Extraction of Hybrid Trace Features with Evolutionary Computation for Face Recognition

Nan Liu, Student Member, IEEE, Han Wang, Senior Member, IEEE

Abstract—The Hybrid Trace Features (HTF), a new face representation, is proposed for face authentication system. Trace transforms of multiple Trace functionals are used to construct the HTF, and Genetic Algorithms is implemented as the data fusion tool. In addition, rotation based Hybrid Trace Features (r-HTF) is also introduced as facial features. The systemic fusion tool. In addition, rotation based Hybrid Trace Features (HTF), and Genetic Algorithms is implemented as the data transforms of multiple Trace functionals are used to construct representation, is proposed for face authentication system. Trace triple features and authentication [3]. Except the invariant but received few attentions [2] in the field of face recognition.

1. INTRODUCTION

The Trace transform [1] was proposed several years ago, but received few attentions [2] in the field of face recognition and authentication [3]. Except the invariant triple features, Trace transform offers an alternative representation of a 2D grey object [4]. Therefore, depending on the choice of functional used, it may be suitable for making more explicit subtle differences between images of the same object. That is why we investigate the potential usage of the Trace transform for face recognition. Some works have been done in recognizing face images using Trace transform [5]. The authors convert the 2D Trace transform to a binary edge map by setting thresholds. In addition, more efforts are put on the shape matching. In our researches, we focus on discovering properties of 2D Trace images, and intend to utilize the advantages of various Trace functionals.

In this paper, Hybrid Trace Features, a novel face representation is proposed. Section II presents a brief introduction of Trace transform together with a list of Trace functionals used in the experiments. The extraction of Hybrid Trace Features is described in details in section III. The variation of HTF, called rotation based Hybrid Trace Features, is proposed in section IV. Followed by the section V with the contents of introduction to k-Nearest Neighbor algorithm and the face recognition system architecture. Experimental results are presented and conclusion is made in the last two sections.

II. THE TRACE TRANSFORM FOR IMAGE REPRESENTATION

A. Brief Introduction

The Trace transform [1], a generalization of the Randon transform, is a new tool for image processing which can be used for recognizing objects under transformations, e.g. rotation, translation and scaling. The statement can be understood as follows: We call the original coordinate system of the image $C_1$ and the distorted $C_2$, $C_2$ is obtained from $C_1$ by rotation by angle $-\theta$, scaling of the axes by parameter $\nu$ and by translation with vector $(-s_0 \cos \psi_0, -s_0 \sin \psi_0)$. Now, suppose that we have a 2D image $F$ which is viewed from $C_1$ as $F_1(x, y)$ and from $C_2$ as $F_2(\tilde{x}, \tilde{y})$. $F_2(\tilde{x}, \tilde{y})$ can be considered as an image constructed from $F_1(x, y)$ by rotation by $\theta$, scaling by $\nu$, and shifting by $(s_0 \cos \psi_0, s_0 \sin \psi_0)$.

Fig. 1. The face image and its Trace transform (Trace functional 6 in Table 1 is used)

It can be shown that in the new coordinate system, straight lines will still appear as straight lines. That is, linear transformations preserve lines, i.e., an image is gliding along lines when it is linearly transformed. If a line in $C_2$ is parameterized by $(\phi, p, t)$, in the old coordinate system, $C_1$, its parameters are:

\[
\phi_{old} = \phi - \theta \quad (1)
\]

\[
p_{old} = \nu[p - s_0 \cos(\psi_0 - \phi)] \quad (2)
\]

\[
t_{old} = \nu[t - s_0 \sin(\psi_0 - \phi)] \quad (3)
\]

Consider scanning an image with lines in all directions. Denote by $T$, the set of all these lines. The Trace transform is a function $g$ defined on $T$ with the help of $T$ which is some functional of the image function when it is considered as a function of variable $t$, $T$ is called the trace transform. If $L(C_1; \phi, p, t)$ is a line in coordinate system $C_1$, then

\[
g(F; C_1; \phi, p) = T(F; C_1; \phi, p, t) \quad (4)
\]

where $F(C_1; \phi, p, t)$ means the values of the image function along the chosen line. Taking this functional, we eliminate variable $t$. The result is a two-dimensional function of the variable $\phi$ and $t$ and can be interpreted as another image defined on $T$. In stead of using a triple feature [1], we make use of the Trace transform of the image as facial features. Next, we will introduce some Trace functionals and show the variations among different Trace transforms.
TABLE I

<table>
<thead>
<tr>
<th>fnl</th>
<th>Trace Functionals $Tf(t)$ and descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$\int_0^p {f(t)}^q dt$, $p,q=1/2,p$</td>
</tr>
<tr>
<td>2</td>
<td>$\int f(t) dt$, one dimensional numerical gradient of $t$, $f(t)=[(t_2-t_1), (t_3-t_2), \ldots, (t_n-t_{n-1})]$</td>
</tr>
<tr>
<td>3</td>
<td>median$_n{f(t),</td>
</tr>
<tr>
<td>4</td>
<td>$\int_0^\infty {\frac{d}{dt}\mathcal{M}{f(t)}} dt$. $\mathcal{M}$ is a median filtering operator, using a local window of length 3, and differentiation means taking the difference of successive samples</td>
</tr>
<tr>
<td>5</td>
<td>$median{c</td>
</tr>
<tr>
<td>6</td>
<td>$\int_0^\infty c</td>
</tr>
<tr>
<td>7</td>
<td>$\int_0^\infty r</td>
</tr>
</tbody>
</table>

B. The Trace Functional

When an image is transformed to the trace transform space, its values are proportional to grey scales whose value is the physical meaning of face images. The resultant Trace transform depends on the functional we used. Different Trace transforms can be produced from an image using different functionals $T$. Eight invariant Trace functionals are used in our implementation, they are shown in Table I. The designation median$_n\{x, w\}$ means the weighted median of sequence $x$ with weights in the sequence $w$. For example, $median\{2, \bar{8}, 1, 6, \bar{2}, 2, 3, 2\}$ indicates the median of numbers 2,8,1, and 6 with corresponding weights 2,2,3, and 2. This means the standard median of numbers 2,8,1,6,8,8, i.e., the median of the ranked sequence 1,1,1,2,2,6,8,8. The result is 2. More Trace functionals are available for researches [5].

III. EXTRACTION OF THE HYBRID TRACE FEATURES

A. Trace Transform Vectorization

In this section, we introduce a novel face representation based on Trace transform, using function $g(F; C; \phi, p)$, hereafter simply called vectorized Trace transform ($\nu$).

As an example, we implement any tracing functional $fnl$ in Table I on the image, then we can obtained a 2-D invariant Trace transform $g^{fnl}(F; C; \phi, p)$ to represent the original image. Along $\phi$ axis, we divide the 2-D function into $\phi$ vectors to form a 1-D string $g$ called vectorized Trace features.

$$g^{fnl} = [P_1, P_2, \ldots, P_o]^T$$
$$= [g^{fnl}(F; C_1, 1, p), \ldots, g^{fnl}(F; C_1, \phi, p)]^T$$ (5)

However, the string is in a very high dimension, thus we will use feature extraction algorithms to reduce the dimensionality and capture the salient facial features.

B. Principal Component Analysis for Dimension Reduction

Principal Component Analysis, also known as Karhunen-Loeve transform (KLT), generates a set of orthonormal basis vectors, considered as principal components, that maximize the variance of itself does not necessarily lead to good face recognition, the computation can be reduced as [6]

$$\Lambda = [g_1, g_2, \ldots, g_N]$$
$$\tilde{G} = [g_1 - \bar{G}, g_2 - \bar{G}, \ldots, g_N - \bar{G}]$$ (7)

where $\bar{G} = 1/N \sum_{i=1}^N g_i$. The covariance matrix of the image set is defined as

$$\Sigma = \frac{1}{N} \sum_{i=1}^N (g_i - \bar{G})(g_i - \bar{G})^T = \frac{1}{N} \tilde{G}\tilde{G}^T$$ (8)

The eigenvector and eigenvalue matrices $\Omega$, $\Lambda$ are computed as

$$\Lambda\Omega = \Sigma\Omega$$ (9)

We notice that $\tilde{G}\tilde{G}^T$ is a $M \times M$ matrix while $\tilde{G}\tilde{G}$ is a $N \times N$ matrix. If the training set size $N$ is much smaller than image dimension $M$ (The small sample problem, e.g., face recognition), the computation can be reduced as [6]

$$\Lambda'=\Omega'$$ (10)

$$\Omega' = \tilde{G}'\tilde{G}'^T$$ (11)

where $\Lambda'$ is the diagonal matrix containing the eigenvalues of covariance matrix $\Sigma$, and $\Omega'_i = [\omega_1, \omega_2, \ldots, \omega_N]$. Assume that the eigenvalues are sorted in decreasing order, then the first $q$ leading eigenvectors define feature space $\Omega_i = [\omega_1, \omega_2, \ldots, \omega_q]$.

Consider a random image vector $g_i \in G$. The principal component vector (feature vector) $\Gamma_k$, is an orthogonal transformation of data $g_k$, described by $\Gamma_k = \Omega_i g_k$, i.e.,

$$\Gamma_k = [\omega_1 g_k, \omega_2 g_k, \ldots, \omega_q g_k]^T$$ (12)

C. Hybrid Trace Features

Previously, the extraction of a single Trace transform is introduced. However, there are no guidelines on which trace functional $T(f(t))$ will produce the best results in terms of suppressing variation in expression, illumination etc, while accentuating differences between different people, the appropriate tracing functionals can be chosen during a training phase [2]. In Eq. (12), though the lower-dimensional vector $\Gamma$ captures the most expressive features of the original data, the high variance of itself does not necessarily lead to good
the feature vector. We can construct the new facial feature \( \Gamma \) performing vTT and PCA be shown as follows.

In order to get a better face representation than the single transforms for the same image, which has the potential to be discriminated 

is multi-modal and the modes correspond to the classes to discriminate [7]. The advantages of Trace transform is its ability to generate several transforms by implementing different Trace functionals. Each Trace transform may reveal diverse aspect of the original image according to the properties of functionals. Fig. 3 shows the diversity of Trace transforms for the same image, which has the potential to form new robust image representation for pattern analysis. In order to get a better face representation than the single Trace transform, we propose a novel facial feature to utilize more than one Trace functionals for face recognition. The new facial feature is named as the Hybrid Trace Features shown as follows.

Let the Trace features of functional \( fnl \) obtained by performing vTT and PCA be \( \Gamma^{fnl} = [\gamma_1^{fnl}, \gamma_2^{fnl}, \ldots, \gamma_q^{fnl}]^T, fnl = \{1, 2, \ldots, 8\} \), here \( q \) is the dimension of the feature vector. We can construct the new facial feature representation Hybrid Trace Features:

\[
\tilde{\Gamma} = \frac{\sum_{i=1}^{\max(fn)} \alpha_i \Gamma^i}{\max(fn)}
\]

where \( \{\alpha_1, \alpha_2, \ldots, \alpha_{\max(fn)}\} \) are the scalars corresponding to different Trace functionals. The calculation of the scalars are the key procedure and will be discussed in the next section.

**D. Scalar Calculation of Hybrid Trace Features with Genetic Algorithms**

To solve the optimization problem in pattern recognition, Genetic Algorithms (GAs) [8], inspired by the biological mechanisms of reproduction, plays an important role. Various applications, such as face recognition [9], clustering techniques [10] [11], and feature selection [12] use GAs as global optimization tool. In this paper, GAs are used as the searching algorithm to find out the global optimal scalars. A randomly generated set of scalars form the the initial population from which the GAs starts its search. The GAs makes its choices via genetic operators as a function of a probability distribution driven by the fitness function.

1) **Chromosome Encoding**: We employ a simple encoding scheme where the chromosome is defined as a string of scalars: \( [\alpha_1, \alpha_2, \ldots, \alpha_{\max(fn)}] \). The number of scalar \( \alpha \) depends on the amount of Trace functionals \( fnl \). In the string, 10 bits (resolution) are used to represent each scalar. Each chromosome represents a combination of scalars.

2) **Initial Population**: In order to initialize the population we need the values: (a) The number of bits in a solution candidate, which is depending on the required precision; (b) Total number of solution candidates (population size). In general, the initial population is generated randomly. Because the population size influences the computation cost and the diversity of the population, we use a population size of 100 to give a trade-off between the above two factors in our experiments.

3) **Fitness Evaluation**: The chromosome selection for the next generation is done on the basis of fitness. The selection mechanism should ensure that fitter chromosomes have a higher probability of survival, i.e., fitness values guide GAs on how to choose offspring for the next generation from the current parent generation. In pattern recognition, numerous works have been done using GAs. Most of them employ classification accuracy (CA) as the fitness [9] [13]. In order to improve the generalization performance, the CA on cross-validation database are set for measuring fitness in our researches.
4) Selection: Based the evolutionary procedure, the selection process will be implemented if the termination criterion is not achieved. The selection operator selects chromosomes from the current generation into a mating pool taking into consideration the fitness of the chromosome. The chromosome strings with higher fitness values will likely be represented in higher numbers in the mating pool. The size of mating pool is usually the same as population size. The roulette wheel is a common technique for selection, however it may not yield expected performance if the population is small. Hence the stochastic universal selection [8] will be our choice.

5) Crossover and Mutation: After chromosome selection, crossover that exchanges information between two parent chromosomes for generating two child chromosome is carried out on the mating pool. There are three basic types of crossovers: one point crossover, two point crossover, and uniform crossover. Different from the other two schemes, in uniform crossover, each gene of the offspring is selected randomly from the corresponding genes of the parents. In general, we do not know how the weights depend on each other, so we choose uniform crossover in our research. Normally, the crossover probability $p_c$ is chosen as $0.65$ to introduce enough varieties to next generation.

Then the mutation is applied to the offspring randomly with the mutation probability $p_m$. The mutation will introduce a degree of diversity to the population, prevent a premature convergence, and help to sample unexplored regions of the search space. Usually the mutation plays a minor role in the evolutionary procedure, thus the $p_m$ is assigned a small value $0.004$ in our experiments.

6) Termination of Evolutionary Process: The termination criterion in this article is that a given number of generations "gen" has been completed. Then the string with maximum fitness in the population of last generation provides the optimal scalars which can achieve best classification performance in training set. The effectiveness of the proposed Hybrid Trace Features will be evaluated on ORL face database in the experimental part.

IV. Rotation based Hybrid Trace Features

Because the proposed Hybrid Trace Features use naive Trace transform, the extracted facial features are lack of investigating the characteristics of Trace transforms themselves. However, the transforms of Trace transforms are yet to be discovered to find useful properties for pattern analysis. In this section, the properties of Trace transform are reviewed in details and a new facial feature is proposed for image representation.

A. Properties of Trace Transform

As stated in [1], a functional $\Xi$ of a function $\xi(x)$ is invariant if

$$\Xi(\xi(x + b)) = \Xi(\xi(x)), \forall b \in \mathbb{R} \quad (I_1)$$

An invariant functional may also have the following additional properties $(i_1)$ $(i_2)$ shown in Table II:

| $(i_1)$ | Scaling the independent variable by $\alpha$, scales the results by some factor $\alpha(\alpha), \Xi(\alpha(\xi(x))) = \alpha(\alpha)\Xi(\xi(x)), \forall \alpha > 0$ |
| $(i_2)$ | Scaling the function by $c$ scales the result by some factor, $\gamma(c), \Xi(c\xi(x)) = \gamma(c)\Xi(\xi(x)), \forall c > 0$ |

One can write $\alpha(\alpha) = a^{\alpha^2}$ and $\gamma(c) = e^{\ln c}$, where parameters $\kappa_\Xi$ and $\lambda_\Xi$ characterize function $\Xi$.

How the Trace transform of an image will be affected by the linear distortion? Suppose we choose the invariant Trace functional $T$ with property $(i_1)$. We start by observing that the Trace transform of the distorted image will be given by:

$$g(F; C_2, \phi, p) = T(F(C_1; \phi_{old}, p_{old}, t_{old}))$$  \hspace{1cm} (15)

Substitute (1), (2) and (3), we obtain

$$g(F; C_2, \phi, p) = T(F(C_1; \phi - \theta, \nu[p - s_0 \cos(\psi_0 - \phi)],$$

$$\nu[t - s_0 \sin(\psi_0 - \phi)])$$  \hspace{1cm} (16)

Due to properties $(I_1)$, $(i_1)$ of $T$, and $T$ is considered as a function of variable $t$, the above equation can be written as:

$$g(F; C_2, \phi, p) = \alpha T(\nu[p - s_0 \cos(\psi_0 - \phi), t])$$  \hspace{1cm} (17)

We can express this condition in terms of the exponents $\kappa$ of the functional $T$, to obtain:

$$g(F; C_2, \phi, p) = \nu^{\kappa_{r+1}} T(F(C_1; \phi - \theta, p - s_0 \cos(\psi_0 - \phi), t))$$  \hspace{1cm} (18)

The Trace transform of the image is invariant if it satisfies the following conditions:

$$\begin{cases} 
\kappa_T = -1 & \text{or} \quad \nu = 1 \\
\theta = 0 \\
s_0 \cos(\psi_0 - \phi) = 0
\end{cases}
$$

where $\kappa_T$ is given in advance according to different functionals. If $\nu = 1$, i.e., there is no scale difference between the objects to be matched, this condition is not necessary and any invariant functionals with necessary properties can be used.

B. Virtual Trace Transform from Rotation

We notice that the above invariant conditions are too restrictive. In most real world applications, not all of these conditions are necessary. e.g., in face recognition, pose variations is know as one of the major challenges. By using the robust face detection techniques [14], localization of the faces and scaling it into a proper size will not be a difficult task. And the detection process will also eliminate useless information in the image. In most face authentication system, the effect of rotation is more dominant than that of shifting and scaling. Consequently, it is worthy to investigate the acquiring and usage of rotation related Trace transforms.

Suppose there are three images with two of them obtained by rotating the original image with angles $90^\circ$ and $270^\circ$ clockwise shown in Fig. 4 (a1)-(a3), where (a1) is the original
Fig. 4. The face images and their Trace transforms (Trace functional 6 in Table I is used). (a1)-(a3) Original and rotated face images; (b1)-(b3) Trace transforms of the images in (a1)-(a3), respectively; (c1)-(c3) Virtual Trace transforms generated from (b1) with rotation angle $-90^\circ$, $-180^\circ$, and $-270^\circ$, respectively.

Known from Eq. (18), the Trace transform is a periodical function. By shifting $-90^\circ$, $-180^\circ$, and $-270^\circ$ along the axis of $\phi$, three virtual Trace transforms are generated shown in Fig. 4 (c1)-(c3). It is obvious that the virtual Trace transforms (c1) and (c3) are exactly matched with (b2) and (b3), which means that the virtual transforms obtained from shift along $\phi$ axis have direct relationship with Trace transforms of real rotated images. In another word, the virtual transform by shifting $-\theta$ is same as the Trace transform of real image with rotation angle $\theta$ clockwise.

For each Trace transform, 359 virtual samples can be generated with shifting angles from $-1^\circ$ to $-359^\circ$ along the $\phi$ axis. More important, the shifting of $\phi$ corresponds to the rotation on real image, and the rotation information is one of key factors in face recognition task as described before. Thus the idea of rotation based Hybrid Trace Features uses the richness of virtual Trace transforms to extend the original training set to consist both genuine and artificial generated transforms. Because rotation information is incorporated into the virtual images, the new constructed training data set has the potential to deal with test data with large pose variations and receive a good generalization performance. Subsequently, the same feature extraction procedure as HTF elaborated in section III is implemented to extract rotation based Hybrid Trace Features for face image representation.

V. CLASSIFICATION WITH HYBRID TRACE FEATURES

In previous section, we have described our proposed Hybrid Trace Features in details. In addition, we should design a robust face recognition structure where an effective classification method is needed to present the discriminatory power of proposed facial features. In order to put emphasis on the improvement brought by features rather than classifiers, a widely used instance-based method called $k$-Nearest
Neural network algorithm (\(k\)-NN) is implemented to the face recognition system. This basic instance-based method assumes patterns can be represented as points in the \(n\)-dimensional Euclidean space \(\mathbb{R}^n\) and simply stores the training examples rather than constructing an explicit description of the target function. When provided a test sample, the classifier will examine the relationship between the new query instance and previously stored training patterns to assign a target function value. Because \(k\)-Nearest Neighbor algorithm never construct an approximation to the target function over the entire instance space, it has the advantages when the target function is complex and can be described by a collection of less complex local approximations [15].

Given the training data \((x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)\) in which each pattern has \(q\) attributes and the target function \(y_i\) is the label of \(x_i\). \(x_r(r)\) denotes the value of \(r\)th attribute of pattern \(x_r\). The distance between two examples \(x_i\) and \(x_j\) is defined in terms of standard Euclidean distance \(d(x_i, x_j)\) as follows

\[
d(x_i, x_j) = \sqrt{\sum_{r=1}^{q} (x_i(r) - x_j(r))^2} \quad (19)
\]

The general classification process is shown as follows where \(x_i\) is our proposed Hybrid Trace Feature \(\Gamma_i\) in Eq. (13).

- Training phase: \(k\)-NN does not construct general description of target function but simply store the training examples \((x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)\).
- Classification phase: Provided the query pattern \(x\), \(k\) instances in training set are selected as nearest neighbors in terms of \(d(x, x_i)\). The pattern \(x\) is assigned to the class having most instances among the \(k\) neighbors.

The Hybrid Trace Features described in previous section together with \(k\)-Nearest Neighbor classifier form a complete face recognition system. The face images in both training set and test set are converted to low dimensionality facial features by implementing proposed feature extraction method, then the \(k\)-NN classifier is trained to predict the labels of cross-validation test samples within training set. The above procedure is incorporated into an evolutionary framework and the weights are obtained to provide potential high generalization performance after reaching the predefined termination criterion. The architecture of the face recognition system is shown in Fig. 5 for better visualizing the contributions of each component. In application, \(k\) is chosen as 1 so that the query pattern is assigned to the same label as that of the nearest training example.

VI. EXPERIMENTS

A. Database

In the experiments, we use the Cambridge ORL face database [16] to evaluate our method. The ORL database contains 400 images of 40 individuals under a high degree of variability in expression and pose. In ORL database, each person has ten different images. There are variations in facial expressions such as glasses or no glasses and open or close eyes. All the images are taken under a dark background largely frontal faces with slight movements, and there are also some variations in scale. The images of three persons are shown in Fig. 6. In the experiments, we randomly select 40 persons with each 5 samples as the training set, and the remaining 200 samples are used as test set, i.e., five randomly selected face images for each pattern are used for algorithm validation. The experimental results are as follows.

B. Practical Feature Extraction for Classification

In previous sections, the ideas of Hybrid Trace Transform (HTF) and rotation-based Hybrid Trace Transform (r-HTF) are proposed. In r-HTF algorithm, number of virtual samples are generated by rotating the Trace transform of original data. However, in practice, the increased training set bring a huge consuming of computer memory. Consequently, longer time is needed to run the program, and in the worst situation, the required memory is out of size of hardware.

In the application of face recognition, normally large memory are needed to store the image and do matrix computation. In ORL database, suppose we want to generate 20 virtual Trace transforms, the training set size will be 4200. Such training set is impossible to be used in most computers, especially during the factorization of covariance matrix, as the required memory is up to 2G bytes. As a result, we propose using original training set to construct subspace, on which the whole training set (including virtual samples) and

![Fig. 6. Sample face images from ORL database. Each person has ten images with variations in facial expressions and poses. In experiments, five images are randomly selected for each person to form the training set. The remaining images of the same pattern are used for testing.](image)
between the real and virtual data. Shown in Fig. 7, the PCA will try to increase the scatters by considering trade-offs samples of each class, in the case of using large subspace, construction errors. Because great variations exist in various both inter- and intra-class variation and minimizing the re-

In order to show the differences between the projection results on two subspaces constructed from original training set and the complete data including the virtual samples, a subset consisting of 3 persons with 5 images each class from ORL database is used for demonstration. In the enlarged training set, two virtual Trace transforms are obtained by rotating original image 90 degree clockwise and counter-clockwise, respectively. In the subset, there are 45 transforms in total. The data will be projected to two subspace (small or large) separately to reduce the dimensionality to 2. In addition, the extracted two-dimensional features are visual-
ized in Fig. 7. It is obvious that the large subspace leads to a over-training phenomena which has the risk of degrading the overall classification performance. The possible reasons are discussed as follows. Principal component analysis is a linear dimensionality reduction algorithm by maximizing both inter- and intra-class variation and minimizing the re-

In general, the proposed method is practical and efficient for extracting robust features with discriminatory power. By projecting the data to small feature space, the original real data features are invariant in spatial distribution, in addition, more supporting features generated from virtual

Trace transform are introduced near the real data features and may contribute to the recognition process in terms of improving the classification accuracy. Of course another advantage of proposed method is the ability to avoid the occurrence of over-fitting in the training phase.

C. Experimental Results

In the experiments, two virtual samples for each Trace transform are generated by shifting $-2^\circ$ and $-358^\circ$ in Trace domain. Thus a training set of total 600 samples are created, in which 200 images are original real data. The proposed Hybrid Trace Features are compared with facial features extracted by benchmark feature extraction algorithms such as PCA, LDA, and Kernel PCA to evaluate its effectiveness. The classification accuracy shown in Table III reveals that proposed Hybrid Trace Features and rotation based Hybrid Trace Features algorithms rank top across all the feature dimensions. Obvious, the HTF and r-HTF methods perform a little better at low dimension than high dimension. The possible reason comes from the less tunable scalars $\alpha$. As one scalar corresponds to the whole data set of one Trace functional, the higher the feature dimension, the more effects that the scalar brings. In another word, the scalar may become negative factors on some less dominant data which has larger discriminatory power for improving the
the HTF can be considered as a data fusion formula where the Genetic Algorithms is the tool. The HTF performs better HTF in the task of classification. General speaking, the HTF is the mean of various Trace transforms, while the HTF is a complex combination of Trace features where optimal scalars are selected in terms of high recognition rate through evolutionary process.

VII. CONCLUSIONS

In this paper, the genuine and rotation based Trace transforms are fused by Genetic Algorithms to generate the Hybrid Trace Features and rotation based Hybrid Trace Features for face recognition. Our contributions in this paper are the proposal of rotation based Trace transform and the introduction of scalars for facial feature fusion in which the evolutionary computation plays a key role. Comprehensive experimental results demonstrate the Hybrid Trace Features and rotation base HTF algorithm robust and discriminant as facial features with high classification accuracy.

ACKNOWLEDGMENT

The authors would like to thank M. Petrou, A. Kadyrov and S. Srisuk who have kindly provided the source code for Trace transform. The authors are also grateful to all the reviewers for their valuable comments and suggestions.

REFERENCES