", IEEE Trans. PAMI, 34(1), pp. 127-143, 2012

[3] O. Pele and M. Werman. "Robust real-time pattern matching using Bayesian sequential hypothesis testing". IEEE Trans. PAMI, 30(8), pp. 1427-1443, 2008

[4] E. Elboher and M. Werman, "Asymmetric correlation: a noise robust similarity measure for template matching", IEEE Trans. Image Processing, 22(8), pp. 3062-3073, 2013

[5] P., Viola, M., Jones, "Robust real-time face detection", International Journal of Computer Vision, 57(2), pp. 137–154, 2004

[6] S. Mattoccia, F. Tombari, and L. D. Stefano, "Reliable rejection of mismatching candidates for efficient ZNCC template matching", in Proc. 15th ICIP, pp. 849-852, 2008

[7] A., Mahmood, S., Khan, "Correlation-coefficient-based fast template matching through partial elimination", IEEE Trans. Image Processing, 21(4), pp. 2099-2108, 2012

[8] L. D. Stefano and S. Mattoccia "Fast template matching using bounded partial correlation", Machine Vision and Applications, 13(4), pp. 213-221, 2003

[9] W. Rudin, Real and complex analysis. McGraw-Hill, New York, 1996

[10] A., Mahmood, S., Khan, "Exploiting transitivity of correlation for fast template matching", IEEE Trans. Image Processing, 19(8), pp. 2190-2200, 2010