A hierarchical algorithm for fuzzy template matching in emotional facial images

Anisha Halder∗, Rajshree Mandal and Amit Konar
Department of Electronics and Tele-Communication Engineering, Jadavpur University, Calcutta, India

Abstract: The paper aims at developing a hierarchical algorithm for matching a given template of \( m \times n \) on an image of \( M \times N \) pixels partitioned into equal sized blocks of \( m \times n \) pixels. The algorithm employs a fuzzy metric to measure the dispersion of individual feature of a block with respect to that of the template. A fuzzy threshold, preset by the user, is employed to restrict less likely blocks from participation in the matching. A decision tree is used to test the feasibility of a block for matching with the template. The tree at each link examines the condition for fuzzy thresholding for one feature of the image. If the block satisfies the condition, it is passed on to the next level in the tree for testing its feasibility of matching with respect to the next feature. If it fails, the block is discarded from the search space, and the next block from the partitioned image is passed on for examination. The process goes on until all the blocks pass through the decision tree. If a suitable block satisfies all the test conditions in the decision tree, the block is declared as the solution for the matching problem. The ordering of features to be examined by the tree is performed here by an entropy measure as used in classical decision tree. The time-complexity of the algorithm is of the order \( MN/mn \), and the elegance of the algorithm lies in its power of approximate matching using fuzzy conditions. The algorithm has successfully been implemented for template matching of human eyes in facial images carrying different emotions, and the classification accuracy is as high as 96%.

Keywords: Template matching, decision tree, hierarchical search, entropy measure, fuzzy threshold

1. Introduction

Template matching is a well known problem in image understanding and interpretation. It has extensive applications in geographical/geological explorations, medical study, and the like. But its application in emotionally expressive facial component recognition is a novel problem. The classical template matching schemes presumes static frames with distortions in the imagery. But when facial expression and in particular emotional expression is concerned, matching becomes a difficult problem. This is because detecting the nearest matched module/block in the image with respect to a static template sometimes gives rise to false indication, and occasionally misses the necessary target. The problem of image matching in dynamic image frames thus is a challenging problem.

There exists extensive works on image matching using correlation [5, 17], feature extraction [35], boosting process [49], distance transforms [10] and sub block coding [8], moment descriptor [1], Harris detector [6], Bayesian approach [9], and other techniques [4, 7, 11–16, 19, 32–34, 37–40, 41–43]. Hierarchical image matching is also addressed in recent papers [10]. However, we are afraid that there are fewer traces of research in the arena of template matching using hierarchical techniques. The scope of hierarchical template matching on emotionally excited faces, however, is few and far
between. A few relevant papers that employ variants of template matching, including hierarchical techniques, are given below for convenience.

In [49], the authors introduce a novel technique for image matching using concentric sampling features and boosting process. In [31], Lin et al. propose a hierarchical part-template matching scheme using Bayesian approach to segment persons in crowded scenes combining local part-based and global template-based schemes. An alternative shape-based, hierarchical part-template matching algorithm to detect humans by considering local part-based and global shape-template-based schemes has been undertaken in [30]. A hierarchical face detection method has been undertaken in [47], where in the first phase the authors consider a template matching algorithm and in the latter phase consider a two-dimensional Principal Component Analysis (2DPCA) algorithm using a rough classifier and a core classifier. In [26], Li et al. develops a new face detection method based on Quantum Fourier Transform (QFT) average template-pair phase correlation matching.

With the existing techniques of pattern recognition and image processing, hundreds or even more approaches [18, 21–25, 27–29, 45, 46, 48] to handle this problem can be addressed. But the objective here is to identify a very robust algorithm, capable of matching in a finite time, whatever small possible. If such an algorithm can be designed, we would be able to use it for real-time matching in movie frames or even with minor modifications on real-time video. Designing a fast algorithm calls for minimum computations, without losing the target blocks. One approach to solve this problem is to consider matching of template features with block features of a partitioned image in a time-staggered manner, so that important features that eliminate the possibility of matching can be used first to reduce the search space. In this paper, we consider a hierarchical matching that explores the search space in such a time-staggered manner with multiple features, taken one at a time. One question that may be raised is how to detect the order of matching of the features. Here, we employ an entropy measure policy like the one considered in decision tree based learning algorithm, to determine the order of features based on which the matching has to be accomplished.

In this paper, simple statistical features like mean, standard deviation and kurtosis are considered to compare the template with the partitioned blocks in the image. Further, instead of directly matching the image attributes, the features are first mapped on the fuzzy plane and then the comparison is carried out. The fuzzy measure reduces the scope of creeping up of noise in the matching process, and thus improves the robustness of the technique.

The paper has been divided into eight sections. Section 2 gives a small introduction on the principle of template matching. Section 3 gives the fuzzy conditions for approximate matching employed in our algorithm. The basis of hierarchical search and the decision tree learning approach is given in Section 4. The algorithm is given in Section 5. The experimental results are given in Section 6. Section 7 gives the Performance analysis and the conclusions are listed in Section 8.

2. Principle of template matching

In template matching, we need to search a template of \(m \times n\) pixels in an image of \(M \times N\) pixels. A pixel-wise matching of the template over the image definitely gives the best result, but is computationally very expensive and time consuming. The pixel-wise matching scheme is thus prohibited for real-time applications, where time is an important factor. This calls for designing an intelligent search algorithm that alleviates the fundamental premise of pixel-wise template matching. One way of formulating the template matching problem in the present context is to design a feature-based search strategy over a partitioned image of equal block size similar to that of a given template. The exploration should continue until the desired block can be identified. During the exploration phase, the template needs to be matched with the partitioned blocks with respect to the selected features. Selection of features for a general image without any knowledge of the background or context is not always easy. Here, we consider simple statistical features to be determined for each block in the image. A distance metric is defined to match the template features with those of individual image blocks. The simplest distance metric is the Euclidean Distance. However, other distance metric is also used, depending on their suitability in applications. In this paper, we select mean, standard deviation and kurtosis as three basic image attributes. To perform template matching in colour images, here, the above features are evaluated in \(r, g\) and \(b\) planes separately.

We now formally define these parameters with respect to an \(\alpha\) plane where \(\alpha \in \{r, g, b\}\). Let

\[x_{i}^\alpha\]

be the intensity of the \(i^{th}\) pixel on the \(\alpha\) plane, \(mean_{\alpha}\) be the mean value of pixel intensities in a block on the \(\alpha\) plane,
σα be the standard deviation of pixel intensities in a block on the α plane,
kα be the kurtosis of pixel intensities in a block on the α plane,
n gives the total number of pixel intensities in the block.

Definition 1. For a particular block, mean gives the arithmetic average of all the pixel intensities in r-, g-, and b-planes respectively, and is formally given by

\[ \text{mean}_α = \frac{1}{n} \sum_{i=1}^{n} x_α^i \] (1)

Definition 2. The standard deviation indicates a measure of deviation of the pixel intensities \( x_α^i \)s from their mean. The standard deviation of a block of pixel intensities is given as

\[ \sigma_α = \left( \frac{1}{n} \sum_{i=1}^{n} (x_α^i - \text{mean}_α)^2 \right)^{\frac{1}{2}} \] (2)

Definition 3. The kurtosis of a block of pixels on the α plane is given by

\[ k_α = \frac{E((x_α^i - \text{mean}_α)^4)}{\sigma_α^4} \] (3)

where \( E(X) \) is the expectation of the random variable \( X \) defined on a probability space.

The template matching algorithm can be realized by matching the template over equally sized partitioned blocks in the image. If the search is performed on non-overlapped regions in the image, the chances of identifying the target block becomes rare. To avoid this problem, we allow overlapped search with an interleaving of few pixels over the previously selected blocks. The more is the overlap, the better is the localization of the target region, at the cost of extra search time.

3. Fuzzy conditions for approximate matching

Feature based matching of template with partitioned blocks usually determines the distance between the measured features of the template with the respective features of a partitioned block. In many cases, it is observed that the template may not be present in the image in its exact form. This raises a fundamental problem, and is addressed here using a transformation of the measurements into fuzzy memberships. It is apparent that the logic of fuzzy sets has its inherent capability to handle approximate matching. We would explore this particular characteristic of fuzzy sets to perform approximate matching of a given block in an image with a fixed template. The definition of fuzzy set and membership is introduced below.

Definition 4. Let, \( X \) be a universe of measurements. For \( x \in X \), we call \( A \) to be a fuzzy set under the universe \( X \), where

\[ A = \{ x, \mu_A(x) \} \] (4)

where \( \mu_A(x) \) is called the membership of \( x \) in \( A \), where \( 0 \leq \mu_A(x) \leq 1 \).

In this paper, we consider Gaussian type membership function, given by

\[ \mu_A(x) = e^{-\frac{(x - \bar{x})^2}{2\sigma^2}} \] (5)

where \( x \) is a linguistic variable in set \( A \), and \( \bar{x} \) and \( \sigma \) are the mean and standard deviation of \( x \) in set \( A \). Figure 1 provides a Gaussian membership distribution curve.

The significance of selecting Gaussian distribution is briefly outlined below.

The features of the template, here, have been modeled as fuzzy linguistic variables. Usually, for similar templates, the probability in deviation of a feature \( \pm \delta(x) \) from its mean value \( \bar{x} \) is presumed to be equal for large samples. This motivated us to use a Gaussian distribution as the membership distribution \( \mu_A(x) \) for the feature \( x \). Such membership distribution has a peak at the centre of the span of the linguistic variable \( x \), and can easily capture the membership of \( x \) in \( A \), where \( A = \text{EQUAL TO} \bar{x} \). In other words, the Gaussian membership distribution indicates the membership of \( x \) to be...
close to $\bar{x}$. Consequently, when $x = \bar{x}$, the membership is 1, and as $x$ is away from $\bar{x}$, the membership falls off.

Because of the inherent non-linearity in the Gaussian membership functions, numerically close linguistic variables are mapped closer in the fuzzy space. Thus a search of the template on a uniformly noisy image with no background information about the noise characteristics can be performed efficiently using the proposed approach.

The feature based fuzzy matching scheme to be proposed attempts to match the features of the template with those of a block respectively in the membership scale. For example, let $(\bar{x} + \delta)$ be the measurement of a feature in a given block, where $\bar{x}$ is the mean value of the feature obtained from several similar templates. Now, we say that the feature $\bar{x}$ of the template will be close enough to the feature $(\bar{x} + \delta)$ of the block, if

$$|\mu_A(\bar{x}) - \mu_A(\bar{x} + \delta)| \leq \epsilon,$$

where $\epsilon$ is a very small pre-assigned positive quantity. The choice of $\epsilon$ is subjective to specific feature under consideration.

Usually, more than one feature is required to perform the template matching operation. Let $f_1, f_2, \ldots, f_n$ be a set of $n$ features used for template matching. Then, we say that the template will be close enough to a given block with respect to the above features if,

$$|\mu_A(f_1) - \mu_A(f_1 + \delta)| \leq \epsilon_1,$$

$$|\mu_A(f_2) - \mu_A(f_2 + \delta_2)| \leq \epsilon_2,$$

$$\vdots$$

and

$$|\mu_A(f_n) - \mu_A(f_n + \delta_n)| \leq \epsilon_n$$

where $f_i$ is the $i^{th}$ feature in fuzzy set $A_i$, $\bar{f}_i$ is the mean value of the $i^{th}$ feature in the template, $\delta_i$ is the offset in measurement of the feature $f_i$ in a given block, and $\epsilon_i$ is the allowed tolerance level in membership matching of a feature between the template and a given block.

In our experiment, we consider three features: $f_1$ for mean, $f_2$ for standard deviation and $f_3$ for kurtosis of a given block and template.

We experimented by varying $\epsilon_i$ for $i = 1$ to 4 in $[0.6, 0.9]$. Table 1 shows the matching accuracy in percentage by varying $\epsilon_i$ from 0.6 to 0.9. The results given in the table are intuitively supported, as with increasing threshold, the classification accuracy increases.

The above measurements of features are carried out in r-, g- and b- planes separately, and the mean, standard deviation and kurtosis in three planes obtained from a set of thirty templates of left/right eye of a subject are evaluated. Table 2 gives the position of the peak value of the Gaussian curve.

One illustrative membership curve for the feature mean in r- plane is given in Fig. 2 for convenience.

4. Template matching by hierarchical search

An exhaustive feature based matching of the template with individual partitioned blocks in the image definitely yields good results but at the cost of excessive
computational complexity. This excessive complexity can however be reduced significantly by hierarchically matching the features of the partitioned blocks in the image with that of the template. This however calls for ordering of the features in a manner so that the failures in matching can be identified earlier in the search process and final localization of the target block can be undertaken by matching other relevant features.

The hierarchical feature matching, introduced here, consists of both coarse and fine search. The coarse search first identifies the approximate location of the target block in a given image. The fine search is required to identify the exact location of the target block in and around the selected location of the block obtained in the coarse search.

The coarse search, here, is accomplished using a decision tree learning algorithm [44]. In a decision tree, the features (attributes) of a set of exemplar observation are used to classify all the data points of n features into two distinct classes. The nodes in a decision tree denote the features, and the arcs emanating from a node denotes possible values of feature. For example, if 'wind velocity', is a feature to predict atmospheric storm, then the possible values of feature. We intentionally want to keep the pixel difference at the last but one level so that the overall search complexity of the algorithm is less. The features for which Information gain are to be determined so as to rank them for mapping in the decision tree according to their relative importance are mean, std. deviation and kurtosis at r-, g- and b- planes together. To measure the possible values of the above three features, we use the following variables, A, B, C and P as given below.

Let,

*pos* denotes the proportion of positive instances in S

*neg* denotes the proportion of negative instances in S.

We now define Information gain (S, A) [20] where, S is number of samples of positive and negative instances for a goal/target decision and A is a given attribute.

\[
\text{Gain}(S, A) = \text{Entropy}(S) - \sum_{\text{values}(A)} \frac{|S_A|}{|S|} \times \text{Entropy}(S_A)
\]

(6)

where

\[
\text{Entropy}(S) = - \text{pos log}_2(\text{pos}) - \text{neg log}_2(\text{neg})
\]

(7)

The decision tree to be developed includes five levels of hierarchy, where the last level are the Boolean variables ‘yes’/’go’ and the last but one level are the pixel wise difference in intensity between the template and the given block. We intentionally want to keep the pixel difference at the last but one level so that the overall search complexity of the algorithm is less. The features for which Information gain are to be determined so as to rank them for mapping in the decision tree according to their relative importance are mean, std. deviation and kurtosis at r-, g- and b- planes together. To measure the possible values of the above three features, we use the following variables, A, B, C and P as given below.

\[
A = \mu_{EQ-TO-180(\text{mean})} \mu_{EQ-TO-120(\text{mean})} \mu_{EQ-TO-90(\text{mean})}
\]

(8)

\[
B = \mu_{EQ-TO-180(\text{std dev})} \mu_{EQ-TO-120(\text{std dev})} \mu_{EQ-TO-90(\text{std dev})}
\]

(9)

\[
C = \mu_{EQ-TO-\text{a}(\text{kurtosis})} \mu_{EQ-TO-\text{b}(\text{kurtosis})} \mu_{EQ-TO-\text{c}(\text{kurtosis})}
\]

(10)

\[
Z_{ij} = [1 - \mu_{EQ-TO-180(\text{mean})}] + [1 - \mu_{EQ-TO-120(\text{mean})}] + [1 - \mu_{EQ-TO-90(\text{mean})}]
\]

\[
+ [1 - \mu_{EQ-TO-180(\text{std dev})}] + [1 - \mu_{EQ-TO-120(\text{std dev})}] + [1 - \mu_{EQ-TO-90(\text{std dev})}]
\]

\[
+ [1 - \mu_{EQ-TO-\text{a}(\text{kurtosis})}] + [1 - \mu_{EQ-TO-\text{b}(\text{kurtosis})}] + [1 - \mu_{EQ-TO-\text{c}(\text{kurtosis})}]
\]

(11)

\[
P = \text{the first n elements of the sorted } Z_{ij} \text{ in ascending order}
\]

Table 3 below shows the Entropy and Information Gain calculation for 10 images. Here, \( p \) denotes the number of positive instances, while \( n \) denotes the number of negative instances. Let us now consider the case...
Table 3

<table>
<thead>
<tr>
<th>No. of blocks</th>
<th>Entropy (std. dev &gt;0.9)</th>
<th>Entropy (mean &gt;0.9)</th>
<th>Entropy (kurtosis &gt;0.9)</th>
<th>Information Gain (mean)</th>
<th>Information Gain (Std. dev)</th>
<th>Information Gain (kurtosis)</th>
</tr>
</thead>
<tbody>
<tr>
<td>609</td>
<td>1 149</td>
<td>1 105</td>
<td>1 452</td>
<td>0.0041–0.0175</td>
<td>0.0033–0.0175</td>
<td>0.000669–0.0179</td>
</tr>
<tr>
<td>599</td>
<td>1 118</td>
<td>1 125</td>
<td>1 420</td>
<td>0.0036–0.0178</td>
<td>0.0040–0.0178</td>
<td>0.000846–0.0178</td>
</tr>
<tr>
<td>406</td>
<td>1 160</td>
<td>1 151</td>
<td>1 390</td>
<td>0.0032–0.0176</td>
<td>0.0031–0.0176</td>
<td>0.00108–0.0176</td>
</tr>
<tr>
<td>710</td>
<td>1 161</td>
<td>1 146</td>
<td>1 499</td>
<td>0.0031–0.0153</td>
<td>0.0029–0.0153</td>
<td>0.00095–0.0153</td>
</tr>
<tr>
<td>610</td>
<td>1 145</td>
<td>1 110</td>
<td>1 445</td>
<td>0.0045–0.0172</td>
<td>0.0032–0.0172</td>
<td>0.00077–0.0175</td>
</tr>
<tr>
<td>603</td>
<td>1 115</td>
<td>1 134</td>
<td>1 415</td>
<td>0.0038–0.018</td>
<td>0.0039–0.018</td>
<td>0.00088–0.018</td>
</tr>
<tr>
<td>702</td>
<td>1 158</td>
<td>1 135</td>
<td>1 376</td>
<td>0.0028–0.0148</td>
<td>0.0025–0.0148</td>
<td>0.00089–0.0148</td>
</tr>
<tr>
<td>714</td>
<td>1 165</td>
<td>1 148</td>
<td>1 502</td>
<td>0.0033–0.0151</td>
<td>0.0031–0.0151</td>
<td>0.00096–0.0151</td>
</tr>
<tr>
<td>653</td>
<td>1 148</td>
<td>1 115</td>
<td>1 462</td>
<td>0.0035–0.0186</td>
<td>0.0045–0.0186</td>
<td>0.00086–0.0186</td>
</tr>
<tr>
<td>352</td>
<td>1 111</td>
<td>1 128</td>
<td>1 410</td>
<td>0.0035–0.0175</td>
<td>0.0033–0.0175</td>
<td>0.00086–0.0175</td>
</tr>
</tbody>
</table>

of searching the eye region in an emotionally aroused face of a subject. In our experiment, we considered 10 facial images of a subject having different emotions. First the image is divided into a number of equal sized partitions called blocks. The number of positive instances in each case is 1 as there can be only one eye block and the rest can be considered to be negative instances. The Information Gain and Entropy value for the three features to be greater than 0.9 are calculated using (6) and (7) respectively. It is apparent from this table that the information gain is the largest for mean followed by that of std. deviation and that of kurtosis. Thus, mean, std. deviation and kurtosis are organized at successive level of the decision tree, starting at the root.

Besides the above, pixel wise difference is included at the lowest decision level of the tree, as given in Fig. 3, this is deliberately kept in the last stage of hierarchical search template matching algorithm because if pixel wise difference is carried out first, then our method will finally narrow down to the classical method of template matching which requires huge computational cost. As the objective of our algorithm is for real time applications, pixel wise difference is carried out on a localized area around the selected blocks. This results in a finer and more accurate search technique. In Fig. 3, we consider matching of a block when the parameters A, B, C exceed 0.9 and n is less than 10. Any time one of the condition listed in the left most leading edge of the tree violates the prescribed conditions the block on which the matching was undertaken is discarded from the list.

In case there exists more than one block of an image that satisfies the matching criteria in the first three steps i.e., A>0.9, B>0.9 and C>0.9, then we need to identify the target block by pixel wise matching between the template and the blocks so far selected by the first three steps. When number of such selected block is high, we select the best ten among selected blocks based on their measure $Z_{ij}$. A block with smaller $Z_{ij}$ is given preference to other blocks having relatively larger $Z_{ij}$. Thus pixel wise matching of the template with 10 blocks reduces the computational overhead of the algorithm.

5. Proposed algorithm

The algorithm includes a coarse search followed by a fine search. The coarse search starts with evaluating A for all blocks of $m \times n$ pixels in a given image of $M \times N$ pixels with an interleaving of $m/4$ pixels for row-blocks and $n/4$ pixels for column blocks.

So, $A$ is evaluated for $\left( \frac{M}{4} - 1 \right) \left( \frac{N}{4} - 1 \right) = M$.
The blocks satisfying \( A > 0.9 \) are passed on to the next stage, and \( B \) is evaluated for these blocks. Those blocks satisfying \( B > 0.9 \) are selected and passed on to the next stage. Now, \( C \) is evaluated for these blocks, and those satisfying \( C > 0.9 \) are considered as the nearest matched block with the template. Finally, the coarse search evaluates \( Z_i \) for the block \( B_j \) that satisfied \( C > 0.9 \). The \( Z_i \) \( \forall \) i, j are sorted in ascending order, and the indices \((i, j)\) for the best 10 blocks are saved in a set \( P \).

The fine search evaluates Euclidean distance between the template and all blocks with indices sorted in \( P \) and their neighborhood.

Let us consider an example to determine the best approximate match for a right eye template in a given image. In order to do so, first the skin region is extracted based on color. This algorithm is executed on r-, g-, and b-planes separately. As the proposed algorithm is based on hierarchical template matching, a coarse to fine search method is employed. The coarse search and the fine search algorithm are formally given next.

---

**Input:** A given template of \( m \times n \) pixels to be searched on an image of \( M \times N \) pixels.

**Output:** The block with minimum pixel-wise unsigned difference with the template.

---

**Coarse Search**

**Begin**

Determine mean, std. deviation and kurtosis for the template;

\[ S \leftarrow \emptyset \] // a set for holding values of \( Z_{ij} \)

For \( j \leftarrow 1 \) to \( (4M/m - 1) \) with an interleaving of \( m \times 4 \) pixels do Begin

For \( j \leftarrow 1 \) to \( (4N/n - 1) \) with an interleaving of \( n \times 4 \) pixels do Begin

Find \( \mu, \sigma, k \) for the block for r-, g-, b- plane

If \( A > 0.9 \) do Begin

Find \( \mu_{r/g/b} \) for the block for r-, g-, b- plane

If \( B > 0.9 \) do Begin

Find \( \mu_{r/g/b} \) for the block for r-, g-, b- plane

If \( C > 0.9 \) do Begin

Find \( Z_i \)

\( S \leftarrow S \cup Z_{ij} \)

End If

End If

End If

End For

End For

Sort \( S \) in ascending order of \( Z_{ij} \)

Save the indices \((i, j)\) for the first 10 elements of \( S \) in set \( P \)

---

**Fine Search**

**Begin**

For \( p \in P \) do Begin

For block, in the neighbourhood \( N_i \) of \( p \)

Find pixel-wise Euclidean distance \( d_{ij} \) between the selected block and the template

Save the smallest distance \( d_{ij} \) in set \( D \)

End For

End For

Find the smallest element in \( D \) and print the block index \((i, j)\)

**End**
Complexity: Given an image of \( M \times N \) pixels, and a template of \( m \times n \) pixels, the template is rolled over the image with an interleaving of \( m/4 \) along the row and \( n/4 \) along the column. So, the total no. of matching of the template with the image is 
\[
\left( \frac{m}{2^{1}} \right) \left( \frac{n}{2^{1}} \right) \approx 16 \left( \frac{MN}{mn} \right)
\]

The coarse search is performed in three steps at different levels of the tree. At the root level, the complexity is \( 16 \left( \frac{MN}{mn} \right) \). However, only a few blocks of the image are passed on to the next level, when the fuzzy membership of the respective statistical variable (here mean), is within 10% fall-off from the peak of the curve. Let the range of variable \( x \) for which the membership is within this 10% fall-off from the peak be \( 2a \). We know that the total \( x \)-span which approximately covers 99% of the range of \( x \) is \( 6\sigma \), where \( \sigma \) is the standard deviation of the variable \( x \). So, the expected number of blocks to be passed on to the next level in the tree is given by \( \frac{M}{a} \left( \frac{MN}{mn} \right) \) of the Gaussian curve. It can be shown that again expected number of blocks to be passed on to the next descendent level is \( \left( \frac{m}{2^{2}} \right)^{2} \left( \frac{MN}{mn} \right) \). So, the expected complexity of the coarse search is given by
\[
TCOARSE = (16 \left( \frac{MN}{mn} \right) \{(1 + \left( \frac{2a}{6\sigma} \right)^{2})\})
\]

To determine the complexity of the fine search, we first determine a zone around the selected block identified by the previous coarse search procedure. In this paper, for a given template size of \( m \times n \), the neighborhood around the centroid of the selected block is of \( (m/2, n/2) \) and the template is searched in this region with an interleaving of \( m/16 \) along the row and \( n/16 \) along the column. The complexity to match the template with the above interleaving is found to be
\[
\left( \frac{3m}{16} \right) \left( \frac{3n}{16} \right) = 529 \text{ (fixed)}
\]
In our experiment, we select 10 such blocks for finer matching. Thus complexity of fine search procedure is \( O(5290) \).
\[
TFINE = O(5290)
\]

The total complexity is given by \( TCOARSE + TFINE \), which is \( O(MN/mn) + O(5290) \). As a specific example, when \( M = 640, N = 480, m = 40, n = 30 \), the total complexity is obtained as \( O(162) + O(5290) \), which means the fine search has much more complexity than the coarse counterpart. It can be verified that the overall complexity of the algorithm is approximately \( 21 \times MN/mn \) for the given settings of \( M, N, m, \) and \( n \). Thus the expected time-complexity of the hierarchical search is \( O(21 \times MN/mn) \). (21)

6. Experiments and computer simulation

The work was undertaken in Artificial Intelligence Laboratory of Jadavpur University. The experiments were conducted with 20 subjects whose facial expressions conveying different emotions such as happiness, anger, fear and sadness, were captured. Before template matching was carried out, the skin region was first detected. This is done in order to localize the search space for template matching. Skin region detection is carried out in the HSV color model. Two parameters namely \( x \) and \( y \) are chosen using the relations:
\[
x = 0.146 \times H - 0.291 \times S + 0.439 \times V + 128 \quad (14)
y = 0.439 \times H - 0.368 \times S - 0.071 \times V + 128 \quad (15)
\]

For each and every pixel, the parameters \( x, y \) and \( H \) are determined, and if they lie in the range given below, it is considered as skin pixel.
\[
140 \leq y \leq 165
140 \leq x \leq 195
0.01 \leq hue \leq 0.1
\]

After the skin region is extracted, our next task is to identify the block with maximum resemblance with the given template. In our experiment, we have taken an eye template of the subject when he/she is emotionally relaxed. With this template, eye region of the subject conveying different emotional expressions is identified with the help of the hierarchical template matching algorithm as mentioned in the previous section. This was repeated for ten individuals and we observed that observed that in most of the cases the eye region matches successfully. The success rate of this experiment is found to be 96.093%.

Figure 4a is an illustrative facial expression of a person in relaxed state. The corresponding left eye template manually extracted is given in Fig. 4b. Template matching is then performed on facial expression of same person, conveying different emotions. It is observed that the left eye in the third column of Fig. 5 is correctly identified for the given emotional expressions.

The right and left eyes of a subject in relaxed state is now identified, and template matching is performed on
their facial expressions conveying different emotions, and the target blocks are shown in Figs. 6 and 7 respectively. The matching score of the right and left eyes with respect to the eye block in relaxed state are given in Tables 4 and 5. The experiment was performed on 128 images. Out of them the eye region was detected correctly for 123 images. The accuracy of our algorithm is measured to be 96.093%.

The experiment is now repeated to identify the target block in group photographs of three people. In Fig. 8, the right eye template of a subject is manually extracted from her facial expression under relaxed state. The eye template is then searched in group photographs of left column in Fig. 9 under different emotional expressions.
We obtained same emotional expression of all the three people by audio-visual stimulus developed in our previous experiment on emotional research [2]. Interestingly, in all the four emotional setting, the target blocks are correctly identified.

Table 4
Matching score of right eyes conveying different emotions with the eye in the relax state

<table>
<thead>
<tr>
<th></th>
<th>Happy</th>
<th>Sad</th>
<th>Fear</th>
<th>Anger</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.708</td>
<td>0.806</td>
<td>0.805</td>
<td>0.567</td>
<td></td>
</tr>
<tr>
<td>0.656</td>
<td>0.772</td>
<td>0.720</td>
<td>0.567</td>
<td></td>
</tr>
<tr>
<td>0.75</td>
<td>0.69</td>
<td>0.712</td>
<td>0.617</td>
<td></td>
</tr>
<tr>
<td>0.701</td>
<td>0.691</td>
<td>0.723</td>
<td>0.567</td>
<td></td>
</tr>
<tr>
<td>0.703</td>
<td>0.612</td>
<td>0.758</td>
<td>0.567</td>
<td></td>
</tr>
</tbody>
</table>

Table 5
Matching score of left eyes conveying different emotions with the eye in the relax state

<table>
<thead>
<tr>
<th></th>
<th>Happy</th>
<th>Sad</th>
<th>Fear</th>
<th>Anger</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.712</td>
<td>0.726</td>
<td>0.608</td>
<td>0.563</td>
<td></td>
</tr>
<tr>
<td>0.725</td>
<td>0.796</td>
<td>0.593</td>
<td>0.607</td>
<td></td>
</tr>
<tr>
<td>0.695</td>
<td>0.726</td>
<td>0.612</td>
<td>0.593</td>
<td></td>
</tr>
<tr>
<td>0.873</td>
<td>0.405</td>
<td>0.72</td>
<td>0.564</td>
<td></td>
</tr>
</tbody>
</table>

The detected blocks are shown in Fig. 10. Experiments reveal that for group photographs, the accuracy is 89.06%.

7. Performance analysis

In image matching, the most popular and well-known technique is the Correlation method. The Correlation method follows the formula

$$ cor = \frac{\sum_{i=0}^{N-1} (T_i - \bar{T})(I_i - \bar{I})}{\left(\sum_{i=0}^{N-1} (I_i - \bar{I})^2 \cdot \sum_{i=0}^{N-1} (I_i - \bar{I})^2\right)^{0.5}} $$  \hspace{1cm} (16)

Where, $T$ is the Template and $I$ is the given image and $N$ is the number of pixels in the Template.

The value of ‘$cor$’ lies in the range -1 to +1. A larger value indicates a stronger relationship between the two images.

Figure 11 shows the complexity comparison of our proposed algorithm with correlation method.

Table 6 shows the complexity comparison of our proposed algorithm with other existing methods. It is apparent from the table that the complexity obtained from our proposed algorithm is less as compared to other existing methods.

We now define accuracy in template matching to compare relative performance of different matching algorithms. The accuracy in the present context is evaluated by determining successful match divided by total number of trials for matching. A factor of 100 is used to convert accuracy in percentage. In Table 7 we present a comparison of our proposed template matching scheme with existing ones with respect to percentage accuracy. It is apparent from the table that the proposed method outperforms the classical techniques.
Fig. 9. Detection of skin region and left eye using template in Figure 2.10.b under different emotional states (H: Happy, S: Sad, A: Anger, F: Fear).

Fig. 10. Detected eye blocks.

for template matching for two distinct class of problems: detection of a single target face in a given image containing a single person, and detection of single target face in an image containing faces of different subjects.

8. Conclusion

The paper introduced a hierarchical approach to template matching. The proposed algorithm for template matching is capable of detecting both noisy and distorted partitioned image blocks similar to that of the template. Because of its hierarchical structure, the algorithms is highly time efficient, and outperforms most of the popular techniques for template matching. Experimental results confirm that percentage matching accuracy of the proposed technique is as high as around 96%. Because of its high matching accuracy and low computational overhead, the proposed algorithm is a good choice for real time image matching. The experimental results further reveal that the proposed technique is capable of approximate identifying the target block in a given image. Further, due to its inherent capability of approximate matching, the algorithm can be employed for matching of salient facial attributes, such as eye or lip with those of partitioned image blocks, where the emotional content of the template need not match with that of the partitioned blocks containing the desired facial attribute.

The hierarchical matching algorithm presented here first identifies the coarse position of the template in the image by a global matching, and later employs a local search/matching to get the correct position of the
template in the given image. The better the result of global search, the lesser is the complexity of the local search in the latter phase. The simplest approach to improve the performance of the global matching is to consider more informative features while matching the template with the image. This calls for identification of the right features. Further, if the selected features are free from rotational invariance, we could employ them to identify skewed target blocks as well. Kernel-based sampling function (KBSF) for instance, is an important parameter free from rotational invariance [49]. Employing KBSF in template matching would have an added advantage in natural face matching in a scene without restricting the subjects to keep their heads vertical. These systems would replace current office automations used for facial signature analysis.

Table 6
Complexity comparison over other techniques

<table>
<thead>
<tr>
<th>Technique</th>
<th>Imaging parameters</th>
<th>Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Template Matching based on Polynomial Approximation [36]</td>
<td>Input image is M × N pixels</td>
<td>O(MN² + L²)</td>
</tr>
<tr>
<td>Image Matching based on Sub-block Coding [8]</td>
<td>Template size: N × N</td>
<td>O(M²)</td>
</tr>
<tr>
<td>Sum of squared differences method</td>
<td>Image size: M × N</td>
<td>(N - M + 1)²</td>
</tr>
<tr>
<td>QFT Phase-only Correlation Template Match [26]</td>
<td>Input image: M × N</td>
<td>O(log(MN))²</td>
</tr>
<tr>
<td>Fast Fourier Transform [26]</td>
<td>Template size: m × n</td>
<td>O(2²m²n²)</td>
</tr>
<tr>
<td>Pattern matching for rotation and scaling space [49]</td>
<td>Template size: m × m</td>
<td>O(m²n²sr)</td>
</tr>
<tr>
<td>Hierarchical Algorithm for Approximate Template Matching</td>
<td>Input image: M × N Template size: m × n</td>
<td>O(MN/mn)</td>
</tr>
</tbody>
</table>

Table 7
Comparison of percentage accuracy in template matching

<table>
<thead>
<tr>
<th>Author’s name/Method used</th>
<th>Accuracy for single face detection (%)</th>
<th>Accuracy for many face detection (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wang et al. [47] template matching with 2DPCA algorithm</td>
<td>92</td>
<td>85.96</td>
</tr>
<tr>
<td>Li et al. [26] QFT</td>
<td>96</td>
<td>91</td>
</tr>
<tr>
<td>Aiping et al. [3] Skin Colour Segmentation and calculation of Hausdorff distance</td>
<td>92.6</td>
<td>86.4</td>
</tr>
<tr>
<td>Proposed method Hierarchical Algorithm for Fuzzy Template Matching</td>
<td>96.093</td>
<td>89.06</td>
</tr>
</tbody>
</table>

References
[31] L. Ma, Y. Sun, N. Feng, Z. Liu, Image fast template matching algorithm based on projection and sequential similarity detect- ing, harbin institute of technology at Weihai, China, 2009 Fifth International Conference on Intelligent Information Hiding and Multimedia Signal Processing, pp. 957–960.