Exploiting Transitivity of Correlation for Fast Template Matching

Arif Mahmood, Sohaib Khan

***Abstract*—Elimination Algorithms are often used in template matching to provide a significant speed-up by skipping portions of the computation while guaranteeing the same best-match location as exhaustive search. In this work, we develop elimination algorithms for correlation-based match measures by exploiting the transitivity of correlation. We show that transitive bounds can result in a high computational speed-up if strong autocorrelation is present in the dataset. Generally strong intra-reference local autocorrelation is found in natural images, strong inter-reference autocorrelation is found if objects are to be tracked across consecutive video frames and strong inter-template autocorrelation is found if consecutive video frames are to be matched with a reference image. For each of these cases, the transitive bounds can be adapted to result in an efficient elimination algorithm. The proposed elimination algorithms are exact, that is, they guarantee to yield the same peak location as exhaustive search over the entire solution space. While the speed-up obtained is data dependent, we show empirical results of up to an order of magnitude faster computation as compared to the currently used efficient algorithms on a variety of datasets.**

I. Introduction

Template matching is the process of evaluating the similarity of a template image at each search location of a larger reference image, to identify the best-match location. If the search for the best-match location is done exhaustively over the entire search space, the process is computationally expensive. To reduce the computational cost while maintaining the exhaustive equivalent accuracy, elimination algorithms are often used, which may be categorized into two types: complete elimination algorithms [1], [2], [3], [4] and partial elimination algorithms [5], [6]. In complete elimination algorithms, the actual similarity measure computation may be skipped completely if an alternate suitability test indicates that the current location cannot be the best-match location. In case of partial elimination algorithms, the similarity measure is partially evaluated at each search location but may be terminated prematurely if the result of the partial computation establishes the unsuitability of the current location as the best-match location. In either case, by skipping computations, elimination algorithms reduce the computational complexity while guaranteeing that the result of the best-match location will not be compromised.

Elimination Algorithms have been well investigated for match measures such as Sum of Squared Differences (SSD) and Sum of Absolute Differences (SAD) (see for example, [1], [2], [3], [4], [5], [6]). However, for correlation-based measures, such as cross-correlation, Normalized Cross Correlation (NCC) and correlation-coefficient, only limited investigations of elimination algorithms are found in literature [7], [8]. This is because of the fact that, the elimination strategies developed for distance measures, are not directly applicable to correlation measures. As a consequence, when computational efficiency is of primary importance, correlation measures are less frequently used. This is despite the fact that correlation-coefficient, being invariant to brightness and contrast variations, is more robust than SAD or SSD.

In this paper, we propose complete elimination algorithms for correlation-based similarity measures including cross-correlation, NCC and correlation-coefficient. The common basis for each of the proposed elimination algorithm is the notion of the transitivity of correlation. That is, if correlation between image blocks r_1 and r_2 is known, and that between r_2 and r_3 is also known, what are the bounds on the correlation between image blocks r_1 and r_3? We present the derivation of these bounds and show how these bounds can be exploited, to yield what we term as *Transitive Elimination Algorithms*.

In transitive elimination algorithms, the required matching computations are divided into two types: *Bounding Correlations* (for example correlation between r_1 and r_2 or that between r_2 and r_3) and *Bounded Correlations* (for example correlation between r_1 and r_3). Bounding correlations are only a small fraction of the total computations, and have to be computed in their entirety. However, bounded correlations, which form the bulk of the computation, do not have to be always computed. Most of the bounded correlations can be skipped by using the transitive elimination algorithms.

In order to get good elimination performance, transitive bounds should be tight enough. We find that, tight bounds require at least one of the two bounding correlations to be of large magnitude. This is ensured by exploiting different forms of autocorrelation found in the dataset. Most of the template matching applications exhibit strong autocorrelation in one of the following three forms: strong intra-reference autocorrelation, strong inter-reference autocorrelation or the strong inter-template autocorrelation. To exploit each of these types, we have proposed variants of transitive elimination algorithms, all exploiting the same underlying principle.

1) *Exploiting strong intra-reference autocorrelation [9]:* Most natural images are low-frequency signals, hence exhibit high local spatial autocorrelation. Let r_1 be the template image, r_2 be a reference image block and r_3 be

Arif Mahmood is with the Punjab University College of Information Technology, Lahore, Pakistan and pursuing his Ph.D. at the Department of Computer Science, LUMS School of Science and Engineering, Lahore, Pakistan (email: arifm@pucit.edu.pk). Sohaib Khan is with the School of Science and Engineering, Lahore University of Management Sciences, Lahore, Pakistan (email: sohaib@lums.edu.pk).

Manuscript received 05-Sep-2008; re-submitted 14-May-2009; revised 11-Oct-2009; again revised 12-Jan-2010; accepted 04-Mar-2010. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Dan Schonfeld.

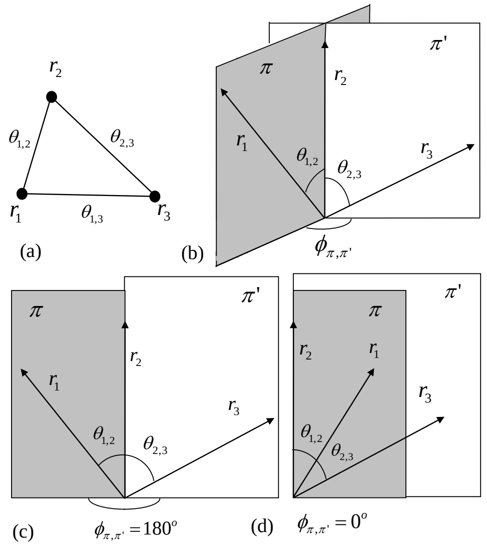


Fig. 1. Triangular inequality for the angular distance measure: **(a)** Image blocks r_1, r_2 and r_3 represented as vertices and the angular distance between them is shown as edges of a triangle. **(b)** $\theta_{1,3}$ depends upon the angle between planes π and π', $\phi_{\pi,\pi'}$. **(c)-(d)** $\theta_{1,3}$ becomes maximum, $\theta_{1,2}+\theta_{2,3}$, when $\phi_{\pi,\pi'} = 180^\circ$ and becomes minimum, $|\theta_{1,2}-\theta_{2,3}|$, when $\phi_{\pi,\pi'} = 0^\circ$.

one of the spatially neighboring blocks of r_2. Since local autocorrelation of the reference image is high, r_2 is often highly correlated with each r_3 block. If these correlations are known, then correlating r_1 with r_2 yields maximum and minimum bounds upon the correlation of r_1 with each r_3. Using these bounds, unsuitable r_3 blocks may be eliminated from the search space, significantly reducing the computations without causing any degradation of accuracy.

The computation of the autocorrelation of r_2 with each of its neighbor r_3 is an algorithmic overhead but it is justified through high elimination of the subsequent computations. Moreover, we also present an efficient algorithm for the computation of local autocorrelation. As a result, this algorithmic overhead turns out to be insignificant as compared to the overall computations.

2) *Exploiting strong inter-reference autocorrelation:* Tracking an object in a surveillance video, checking for missing components on a PCB production line or object inspection over conveyor belts require one template image to be correlated across multiple reference frames. In such an application, the reference images are often highly correlated with each other, because the camera is often static, a fact which can be exploited for high elimination. Let r_1 be the template image and r_2 be a reference image block and r_3 be one of the temporal neighboring blocks, in another reference image. Since inter-reference autocorrelation will be high, correlation of r_1 with r_2 yields tight transitive bounds upon the correlation between r_1 and r_3. Those r_3 blocks for which elimination test is found to be positive may be skipped from computations without any loss of accuracy.

3) *Exploiting strong inter-template autocorrelation [10]:* Certain applications require a set of template images to be correlated with a single reference image, for example, matching an aerial video with a satellite image or exhaustive rotation-scale invariant template matching. In such cases, if the set of templates has high autocorrelation, correlation of one template with the reference image yields tight bounds upon the correlation of all other templates within the set.

The proposed algorithms are implemented in C++ and compared with current known efficient algorithms including Enhanced Bounded Correlation [8], Bounded Partial Correlation [7], SAD [1], [6], FFT based frequency domain implementation [11] and an efficient spatial domain implementation [12]. Experiments are performed on a variety of real image datasets. The exact speed-up of the proposed algorithms varies from experiment to experiment, ranging from multiple times to more than an order of magnitude.

II. Transitive Inequality for Correlation Based Similarity Measures

Let r_1 and r_2 be two image blocks, each of size $m \times n$ pixels, and $\psi_{1,2}$ be the cross-correlation between these blocks:

$$\psi_{1,2} = \sum_{i=0}^{m-1}\sum_{j=0}^{n-1} r_1(i,j) r_2(i,j). \tag{1}$$

r_1 and r_2 may also be considered as vectors in $\mathcal{R}^{m\times n}$ space. Let $\theta_{1,2}$ be the angular distance between these vectors. Using the definition of scalar product, $\theta_{1,2}$ can be related with cross-correlation, $\psi_{1,2}$:

$$\theta_{1,2} = \cos^{-1} \frac{\psi_{1,2}}{||r_1||_2 ||r_2||_2}, \tag{2}$$

where $||.||_2$ denotes the L_2 norm. The angular distance is symmetric, i.e. $\theta_{1,2} = \theta_{2,1}$, and bounded between 0° and 180°. In addition, the angular distance also follows the triangular inequality of distance measures [10], that is for three image blocks r_1, r_2 and r_3 (Figure 1):

$$\theta_{1,2} + \theta_{2,3} \geq \theta_{1,3} \geq |\theta_{1,2} - \theta_{2,3}|, \tag{3}$$

where $\theta_{1,3}$ is the angular distance between r_1, r_3 and $\theta_{2,3}$ is the angular distance between r_2, r_3. The minimum and the maximum angular distance between r_1 and r_3 occurs when r_3 lies in the same plane as r_1 and r_2 (Figure 1). Therefore the upper and lower triangular bounds are also bounded between 0° and 180° and the triangular inequality may be written as: $\min\{360^\circ - (\theta_{1,2}+\theta_{2,3}), (\theta_{1,2}+\theta_{2,3})\} \geq \theta_{1,3} \geq |\theta_{1,2} - \theta_{2,3}|$. We observe that the cosine function monotonically decreases from +1 to -1 as θ varies from 0° to 180°. Taking the cosine of the triangular inequality, we get the basic form of the transitive inequality for cross-correlation:

$$\cos(\theta_{1,2} + \theta_{2,3}) \leq \cos(\theta_{1,3}) \leq \cos(\theta_{1,2} - \theta_{2,3}). \tag{4}$$

This may be rearranged using trigonometric identities as:

$$\cos\theta_{1,2}\cos\theta_{2,3} - \sqrt{1-(\cos\theta_{1,2})^2}\sqrt{1-(\cos\theta_{2,3})^2} \leq \cos\theta_{1,3}$$
$$\leq \cos\theta_{1,2}\cos\theta_{2,3} + \sqrt{1-(\cos\theta_{1,2})^2}\sqrt{1-(\cos\theta_{2,3})^2} \tag{5}$$

Multiplying Equation (5) with $(||r_1||_2||r_2||_2)(||r_2||_2||r_3||_2)$ and simplifying using Equation (2), the transitive inequality in terms of the cross-correlation measure ψ:

$$\frac{\psi_{1,2}\psi_{2,3} + \sqrt{(||r_1||_2||r_2||_2)^2 - \psi_{1,2}^2}\sqrt{(||r_2||_2||r_3||_2)^2 - \psi_{2,3}^2}}{(||r_2||_2)^2} \le \psi_{1,3} \le \frac{\psi_{1,2}\psi_{2,3} - \sqrt{(||r_1||_2||r_2||_2)^2 - \psi_{1,2}^2}\sqrt{(||r_2||_2||r_3||_2)^2 - \psi_{2,3}^2}}{(||r_2||_2)^2} \quad (6)$$

This inequality provides transitive bounds on the cross-correlation between r_1 and r_3, if the cross-correlation between r_1 and r_2 and that between r_2 and r_3 is already known.

Cross-correlation is often used in its normalized form to remove bias towards brighter regions. Normalized cross-correlation between image blocks r_1 and r_2 is defined as:

$$\phi_{1,2} = \frac{\psi_{1,2}}{||r_1||_2||r_2||_2}, \quad (7)$$

The angular distance between two image blocks may also be written in terms of ϕ: $\theta_{1,2} = \cos^{-1}(\phi_{1,2})$. Transitive inequality given by Equation (5) gets modified for NCC as follows:

$$\phi_{1,2}\phi_{2,3} + \sqrt{1-\phi_{1,2}^2}\sqrt{1-\phi_{2,3}^2} \le \phi_{1,3} \le \phi_{1,2}\phi_{2,3} - \sqrt{1-\phi_{1,2}^2}\sqrt{1-\phi_{2,3}^2} \quad (8)$$

This inequality yields transitive bounds upon NCC between image blocks r_1 and r_3, if the NCC between r_1 and r_2 and that between r_2 and r_3 is already known.

NCC is robust to contrast variations, but it is not robust to the brightness variations. A more robust measure, invariant to any linear change in the signal, is the correlation coefficient:

$$\rho_{1,2} = \frac{\psi_{1,2} - mn\mu_1\mu_2}{||r_1-\mu_1||_2||r_2-\mu_2||_2}, \quad (9)$$

where μ_1 and μ_2 are the means of r_1 and r_2 respectively. Correlation-coefficient can also be written in terms of the angular distance as: $\rho_{1,2} = \cos(\hat{\theta}_{1,2})$, where $\hat{\theta}_{1,2}$ is the angular distance between $r_1 - \mu_1$ and $r_2 - \mu_2$. The transitive inequality in terms of $\hat{\theta}$ can be derived by following the same steps as that for θ, and yields:

$$\cos(\hat{\theta}_{1,2} + \hat{\theta}_{2,3}) \le \cos(\hat{\theta}_{1,3}) \le \cos(\hat{\theta}_{1,2} - \hat{\theta}_{2,3}). \quad (10)$$

This can be expanded to the transitive inequality for the correlation-coefficient:

$$\rho_{1,2}\rho_{2,3} + \sqrt{1-\rho_{1,2}^2}\sqrt{1-\rho_{2,3}^2} \le \rho_{1,3} \le \rho_{1,2}\rho_{2,3} - \sqrt{1-\rho_{1,2}^2}\sqrt{1-\rho_{2,3}^2}. \quad (11)$$

Transitive bounds can also be derived by exploiting the relationship between correlation and distance measures other than the angular distance. For example, we have derived transitive inequality through Euclidean distance as shown in Appendix I. However, we find that the bounds based upon angular distance are tighter than the bounds based upon Euclidean distance (see Appendices II and III for proof) and therefore more useful for elimination algorithms. In the next section, we will show how the transitive bounds for correlation measures can be exploited algorithmically, to speed-up different template matching applications.

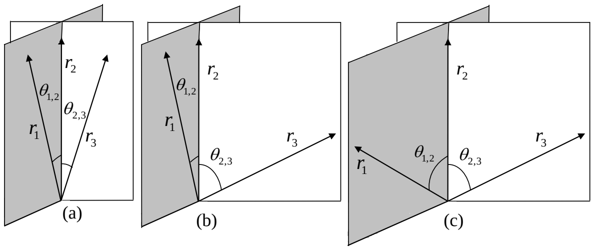


Fig. 2. Tightness of the transitive bounds: **(a)** *Case 1:* Both angles, $\theta_{1,2}$ and $\theta_{2,3}$, have small magnitude. **(b)** *Case 2:* One angle is small and the other is large. **(c)** *Case 3:* Both angles are large.

III. Transitive Elimination Algorithms

Transitive Elimination algorithms are developed to exploit the transitive bounds for fast template matching. For a particular search location, transitive bounds indicate the maximum and the minimum limits upon correlation, which can be used to discard unsuitable search locations. For example, at a specific location, if the maximum limit is less than the correlation value at some previous location, correlation computation becomes redundant and may be skipped without any loss of accuracy. As the percentage of skipped search locations increases, the template matching process accelerates accordingly.

In order to compute the transitive bounds, three transitive inequalities were presented in the last section, Equations (6), (8) and (11). In each of these inequalities, there are two *Bounding Correlations* which must be known in order to find bounds upon the third *Bounded Correlation*. For example, in Equation (11), $\rho_{1,2}$ and $\rho_{2,3}$ are the two bounding correlations which constrain the upper and the lower limits upon the bounded correlation $\rho_{1,3}$. In a template matching problem, this concept may be exploited for significant speed-up, by designing an algorithm such that bounded correlations comprise a large percentage of the total computations. Most of the bounded correlations may be skipped if tight transitive bounds are available.

The tightness of the transitive bounds depends upon the magnitude of the two bounding correlations, and requires the upper bound to be low and the lower bound to be high. This dependency may be more clearly understood by considering transitive inequalities in terms of angular distances as given by Equations (4) or (10). In these equations, a tight upper bound means $\cos(\theta_{1,2} - \theta_{2,3})$ having a value significantly lesser than +1, which implies $|\theta_{1,2} - \theta_{2,3}|$ has a value significantly larger than $0°$. Similarly, the lower bound will be tight if $\cos(\theta_{1,2} + \theta_{2,3})$ has a higher value, which implies that $\theta_{1,2} + \theta_{2,3}$ should have a value close to $0°$. Considering different ranges of values which $\theta_{1,2}$ and $\theta_{2,3}$ may assume, three possible cases are shown in Figure 2:

1) *Case I:* If both angles are small (Figure 2a), their difference will be even smaller and their sum will also

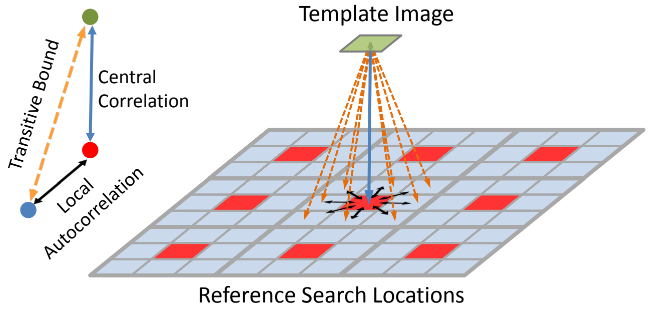


Fig. 3. Groups of Search Locations in `Intra-Ref-TEA` algorithm. A ‘search location’ is the central pixel of a possible matching location of the template, within the reference image. Small squares show 81 search locations divided into non overlapping 3×3 groups. Each group has a central search location shown in red and neighboring search locations shown in blue. The template always has to be correlated with central locations while its correlation with the neighboring locations may be eliminated based upon the transitive bounds.

be a relatively small number. Therefore both upper and lower transitive bounds will approach +1. This ensures tight upper and lower bounds because in this case, the bounded correlation will also be very high.

2) *Case II:* If one angle is small while the other is large (Figure 2b), then their difference will be large, resulting in a tight upper bound, and their sum will also be a relatively large number, resulting in a loose lower bound.

3) *Case III:* If both of the angles are large (Figure 2c), then their difference will be a small number, resulting in a very loose upper bound while their sum will be a significantly larger number, resulting in a very loose lower bound.

In these three cases, Case I yields tight upper and lower bounds and can potentially be exploited for computation elimination. However, practically, this case occurs infrequently because it is less likely to get all of the three image patches to be highly correlated. Case III yields loose upper and lower bounds therefore this case cannot be exploited for computation elimination. Case II yields a tight upper bound, and requires that one of the two bounding correlations has high magnitude. Since in most of the template matching problems, strong autocorrelation is present in one form or the other, therefore choosing autocorrelation as one of the two bounding correlations ensures that Case II occurs frequently.

For a standard single template and single reference matching problem, local spatial autocorrelation of the reference may be exploited to ensure one high bounding correlation, as required by Case II. For a problem in which one template has to be correlated with a sequence of reference images, temporal autocorrelation of the reference images may be exploited. Finally if a sequence of template images is to be correlated with a single reference image, then temporal autocorrelation of templates may be exploited to obtain speed-up. We discuss these three cases in detail in the following subsections.

A. Exploiting Strong Intra-Reference Autocorrelation

Many template matching applications may require a single template to be correlated with a single reference image. In such applications, local spatial autocorrelation of the reference image may be exploited for fast template matching. For this purpose, we divide the search locations within the reference image into non overlapping rectangular groups and compute local autocorrelation (A_S) of the central location with the neighboring locations of the group (Figure 3).

In each group, the template image is correlated with the central search location, to yield *Central Correlation* (C_C) and the correlation of the template with the remaining locations is delayed until the evaluation of the elimination test. As shown in Figure 3, both local autocorrelation and central correlation are used as bounding correlations to compute transitive bounds for the remaining locations, and those with upper bounds less than a current known maximum (or less than a conservative initial threshold) may be skipped, without any loss of accuracy. Since the spatial autocorrelation with close neighbors is often high for natural images, this results in a tight upper bound and hence high elimination at most locations. Complete pseudo-code for this algorithm is shown as `Intra-Ref-TEA`.

Algorithm 1 `Intra-Ref-TEA`

```
$A_S \Leftarrow$ Local Spatial Auto-correlation
$C_{\max} \Leftarrow$ Initial correlation threshold
for all Groups of search locations do
    $C_C \Leftarrow \texttt{correlate}(\textit{template,central search location})$
    if $C_C > C_{\max}$ then
        $(C_{\max}, i_{\max}, j_{\max}) \Leftarrow (C_C,$ Central location indices$)$
    end if
    for all Remaining locations within current group do
        $UpperBound \Leftarrow A_S C_C + \sqrt{(1-{A_S}^2)(1-{C_C}^2)}$
        if $UpperBound < C_{\max}$ then
            Skip current location
        else
            $C \Leftarrow \texttt{correlate}(\textit{template,current search location})$
            if $C > C_{\max}$ then
                $(C_{\max}, i_{\max}, j_{\max}) \Leftarrow (C,$ Current location indices
                $)$
            end if
        end if
    end for
end for
print $i_{\max}, j_{\max}, C_{\max}$
```

In Algorithm 1 the speed-up is obtained from bounded correlations, shown as dotted arrows in Figure 3, whereas the bounding correlations constitute an overhead for the algorithm. There are two types of overheads: the computation of the local spatial autocorrelation of the reference image and the computation of the central correlation in each group. For the first type, the standard implementation has computational complexity of the order of $O(mnpq)$ [9], where $m \times n$ is the template size and $p \times q$ is the reference image size. However, redundant computations can be eliminated by using a more efficient algorithm, which reduces the computational complexity to $O(s_h s_w pq)$ (as discussed later in this section), where $s_h \times s_w$ is the size of the group of locations.

For the overhead due to central correlation, we observe that at least one correlation is a must for each group. Since the number of groups are pq/s_hs_w, and one correlation of the template of size $m\times n$ is must for each group, the overhead cost is given as $O(mnpq/s_hs_w)$. The total overhead for both types can be written as the summation of the two overheads:

$$\eta = \xi(s_hs_w + \frac{mn}{s_hs_w})pq, \tag{12}$$

where ξ is a machine dependent constant. If k templates are to be matched with the same reference image, the local autocorrelation overhead is further amortized to yield a total overhead of

$$\eta = \xi(\frac{s_hs_w}{k} + \frac{mn}{s_hs_w})pq, \tag{13}$$

Assuming the cost of spatial domain template matching to be $\xi mnpq$, a theoretical upper bound upon the speed-up of `Intra-Ref-TEA` may be written as:

$$\text{Speed-Up} \leq \frac{mn}{(\frac{s_hs_w}{k} + \frac{mn}{s_hs_w})}. \tag{14}$$

As an illustration, if 10 templates each of size 64×64 pixels are to be matched with a reference image (of any size) and the group size is 5×5, the upper bound upon maximum achievable speed-up over spatial domain is 24.624.

Equation (14) indicates that more speed-up is possible on larger group sizes. However, on larger sizes the local autocorrelation may decay down to a small value, hence reducing the tightness of the transitive bounds and therefore resulting in reduction in elimination. The proper choice of the group-size parameter, therefore, depends upon the spread of the local autocorrelation function in the reference image and the magnitude of the known correlation maxima. The smallest size of a symmetrical group is 3×3 search locations, which means that the central search location will be correlated with its eight neighbors only. Practically one may adapt to the proper group size by observing the computation elimination. For $s_h\times s_w$ group size, if percentage of eliminated computations approach the maximum limit $(s_hs_w-1)/(s_hs_w)\times 100$, the group size may be increased to $(s_h+1)\times(s_w+1)$. This is because, approaching the maximum limit of elimination indicates that the reference image may have a wider autocorrelation that may allow even larger group size to get more speed-up. On the other hand, if the computation elimination reduces to less than $((s_h-1)(s_w-1)-1)/((s_h-1)(s_w-1))\times 100$, then the size may be reduced to $(s_h-1)\times(s_w-1)$.

As mentioned earlier, the computation of local autocorrelation can be made more efficient than its standard implementation by exploiting the redundancy in its computation. We propose an algorithm in which the correlation between central location r_c, and another location r_n, is computed simultaneously over all groups, through pixel by pixel multiplication of the reference image with its (w_r,w_c) translated version, where (w_r,w_c) is the row, column difference between r_c and r_n. Then using the running-sum approach, we compute the sum of all $m\times n$ blocks in the product array, in just four operations per block. This results in correlation of each search location with a (w_r,w_c) translated location. We copy only the required values in a final *LA-Array* as shown in `LA-Algorithm`. The same process is repeated s_hs_w times, and each time pq integer multiplications and $4pq$ additions are done. Therefore the overall complexity of this approach is $O(s_hs_wpq)$. In terms of space, this algorithm requires additional storage of three arrays: P_r, S_f and L_A, each of size equal to that of the reference image.

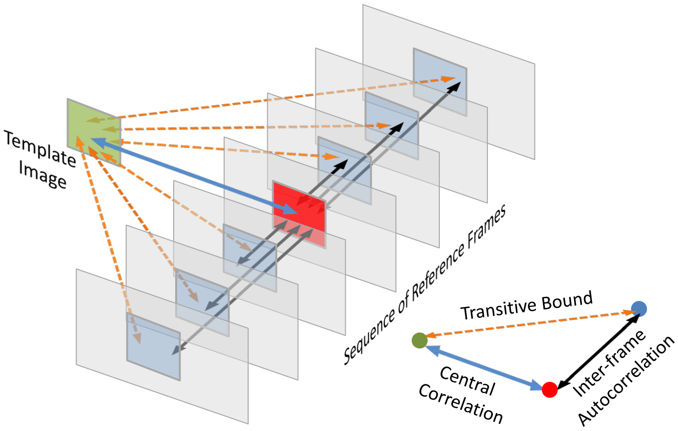


Fig. 4. Exploiting strong inter-frame autocorrelation for fast template matching in `Inter-Ref-TEA`. The template is fully correlated with only one frame (shown red), while for the remaining frames transitive bounds are computed.

Algorithm 2 `LA-Algorithm`

```
$I_{ref} \Leftarrow$ Reference image
$(m,n) \Leftarrow$ Template image size
$(s_h,s_w) \Leftarrow$ Size of group of locations
for $w_r = 1$ to $s_h$ do
  for $w_c = 1$ to $s_w$ do
    for all pixels $(i,j)$ in Reference-Image do
      $P_r(i,j) \Leftarrow I_{ref}(i,j)I_{ref}(i+w_r,j+w_c)$
    end for
    $S_f \Leftarrow$Running sum of all $m\times n$ patches in $P_r$
    ▷Copy Only Required Values From $S_f$ to LA-array
    for all $(i,j)$ in final LA-array do
      $LA(i+w_r,j+w_c) \Leftarrow S_f(i+m,j+n)$
      $i \Leftarrow i+s_h$
      $j \Leftarrow j+s_w$
    end for
  end for
end for
```

B. Exploiting Strong Inter-Reference Auto-Correlation

In some template matching applications, for example tracking objects across a video sequence, one template image has to be correlated with multiple reference frames. If the reference frames are highly temporally correlated, such as in the case of a static surveillance camera, we can exploit their temporal autocorrelation (A_T) to get tight transitive bounds. The concept is illustrated in Figure 4. In this case, the central correlation (C_C) is obtained by correlating the template with a specific reference frame. The correlation with the remaining frames is delayed until evaluation of the transitive elimination test.

Using A_T and C_C as bounding correlations, we compute the transitive upper and lower bounds upon all search locations in the remaining frames. The locations with upper bound less than the current known maximum (or an initial correlation threshold), may be discarded without any loss of accuracy.

In some applications, for example checking for missing components in a circuit board manufacturing facility, the three image patches may be very similar. Therefore we may get both upper and lower bounds to be tight as given by Case I. In such applications, search locations where the upper bound is less than the maximum of the lower bound, may also be skipped without any loss of accuracy. The pseudo code for this algorithm is given as `Inter-Ref-TEA`.

This algorithm carries the overhead of computing the temporal autocorrelation of the sequence of reference frames. We employ a similar strategy as in the previous case and compute this overhead in $O(pq)$, where $p \times q$ is size of the reference image. This is done by multiplying, pixel by pixel, the two reference frames and then using the running sum approach to compute the summation of all patches of size $m \times n$ in the product array. Since the complexity of running sum algorithm is $O(pq)$ and pq integer multiplications were carried out, overall complexity of this overhead is $O(pq)$, which is significantly smaller than even the complexity of a single template correlation in $O(mnpq)$. Hence the computational cost of inter frame autocorrelation computation is insignificant compared to the over all cost of template matching.

Algorithm 3 `Inter-Ref-TEA`

```
$f_c \Leftarrow$ Fully correlated reference frame
$C_C \Leftarrow$ correlate( $template$, $f_c$)
print  $f_c, i_{\max}, j_{\max}, \max(C_C)$
for all  remaining frames, $f_k$ do
  $A_T \Leftarrow$ Autocorrelate $f_c$ with $f_k$
  $L_{\max} \Leftarrow$ Maximum of lower bound over $f_k$
  $C_{\max} \Leftarrow$ Initial correlation threshold
  if $L_{\max} > C_{\max}$ then
    $C_{\max} = L_{\max}$
  end if
  for all Search locations in $f_k$ do
    $UpperBound \Leftarrow A_T C_C + \sqrt{(1-{A_T}^2)(1-{C_C}^2)}$
    if  $UpperBound < C_{\max}$ then
      Skip current location
    else
      $C \Leftarrow$ Correlate template with current Location
      if $C > C_{\max}$ then
        $(C_{\max}, i_{\max}, j_{\max}) \Leftarrow (C,$ Current location indices )
      end if
    end if
  end for
  print  $f_k, i_{\max}, j_{\max}, C_{\max}$
end for
```

C. Exploiting Strong Inter-Template Auto-Correlation

In some template matching applications, for example registration of an aerial video with a satellite image [13], a sequence of template frames is to be correlated with the same reference image. In such applications, if consecutive template frames exhibit strong inter-template auto-correlation, the transitive bounds may be used to speed-up the template matching process. For this purpose, we divide the sequence of template frames into groups such that all templates within each group exhibit strong autocorrelation A'_T with the temporally central frame. One such group of templates is shown in Figure 5, in which the central correlation C_C is obtained by correlating the central template with the reference image. Then using A'_T and C_C as bounding correlations, we compute the transitive bounds upon the correlation of each remaining template in the group. All match locations with upper transitive bounds less than the current known maximum or an initial correlation threshold, may be discarded without any loss of accuracy.

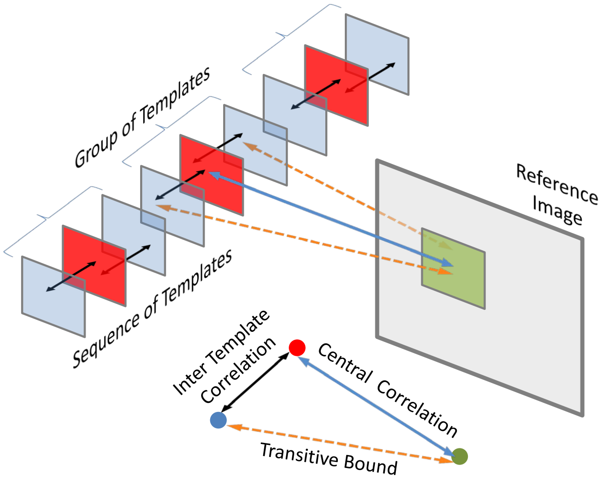


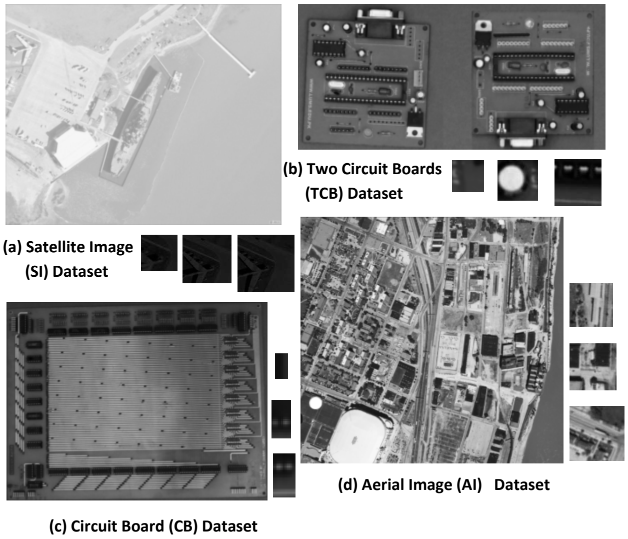
Fig. 5. Exploiting strong inter-template autocorrelation for fast template matching in `Inter-Template-TEA`.

In large template video sequences, the temporal autocorrelation may significantly vary over time, requiring different group lengths. To find the appropriate group length at runtime, we adapt the length of the current group using the percentage computation elimination results of the previous group. Let actual elimination obtained in the $k-1^{st}$ group be e^{k-1}_{act}, and the maximum possible elimination be $e^{k-1}_{\max} = (L[k-1]-1)/L[k-1]$, where $L[\cdot]$ denotes the length of a group. If both of these eliminations are close to each other, then autocorrelation may be under utilized and the group length may be increased, while if e^{k-1}_{act} is significantly less than $e^{k-1}_{\max}$, then autocorrelation is less than expected, therefore group length, should be decreased for the next group:

$$L[k] = \begin{cases} L[k-1]+2, & \text{if } e^{k-1}_{\max} - e^{k-1}_{act} < \delta_l \\ L[k-1]-2, & \text{if } e^{k-1}_{\max} - e^{k-1}_{act} > \delta_h \\ L[k-1], & \text{otherwise} \end{cases} \qquad (15)$$

where δ_l and δ_h are low and high thresholds upon elimination.

The only overhead in this algorithm is the computation of inter-template autocorrelation which is of the order of $O(mn)$, where $m \times n$ is the template size. The cost of this overhead is negligibly small as compared to the overall computations.



(a) Satellite Image (SI) Dataset

(b) Two Circuit Boards (TCB) Dataset

(c) Circuit Board (CB) Dataset

(d) Aerial Image (AI) Dataset

Fig. 6. Four datasets used for experiments on exploiting intra reference autocorrelation. See Table I for more dataset details.

TABLE I
DATASET DESCRIPTION FOR EXPERIMENTS WITH INTRA-REF-TEA

Dataset	Template Sizes			Total Frames	Reference Size
	a	b	c		
SI	64×64	112×112	128×128	711	800×1000
TCB	34×34	51×51	68×68	579	807×1716
CB	16×8	24×12	32×16	328	762×1000
AI	95×95	97×97	99×99	171	1453×1548

TABLE II
TOTAL EXECUTION TIME IN SECONDS TAKEN BY INTRA-REF-TEA AND OTHER ALGORITHMS UPON DATASETS DESCRIBED IN TABLE I

Dataset	IR-TEA	ZBPC	ZNccEbc	FFT	SAD	Spat
AI.a	108.89	1176.6	306.91	368.05	460.89	2375.9
AI.b	141.61	1604.3	632.19	474.49	625.14	3099.4
AI.c	185.53	2194.7	413.00	639.87	812.38	4207.8
CB.a	9.71	24.14	29.34	193.07	1.45	35.62
CB.b	17.14	51.86	28.76	188.03	2.91	70.66
CB.c	26.58	81.72	31.94	191.25	5.87	118.58
TCB.a	63.80	675.70	249.03	880.89	50.23	870.78
TCB.b	103.10	1426.0	263.64	848.97	130.38	1827.0
TCB.c	160.75	2499.5	278.67	838.24	267.31	3260.90
SI.a	352.68	2332.0	460.55	1307.5	5.13	2557.0
SI.b	449.84	5717.9	831.12	1152.7	12.13	6356.6
SI.c	465.31	6882.4	961.84	1108.9	15.76	7667.3

IV. EXPERIMENTS AND RESULTS

We have performed extensive empirical evaluation of the three types of template matching problems described in the previous section. Our experiments are done on ten different datasets, consisting of 424 reference images and 8465 templates. The size of reference images ranges from 240 × 320 to 1394×2194 pixels, while the smallest template is 16×8 and the largest 128×128 pixels. None of the templates is generated by simply cropping the reference image; rather, each template is from an independently captured image, containing natural, and in some cases, synthetically generated distortions.

The proposed algorithms are implemented in C++ and compared with the currently known fast exhaustive template matching techniques including FFT-based frequency domain implementation [14], *Zero-mean Bounded Partial Correlation* (ZBPC) [7], *Zero-mean Enhanced Bounded Correlation*(ZNccEbc) [8] and an exhaustive spatial domain implementation (Spat) [12]. We have implemented ZBPC algorithm and all experiments are carried out with the correlation area of 20% and bound area of 80% [7]. Implementation of ZNccEbc algorithm was provided by the original authors [8]. Other than correlation based measures, we have also implemented Sum of Absolute Differences with *Partial Distortion Elimination* [6] and *Successive Elimination Algorithm* [1] optimizations.

In order to ensure a realistic comparison, we have used only sequential implementations of all algorithms. The execution times are measured on an IBM machine with Intel Core 2 CPU 2.13 GHz processor and 1GB RAM. The datasets, implementation and detailed results are available on our web site: *http://cvlab.lums.edu.pk/tea*.

A. Exploiting Intra-Reference Auto-correlation

Experiments corresponding to Section III-A are performed on four datasets: Satellite Images (SI) dataset, Aerial Images (AI) dataset, Circuit Board (CB) dataset and Two Circuit Boards (TCB) dataset (see Table I and Figure 6). The images to be matched have projective distortions due to difference in viewing geometry. In addition, the reference image of SI dataset has very high brightness while the templates have low brightness and contrast. In CB and TCB datasets, templates and the reference images are taken from different boards. In AI dataset, available from flicker.com under Creative Commons license, templates and the reference are aerial images of the same scene, taken from two different locations.

For the Intra-Ref-TEA algorithm, a group size of 5×5 was used for all datasets. For ZNccEbc algorithm, we used r = 8 where-ever possible; when the rows of the template were not divisible by 8, all factors were tried and results were reported for factor generating maximum speed-up. Therefore, r = 8 was selected for SI and CB, 17 for TCB and 5, 97, 9 for AI(a, b, c) datasets.

The speedup is dependent upon the initial threshold value, which may be specified by the user or estimated by several approaches, such as the coarse-to-fine technique [15], two stage template matching [16], three-step search [17] or two dimensional logarithmic search [18]. However, to keep the analysis focused on the current problem, we used a uniform initial threshold of $\rho = 0.8$ for all algorithms. The execution time, reported in Table II, includes the local autocorrelation overhead, which is {1.463s, 0.270s, 0.505s, 0.963s} for AI, CB, SI and TCB datasets respectively.

TABLE III
INTRA-REF-TEA: EFFECT OF GROUP SIZE PARAMETER (GRSZ) UPON EXECUTION TIME (SEC) AND PERCENT COMPUTATION ELIMINATION (%E)

GrSz	3×3		5×5		7×7		9×9	
DSet	T	%E	T	%E	T	%E	T	%E
AI.a	7.01	87.2	2.74	95.1	2.99	94.6	4.69	91.4
CB.c	0.18	85.3	0.24	80.3	0.25	79.2	0.27	77.3
TCB.c	2.04	88.8	0.86	95.4	0.96	94.8	1.79	90.1
SI.c	3.23	88.9	1.24	95.8	1.13	96.1	1.6	94.4

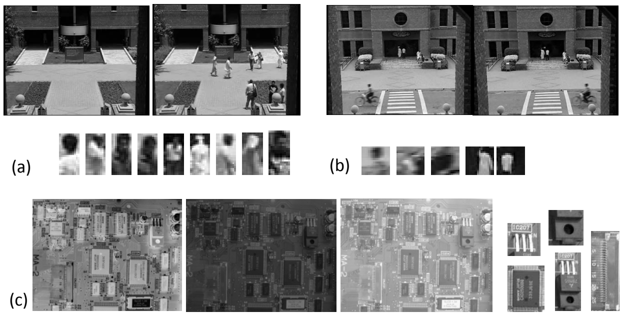


Fig. 7. **(a)** Pedestrian dataset: four reference frames and 9 feature templates. **(b)** Cyclist dataset: four reference frames and 5 feature templates. **(c)** Fast Component Tracking (CT) dataset: 3 reference frames and 5 templates [19].

TABLE IV
DATASET DESCRIPTION FOR FAST FEATURE TRACKING/FAST COMPONENT TRACKING EXPERIMENTS

Dataset	# of Feat.	Feat. Size	# of Frames	Frame Size
PED	21	23 × 11	325	240 × 320
CYC	5	17 × 17	38	240 × 320
CT.a	6	63 × 63	16	479 × 640
CT.b	1	178 × 62	16	479 × 640
CT.c	1	136 × 104	16	479 × 640
CT.d	1	147 × 63	16	479 × 640
AT	20	95 × 95	25	1453 × 1548

The execution time speed-up of `Intra-Ref-TEA` over other algorithms is dataset dependant. Maximum observed speed-up over `ZBPC` is 15.549 times, over `ZNccEbc` is 4.464, over FFT is 24.626 and over `Spat` is 22.680 times. `Intra-Ref-TEA` has remained faster than other correlation coefficient based algorithms, while for CB and SI datasets SAD has exhibited highest speed. However SAD exhibits poor accuracy over these datasets, due to brightness and contrast variations. For SI, none of the templates matched at the correct location, while for CB, only 25 of 328 templates matched correctly. Whereas, the accuracy of all correlation coefficient based algorithms has remained 100%.

Over a portion of the four datasets, variation of % computation elimination and average execution time per template has been studied by varying the group size parameter to {3×3, 5×5, 7×7 and 9×9} (see Table III). The datasets AI.a and TCB.c have shown the best performance at group size of 5×5 while CB.c and SI.c performed best at 3×3 and 7×7, respectively. Thus by tuning the group size parameter, speed-up reported in Table II may further be improved for CB and SI datasets, even though all experiments reported in Table II are for 5×5 group size.

B. Exploiting Inter-Reference Auto-correlation

TABLE V
TOTAL TIME IN SECONDS FOR DATASETS DESCRIBED IN TABLE IV FOR `INTER-REF-TEA` AND OTHER ALGORITHMS

Data	IRTEA	`ZNccEbc`	`ZBPC`	FFT	`SAD`	`Spat`
PED	58.30	548.61	340.38	268.91	110.27	374.51
CYC	1.50	14.04	10.72	22.61	3.03	11.1
CT.a	12.76	65.90	198.50	166.08	86.27	263.45
CT.b	4.06	27.92	82.05	27.59	49.53	88.53
CT.c	3.31	29.88	101.64	27.69	73.47	125.72
CT.d	2.05	8.62	57.70	27.72	38.08	81.68
AT	223.43	5235.1	22668	4171.3	6105.7	27099

TABLE VI
PERCENTAGE COMPUTATION ELIMINATION IN `INTER-REF-TEA` AND OTHER ELIMINATION ALGORITHMS

Dataset	`IR-TEA`	`ZNccEbc`	`ZBPC`	`SAD`
PED	80.496	93.839	12.571	75.583
CYC	93.749	89.607	8.451	77.259
CT.a	92.250	97.691	24.957	69.192
CT.b	88.150	93.047	8.103	49.063
CT.c	91.029	98.631	19.751	43.889
CT.d	93.162	99.560	29.585	57.397
AT	95.755	98.947	17.733	78.318

1) Experiment on Fast Feature Tracking: In this experiment, manually extracted features are tracked across Pedestrian (PED) and Cyclist (CYC) datasets. Both videos were acquired in a typical surveillance scenario (see Table IV and Figure 7). Both datasets contain dissimilarities produced by human motion and illumination variations. Initial correlation threshold is set to 0.70 for each of the algorithm. The parameter r in `ZNccEbc` is 23 for PED and 17 for CYC. The total execution time for the `Inter-Ref-TEA` given in Table V, includes all overheads.

In these experiments `Inter-Ref-TEA` remained significantly faster than other algorithms. The maximum speed-up over `ZBPC` is 7.15 times, over `ZNccEbc` is 9.41, over FFT is 15.07, over `SAD` is 2.02 and over `Spat` is 7.40 times. The slow execution times for the `ZNccEbc` algorithm is due to unfavorable template sizes, which increases the bound computation overhead. Percentage of eliminated computations is reported in Table VI. For the PED dataset the `ZNccEbc` algorithm has obtained maximum elimination, while for the CYC dataset `Inter-Ref-TEA` has obtained highest elimination.

2) Experiment on Fast Component Tracking: In this dataset there is no local motion and the component templates are significantly larger in size as compared to the feature templates. Two types of datasets are used: Component Tracking (CT) and Aerial Tracking (AT) (see Figure 7 and Table IV). Original images in CT were taken from [19] and AT dataset is a portion of the AI dataset used in Subsection IV-A. The following frame to frame variations were synthetically produced: affine photometric variations, non-linear photometric variations, complementing, sharpening by edge-enhancements and geometrically transforming the original images.

Initial correlation threshold of 0.70 has been used for all algorithms. The central correlations in `Inter-Ref-TEA` has been computed by the FFT based implementation. For `ZNccEbc`, the r parameter has been selected to be {7, 89, 8, 7, 5} for CT(a, b, c, d) and AT. For fast feature tracking experiment, the percent computation elimination comparison is shown in Table VI, in which `ZNccEbc` has obtained maximum elimination. However, in the total execution times reported in Table V, `Inter-Ref-TEA` has remained faster than all algorithms. This is because the cost of bound computation in `ZNccEbc` has exceeded the benefit obtained by elimination. The maximum speed-up observed by `Inter-Ref-TEA` over `ZNccEbc` algorithm is 23.43 times, over `ZBPC` is 101.46, over

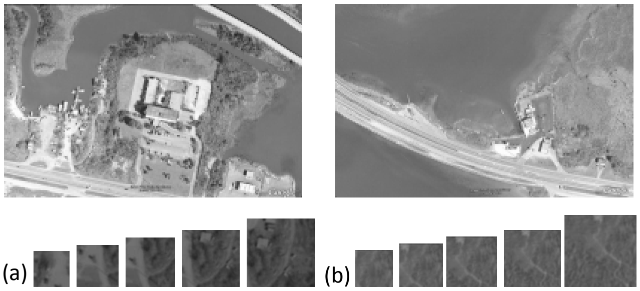


Fig. 8. Video geo-registration dataset: **(a)** DS1 **(b)**DS2. Reference images are taken from earth.google.com and templates from terraserver.microsoft.com.

TABLE VII
DATASET DETAILS USED FOR VIDEO GEO-REGISTRATION EXPERIMENTS

Dataset	# of Frames	Frame Size	Ref. Size	Avg. $\rho_{\max}$
DS1.a	734	64 × 64	736 × 1129	0.939
DS1.b	744	80 × 80	736 × 1129	0.961
DS1.c	694	96 × 96	736 × 1129	0.963
DS1.d	641	112 × 112	736 × 1129	0.961
DS1.e	594	128 × 128	736 × 1129	0.958
DS2.a	659	64 × 64	1394 × 2152	-0.935
DS2.b	645	80 × 80	1394 × 2152	-0.921
DS2.c	648	96 × 96	1394 × 2152	-0.874
DS2.d	632	112 × 112	1394 × 2152	-0.924
DS2.e	616	128 × 128	1394 × 2152	-0.794

FFT is 18.67, over `SAD` is 27.33 and over `Spat` is 121.29 times.

C. Exploiting Inter-Template Auto-correlation

1) Fast Video Geo-registration: These experiments are performed on two datasets, DS1 and DS2 (see Table VII and Figure 8). The images to be matched contain dissimilarities due to difference in imaging sensor and viewing geometry. Additional dissimilarities were generated by reducing the dynamic range of templates in DS1 to one third of the original range and the templates in DS2 were contrast reversed. Contrast reversals are frequently observed in practical situations, if matching is to be done across infra-red and optical imagery.

The initial correlation threshold is 0.80 for DS1 and -0.85 for DS2. For `ZNccEbc`, $r = 8$ has been used. In `IT-TEA`, the correlation of the central templates with the reference images has been done by using the FFT based implementation and length of the group of templates is initialized to 7 for DS1 and 5 for DS2. For the remaining groups, length was automatically adapted by using $\delta_l = 3\%$ and $\delta_h = 10\%$ in Equation (15). Average group length has remained {8.6, 10.9, 11.6, 12.1, 12.4} for DS1(a-e) and { 7.8, 7.2, 7.7, 8.2, 7.7} for DS2(a-e).

TABLE VIII
VIDEO GEO-REGISTRATION: AVERAGE EXECUTION TIME IN SECONDS PER TEMPLATE FRAME

Dataset	`IT-TEA`	`ZNccEbc`	`ZBPC`	FFT	`SAD`	`Spat`
DS1.a	1.217	1.366	6.415	4.223	0.107	8.455
DS1.b	1.156	1.675	8.575	4.173	0.156	13.587
DS1.c	1.413	2.314	12.736	4.161	0.258	18.553
DS1.d	1.669	2.787	16.310	4.261	0.436	24.018
DS1.e	1.977	3.333	16.715	4.266	0.610	29.855
DS2.a	6.394	16.725	32.163	19.547	2.969	32.848
DS2.b	8.614	28.378	53.303	19.552	4.976	53.760
DS2.c	12.290	42.751	74.399	19.606	7.030	74.933
DS2.d	15.534	58.995	98.432	19.458	9.374	99.027
DS2.e	12.250	78.110	125.170	19.563	11.959	125.740

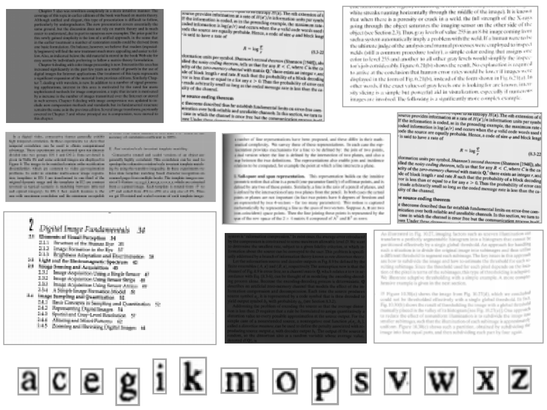


Fig. 9. Rotation and Scale invariant template matching: Nine reference images and 14 templates.

TABLE IX
ROTATION AND SCALE INVARIANT TEMPLATE MATCHING: DATASET FOR CHARACTER RECOGNITION

Letter	Tmp. Size	Ref. Size	Letter	Tmp. Size	Ref. Size
a	19×14	679×889	o	18×17	671×1215
c	19×15	755×977	p	25×17	702×1206
e	17×15	552×1005	s	18×12	711 × 1224
g	26×16	593×1209	v	18×17	681 × 1271
i	25×8	907×1263	w	19×23	756 × 1341
k	25×17	684×1031	x	18×16	475 × 1463
m	18×24	647×1046	z	19×15	291 × 758

Execution time comparison of `IT-TEA` and other algorithms is given in Table VIII. For DS1, maximum execution time speed-up of `IT-TEA` over `ZBPC` is 9.77 times, over `ZNccEbc` is 1.69, over FFT is 3.61 and over `Spat` is 15.10 times. For DS2, maximum observed speed-up of `IT-TEA` over `ZBPC` is 10.22, over `ZNccEbc` is 6.38, over FFT is 3.06 and over `Spat` is 10.26 times. Although SAD has remained faster than all correlation coefficient based algorithms, it failed to match any template at the correct location. High execution times of `ZBPC` and `ZNccEbc` on DS2 can be attributed to the fact that these algorithms have been developed to find only positive maxima, where as in case of DS2 negative peaks have to be searched. Transitive elimination algorithm does not require any modification to search for negative peaks.

2) Fast rotation/scale invariant template matching: Consecutive rotated and scaled versions of an object are generally highly correlated. We have used this correlation to speed-up the exhaustive rotation/scale invariant template matching by using `IT-TEA`. These experiments are performed upon optical character recognition dataset using scanned pages from multiple books. The template images consist of 14 individual characters, which were extracted from one of the scanned image (see Table IX and Figure 9). Each template is rotated from -5° to +5° and scaled from -8% to +8% at a step size of 2%, resulting in 99 rotated/scaled versions. All of these rotated/scaled versions are exhaustively correlated with each of the 14 reference images, with varying background

TABLE X
Rotation and Scale invariant template matching: Total execution time (in seconds) for IT-TEA and other algorithms

Dataset	IT-TEA	ZNccEbc	ZBPC	FFT	SAD	Spat
a	43.06	1164.7	849.77	4975.6	322.97	836.80
c	41.20	1187.2	880.01	4769.8	372.29	891.00
e	40.12	1091.7	784.39	4808.9	308.02	807.84
g	50.37	974.53	1230.8	4761.0	516.43	1245.2
i	47.64	445.77	682.37	4804.7	253.40	679.14
k	45.45	643.08	1285.9	4756.3	578.13	1289.0
m	63.67	760.49	1250.7	5132.5	467.56	1311.3
o	42.67	699.70	921.67	4803.1	370.72	955.35
p	46.43	559.51	1286.8	4845.2	546.72	1288.7
s	38.01	680.66	712.88	4815.8	260.81	706.61
v	39.75	682.33	927.68	4829.8	387.66	954.36
w	45.21	1311.5	1264.9	5046.7	571.79	1322.9
x	41.85	733.22	878.91	4886.5	399.99	888.03
z	40.22	1219.6	900.72	4816.2	457.56	893.48

colors, arbitrary rotations, arbitrary scaling, aliasing effects due to poor scanner resolution and with broken and irregular character boundaries.

Out of 99 rotated/scaled versions of each template, only one template (with zero rotation and unit scaling) is fully correlated with the complete reference image while for all of the remaining templates, transitive bounds are computed. The initial correlation threshold is set to 0.80. In ZNccEbc, partition parameter r is set to be: $\{19,19,17,13,5,5,9,9,5,9,9,19,9,19\}$ respectively for the 14 templates given in Table IX.

The total execution time including all overheads is shown in Table X. The maximum execution time speed-up obtained by IT-TEA is 28.29 times over ZBPC, 30.32 times over ZNccEbc, 126.70 times over FFT, 12.67 times over SAD and 29.26 over Spat. On this dataset, the speed-up obtained by IT-TEA over other algorithms is enhanced because of the small template sizes and high autocorrelation between consecutive rotated/scaled template versions.

D. Speed-Up Comparison of Correlation Measures

We compared the execution times and the computation elimination performance of the three correlation based similarity measures: cross-correlation, NCC and correlation-coefficient on six datasets: DS1 (a, b, c, d, e) and PED. For DS1, IT-TEA and for PED Inter-Ref-TEA has been used for comparison. The total execution time and the average computation elimination per frame is reported in Table XI.

In these experiments we observe that cross-correlation is the fastest of the three measures. NCC was found to be faster than correlation coefficient over PED datasets while slower on DS1 datasets. This may be because of the fact that NCC is not robust to additive intensity variations and therefore in the presence of such variations the magnitude of NCC maxima may reduce, causing a reduction in elimination and an increase in execution time. However, the relative speed-up is data dependent and may vary for other datasets.

V. Conclusion

In this work, we have demonstrated that the transitive property of correlation can be exploited for fast template matching. Three variations of transitive elimination algorithms are presented to cater for different types of the template matching problems. The proposed algorithms have exhaustive equivalent accuracy and are compared with fast exhaustive techniques on a wide variety of real image datasets. Our empirical results based upon the correlation of 8465 templates with 424 reference images, demonstrate that the proposed algorithms are faster than current techniques by a significant margin.

TABLE XI
Total execution time (T) (sec) and average percent elimination (E) for cross-correlation (ψ), NCC (ϕ) and ρ

Dataset	T_ψ	T_ϕ	T_ρ	E_ψ	E_ϕ	E_ρ
PED	23.614	45.75	58.318	99.36	82.24	80.5
DS1.a	213.66	967.32	732.12	95.13	75.4	83.21
DS1.b	316.42	1321.6	861.47	94.66	70.5	85.32
DS1.c	548.14	1387.9	980.37	92.56	71.42	85.79
DS1.d	777.27	1462.1	1069.9	89.13	72.34	86.29
DS1.e	931.09	1572.1	1174.4	85.26	72.6	86.49

Appendix A
Transitive Inequality Based Upon Euclidean Distance

Considering three image blocks r_1, r_2 and r_3, let $\Delta_{i,j}$ be the Euclidean distance between any two of them, where $i,j \in \{1,2,3\}$:

$$\Delta_{i,j} = \sqrt{\sum_{x=1}^{n}\sum_{y=1}^{m}\Big(\frac{r_i(x,y)-\mu_i}{\sigma_i} - \frac{r_j(x,y)-\mu_j}{\sigma_j}\Big)^2}. \tag{16}$$

Triangular inequality for Euclidean distance is given by:

$$|\Delta_{1,2} - \Delta_{2,3}| \le \Delta_{1,3} \le \Delta_{1,2} + \Delta_{2,3}. \tag{17}$$

Squaring all sides and using relationship from [9]:

$$\rho_{i,j} = 1 - \frac{1}{2}\Delta_{i,j}^2, \tag{18}$$

we get Euclidean distance based transitive bounds upon correlation coefficient:

$$\begin{aligned}(\rho_{1,2}+\rho_{2,3}-1)+2\sqrt{(1-\rho_{1,2})(1-\rho_{2,3})} \ge \rho_{1,3}\\ \ge (\rho_{1,2}+\rho_{2,3}-1)-2\sqrt{(1-\rho_{1,2})(1-\rho_{2,3})}.\end{aligned} \tag{19}$$

Appendix B
Comparison of Upper Transitive Bounds

Since $0 \le \Delta_{i,j} \le 2$ and $-1 \le \rho_{i,j} \le +1$, therefore inequalities

$$1 - \frac{\Delta_{1,2}\Delta_{2,3}}{4} - \frac{1}{2}\sqrt{(1+\rho_{1,2})(1+\rho_{2,3})} \ge 0 \tag{20}$$

and

$$\Delta_{1,2}\Delta_{2,3} \ge 0, \tag{21}$$

always hold. By multiplying these two inequalities:

$$\Delta_{1,2}\Delta_{2,3}\big[1 - \frac{\Delta_{2,3}\Delta_{1,2}}{4} - \frac{1}{2}\sqrt{(1+\rho_{1,2})(1+\rho_{2,3})}\big] \ge 0, \tag{22}$$

where:

$$\Delta_{1,2}\Delta_{2,3} = 2\sqrt{(1-\rho_{2,3})(1-\rho_{1,2})}. \tag{23}$$

From (22) and (23):

$$\rho_{1,2}\rho_{2,3}+\sqrt{(1-\rho_{1,2}^2)(1-\rho_{2,3}^2)}\leq$$
$$(\rho_{1,2}+\rho_{2,3}-1)+2\sqrt{(1-\rho_{1,2})(1-\rho_{2,3})} \tag{24}$$

Appendix C
Comparison of Lower Transitive Bounds

Since $0\leq\Delta_{i,j}\leq 2$ and $-1\leq\rho_{i,j}\leq+1$, therefore inequalities

$$\Delta_{1,2}\Delta_{2,3}\left[1-\frac{1}{2}\sqrt{(1+\rho_{1,2})(1+\rho_{2,3})}\right]\geq 0 \tag{25}$$

and

$$\rho_{1,2}\rho_{2,3}-(\rho_{1,2}+\rho_{2,3}-1)\geq 0, \tag{26}$$

always hold. Adding these two inequalities and rearranging the terms:

$$\rho_{1,2}\rho_{2,3}-\Delta_{1,2}\Delta_{2,3}\frac{1}{2}\sqrt{(1+\rho_{1,2})(1+\rho_{2,3})}]$$
$$\geq(\rho_{1,2}+\rho_{2,3}-1)-\Delta_{1,2}\Delta_{2,3}, \tag{27}$$

substituting the value of $\Delta_{1,2}\Delta_{2,3}$ from (23):

$$\rho_{1,2}\rho_{2,3}-\sqrt{(1-(\rho_{1,2})^2)(1-(\rho_{2,3})^2)}$$
$$\geq(\rho_{1,2}+\rho_{2,3}-1)-2\sqrt{(1-\rho_{1,2})(1-\rho_{2,3})}. \tag{28}$$

Acknowledgement

The authors are grateful to Dr. S. Mattoccia, Department of Electronics Computer Science and Systems, University of Bologna, Italy, for providing C++ code of `ZNccEbc`.

References

[1] W. Li and E. Salari, "Successive elimination algorithm for motion estimation," *IEEE Trans. Image Processing*, vol. 4, pp. 105–107, 1995.

[2] H.S. Wang and R.M. Mersereau, "Fast algorithms for the estimation of motion vectors," *IEEE Trans. Image Processing*, vol. 8, no. 3, pp. 435–438, 1999.

[3] T. Kawanishi, T. Kurozumi, K. Kashino, and S. Takagi, "A fast template matching algorithm with adaptive skipping using inner-subtemplates distances," in *ICPR*, 2004.

[4] M. Brunig and W. Niehsen, "Fast full-search block matching," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, no. 2, pp. 241–247, 2001.

[5] C.H. Cheung and L.M. Po, "Adjustable partial distortion search algorithm for fast block motion estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 1, pp. 100–110, 2003.

[6] B. Montrucchio and D. Quaglia, "New sorting-based lossless motion estimation algorithms and a partial distortion elimination performance analysis," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 2, pp. 210–220, 2005.

[7] L. Di Stefano, S. Mattoccia, and F. Tombari, "ZNCC-based template matching using bounded partial correlation," *Pattern Recognition Ltr.*, vol. 26, no. 14, pp. 2129–2134, 2005.

[8] S. Mattoccia, F. Tombari, and L. Di Stefano, "Reliable rejection of mismatching candidates for efficient ZNCC template matching," *IEEE ICIP*, 2008.

[9] A. Mahmood and S. Khan, "Exploiting local auto-correlation function for fast video to reference image alignment," in *IEEE ICIP*, 2008.

[10] A. Mahmood and S. Khan, "Exploiting inter-frame correlation for fast video to reference image alignment," in *ACCV*, 2007.

[11] William P, Saul T, William V, and Brian F, *Numerical Recipes: The Art of Scientific Computing*, Cambridge University Press, Cambridge, UK, 3nd edition, 2007.

[12] R. M. Haralick and L. G. Shapiro, *Computer and Robot Vision*, vol. 2, Addison-Wesley, 1992.

[13] M. Shah and R. Kumar, *Video Registration*, Kluwer Academic Publishers, Boston, 2003.

[14] J.P. Lewis, "Fast normalized cross-correlation," *Vis. Inter.*, pp. 120–123, 1995.

[15] A. Rosenfeld and G. J. Vanderburg, "Coarse to fine template matching," *IEEE Trans. Syst., Man, Cybern.*, vol. 7, no. 2, pp. 104–107, 1977.

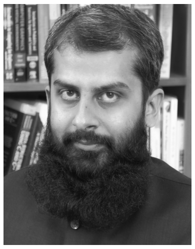
[16] A. Goshtasby, S. H. Gage, and J. F. Bartholic, "A two-stage cross correlation approach to template matching," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 6, no. 3, pp. 374–378, 1984.

[17] T. Koga, K. Iinuma, A. Hirano, Y. Iijima, and T. Ishiguro, "Motion compensated interframe coding for video conferencing," *Proc. National Telecom. Conf.*, pp. G5.3.1–G5.3.5, December 1981.

[18] J.R. Jain and A.K. Jain, "Displacement measurement and its application in interframe image coding," *IEEE Trans. Commun.*, vol. 29, no. 12, pp. 1799–1808, December 1981.

[19] S. Mattoccia, F. Tombari, and L. Di Stefano, "Fast full-search equivalent template matching by enhanced bounded correlation," *IEEE Trans. Image Processing*, vol. 17, no. 4, pp. 528–538, 2008.

Arif Mahmood did B.Sc. Engineering from UET Lahore (www.uet.edu.pk), in 1994. He completed MS in Computer Science from Lahore University of Management Sciences (www.lums.edu.pk) in 2003, where he received gold medal for top academic merit. Currently, he is close to the completion of PhD in Computer Science from LUMS School of Science and Engineering, Lahore, Pakistan. His PhD research is focused on enhancing the speed of computational algorithms, in particular for template matching and classification applications. He was the winner of the Best Student Paper Award at International Conference on Machine Vision, 2007. He has recently joined Punjab University College of Information Technology (www.pucit.edu.pk), Lahore, Pakistan as Assistant Professor.



Sohaib Khan is Associate Professor of Computer Science at the LUMS School of Science and Engineering, Lahore, Pakistan and the founding director of the Computer Vision Lab (http://cvlab.lums.edu.pk) at LUMS. His research interests broadly span the areas of image and video analysis, including image registration, multiple camera surveillance systems, structure from motion and satellite and aerial image processing. Dr Khan earned his PhD degree in Computer Science in 2002 from University of Central Florida, specializing in computer vision. He received BE degree in Electronics Engineering from GIK Institute of Engineering Sciences and Technology, Topi, Pakistan, in 1997.