

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 speedup 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 speedup 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 are guaranteed to yield the same peak location as exhaustive search over the entire solution space. While the speedup obtained is data dependent, we show empirical results of more than 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

Arif Mahmood and Sohaib Khan are with the Department of Computer Science, LUMS School of Science and Engineering, Lahore, Pakistan (<http://cvlab.lums.edu.pk>).

Manuscript received September 05, 2008; Resubmitted May 14, 2009.

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 the 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 algorithmically, 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 one of the spatially neighboring blocks of r_2 . Since local autocorrelation of the reference image is high, r_2 will be 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 a very 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 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

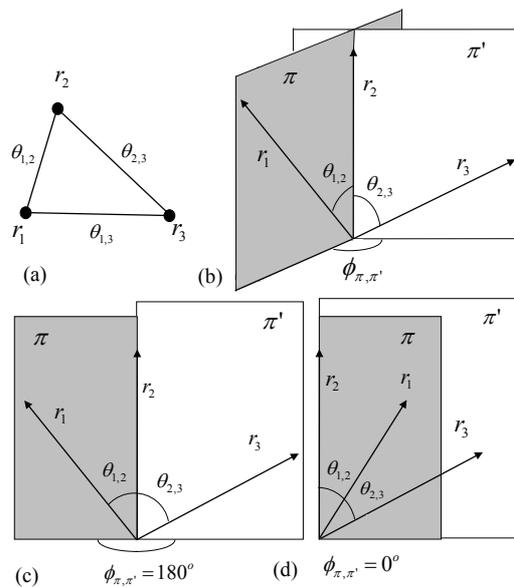


Fig. 1. Triangular inequality for the angular distance measure: (a) Image blocks r_1 , r_2 and r_3 represented as vertices and the angular distance between them is shown as edges of a triangle. (b) $\theta_{1,3}$ depends upon the angle between planes π and π' . (c)-(d) $\theta_{1,3}$ becomes maximum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

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 as described in [12]. Experiments are performed on a wide variety of real image datasets. While the exact speedup of the proposed algorithms varies from experiment to experiment, we have observed speedups 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 the two image blocks, each of size $m \times n$ pixels, and $\psi_{1,2}$ be the cross-correlation between these vectors:

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

r_1 and r_2 may also be considered as vectors in $\mathcal{R}^{m \times n}$ space, and 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, $\psi_{1,2}$:

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

where $\|\cdot\|_2$ denotes the L_2 norm. The angular distance is symmetric, $\theta_{1,2} = \theta_{2,1}$, and bounded between 0° and 180° . In addition, 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):

$$\theta_{1,2} + \theta_{2,3} \geq \theta_{1,3} \geq |\theta_{1,2} - \theta_{2,3}| \quad (3)$$

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 . Therefore the upper and lower triangular bounds are also bounded between 0° and 180° and the triangular inequality in Equation (3) may be written as:

$$\min\{360^\circ - (\theta_{1,2} + \theta_{2,3}), (\theta_{1,2} + \theta_{2,3})\} \geq \theta_{1,3} \geq |\theta_{1,2} - \theta_{2,3}| \quad (4)$$

To convert this inequality in correlation terms, 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:

$$\cos(\theta_{1,2} + \theta_{2,3}) \leq \cos(\theta_{1,3}) \leq \cos(\theta_{1,2} - \theta_{2,3}). \quad (5)$$

This may be rearranged using trigonometric identities to:

$$\begin{aligned} \cos \theta_{1,2} \cos \theta_{2,3} - \sqrt{1 - (\cos \theta_{1,2})^2} \sqrt{1 - (\cos \theta_{2,3})^2} &\leq \cos \theta_{1,3} \\ &\leq \cos \theta_{1,2} \cos \theta_{2,3} + \sqrt{1 - (\cos \theta_{1,2})^2} \sqrt{1 - (\cos \theta_{2,3})^2} \end{aligned} \quad (6)$$

Multiplying this inequality with $(\|r_1\|_2 \|r_2\|_2)(\|r_2\|_2 \|r_3\|_2)$ and simplifying using Equation (2), we get the transitive inequality for cross-correlation:

$$\begin{aligned} \frac{\psi_{1,2} \psi_{2,3} + \sqrt{(\|r_1\|_2 \|r_2\|_2)^2 - \psi_{1,2}^2} \sqrt{(\|r_2\|_2 \|r_3\|_2)^2 - \psi_{2,3}^2}}{(\|r_2\|_2)^2} \\ \leq \psi_{1,3} \leq \end{aligned}$$

$$\frac{\psi_{1,2}\psi_{2,3} - \sqrt{(\|r_1\|_2\|r_2\|_2)^2 - \psi_{1,2}^2}\sqrt{(\|r_2\|_2\|r_3\|_2)^2 - \psi_{2,3}^2}}{(\|r_2\|_2)^2} \quad (7)$$

This inequality provides transitive bounds upon 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 its bias towards brighter regions. Normalized Cross-Correlation (NCC) between image blocks r_1 and r_2 is defined as:

$$\phi_{1,2} = \frac{\psi_{1,2}}{\|r_1\|_2\|r_2\|_2}, \quad (8)$$

Angular distance between two image blocks may also be written in terms of NCC, $\phi: \theta_{1,2} = \cos^{-1}(\phi_{1,2})$.

Transitive inequality given by Equation (6) gets modified for NCC as follows:

$$\begin{aligned} \phi_{1,2}\phi_{2,3} + \sqrt{1 - \phi_{1,2}^2}\sqrt{1 - \phi_{2,3}^2} &\leq \phi_{1,3} \leq \\ \phi_{1,2}\phi_{2,3} - \sqrt{1 - \phi_{1,2}^2}\sqrt{1 - \phi_{2,3}^2} & \end{aligned} \quad (9)$$

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 all linear changes in the signal, is correlation coefficient, defined as:

$$\rho_{1,2} = \frac{\psi_{1,2} - mn\mu_1\mu_2}{\|r_1 - \mu_1\|_2\|r_2 - \mu_2\|_2}, \quad (10)$$

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 follows: $\rho_{1,2} = \cos(\hat{\theta}_{1,2})$, 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:

$$\cos(\hat{\theta}_{1,2} + \hat{\theta}_{2,3}) \leq \cos(\hat{\theta}_{1,3}) \leq \cos(\hat{\theta}_{1,2} - \hat{\theta}_{2,3}). \quad (11)$$

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

$$\begin{aligned} \rho_{1,2}\rho_{2,3} + \sqrt{1 - \rho_{1,2}^2}\sqrt{1 - \rho_{2,3}^2} &\leq \rho_{1,3} \leq \\ \rho_{1,2}\rho_{2,3} - \sqrt{1 - \rho_{1,2}^2}\sqrt{1 - \rho_{2,3}^2} & \end{aligned} \quad (12)$$

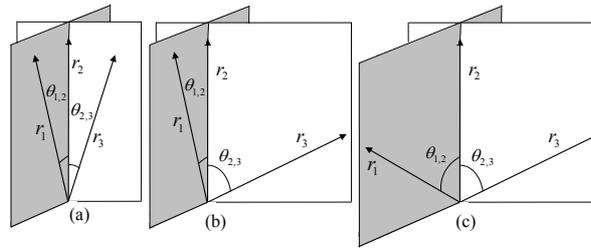


Fig. 2. The tightness of Transitive bounds: (a) Case 1: Both angles are small (b) Case 2: One angle is small and the other is large (c) Case 3: Both angles are large.

This inequality gives bounds upon the correlation coefficient between image blocks r_1 , r_3 , if the values of $\rho_{1,2}$ and $\rho_{2,3}$ are known.

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 (Appendices II and III) 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 speedup different template matching applications.

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. 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 (12), $\rho_{1,2}$ and $\rho_{2,3}$ are the two bounding correlations which constraint 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 speedup, 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 that 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 (5) or (11). In these equations, a tight upper bound means $\cos(\theta_{1,2} - \theta_{2,3})$ assuming a value significantly lesser than +1, which implies $|\theta_{1,2} - \theta_{2,3}|$ has a value significantly larger than 0° . Similarly, lower bound will be tight if $\cos(\theta_{1,2} + \theta_{2,3})$ assumes 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 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.

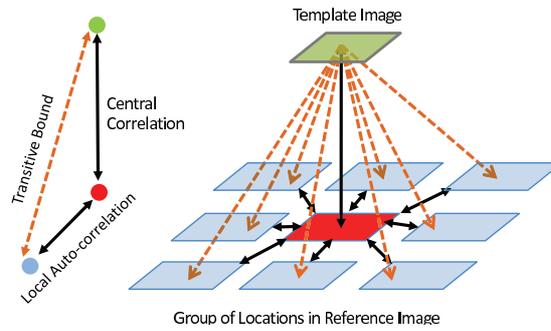


Fig. 3. A 3×3 group of search locations with central location shown red. Solid line arrow from template to red patch represents central correlation and from red patch to blue patches show local autocorrelation, while the dotted arrows represent transitive bounds.

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 speedup. 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 small rectangular groups and compute local autocorrelation (A_S) of the central location with the neighboring locations (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-Reference-TE-Algorithm.

Algorithm 1 Intra-Reference-TE-Algorithm

```

 $A_S \leftarrow$  Local Spatial Auto-correlation

 $C_{\max} \leftarrow$  Initial correlation threshold

for all Groups of search locations do

   $C_C \leftarrow$  correlate(template, central search location)

  if  $C_C > C_{\max}$  then

     $(C_{\max}, i_{\max}, j_{\max}) \leftarrow (C_C, \text{Central location indices})$ 

  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

       $C \leftarrow$  correlate(template, current search location)

      if  $C > C_{\max}$  then

         $(C_{\max}, i_{\max}, j_{\max}) \leftarrow (C, \text{Current location indices})$ 

      end if

    end if

  end for

end for

print  $i_{\max}, j_{\max}, C_{\max}$ 

```

In Algorithm 1, the computation of local autocorrelation is an algorithmic overhead. A standard implementation of the computation of this overhead 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 $O(s_h s_w pq)$, where $s_h \times s_w$ is the size of the group of locations. Since $s_h \times s_w$ is significantly smaller than

$m \times n$, therefore $O(s_h s_w pq) \ll O(mnpq)$. As an example, for a template of 11×11 pixels and using group size of 3×3 search locations, the cost of local autocorrelation computations is only 7.438% of the cost of one template matching in spatial domain. In many template matching applications more than one templates are to be correlated with the same reference image, therefore the cost of local autocorrelation computations being one time cost, is even less significant in such cases.

In our proposed algorithm for A_S computation, correlation between central location, r_c , and an other location, r_n , is computed simultaneously for 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) pixels translated location. We copy only required values in a final *LA-Array* as shown in *LA-Algorithm*. The same process is repeated $s_h s_w$ times, and each time pq integer multiplications and $4pq$ additions are done. Therefore the overall complexity of the proposed algorithm for local spatial autocorrelation computation is $O(s_h s_w pq)$.

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 or nominally moving camera we can exploit their temporal autocorrelation (A_T) to get tight transitive bounds. The concept is illustrated in Figure 4. In this scenario, the central correlation (C_C) is obtained by completely correlating the template image 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 transitive upper and lower bounds upon all search locations in the remaining frames and those match 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 automatically checking the missing components in a circuit board manufacturing facility, the three image patches may happen to be very similar. Therefore we may get both

Algorithm 2 Local Autocorrelation (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  $P_r$ 
     $\triangleright$ 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

```

upper and lower bounds to be tight as given by Case I. In such applications, all search locations where upper bound is less than maximum of the lower bound, may also be skipped without any loss of accuracy. The pseudo code for the complete algorithm is given as Inter-Reference-TE-Algorithm.

This algorithm also carries an overhead but this time it is 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 pq is the size of 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. This summation of products is the cross-correlation between corresponding blocks of the two frames. Since the complexity of running sum algorithm is $O(pq)$ and before that pq integer multiplications were carried out, therefore overall complexity of inter-frame autocorrelation computation

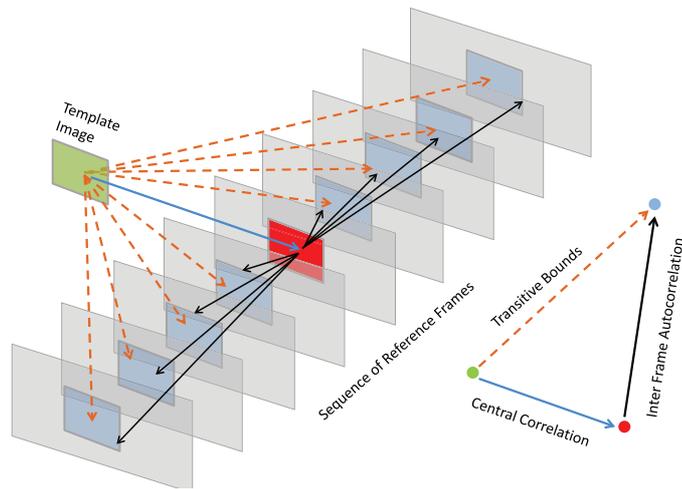


Fig. 4. Exploiting strong inter-frame autocorrelation for fast template matching. Template is fully correlated with only one frame (shown dark red), while for the remaining frames transitive bounds are computed.

is of the order of $O(pq)$, which is significantly smaller than the complexity of one template correlated with one reference frame in $O(mnpq)$. Hence the computational cost of inter frame autocorrelation computation is insignificant as compared to the over all cost of template matching.

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 are 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 speedup 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 central template is shown red and central correlation C_C , is obtained by correlating central template with complete 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 the initial correlation threshold, may be discarded without any loss of accuracy.

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 have developed

Algorithm 3 Inter-Reference-TE-Algorithm

```

 $f_c \leftarrow$  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
         $(C_{\max}, i_{\max}, j_{\max}) \leftarrow (C, \text{Current location indices})$ 
      end if
    end if
  end for

  print  $f_k, i_{\max}, j_{\max}, C_{\max}$ 
end for

```

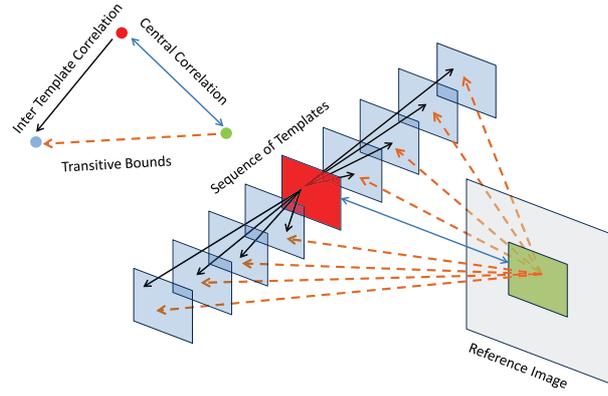


Fig. 5. Exploiting strong inter-template autocorrelation for fast template matching.

a simple algorithm which adapt the length of current group using the percentage computation elimination results of the previous group. Let actual elimination obtained in the $k - 1^{st}$ group be e_{act}^{k-1} , and the maximum possible elimination be e_{max}^{k-1} :

$$e_{max}^{k-1} = (L[k - 1] - 1) / L[k - 1], \quad (13)$$

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 group length may be increased, while if e_{act}^{k-1} is significantly less than e_{max}^{k-1} , then autocorrelation is less than expected, therefore group length, $L[k - 1]$, must be decreased for the next group:

$$L[k] = \begin{cases} L[k - 1] + 2, & \text{if } e_{max}^{k-1} - e_{act}^{k-1} < \delta_l \\ L[k - 1] - 2, & \text{if } e_{max}^{k-1} - e_{act}^{k-1} > \delta_h \\ L[k - 1], & \text{otherwise} \end{cases} \quad (14)$$

where δ_l and δ_h are low and high thresholds upon elimination. Keeping a very low value of δ_l will result an increase in number of groups and hence the number of fully correlated templates, and keeping a high value of δ_h may cause an increase in computational cost due to reduction in elimination.

IV. EXPERIMENTS AND RESULTS

We have implemented the proposed algorithms in C++ and compared with the currently known fast exhaustive template matching techniques including FFT based frequency domain implementation [11], [12] *Zero-mean Bounded Partial Correlation* (ZBPC) [7], *Zero-mean Enhanced Bounded Correlation* (ZNCCbC) [8]

and an exhaustive spatial domain implementation (S_{pat}) [14]. In order to ensure a realistic comparison, we have used only sequential implementations of all algorithms.

Other than correlation based measures, we have also implemented Sum of Absolute Differences (SAD) with *Partial Distortion Elimination* [6] and *Successive Elimination Algorithm* [1] optimizations. Execution times are measured on an IBM machine with Intel Core 2 CPU 2.13 GHz processor and 1GB RAM.

Experiments are divided into four sub-sections. First three sub-sections correspond to the three proposed elimination algorithms applied on the correlation coefficient match measure and in forth sub-section the elimination performance of different correlation based measures is compared with each other. In each sub-section, different datasets are used which are described first, followed by a brief discussion of the experimental setup and then the results of a particular proposed algorithm are compared with existing algorithms in terms of both execution time and computation elimination percentage.

The datasets used in these experiments, implementation codes, experimental setup details and complete results are available on our web site: <http://cvlab.lums.edu.pk/tea>.

A. Exploiting Intra-Reference Auto-correlation

These experiments are performed upon 9 datasets, divided into two groups: Circuit Board (CB) and the Satellite Image (SI) (see Table I). The images to be matched have projective distortions due to difference in viewing geometry. In addition, the reference image for CB group is a very low contrast image (Figure 6) and the reference image for SI group is a washed out image with very high brightness.

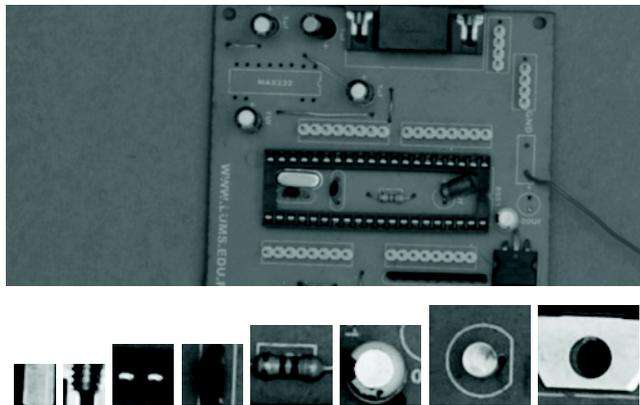
For ZBPC, ZNccEbc and Intra-Reference-TE- Algorithm, initial threshold of 0.80 is used. Average execution time per template for all algorithms is given in Table II. For CB datasets, maximum execution time speedup obtained by Intra-Reference-TE-Algorithm over ZBPC is 13.83 times, over ZNccEbc is 4.63 times, over FFT is 8.00 times, over SAD is 7.15 and over S_{pat} is 14.00 times. For SI group, maximum speedup of Intra-Reference-TE-Algorithm over ZBPC is 14.81, over ZNccEbc is 3.38, over FFT is 3.71 and over S_{pat} is 16.50 times. For SI group, SAD has remained faster than all correlation-coefficient based algorithms. However because of variations in brightness, the accuracy of SAD is less than 2%, while the accuracy of correlation-coefficient has remained 100%.

The local autocorrelation function computations by LA-Algorithm is many times faster than the

TABLE I

DATASET DESCRIPTION FOR EXPERIMENTS WITH INTRA-REFERENCE-TE-ALGORITHM

Dataset	# of Frames	Frame Size	Ref. Size	Avg. ρ_{\max}
CB.1	15	33×33	1429×1796	0.942
CB.2	15	49×49	1429×1796	0.944
CB.3	15	65×65	1429×1796	0.948
CB.4	15	81×81	1429×1796	0.954
SI.1	598	64×64	1000×800	0.859
SI.2	291	80×80	1000×800	0.891
SI.3	283	96×96	1000×800	0.907
SI.4	275	112×112	1000×800	0.944
SI.5	267	128×128	1000×800	0.942

Fig. 6. Circuit Board (CB) dataset: reference image (1429×1796) and templates of various sizes: 33×33 , 49×49 , 65×65 , 81×81 .

Correlation-Elimination-Algorithm (CEA) [9]. An execution time comparison for multiple template sizes is shown in Table III, in which maximum observed speedup of LA-Algorithm over CEA is 62.37 times.

B. Exploiting Inter-Reference Auto-correlation

1) *Experiment on Fast Feature Tracking*: In this experiment, manually extracted features are tracked across Pedestrian (PED) and Cyclist (CYC) video datasets. Both videos were acquired in a typical surveillance scenario. See Table IV for dataset description and Figure 7 for selected dataset display. Both datasets contain dissimilarities produced by human motion and frame to frame illumination variations.

In these experiments, initial correlation threshold is set to 0.70. Average execution time per frame

TABLE II

AVERAGE TIME (SEC) PER TEMPLATE FRAME FOR INTRA-REFERENCE-TE-ALGORITHM AND OTHER EXHAUSTIVE ALGORITHMS

Dataset	IR-TEA	ZBPC	ZNccEbc	FFT	SAD	Spat
CB.1	1.077	7.636	4.991	8.625	2.861	7.817
CB.2	1.667	16.312	7.605	8.610	8.063	16.705
CB.3	2.54	30.807	10.050	8.597	15.373	31.224
CB.4	3.332	46.082	12.175	8.596	23.852	46.663
SI.1	1.120	7.427	3.021	4.164	0.0163	8.143
SI.2	1.353	11.805	3.859	4.185	0.0283	13.045
SI.3	1.485	16.132	4.555	4.172	0.0378	17.823
SI.4	1.633	20.791	5.181	4.192	0.0441	23.117
SI.5	1.740	25.782	5.880	4.154	0.0590	28.715

TABLE III

LOCAL AUTOCORRELATION COMPUTATION TIME IN SECONDS FOR LA-ALGORITHM AND CEA ALGORITHM [9].

SI DataSet	SI.1	SI.2	SI.3	SI.4	SI.5
LAF Time	0.499	0.499	0.499	0.484	0.484
CEA Time	8.937	13.780	18.639	24.216	30.184
CB DataSet	CB.1	CB.2	CB.3	CB.4	-
LAF Time	1.671	1.671	1.671	1.656	-
CEA Time	8.390	18.013	32.730	49.260	-

taken by different algorithms is shown in Table V. Maximum observed execution time speedup of Inter-Reference-TE-Algorithm over ZBPC is 12.47 times, over ZNccEbc is 13.71 times, over FFT is 21.51 times, over SAD is 4.04 times and over Spat is 13.72 times. Computation elimination comparisons are shown in Table VI. Percentage computation elimination achieved by ZNccEbc are larger than Inter-Reference-TE-Algorithm in both experiments, however execution time of ZNccEbc is larger than almost all algorithms. It is because of the fact that on small template sizes the ZNccEbc bound evaluation cost exceeds the savings achieved by the eliminated computations.

2) *Experiment on Fast Component Tracking*: As compared to feature tracking, in this dataset there is no local motion and the component templates are significantly larger in size as compared to the feature templates. Depending upon the template sizes, there are four categories of Component Tracking (CT)

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.1	6	63×63	16	479×640
CT.2	1	178×62	16	479×640
CT.3	1	136×104	16	479×640
CT.4	1	147×63	16	479×640

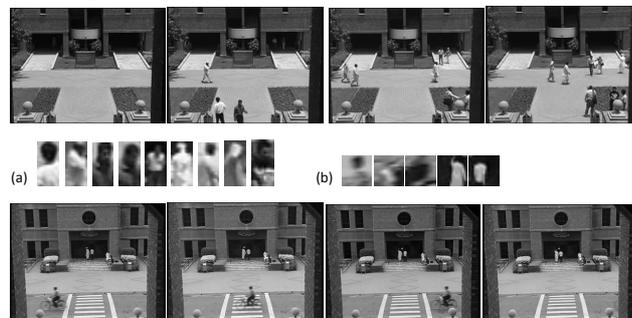


Fig. 7. Fast feature tracking Dataset: (a) Pedestrian dataset: four frames and 9 feature templates. (b) Cyclist dataset: four frames and 5 feature templates.

TABLE V

AVERAGE TIME IN SECONDS PER FRAME FOR INTER-REFERENCE-TE-ALGORITHM AND THE OTHER ALGORITHMS.

Dataset	IR-TEA 3	ZNccEbc	ZBPC	FFT	SAD	Spat
PED	0.159	1.686	1.047	2.877	0.338	1.150
CYC	0.0269	0.369	0.282	0.590	0.0795	0.292
CT.1	0.792	6.66	12.406	10.380	5.388	16.464
CT.2	0.253	2.563	5.127	1.724	3.095	5.533
CT.3	0.206	2.447	6.352	1.730	4.591	7.857
CT.4	0.127	1.18	3.606	1.732	2.379	5.105

TABLE VI

PERCENTAGE COMPUTATION ELIMINATION IN FAST FEATURE TRACKING AND FAST COMPONENT TRACKING EXPERIMENTS

Dataset	TE-Algorithm 3	ZNccEbc	ZBPC	SAD
PED	81.976	89.607	12.570	75.583
CYC	93.748	93.839	8.451	77.259
CT.1	92.249	89.454	24.957	69.192
CT.2	88.149	87.569	8.102	49.063
CT.3	91.029	87.957	19.750	43.889
CT.4	93.162	97.58	29.585	57.396

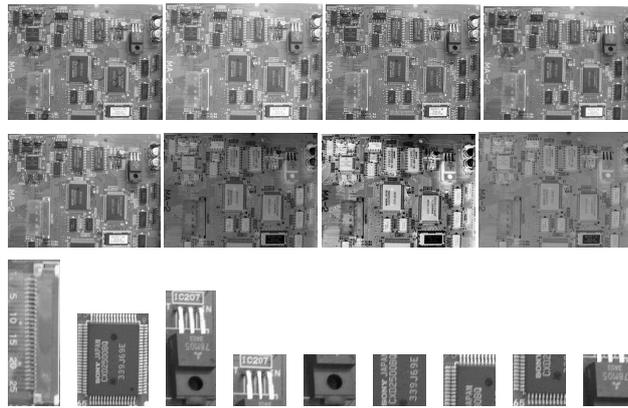


Fig. 8. Fast component tracking experiment datasets: 8 frames of circuit-board images, each of size 479×640 pixels, and component templates of different sizes: 178×62 , 136×104 , 147×63 , 63×63 . Original dataset taken from [15].

dataset: CT1, CT2, CT3 and CT4 (Figure 8 and Table IV). Original images were taken from [15] and following frame to frame variations were produced: affine photometric variations, non-linear photometric variations, complementing, sharpening by edge-enhancements and geometrically transforming the original images.

For all algorithms, average execution time per reference frame is given in Table V. Maximum speedup observed by Inter-Reference-TE-Algorithm over ZBPC is 30.70 times, over ZNccEbc algorithm is 11.88 times, over FFT is 13.54 times, over SAD is 22.19 times and over Spat is 39.92 times. Average computation elimination for each of these algorithms is given in Table VI. Computation elimination achieved by Inter-Reference-TE-Algorithm is significantly larger than any of the other algorithms.

C. Exploiting Inter-Template Auto-correlation

1) *Fast Video Geo-registration*: These experiments are performed upon ten datasets divided into two groups: DS 1 and DS 2 (Table VII and Figure 9). The images to be matched contain dissimilarities due to difference in imaging sensor and geometry. Additional dissimilarities were generated by reducing the dynamic range of templates in DS 1 to one third of the original range and the templates in DS2 were contrast reversed. Contrast reversals are frequently observed if matching is to be done between infra-red and optical imagery.

In DS 1, best match location is the location with maximum correlation and the minimum acceptable threshold is set to 0.80. In case of DS 2, the best match location is the location with minimum negative correlation and the maximum acceptable correlation is set to -0.85. Execution time comparison of Inter-Template-TE-Algorithm (IT-TEA) and other algorithms are given in Table VIII. For DS1, maximum execution time speedup of IT-TEA over ZBPC is 9.77 times, over ZNccEbc is 2.98 times, over FFT is 4.24 times and over Spat is 15.10 times. For DS2, maximum observed speedup of IT-TEA over ZBPC is 10.20 times, over ZNccEbc is 11.65 times, over FFT is 3.022 times and over Spat is 10.21 times. Poor performance of ZBPC and ZNccEbc on DS2 is due to the fact that these algorithms have been developed to find maxima of correlation coefficient, where as in case of DS2 correlation minima has to be searched. Transitive elimination algorithms, however can be used to find the correlation maxima as well as the minima, efficiently.

In some cases, SAD has been found to be faster than all correlation coefficient based algorithms, however we find that, because of intensity distortions, accuracy of SAD over these datasets is less than 1%, where as the accuracy of correlation coefficient has remained 100%.

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 speedup the exhaustive rotation/scale invariant template matching by using the Inter-Template-TE-Algorithm (IT-TEA). These experiments are performed upon optical character recognition dataset using scanned pages from multiple books. The template images consist of 14 letters: {a, c, e, g, i, k, m, o, p, s, v, w, x, z}, which were extracted from one of the scanned image (see Table IX and Figure 10). Each template is rotated from -5° to $+5^\circ$ and scaled from -8% to +8% at a step size of 2%, resulting in 99 rotated/scaled versions. All of these

TABLE VII

DATASET DETAILS USED FOR VIDEO GEO-REGISTRATION EXPERIMENTS

Dataset	# of Frames	Frame Size	Ref. Size	Avg. ρ_{\max}
DS 1.a	734	64 × 64	736 × 1129	0.939
DS 1.b	744	80 × 80	736 × 1129	0.961
DS 1.c	694	96 × 96	736 × 1129	0.963
DS 1.d	641	112 × 112	736 × 1129	0.961
DS 1.e	594	128 × 128	736 × 1129	0.958
DS 2.a	659	64 × 64	1394 × 2152	-0.935
DS 2.b	645	80 × 80	1394 × 2152	-0.921
DS 2.c	648	96 × 96	1394 × 2152	-0.874
DS 2.d	632	112 × 112	1394 × 2152	-0.924
DS 2.e	616	128 × 128	1394 × 2152	-0.794

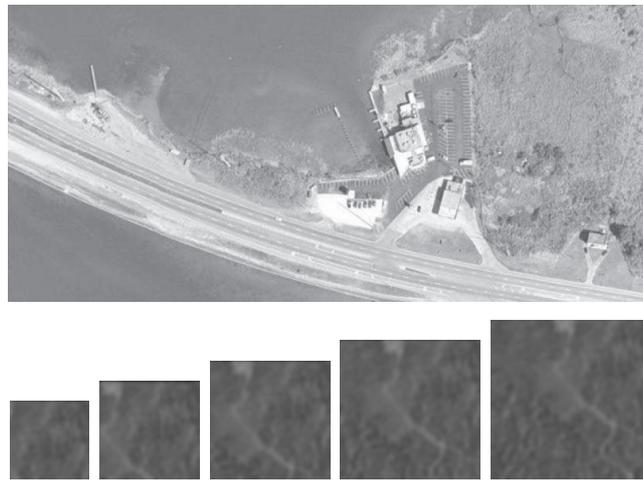


Fig. 9. Video geo-registration experiment dataset DS2. Reference images are taken from earth.google.com and templates from terraserver.microsoft.com.

rotated/scaled versions are exhaustively correlated with each of the 14 reference images, with varying background colors, arbitrary rotations, arbitrary scaling and aliasing effects due to poor scanner resolution with broken and irregular character boundaries.

Out of 99 rotated/scaled versions of each template, only one template is fully correlated with the complete reference image while for all of the remaining templates transitive bounds are computed. In these experiments, initial correlation threshold is set to 0.80. The execution time speedup obtained by IT-TEA is 28.26 times over ZBPC, 36.03 times over ZNCC_{Ebc}, 126.7 times over FFT, 12.71 times over

TABLE VIII

VIDEO GEO-REGISTRATION: EXECUTION TIME IN SEC FOR INTER-TEMPLATE-TE-ALGORITHM AND OTHER ALGORITHMS

Dataset	IT-TEA	ZNccEbc	ZBPC	FFT	SAD	Spat
DS1.a	0.996	2.962	6.413	4.222	0.107	8.455
DS1.b	1.156	3.437	8.575	04.173	0.156	13.587
DS1.c	1.412	4.209	12.738	04.161	0.257	18.552
DS1.d	1.669	4.837	16.315	04.261	0.436	24.017
DS1.e	1.977	4.905	16.710	04.266	0.609	29.855
DS2.a	6.466	19.048	32.163	19.546	2.969	32.847
DS2.b	8.950	32.028	53.303	19.551	4.975	53.760
DS2.c	12.29	48.777	74.399	19.605	7.030	74.932
DS2.d	5.838	68.036	98.432	19.458	9.374	99.027
DS2.e	12.31	87.991	125.17	19.562	11.958	125.74

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

SAD and 29.26 over Spat (Table X). As observed in feature tracking experiment, the performance of ZNccEbc have been significantly degraded in this experiment as well, because on small template sizes, the cost of ZNccEbc elimination test is larger than the benefit from eliminated computations.

D. Performance Comparison of Different Correlation Based Measures

In this subsection we compare the computation elimination performance and the execution times of the three correlation based measures: cross-correlation, NCC and correlation-coefficient, with each other. Comparison is done on seven datasets, five from video geo-registration DS1: a, b, c, d, e (Table VII

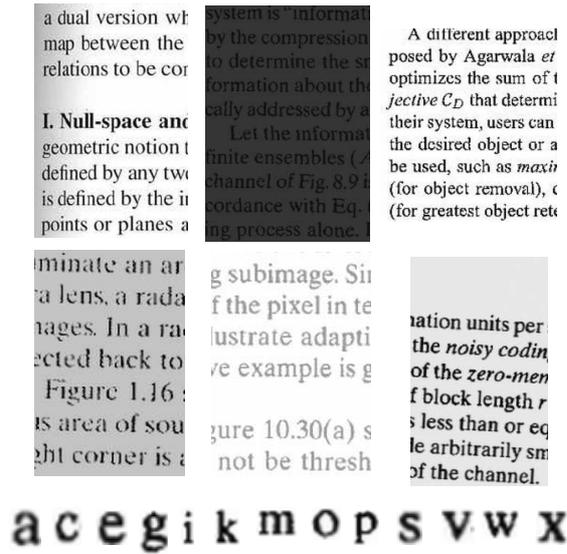


Fig. 10. Rotation and Scale invariant template matching: some portions from different reference images and enlarged view of template images containing small case letters.

TABLE X

ROTATION AND SCALE INVARIANT TEMPLATE MATCHING: AVERAGE EXECUTION TIME IN SEC FOR IT-TEA AND OTHER ALGORITHMS.

Dataset	IT-TEA	ZNccEbc	ZBPC	FFT	SAD	Spat
a	0.0311	0.837	0.613	3.589	0.233	0.603
c	0.0297	0.853	0.634	3.441	0.268	0.642
e	0.0289	0.786	0.565	3.469	0.222	0.582
g	0.0363	0.953	0.888	3.435	0.372	0.898
i	0.0344	0.982	0.492	3.466	0.182	0.490
k	0.0328	1.182	0.927	3.431	0.417	0.930
m	0.0459	0.883	0.902	3.703	0.337	0.946
o	0.0308	0.853	0.665	3.465	0.267	0.689
p	0.0335	1.118	0.928	3.495	0.394	0.929
s	0.0274	0.827	0.514	3.474	0.188	0.509
v	0.0287	0.833	0.669	3.484	0.279	0.688
w	0.0326	0.944	0.912	3.641	0.412	0.954
x	0.0302	0.882	0.634	3.525	0.288	0.640
z	0.0290	0.879	0.649	3.474	0.330	0.644

TABLE XI

EXECUTION TIME (T) (SEC) AND PERCENT ELIMINATION (E) FOR CROSS-CORRELATION (ψ), NCC (ϕ) AND THE ρ .

Dataset	T_ψ	T_ϕ	T_ρ	E_ψ	E_ϕ	E_ρ
DS 1.a	0.278	1.247	0.996	95.369	76.697	83.201
DS 1.b	0.425	1.774	1.156	94.66	70.495	85.322
DS 1.c	0.793	2.002	1.412	92.494	71.425	85.793
DS 1.d	1.205	2.270	1.669	89.251	72.511	86.292
DS 1.e	1.576	2.651	1.977	85.137	72.569	86.495
PED	0.00524	0.00668	0.00758	99.358	82.237	81.976
CYC	0.00345	0.00407	0.00538	97.298	94.315	93.748

) using Inter-Template-TE-Algorithm and two datasets from fast feature tracking experiment: pedestrian (PED) and cyclist (CYC) (Table IV) using Inter-Reference-TE-Algorithm.

In these experiments we observe that cross-correlation is the fastest of the three measures. Maximum speedup obtained by cross-correlation over NCC is 4.48 times and over correlation coefficient is 3.58 times.

V. CONCLUSION

In this work, we have demonstrated that transitive property of correlation can be used to exploit autocorrelation for fast template matching. In order to exploit three different forms of autocorrelation, three different elimination algorithms are formulated. The proposed algorithms have exhaustive equivalent accuracy and are compared with current known fast exhaustive techniques on a wide variety of real image datasets. In our experiments, the proposed algorithms have outperformed the current known algorithms by a significant margin.

APPENDIX I

TRANSITIVE INEQUALITY BASED UPON EUCLIDEAN DISTANCE

Considering the image blocks r_1 , r_2 and r_3 as points in $\mathcal{R}^{m \times n}$, let $\Delta_{i,j}$ be the Euclidean distance between any two of them, where $i, j \in \{1, 2, 3\}$:

$$\Delta_{i,j} = \sqrt{\sum_{x=1}^n \sum_{y=1}^m \left(\frac{r_i(x,y) - \mu_i}{\sigma_i} - \frac{r_j(x,y) - \mu_j}{\sigma_j} \right)^2}. \quad (15)$$

Triangular inequality for Euclidean distance is given by:

$$|\Delta_{1,2} - \Delta_{2,3}| \leq \Delta_{1,3} \leq \Delta_{1,2} + \Delta_{2,3}. \quad (16)$$

Squaring all sides and using following relationship [9]:

$$\rho_{i,j} = 1 - \frac{1}{2}\Delta_{i,j}^2, \quad (17)$$

and simplifying the expression, we get Euclidean distance based transitive inequality:

$$\begin{aligned} (\rho_{1,2} + \rho_{2,3} - 1) + 2\sqrt{(1 - \rho_{1,2})(1 - \rho_{2,3})} &\geq \rho_{1,3} \\ &\geq (\rho_{1,2} + \rho_{2,3} - 1) - 2\sqrt{(1 - \rho_{1,2})(1 - \rho_{2,3})}. \end{aligned} \quad (18)$$

In [9] it has been shown that upper bound based upon angular distance is tighter than the Euclidean distance. Here we show that the angular distance based lower bound is also tighter than the Euclidean distance based lower bound.

Since $0 \leq \Delta_{i,j} \leq 2$ and $-1 \leq \rho_{i,j} \leq +1$, therefore following inequality always holds:

$$1 - \frac{\Delta_{1,2}\Delta_{2,3}}{4} - \frac{1}{2}\sqrt{(1 + \rho_{1,2})(1 + \rho_{2,3})} \geq 0, \quad (19)$$

and from the non-negativity property of the distance measures: $\Delta_{1,2}\Delta_{2,3} \geq 0$. Therefore:

$$\Delta_{1,2}\Delta_{2,3} \left[1 - \frac{\Delta_{2,3}\Delta_{1,2}}{4} - \frac{1}{2}\sqrt{(1 + \rho_{1,2})(1 + \rho_{2,3})} \right] \geq 0. \quad (20)$$

From Equation (17):

$$\Delta_{1,2}\Delta_{2,3} = 2\sqrt{(1 - \rho_{2,3})(1 - \rho_{1,2})}. \quad (21)$$

From (21) and (20):

$$\begin{aligned} &\rho_{1,2}\rho_{2,3} + \sqrt{(1 - \rho_{1,2}^2)(1 - \rho_{2,3}^2)} \\ &\leq (\rho_{1,2} + \rho_{2,3} - 1) + 2\sqrt{(1 - \rho_{1,2})(1 - \rho_{2,3})}. \end{aligned} \quad (22)$$

APPENDIX II

COMPARISON OF LOWER TRANSITIVE BOUNDS

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

$$\Delta_{1,2}\Delta_{2,3} \left[1 - \frac{1}{2} \sqrt{(1 + \rho_{1,2})(1 + \rho_{2,3})} \right] \geq 0, \quad (23)$$

$$\rho_{1,2}\rho_{2,3} - (\rho_{1,2} + \rho_{2,3} - 1) \geq 0. \quad (24)$$

Adding these two inequalities and rearranging the terms:

$$\begin{aligned} & \rho_{1,2}\rho_{2,3} - \Delta_{1,2}\Delta_{2,3} \frac{1}{2} \sqrt{(1 + \rho_{1,2})(1 + \rho_{2,3})} \\ & \geq (\rho_{1,2} + \rho_{2,3} - 1) - \Delta_{1,2}\Delta_{2,3}, \end{aligned} \quad (25)$$

substituting the value of $\Delta_{1,2}\Delta_{2,3}$ from (21):

$$\begin{aligned} & \rho_{1,2}\rho_{2,3} - \sqrt{(1 - (\rho_{1,2})^2)(1 - (\rho_{2,3})^2)} \\ & \geq (\rho_{1,2} + \rho_{2,3} - 1) - 2\sqrt{(1 - \rho_{1,2})(1 - \rho_{2,3})}. \end{aligned} \quad (26)$$

ACKNOWLEDGEMENT

Thanks to Dr. S. Mattocchia, Department of Electronics Computer Science and Systems, University of Bologna, Italy, for providing a sequential implementation of ZNccEbc algorithm.

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," in *IEEE Trans. Circuits Syst. Video Technol.*, 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] Montrucchio and B. Quaglia, "New sorting-based lossless motion estimation algorithms and a partial distortion elimination performance analysis," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, pp. 210–212, 2005.

- [7] di Stefano L., S. Mattoccia, and F. Tombari, "ZNCC-based template matching using bounded partial correlation," *Patr. Rec. Ltr.*, vol. 26, no. 14, 2005.
- [8] S. Mattoccia, F. Tombari, and L. di Stefano, "Reliable rejection of mismatching candidates for efficient ZNCC template matching," *IEEE ICIP*, pp. 849–852, 2008.
- [9] A. Mahmood and S. Khan, "Exploiting local auto-correlation function for fast video to reference image alignment," in *IEEE ICIP*, September 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, 3rd edition, 2007.
- [12] J.P. Lewis, "Fast normalized cross-correlation," *Vis. Inter.*, pp. 120–123, 1995.
- [13] M. Shah and R. Kumar, *Video Registration*, Kluwer Academic Publishers, Boston, 2003.
- [14] R. M. Haralick and L. G. Shapiro, *Computer and Robot Vision*, vol. 2, Addison-Wesley, 1992.
- [15] S. Mattoccia, F. Tombari, and L. di Stefano, "Fast full-search equivalent template matching by enhanced bounded correlation," *IEEE TIP*, pp. 528–538, 2008.