Abstract—We propose Facial Trait Code (FTC) to encode human facial images and apply it to face recognition. Extracted from an exhaustive set of local patches cropped from a large stack of faces, the facial traits and the associated trait patterns can accurately capture the appearance of a given face. The extraction has two phases. The first phase is composed of clustering and boosting upon a training set of faces with neutral expression, even illumination, and frontal pose. The second phase focuses on the extraction of the facial trait patterns from the faces with variations in expression, illumination, and poses. To apply the FTC to face recognition, two types of codewords, the hard and the probabilistic codewords, with different metrics for characterizing the facial trait patterns are proposed. The hard codeword offers a concise representation to a face, while the probabilistic codeword enables matching with better accuracy. Our experiments compare the proposed FTC to other algorithms on several public datasets, all showing promising results.

Index Terms—face recognition, feature extraction, error-correcting code

I. INTRODUCTION

MOST 2D face recognition algorithms can be classified into global approach, part-based approach, and some hybrid of both. A comprehensive survey is given in [1]. The global approach considers the whole facial area, mostly from eyebrows to chin and ear to ear; while the part-based extracts some local features, mostly eyes, nose, mouth, and maybe others [2] [3] [4] [5] [6] [7]. Besides the aforementioned facial features that are describable to humans, some work used the local facial features on faces that can not be straightforwardly described. For example, such local features can be a region covering part of the left eye and eyebrow, or one covering part of the upper lip and nose. These local features can be defined by exhaustive search on human faces [8], just like what was done in the face detection work by Viola and Jones [8], or spatial segmentation of a face into blocks of equal size [9] [10] [11]. Besides rectangular patches, Bart et al. [12] found triangular sub-images on human faces were robust under viewpoint changes for face recognition.

Part-based approaches typically involves the selection and fusion of facial features [4] [12] [9] [10] [11] [7]. Viola and Johns [8] selected a bunch of facial features best for face detection. Our previous work [13] extracted facial features good for discriminating different faces, and used the patterns of the extracted facial features to define a facial coding scheme for face recognition. This work was known as the Facial Trait Code, or FTC for short. The development of the FTC was motivated by the observations that typical patterns seemed to exist in many local patches, and the number of patterns in one patch varied from one to another. Furthermore, if we associate each pattern in each patch with a number, a face can be represented by a series of numbers, each of which shows the specific pattern of a specific patch on that face. More importantly, the series of the numbers seemed different for a different face. These observations made us carry out a thorough study on local patches, and proposed the FTC. From an exhaustive set of the patches collected from a large number of faces, FTC extracted those with relatively strong strength in discriminating different faces. The extracted patch features were called facial traits, and each facial trait had its own specific location, size, and orientation on a face. Each facial trait had a certain number of patterns, which allowed a trait-based encoding and decoding phases to be defined. Given a face, one can encode it using the numbers of the patterns at the facial traits that best describes the appearance of this face. These numbers constitute the facial trait code for this face. An example is given in Fig. 1, where three facial traits with numbered patterns are shown in the top, and the three faces show how the corresponding traits give the three facial trait codes.

Putting the facial trait patterns into codes has the following advantages. Firstly, a face can be effectively denoted by a finite length of codeword in which each symbol gives not
just a specific facial trait with a fixed size and location, but also the pattern in this facial trait that best describes the face in that specific facial trait. This implies that the FTC offers an effective descriptor to a given face. Secondly, face identification and verification problems can be formulated as code matching problems, and thus some merits from coding perspectives can be preserved. Error correcting is one such merit exploited in the development of the FTC. Furthermore, since each number in a facial trait code refers to a trait pattern, we can render a facial image by mosaicing the trait pattern patches from different locations and sizes. Please refer to our previous work for the application of FTC to face synthesis [14].

The distance measure in the code space of the early version of FTC [13] was given by Hamming distance, hence the similarity between one facial trait patch and all facial trait patterns except the most similar one is completely ignored. This refrains the FTC from discriminating similar faces or faces with similar local features. To address this issue, we proposed a new type of FTC codeword that employed a probabilistic distance measure in its code space [15]. Instead of keeping only the most similar facial trait pattern, the new FTC codeword kept the similarity to all the facial trait patterns. This work is known as the Probabilistic Facial Trait Code, or PFTC for short. Fig. 2 shows the training and the application of the proposed FTC to face recognition using hard and probabilistic codewords.

This paper reports a complete version of the Facial Trait Code. Both the codeword in early version of FTC [13] and the codeword in PFTC [15] are included and compared in this paper. In this paper, we elaborate the construction of the FTC encoder to achieve better performance when recognizing faces under variations in illumination or facial expression. Several practical issues concerning the building of FTC are also discussed, such as the code length issue and the pre-selection of discriminating facial traits. Extensive experimental results are given that evaluate the performance of the proposed FTC under several scenarios with different variations among facial samples or different settings of the training, gallery and probe sets. The result on FERET data set is also reported that enables a direct comparison between the proposed algorithm and other ones.

A. Related Works

The proposed FTC encodes human faces based on a bunch of local facial features, and it can be regarded as a part-based approach. Part-based approaches often start with some feature extraction methods and are followed by classification schemes [16] [3]. Different feature extraction methods result in different descriptors of local features. Deformable graphs and dynamic programming were used in [17] to determine eyes, nose, mouth, and chin. Liao and Li [18] extracted 17 local features using the Elastic Graph Matching (EGM), and each of these 17 features had its own specific spot on a face, for example, the corners of eyes, the ends of eyebrows, and the centers of lips. Inspired by the EGM, Arca et. al. located eyes, nose and mouth based on 24 facial fiducial points, and used these local features for face recognition [6]. Huang et al. proposed the component-based LDA (Linear Discriminant Analysis) method for face recognition [4]. Five components, two eyes, nose, and left and right parts of mouth were used in their work. A two-level hierarchical component classifier was proposed in [2] to locate 14 feature patches in a face, and [3] showed that face recognition using these 14 feature patches outperformed the same recognition method but using the whole face as the feature. Ivanov et. al. [5] extended this study by experimenting with a few different recognition schemes using the same set of 14 feature patches. Recently, an iterative growing process was proposed to further improve the localization of these 14 feature patches [7]. Besides manually defining eyes, noses or mouths as local facial features, one other way is to blindly segment a face into non-overlapping blocks of equal size [9]. Our work is different from all the aforementioned part-based works on the definition of facial parts. Instead of manually defining facial features such as eyes, noses or mouths as local facial features, one other way is to blindly segment a face into non-overlapping blocks of equal size [9]. Our work is different from all the aforementioned part-based works on the definition of facial parts. Instead of manually defining facial features such as eyes, noses or mouths as local facial features, one other way is to blindly segment a face into non-overlapping blocks of equal size [9]. Our work is different from all the aforementioned part-based works on the definition of facial parts. Instead of manually defining facial features such as eyes, noses or mouths as local facial features, one other way is to blindly segment a face into non-overlapping blocks of equal size [9].
the binary string for the determination of the class label. [20] aimed to fix the computationally expensive one-classifier-per-subject problem when the number of subjects is large. Given an \( N \)-subject recognition task that generally requires \( N \) binary classifiers, FCC only creates \( \log_2(N) \) binary classifiers. The FTC is distinctive from all of the previous works upon coding for facial recognition in following aspects. Firstly, the codes are developed from local features. Secondly, each symbol in a codeword denotes some specific pattern existing in a facial trait, which is a local rectangle patch of a certain size and location. Thirdly, the FTC is an \( n \)-ary or \( n \)-by-\( k \) array, instead of the binary ones in all of the previous works.

Although the proposed Facial Trait Code shares a similar concept of codebook with the aforementioned works, it is different form them for the following aspects: 1) As different patterns seem to exist in different local facial part, the proposed FTC builds different codebooks for patches of different facial parts while the aforementioned works build a single codebook for patches from the whole face; 2) As some local facial parts are more discriminative for face recognition purpose, the proposed FTC selects the set of the most discriminating local facial parts. [23] lacks this feature selection mechanism and basically use all the facial parts for face recognition regardless of their effectiveness for face recognition.

**B. Requirements of FTC and Outline of this Paper**

Similar to the learning-set-based face recognition, for instance Eigenface [24] and Fisherface [25], the FTC requires a large collection of faces as its learning set. Three datasets are required in making the basis of the FTC, namely the Trait Extraction Set (TES), the Trait Variation Set (TVS), and the Trait Enrichment Set (TRS).

1) The TES consists of a large number of frontal faces with neutral expression and evenly distributed illumination. All faces are aligned by the centers of both eyes. This set is used in the first half of the development of the FTC that determines the facial traits and the patterns in each trait.

2) The TVS consists of facial images taken under various illumination conditions, and/or with different expressions and poses. The pose variation is limited to 20° in all directions, making the alignment by both eyes possible. The TVS is used in the second half of the development of the FTC that defines the modes of variation in each facial trait.

3) The TRS aims to further enrich the facial variations by synthesizing new facial images given images in TES and TVS. For example, we generated imprecisely aligned facial images by adding some perturbation to their alignment. It is possible to further generate facial images taken under different illumination conditions or poses. TRS is also used in the second half of the development of the FTC that defines some additional modes of the variation in each facial trait.

TES, TVS and TRS have to be different sets. When defining trait patterns, we want to find the patterns that exist among faces of different identities. In this case, we have to use only faces without variations in illumination conditions, poses, or expression (i.e. TES). After the trait patterns are defined and we want to train SVM classifiers to distinguish these patterns, and we have to consider facial images under all kinds of variations, as in practice the faces to be recognized may undergo these variations. In this case, we use additional TVS and TRS for SVM training.

This paper is organized as the following. Given a TES, it is elaborated in Sec. II-A how patches are generated, and the patterns in each patch are extracted and numbered. To assess each patch’s strength to discriminate faces, we create the Patch Pattern Map in Sec. II-B so that the patches with relatively strong strength can be effectively selected to form the facial traits. Each facial trait comes with a SVM classifier able to classify the corresponding trait cropped from a given face to a specific pattern. The collection of all of these SVM trait classifiers makes up the FTC encoder in Sec. III-A. The training of the SVM classifiers is based on all three datasets, TES, TVS, and TRS, with features in terms of the variation patterns extracted from these three datasets. When applying the FTC to face recognition, in the enrollment phase each gallery face, i.e., the face of a subject enrolled to the gallery set, is encoded into a gallery code and stored in the database. In the recognition phase, a probe face is encoded into a probe code, and then matched against the gallery codes in the database. Two types of codewords, namely the hard codewords and the probabilistic codewords are used. The former is more concise and the latter achieves superb recognition performance. The encoding and decoding schemes for both of the hard and the probabilistic codewords are detailed in Sec. III-B and Sec. III-C, respectively. A comprehensive experimental study is presented in Sec. IV. The conclusion of this study is given in Sec. V, along with possible directions for continuing study.

**II. EXTRACTION OF FACIAL TRAITS AND THEIR PATTERNS**

**A. Patches and Patch Patterns**

Given a facial image from the TES, one can specify a local patch by a bounding box \( \{x, y, w, h\} \), where \( x \) and \( y \) are the 2-D pixel coordinates of this bounding box’s upper-left corner, and \( w \) and \( h \) are the width and height of this bounding box, respectively. If this bounding box is moved from left to right and top to bottom in the face with a step size of \( \Delta x \) and \( \Delta y \) pixels in each direction, one can obtain many patches with the same size but different locations. If \( w \) and \( h \) can further change from some small values to large values, we will end up with some exhaustive set of local patches across the face. Some similar set of such an exhaustive collection of local patches was used in [8] determine the features good for facial detection. It can be seen that the number of the patches grows with (1) the size of the facial image, (2) the size range of local patches, and also (3) the reduction in the step size. The above items (2) and (3) are the size-and-step parameters. Concerning the size and step parameters, the size of local patch should not be too small, for small patches generally have less discriminating power and suffer more from imprecise localization. A smaller step size increases the number of the total patches to be evaluated, at the cost of
higher computational complexity in off-line training process. The relationship between the size-and-step parameters and the recognition rates has been studied in our experiments, and the results are given in Sec. IV-A.

Assuming that \( K \) frontal faces are available from the TES and all aligned to the centers of both eyes, we can crop a stack of \( K \) corresponding patch samples for each patch following the above patch generation scheme. The next step is to cluster the \( K \) patch samples in each patch stack into \( M \) salient patch patterns. By our construction, patches belong to the same person should be clustered into the same patch pattern. To enforce this constraint, we apply Linear Discriminant Analysis (LDA) to make the sample belong to the same person as similar as possible, then we average all the sample of the same person to obtain the mean patch in the LDA subspace. In practice, we first apply the Principal Component Analysis (PCA) to \( K \) patch samples to avoid the singularity problem of LDA. Considering the case that the \( K \) facial images can be from \( L \) subjects (\( L \leq K \), i.e., one subject can have multiple facial samples), in each patch stack the LDA is applied, and the \( L \) low dimensional mean patch features for the \( L \) subjects can be obtained.

If there are \( M \) patch stacks, then we apply an unsupervised clustering algorithm to cluster the \( L \) low dimensional patch features into \( k_i \) clusters in the \( i \)-th patch stack, where \( i = 1, 2, ..., M \). The \( k_i \) clusters in the \( i \)-th patch stack are considered the patterns existing in the \( i \)-th patch stack, and will be called patch patterns. Basically any unsupervised clustering algorithm can be applied in this step, as long as it can determine a proper number of patch patterns for each patch. The resulting facial patterns have effect on the overall FTC performance.

B. From Patches to Facial Traits

It is often the case that \( M \), the number of the patches across a face, is large. However, that the effectiveness of each patch in the discrimination of the faces in the TES (Trait Extraction Set) can be compared and ranked, this can give a way to reduce \( M \). In this work, the effectiveness of patches were evaluated in two aspects: (1) given a single patch, we evaluate the discriminability of its patch patterns; (2) a set of patches provide a way to encode human faces. We evaluate the discriminability of human faces using the associating codewords. The patches with good discriminabilities in both aspects will be called Facial Trait.

For the first aspect, we train a Support Vector Machine (SVM) and apply a five-fold cross validation to measure the discriminability of patch patterns. We randomly segment the subjects in the TES into five subsets. For each patch, we train a SVM to classify its patterns (extracted in Sec. II-A) using samples in four subsets as training samples. This SVM is then used to encode patch samples in the remaining subset. Given multiple samples per subject in the remaining subset, we define a measure

\[
\alpha \equiv \frac{1}{L} \sum_{i}^{L} \frac{n_i}{N_i}.
\]  

where \( n_i \) is the maximum number of the samples of the \( i \)-th subject that are encoded identically; \( N_i \) is the number of samples of the \( i \)-th subject; \( L \) is the number of the subjects. If all the facial samples belonging to the same subject are encoded identically, then \( \alpha = 1 \). On the other hand, a low \( \alpha \) value indicates that the pattern-specific classifier cannot classify patch patterns well. The process is repeated five times with different training subsets, and an averaged \( \alpha \) is calculated. Patches with \( \alpha \)s below a pre-defined threshold \( T_\alpha \) will not be considered in the following. (1) will be used in Section IV-A to ignore the patches with bad discriminability of their patch patterns.

For the second aspect, we define a measure able to assess each patch’s effectiveness for face discrimination. This measure can be defined via the Patch Pattern Map or PPM for short. The PPM is defined for each different patch, and it shows which subjects’ faces reveal the same patch pattern at that specific patch. Let \( PPM_i \) denote the PPM for the \( i \)-th patch, \( i = 1, 2, ..., M \). Assuming \( L \) subjects in the TES, \( PPM_i \) will be \( L \times L \) in dimension, and the entry at \( (p,q) \), denoted as \( PPM_i(p,q) \), is defined as follows:

\[
PPM_i(p,q) = \begin{cases} 
0 & \text{if the patches on the faces of the } p \text{-th and the } q \text{-th subjects are clustered into the same patch pattern} \\
1 & \text{otherwise}
\end{cases}
\]

Given \( N \) patches and their associated \( PPM_i \)’s stacked to form a \( L \times L \times N \) array, there are \( \frac{L(L-1)}{2} \) \( N \)-dimensional binary vectors along the depth of the array.\(^1\) Let \( v_{p,q} \) denote one of the \( N \)-dimensional binary vectors, \( v_{p,q} \) reveals the local similarity between the \( p \)-th and the \( q \)-th subjects in terms of the \( N \) local patches. More unities in \( v_{p,q} \) indicates more differences between this pair of subjects, and on the contrary, more zeros shows more similarities.

The binary vector \( v_{p,q} \) motivates the application of the Error Correcting Output Code (ECOC) [22] to this research. If each subject’s face is encoded using the most discriminant patches, or the facial traits, then the induced set of \( [v_{p,q}]_{1 \leq q < p \leq L} \) can be used to define the minimum and maximum Hamming distance among all encoded faces in the corresponding code space. The \( v_{p,q} \) with the least (most) of unities gives the minimum (maximum) Hamming distance. To maximize the robustness against possible recognition errors in the decoding phase, we formulate an Adaboost algorithm to maximize \( d_{min} \), which denotes the minimum Hamming distance for the determination of the facial traits from the patches. The idea behind the Adaboost algorithm is to select a bunch of weak classifier to form a strong classifier. The algorithm selects weak classifier one by one. In each round of the selection, the weak classifier that has the maximum accuracy on the current data is selected. Then the data on which the selected classifiers make incorrect classification are assigned larger weights. Then in the next round of the selection, the algorithm will tend to select the classifier that makes the correct classification on

\(^1\)Because each \( PPM_i \) is a symmetric matrix, one only needs to consider the lower triangular part of it.
these data. Our modified Adaboost algorithm is summarized in Algorithm 1.

Algorithm 1 Extraction of Facial Trait Patterns

Require: \( PPM_i, i = 1 \sim M \)

Ensure: selected \( N \) facial traits that yield maximum \( d_{\min} \)

\( F = \{ \text{the set of } M \text{ patches} \}; F = \emptyset \)

\( C(p, q) = 0; \omega (p, q) = 1, \) where \( p = 1 \sim L - 1, q = 1 \sim L - 1, \) and \( p > q \)

for \( t = 1 \) to \( N \) do

Normalizing the weight \( \omega (p, q) = \frac{\omega (p, q)}{\sum_{p,q}\omega (p,q)} \).

For every element \( f_i \) \( \in F \), \( \alpha (f_i) = \sum_{p,q} \omega (p,q)\omega (p,q) \).

Select \( \hat{f}_i = \arg \max \alpha (f_i) \).

Update \( C(p, q) = C(p, q) + \sum_{p,q} \omega (p,q)\omega (p,q) \).

Calculate \( d_{\min}(d_{\max}) \), which is the minimum(maximum) element in \( C(p, q) \).

\[
\omega (p, q) = \begin{cases} 
L & \text{if } C(p, q) = d_{\min} \\
0 & \text{if } C(p, q) = d_{\max} \\
1 & \text{otherwise}
\end{cases}
\]

Update sets \( F = F - \hat{f}_i \) and \( \hat{F} = \hat{F} \cup \hat{f}_i \).

end for

In Algorithm 1, \( F \) denotes the set of the overall patches, initially \( F \) contains \( M \) patches. \( \hat{F} \) is the set of the facial traits selected from \( F \), and the number of the facial traits in \( \hat{F} \) will finally reach \( N \). \( C \) is a \( L \times L \) dimensional array where \( C(p, q) \) gives the number of ones in \( v_{p,q} \). \( \omega \) is a weight array with the same dimension as that of \( C \), and \( \omega (p, q) \) is the weight for the pair of Subject-\( p \) and Subject-\( q \). In each run, the patch able to maximize the updated \( d_{\min} \) is selected as one new facial trait.

With the increasing of \( N \), the minimum codeword distance between codewords of training faces, or \( d_{\min} \), and the number of correctable digit, which is \( \frac{d_{\min} - 1}{2N} \), will increase as well. An important quantity that affects the FTC recognition rate is the ratio of correctable digits, and it can be expressed as

\[
\beta = \frac{(d_{\min} - 1)}{2N}. \tag{2}
\]

A bigger \( \beta \) is desirable, for it grants the FTC better error-correcting capability. \( \beta \) can be used to determine the most appropriate \( N \). This issue will be studied in Sec. IV-A.

The above scheme determines \( N \) facial traits, \( \{T_j\}_{j=1}^N \), which with their associated trait patterns \( \{[P_{j,q}]_{q=1}^N\}_{j=1}^N \) define the following basic specification of the FTC:

- \( \{T_j\}_{j=1}^N \) specifies a way to decompose a given face into \( N \) trait patches, \( \{f_j\}_{j=1}^N \). The location, size, and orientation of each different trait patch \( f_j \) are all specified by the corresponding \( T_j \).

- To convert \( \{f_j\}_{j=1}^N \) into a FTC codeword, i.e., to encode the given face, each \( f_j \) needs to be classified into a trait pattern \( P_{j,q}^* \) with some specific \( q^* \).

\( P_{j,q}^* \) is a collection of neutral trait samples belong to this pattern. Because \( \{T_j\}_{j=1}^N \) and \( \{[P_{j,q}]_{q=1}^N\}_{j=1}^N \) are extracted from the TES, which is composed of neutral faces only, the trait patterns \( \{[P_{j,q}]_{q=1}^N\}_{j=1}^N \) must be extended such that the extended trait patterns can encode a face with variation caused by different parameters, such as different illumination conditions and different expressions. Let \( \{[\Phi_{j,q}]_{q=1}^N\}_{j=1}^N \) be the extension of \( \{[P_{j,q}]_{q=1}^N\}_{j=1}^N \) for \( q^* \) with variant trait samples added on, using the TVS (Trait Variation Set) and TRS (Trait Enrichment Set) for capturing the patterns of the traits with the aforementioned variations. The second half development of the FTC that focuses on (1) the extension of \( \{[P_{j,q}]_{q=1}^N\}_{j=1}^N \) to \( \{[\Phi_{j,q}]_{q=1}^N\}_{j=1}^N \), (2) the use of \( \{[\Phi_{j,q}]_{q=1}^N\}_{j=1}^N \) for encoding a face into a FTC codeword in the encoding phase, and (3) the determination of a match between a pair of FTC codewords in the decoding phase will be presented in the next section.

III. FTC Encoding and Decoding for Face Recognition

A. Encoding with SVM Classifiers Built on Extended Trait Samples

Because each face in the TVS and TRS must have at least one corresponding (neutral) face in the TES, for any trait sample from the TVS and TRS there must also be at least one corresponding neutral trait sample in the TES which has been clustered into a neutral trait pattern \( P_{j,q}^* \). Assuming that \( P_{j,q}^* \) can be extended in its features by the corresponding trait samples from the TVS and TRS, the extended trait pattern \( \Phi_{j,q} \) can characterize the corresponding trait pattern with variations caused by, for example, illumination conditions, expressions, poses, and others. This assumption further postulates that the pattern of each trait sample from the TVS and TRS can be labeled the same as that of its corresponding trait sample in the TES. This labeling combines the neutral trait samples from the TES and the variant trait samples from the TVS and TRS for a specific and extended trait pattern \( \Phi_{j,q} \) in the facial trait \( T_j \).

When encoding a face, it is first decomposed into \( N \) trait patches, \( \{f_j\}_{j=1}^N \), according to \( \{T_j\}_{j=1}^N \), and each \( f_j \) is then classified into a trait pattern \( \Phi_{j,q}^* \) for some \( q^* \). The classification is achieved using a trait pattern classifier trained upon the combined neutral and variant trait samples from the TES, TVS, and TRS. This winds up with \( N \) classifiers, and each aims at the classification of a trait patch \( f_j \) into one of the trait patterns \( \{[\Phi_{j,q}]_{q=1}^N\}_{j=1}^N \) for the facial trait \( T_j \). Typical pattern classification technique can be apply to build the trait pattern classifier. For example, the LDA (Linear Discriminant Analysis) that learns the feature subspace in which the between-pattern scatter is maximized while the within-pattern scatter is minimized, or SVM that projects the feature to a higher dimensional space in which data are linearly-separable. We conducted an experiment to compare the performance of the trait pattern classifier using the combination of some common pattern recognition techniques such as PCA, LDA, nearest-neighbour classifier and SVM. 2091 images were used for training and 2714 images were used for testing. These images included frontal, uniformly-lit images without facial expression, as well as images taken under variations in illumination and facial expression. Using PCA features with nearest-neighbour classifier yielded 66.3%; LDA features with nearest-neighbour classifier yielded 75.4%; PCA features with SVM yielded...
85.2%; LDA features with SVM yielded 87.0%. Although other features and classifiers are possible, we found that SVM with LDA features delivered promising results, and was thus selected.

The SVM classifier for $T_j$ is then trained on the LDA features extracted from the neutral and variant trait samples which are all labeled in $\Phi_{j,1}, \Phi_{j,2}, ..., \Phi_{j,k_j}$. The set of N SVM classifiers constitutes the FTC encoder. We use the C-SVC with RBF kernel in the libsvm offered by Lin [26]. The grid search resulted in the parameters $C = 8$ and $\gamma = 0.5$ for most SVMs in our experiments. If the trait patch $f_j$ is classified as $\Phi_{j,q^*}$ (from $F_{j,q^*}^{k_j}$) for a certain class label $q^*$, which is an integer between 1 and $k_j$, one can simply use $c_j \equiv q^*$ to symbolize $f_j$. The whole trait patches $[f_j]^{N_{j=1}}$ from a face can then be symbolized as $[c_1, c_2, ..., c_N]$, an FTC hard codeword for the face.

B. Hard Codewords to Probabilistic Codewords

The FTC hard codewords are established on the condition that two trait patches, $f_j^{p_1}$ and $f_j^{p_2}$, corresponding to the same facial trait $T_j$ but from two different subjects, $S_{p_1}$ and $S_{p_2}$, respectively, will be encoded into the same symbol if both trait patches are classified into the same trait pattern. No measurement is made for differentiating the case that trait patches are classified into the same trait pattern. It will be much more informative than far apart different but close different but far apart trait patterns. In this case, we build the similarity/dissimilarity. This property makes the signed distance to the separating hyperplane of the SVM classifier a good candidate for $c_{j,q}$. To make $\bar{c}_j$ a distribution, it is normalized so that $||\bar{c}_j|| = 1$. Fig. 3 shows an example of encoding faces using the hard and the probabilistic codewords, as well as their distances.

C. Decoding for Face Recognition

Face recognition is generally composed of an enrollment phase and an recognition phase. In the enrollment phase, subjects are being enrolled to the database with their facial images collected in a gallery or target set. The difference between a gallery set and a target set is that the former can only have one facial image per subject, but the latter can have multiple facial images per subject [27]. At enrollment, each face in the gallery set (or target set) is encoded into a gallery code (or target code), and then stored in the database. In the recognition phase, a probe is presented for the determination of whether a match in the enrollment database exists or not. The probe face is also firstly encoded into a probe code, and then matched against a gallery code from the database. The decoding phase in the FTC aims at solving this code-match problem, in which some metrics able to measure the similarity between two codewords are needed.

The Hamming distance can be used to measure the similarity between two hard codewords. Given two hard codewords, one is a gallery code $g_b = [g_1, g_2, ..., g_N]$ and the other a probe code $p_h = [p_1, p_2, ..., p_N]$, the difference in between is measured by $d_h = [d_1, d_2, ..., d_N]$ where

$$d_j = \begin{cases} 0 & \text{if } p_j = g_j \\ 1 & \text{otherwise.} \end{cases}$$

$D_h(g_b, p_h)$, the Hamming distance between $g_b$ and $p_h$, is given by the sum of the entries in $d_h$, i.e.,

$$D_h(g_b, p_h) = \sum_{j=1}^{N} d_j. \quad (3)$$

When encoded into probabilistic codewords, the gallery code becomes $g_p = [g_1, g_2, ..., g_N]$ and the probe code becomes $p_p = [p_1, p_2, ..., p_N]$. The distance in between, $D_p(g_p, p_p)$, can
be measured by the sum of the Bhattacharyya distances across all \( N \) entry pairs:

\[
D_p(\bar{g}_j, \bar{p}_j) = \sum_{j=1}^{N} \left( -\ln \frac{1}{N} \sum_{q=1}^{k_j} \sqrt{g_{jq}p_{jq}} \right)
\]

where \( \bar{g}_{jq} \) and \( \bar{p}_{jq} \) is respectively the probability of the gallery face and the probe face belonging to the \( j \)-th trait, and \( k_j \) is the number of the pattern in the \( j \)-th trait.

**D. Computational Complexity of FTC**

1) **Encoding Complexity:** The encoding complexity of the FTC depends on the number of facial trait used, the number of the trait patterns in facial traits, and the type of the classifier applied for trait pattern classification. Basically we can apply any classifier under the FTC framework. As this paper applies LIBSVM [26] as trait pattern classifier, we give the computational complexity using LIBSVM. LIBSVM implements the one-against-one approach for multi-class classification, and it requires \( \frac{k(k-1)}{2} \) binary SVM classifications for a \( k \)-class classification problem. For each binary classification, the following equation has to be evaluated:

\[
f(x) = \sum_{i=1}^{N_s} \alpha_i y_i K(s_i, x) + b
\]

where \( x \) is the input vector, \( s_i \)s are the support vectors, \( N_s \) is the number of the support vector used, \( K(\cdot) \) is the kernel function, which is selected as a radial basis function (RBF), \( \alpha_i s \) are solved in the dual problem and \( y_i \) is the corresponding label. The complexity of evaluating a RBF kernel is \( d \), which is the dimension of \( \cdot \) of the input vector (i.e. \( x \)). The complexity of a binary SVM classification is thus \( O(N_s \cdot d) \). Assuming the number of facial trait used is \( N \), the average number of the trait patterns in facial traits is \( K \), the complexity of FTC encoding is \( O(N \cdot K^2 \cdot N_s \cdot d) \).

2) **Decoding Complexity and Memory Requirement:** The complexity of the FTC decoding depends on the codewords used. For the hard codeword, it simply requires \( N \) integer comparisons, while the probabilistic codeword requires \( N \) Bhattacharyya distance calculations. A probabilistic codeword also needs more space to store than a hard codeword does. It requires \( N \) k-by-1 real arrays, while a hard code requires only \( N \) integers. In summary, the probabilistic codewords renders superior recognition accuracy in exchange of significantly more storage space and computation. TABLE I gives a comparison between the hard and the probabilistic codewords.

**TABLE I**

<table>
<thead>
<tr>
<th>codeword type</th>
<th>hard</th>
<th>probabilistic</th>
</tr>
</thead>
<tbody>
<tr>
<td>data structure</td>
<td>( N ) integers</td>
<td>( N ) k-by-1 real arrays (typically, ( k &lt; 100 ))</td>
</tr>
<tr>
<td>encoding complexity</td>
<td>( N ) SVM classifications</td>
<td></td>
</tr>
<tr>
<td>decoding complexity</td>
<td>( N ) integer comparisons</td>
<td>( N ) Bhattacharyya distance calculations</td>
</tr>
</tbody>
</table>

Consider a FTC with 64 facial traits, each facial trait has exactly 64 trait patterns, the feature dimension for each trait is 100, and there are 1000 faces in the gallery. A corresponding hard codeword requires only 48 Bytes for storage, while a probabilistic codewords requires 16 KBytes, when single precision numbers are used. In this case, it takes 1.52 milliseconds for each LIBSVM classification on average, and it takes 97.3 milliseconds to encode a face. Note that the encoding complexity does not change with the gallery size, while the decoding complexity does. It takes 0.11 milliseconds on average to decode a face using the hard codeword, while it takes 85 milliseconds on average to decode a face using the probabilistic codeword. These results are reported based on C code using a laptop with Intel(R) core(TM)2 Duo CPU P8400 runs at 2.26 GHz and 1.58 GHz 2.96 GB RAM. As expected, decoding using the hard codeword is extremely fast, which is 773 times faster than using the probabilistic codeword. The low complexity of hard codeword makes it useful for a very large gallery. For example, it can be inferred that it takes about one second to match a face against ten million faces using hard codeword.

**IV. PERFORMANCE EVALUATION: A COMPARATIVE STUDY**

The training set, the gallery (or target) set, and the probe set are generally three disjoint sets. The training set is composed of the TES, TVS and TRS datasets for building up the facial traits and their associated trait patterns. If the gallery set happens to be the training set, i.e., the trait patterns are all learned from the gallery set, the performance of the FTC is expected to reach its best. It would be interesting to study the difference in performance between this best case and the general case that the trait patterns are already defined from the training set, and the gallery set can only be encoded using the training-set defined trait patterns. Therefore, two test protocols are considered as follows:

- **Protocol-1:** the training set and gallery set are the same;
- **Protocol-2:** the training set and gallery set are two disjoint sets.

Both protocols were tested on the AR face database [28] and a dataset composed of samples from several face databases. The tests on the AR database gave a way to compare the FTC against other algorithms with reported performance on AR database. However, to reveal that the facial traits can be better defined from a large set of faces collected from different resources, and compared with other algorithms in performance, a mixed dataset was used. The mixed dataset is composed of images collected from AR [28], FERET [29], FRGC [30], FV1 [31], PIE [32] and XM2VTS [33]. The mixed dataset includes 6405 facial images from 903 individuals taken under uncontrolled illumination conditions, facial expressions, and poses, or using different image acquisition devices. The pose variations cover at most 20° toward both sides, up and down. Please refer to the supplementary materials for these facial images. All facial images in both datasets were aligned to the centers of the eyes, and normalized to 80x100 pixels in size. To alleviate the impacts made by illumination variations, all samples were processed to have mean 128 and variance 15.
Two typical face recognition tasks were carried out: identification and verification. In identification, each probe image had one unique match to identify in the gallery set. In verification, each probe image with a claimed subject were both presented to the verification algorithm, which would either accept or reject the claim. A claim would be rejected when the probe failed to match the claimed subject, no matter whether the subject of the probe was in the gallery set or not.

The selected facial traits and their trait patterns define the FTC codebook. The FTC code length is determined by the number of the facial traits. When this number increases, more facial traits will be used to encode a face. In Sec. IV-A, an experimental study will determine this number, and thus the most appropriate FTC code length. Once the code length is determined, the performance of the resulting FTC will be compared against other algorithms in Sec. IV-B. We also report the result on the proposed FTC under the FERET dataset in Sec. IV-C.

A. Study on the Code Length Issue

An experiment with different size of patches, scanning steps, numbers of patches (M) and number of facial traits (N) was conducted to determine the FTC code length. The patches were generated and processed as the following:

1) The minimum patches were cropped by a $0.1W \times 0.1H$ bounding box with a scanning step $t_c \cdot W$ from left to right, and then $t_c \cdot H$ top to bottom, where $t_c$ is the step factor.

2) The bounding box increased with increments $\Delta w$ and $\Delta h$ in the horizontal and vertical direction, respectively. The increment was first added on in one direction only, then in the other direction, and then in both directions. The smallest increments in both directions are $(\Delta w, \Delta h) = (s_c \cdot W, s_c \cdot H)$, where $s_c$ is the size incremental factor. Patches of each size were also cropped by scanning the corresponding bounding box over the face with the step factor $t_c$.

3) The following size incremental factors were considered: $s_c = \{0.35, 0.3, 0.25, 0.2, 0.15, 0.1\}$, and the resulting patch sets were denoted as A, B, C, D, E and F, respectively, as shown in TABLE II.

4) Patches of size less than 5% of the face were ignored as they would introduce noises caused by local misalignment. This reduced the number of patches from $M$ to $M_1$.

5) The patches with poor discriminability were removed by checking the measure in (1) was against a predefined threshold $T_\alpha$. $T_\alpha$ was determined such that the patches with poor discriminability were ignored, while the number of the remaining patches was kept large enough to define a FTC with a large ratio of correctable digits (i.e. $\beta$ in equation (2)). The $T_\alpha$ that gave the best recognition performance was affected by the average number of the patch patterns in each patch and the patch pattern classifier. The former was found to be around 40, and we used the SVM for the latter. Based on our experiments, the best $T_\alpha$ was selected as 0.6, and this reduced $M_1$ to $M_2$, as shown in TABLE II.

In summary, the size of the minimum patches and the step factors were kept the same for all six size incremental factors, which led to the generation of six sets of patches. The numbers of patches in the six sets are given in TABLE II. For each set in TABLE II, we selected $N$ facial traits among the $M_2$ patches using Algorithm 1.

To study the impact made by $N$ on the identification rate, a set of Protocol-1 tests were carried out. We used 3907 neutral images (i.e. nearly frontal, uniformly lit faces without facial expressions) from 903 identities. These images were divided into two disjoint subsets according to the time images were taken. The first subset made up the TES used for patch pattern extraction and facial trait selection; the second subset was used for evaluation of the identification rates. Fig. 4(a) shows the relationship between $N$ and the identification rate for the seven parameter sets.

TABLE II and Fig. 4(a) give the following observations:

- The identification rate increased prominently with $N$ when $N \leq 64$. But the increasing rate was becoming flattened when $N$ increased from 64 to 128 and over.
- Finer step sizes and finer increments gave more patches (i.e. a larger $M$), leading to better identification rates. However, when the step size and increment were reduced to below some values, it can hardly improve the identification rate. The set E in which $M = 1369$ led to the best identification rate, which is 89.1% when 64 facial traits were used (i.e. $N = 64$). In the rest of this paper, we will use $M = 1369$ and $N = 64$ unless otherwise mentioned.

B. Performance Comparison with Other Algorithms

We compared the FTC’s performance with LBP [10] and the algorithm using Sparse Representation (SRC) [34]. SRC is acknowledged as one of the most promising approaches for face recognition published recently. Our comparison also included one algorithm using ECOC [19], as well as two baseline methods, Eigenface [24] and Fisherface [25]. The FTC [13] with hard codewords was dubbed as $FTC_h$, and the FTC with probabilistic codewords was dubbed as $FTC_p$.

Both Protocol-1 and Protocol-2 can be applied in this experiment. However, under Protocol-1 the trait patterns can be learned better than under Protocol-2, since the learned trait patterns are specific to the identities in the gallery. In order to focus only on the code length issue in this experiment, we applied Protocol-1 to reduce the effect caused by trait pattern learning.
TABLE II

SEVEN SIZE-AND-STEP PARAMETER SETS AND THEIR IDENTIFICATION RATES WHEN \(N = 32, 64, 128, \text{and} 256\). The 'ident' stands for identification rate, \(t_c\) and \(s_c\) are the step and the size incremental factors, respectively. The minimum size of the patch is \(0.1W \times 0.1H\).

<table>
<thead>
<tr>
<th>set</th>
<th>(t_c)</th>
<th>(s_c)</th>
<th>(M)</th>
<th>(M_1)</th>
<th>(M_2)</th>
<th>(N = 32)</th>
<th>(N = 64)</th>
<th>(N = 128)</th>
<th>(N = 256)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0.35</td>
<td>225</td>
<td>104</td>
<td>79</td>
<td>.86</td>
<td>.818</td>
<td>.829</td>
<td>.850</td>
<td>-</td>
</tr>
<tr>
<td>B</td>
<td>0.20</td>
<td>324</td>
<td>160</td>
<td>160</td>
<td>.84</td>
<td>.867</td>
<td>.800</td>
<td>.828</td>
<td>.801</td>
</tr>
<tr>
<td>C</td>
<td>0.25</td>
<td>376</td>
<td>229</td>
<td>.359</td>
<td>.880</td>
<td>.884</td>
<td>.859</td>
<td>.905</td>
<td>-</td>
</tr>
<tr>
<td>D</td>
<td>0.20</td>
<td>900</td>
<td>361</td>
<td>.344</td>
<td>.809</td>
<td>.900</td>
<td>.884</td>
<td>.918</td>
<td>.918</td>
</tr>
<tr>
<td>E</td>
<td>0.15</td>
<td>1369</td>
<td>650</td>
<td>.344</td>
<td>.835</td>
<td>.398</td>
<td>.891</td>
<td>.937</td>
<td>.918</td>
</tr>
<tr>
<td>F</td>
<td>0.10</td>
<td>3025</td>
<td>1418</td>
<td>.344</td>
<td>.811</td>
<td>.414</td>
<td>.872</td>
<td>.910</td>
<td>.912</td>
</tr>
<tr>
<td>G</td>
<td>0.06</td>
<td>4900</td>
<td>2062</td>
<td>.375</td>
<td>.825</td>
<td>.414</td>
<td>.864</td>
<td>.414</td>
<td>.913</td>
</tr>
</tbody>
</table>

TABLE III

THE TEST PROTOCOLS. 'E', 'P', 'D', AND 'SPE' STANDS FOR 'EXPERIMENT', 'PROTOCOL', 'DATASET', AND THE NUMBER OF SAMPLE-PER-ENROLLEE, respectively. 'S' AND 'F' STANDS FOR 'SUBJECT' AND 'FACE' respectively.

<table>
<thead>
<tr>
<th>E</th>
<th>P</th>
<th>D</th>
<th>training</th>
<th>gallery</th>
<th>probe</th>
<th>SPE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>AR</td>
<td>A1</td>
<td>126S, 882F</td>
<td>A2</td>
<td>126S, 882F</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>mixed</td>
<td>B1</td>
<td>63S, 882F</td>
<td>B2</td>
<td>63S, 44F</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>AR</td>
<td>C1</td>
<td>397S, 257S</td>
<td>C2</td>
<td>850F</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>mixed</td>
<td>C2</td>
<td>257S, 257F</td>
<td>C2</td>
<td>257S, 257F</td>
</tr>
</tbody>
</table>

TABLE IV

RESULTS OF EXPERIMENT 1 (IN PERCENTAGE). THE 'IDT' AND 'EER' STAND FOR 'IDENTIFICATION RATE' AND 'EQUAL-ERROR RATE MEASURED WHEN FAR EQUALS TO 1%', RESPECTIVELY.

<table>
<thead>
<tr>
<th>algorithm</th>
<th>IDT</th>
<th>HIT at FAR equals to 1%</th>
<th>EER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Eigenface</td>
<td>77.8</td>
<td>90.1 67.8 48.5 10.2</td>
<td></td>
</tr>
<tr>
<td>Fisherface</td>
<td>80.5</td>
<td>89.7 69.9 52.7 10.0</td>
<td></td>
</tr>
<tr>
<td>ECOC</td>
<td>85.3</td>
<td>92.5 88.9 82.8 7.26</td>
<td></td>
</tr>
<tr>
<td>LBP</td>
<td>93.0</td>
<td>93.0 78.6 57.8 8.31</td>
<td></td>
</tr>
<tr>
<td>SRC</td>
<td>90.5</td>
<td>99.7 96.9 91.0 2.0</td>
<td></td>
</tr>
<tr>
<td>FTCp [13]</td>
<td>96.8</td>
<td>99.8 99.4 97.2 1.1</td>
<td></td>
</tr>
<tr>
<td>FTCp</td>
<td>95.0</td>
<td>99.8 96.2 85.6 2.4</td>
<td></td>
</tr>
</tbody>
</table>

The feature dimension that achieved the best performance for Eigenface and Fisherface was 130 and 80, and the corresponding results were reported. The implementation of ECOC applied [127 , 15 , 27] binary BCH code to generate the codewords. The ECOC with codewords of length 127 bit achieved the best performance in our experiment. For LBP, the \([8, 1]\) LBP descriptor was applied, as suggested in [10]. For the SRC algorithm, we implemented the Eigen+SRC that achieved promising results in [34], which applied the PCA features.

TABLE III summarizes the four experiments performed in this study. The training set is the set of images for training algorithms: it was used to extract the eigen-component in Eigenface, Fisherface, and SRC; it was divided into TES and TVS in FTC-based algorithms. The probe set contains faces for testing. If a probe, a face in the probe set, also exists in the gallery, then it is known as an enrollee; otherwise, it will be referred to as an imposter. Please refer to the supplementary material for the images used in our experiment.

Experiment 1 studied the performance under Protocol-1, where the gallery set was the same as the training set. Experiment 2 studied performance under Protocol-2, where the gallery set and the training set were disjoint. It also studied how recognition performance varied with SPE (number of Sample-Per-Enrollee). TABLE IV gives the results of

3The [127 , 15 , 27] binary BCH code was able to correct up to 27 error bits. For the proposed FTC, we used 64 facial traits following the conclusion given in Section IV-A. In TABLE II, \(N = 64\) with set E was applied. Given \(\beta = 0.398\) in this case, the FTC can correct up to \(64*0.398 = 25\) digits.
Experiment 1. Both FTC\textsubscript{h} and FTC\textsubscript{p} outperformed most algorithms, except for SRC. FTC\textsubscript{h} and FTC\textsubscript{p} outperformed SRC in identification rate, but SRC appeared slightly better than FTC\textsubscript{p} in hit rate and EER. However, FTC\textsubscript{h} appeared to give the best overall performance. The identification and hit rates for Experiment 2 are shown in Fig. 5 (a) and (b), respectively. The performance of most algorithms degraded significantly when SPE decreased, except FTC\textsubscript{h} and FTC\textsubscript{p}. FTC\textsubscript{p} (the red dotted lines) gives the best overall performance in this test. From TABLE IV and Fig. 5, it can be seen that FTC\textsubscript{h} performed exceptionally well when it was allowed to build the codebook using the gallery samples (i.e. Protocol-1). However, for a more practical scenario when the gallery and probe were disjoint (i.e. Protocol-2), FTC\textsubscript{p} performed better. The reason beyond is that the hard codeword matching in FTC\textsubscript{h} keeps the best matched pattern only and ignores other similar patterns. Under Protocol-1 when the trait patterns are learned specifically for the gallery identities and thus the trait patterns among them are known, using only the best matched pattern seems to be a slightly better strategy. However, under Protocol-2 when the gallery identities are unknown during the trait pattern learning stage and there are possibly some unseen patterns among them, it seems a better strategy is not to totally trust the best matched pattern but also consider similar patterns.

Experiment 3 studied the performance variations caused by illumination, expression and pose. Experiment 4 applied the same setup as for Experiment 3, but considering the most challenging scenario where SPE=1. Face recognition from a single image remains an important task in many practical applications and some works tried to handle this problem. For example, [35] argued that the appearance seen through a sliding window overlaid over an image of a face, traces a trajectory over a 2D manifold embedded in the image space, and demonstrated this concept can be exploited in the single image based recognition. Our work also handle the single image based recognition, but in a different way to [35]. Instead of learning the trajectory over a 2D manifold embedded in the image space, we conceptually learn the manifold embedded in each local facial features, and select the most discriminating ones for face recognition. TABLE V summarizes the results of Experiment 3 and 4, in terms of identification and verification rates. In Experiment 3, SRC and LBP yielded good results when probe samples under variations in illumination, expression and pose. The performance of FTC\textsubscript{h} degraded considerably in this case, yielding slightly worse overall performance than SRC and LBP. FTC\textsubscript{p} yielded significantly better results than all other algorithms in this experiment. In Experiment 4 when each enrollee has only one facial sample for enrollment (SPE=1), the performance of all algorithms again degraded dramatically except FTC\textsubscript{p}. FTC\textsubscript{p} outperformed FTC\textsubscript{h}, SRC and LBP by 14.9% 38.3% 13.4% respectively in identification rate. Similar performance was observed in hit rates and ERRs as well. Note that both SRC and LBP do not need a training phase while the proposed FTC does, and SRC typically requires gallery set contains an adequate amount of samples to represent the probe sample. The Experiment 4 with SPE=1 demonstrated that when we were allowed to train our model offline but we had only a single image for each enrollee in the gallery online, which is common in the real applications, the proposed FTC performed much better than some algorithms that did not require the training phase. Please refer to the Appendix B for the rank-N identification rates and the receiver operating characteristic (ROC) curves for Experiment 3 and 4.

C. Results on FERET Dataset

To allow readers to have more clear understanding of the performance of the proposed method while comparing with the state-of-the-art face recognition methods, we also report the results of the proposed algorithm under the FERET [29] dataset using the public training/testing protocols. Four image sets were evaluated, which were dubbed as fabf, fabc, DupI and DupII. fabf included 1195 images with alternative facial expressions. fabc included 194 images taken under different illumination conditions. DupI included 722 obtained anywhere between one minute and 1031 days after their respective gallery matches. The harder DupII 234 probe images were a strict subset of the DupI; they were those taken only at least 18 months after their gallery entries. All the tests used a single gallery containing 1196 images. The results are shown in TABLE VI.

We included the result from a dictionary learning method reviewed in the introduction for comparison, which was known SOM face [23]. As reviewed in the introduction, SOM face extended the bag-of-features representation to face recognition in some sense using the Self-Organizing Map (SOM) [36]. We also included the results of LBP [10] and LGBPS [37] for comparison. The algorithm described in [10] divided a face into 7*7 windows. The results of ‘LBP’ in TABLE VI was obtained using equal weights for all the 49 facial regions, while the ‘Weighted LBP’ assigned different weights for different facial regions. The algorithm described in [37], known as the Local Gabor Binary Pattern Histogram Sequence (LGBPHS), modelled a face image as a `histogram sequence’ by concatenating the histograms of all the local facial regions of all the local Gabor magnitude binary pattern maps. For recognition, histogram intersection was used to measure the similarity between different LGBPHSes and the nearest neighbour was exploited for final classification. The results of ‘Direct LGBPHS’ in TABLE VI was obtained using equal weights for all the facial regions. The results of the ‘Weighted LGBPHS’ was obtained by weighting facial regions using the weights specifically learned for the FERET data set.

From TABLE VI we can see that the proposed FTC significantly outperformed LBP and its weighted version for all the subsets. Tan et al. only reported their result for fabf subset in [36], and FTC outperformed SOM face in this subset. FTC had better performance than ‘Direct LGBPHS except the fabc subset. It seems that Gabor features employed in [37] is particularly robust to illumination variation. This results indicates that using illumination-robust features such as Gabor features could potentially boost the FTC performance, as the current FTC implementation uses image intensity. The Weighted LGBPHS had the same recognition rate with the
## V. Conclusion and Future Work

We propose a novel human face coding scheme, Facial Trait Code (FTC), for face recognition. FTC extracts various sizes of patches across a face, and encodes the extracted patches into two different forms of codewords good for denoting the uniqueness of the face. The hard codeword extracts the coarse patterns from each trait, and combines the discriminative power of the coarse patterns from many traits to compose a strong classifier. It offers a low-cost while highly efficient solution for face recognition. On the contrary, the probabilistic codeword extracts the fine features from the coarse patterns in each trait, and measures the subtle difference between the gallery and probe using the fine features. It offers a high-capacity and highly accurate alternative to the hard codeword. Both are experimentally proven competitive to the state-of-the-art.

An immediate extension of our work can be on video-based face recognition. It is common in video-based recognition that a video clip is used for enrollment and a disjoint clip is used as a probe [38]. A video clip generally contains multiple images. Using multiple images for enrollment and recognition typically achieves better performance than using single image. Concerning the benefit of using multiple enrollment images, Table V shows using single image for enrollment (SPE=1) achieves 87.8% on the neutral set, while using multiple images (average SPE=2.8) achieves 95.3% on the same set. Similar results can be observed on the variant set. Concerning the benefit of using multiple probe images, the majority vote can be applied to obtain better accuracy. One possible issue in video-based recognition is the low resolution image sequences. As the proposed facial trait code classifies the local human facial patches into few major patterns, we need only the information of rough shapes instead of facial details. Hence we believe that FTC is also applicable to low-resolution videos.

We plan to improve the performance using some latest developments. For example, the Local Ternary Patterns proposed
in [39] can be applied to deal with illumination. The nonlinear discriminant features, such as those extracted by Generalized Discriminant Analysis (GDA) [40], can be applied to handle expressions. 3D facial model can be effective in handling large poses. The FTD framework can incorporate range data while maintaining the rest of the algorithm the same.

ACKNOWLEDGMENT

This work was supported in part by the National Science Council, Taiwan, under Grants NSC 98-2221-E-002-127-MY3, and by the Ministry of Economic Affairs, Taiwan, under Grant 100-EC-17-A-02-S1-032.

REFERENCES

Ping-Han Lee  Ping-Han Lee received his B.Sc. and M.Sc. from Naval Architecture and Ocean Engineering, National Taiwan University in 2000 and 2002, respectively. He received his Ph.D degree from the Graduate Institute of Computer Science and Information Engineering, National Taiwan University in 2010. He works in MediaTek Inc. now as a senior engineer. His research interests include face recognition, video surveillance, computer vision and image processing.

Gee-Sern Hsu  Gee-Sern Hsu received the dual M.S. degree in electrical and mechanical engineering, and the Ph.D. degree in mechanical engineering from the University of Michigan (Ann Arbor) in 1993 and 1995, respectively. He was a postdoctoral fellow at the University of Michigan from 1995 to 1996, and then a senior research staff at the National University of Singapore from 1997 to 2000. In 2001, he joined Penpower Technology for leading research on face recognition and intelligent video surveillance. The products designed by his team won the best innovation awards in SecuTech Expo in 2005, 2006 and 2007. In 2007, he joined the Department of Mechanical Engineering at the National Taiwan University of Science and Technology as an assistant professor. His research interests are in the area of computer vision, especially on face recognition, visual tracking, object and event recognition.

Yi-Ping Hung  Yi-Ping Hung received his B.S. in Electrical Engineering from the National Taiwan University in 1982 and Ph.D. from the Division of Engineering at Brown University in 1990. He is currently a professor in the Graduate Institute of Networking and Multimedia, and in the Department of Computer Science and Information Engineering, at the National Taiwan University. From 1990 to 2002, he was with the Institute of Information Science, Academia Sinica, Taiwan, where he became a tenured research fellow in 1997 and is now an adjunct research fellow. He served as a deputy director of the Institute of Information Science from 1996 to 1997, and received the Young Researcher Publication Award from Academia Sinica in 1997. Since 2007, he has served as the director of the Graduate Institute of Networking and Multimedia at the National Taiwan University. He was the program co-chairs of ACCV’00 and ICAT’00, as the workshop co-chair for ICCV’03. He has served in the editorial board of the International Journal of Computer Vision since 2004. His current research interests include computer vision, image processing, multimedia system and human-computer interaction.

Tsuhan Chen  Tsuhan Chen has been with Cornell University, Ithaca, New York, since January 2009, where he is the David E. Burr Professor of Engineering and Director of the School of Electrical and Computer Engineering. From October 1997 to December 2008, he was with the Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, Pennsylvania, as Professor, and as Associate Department Head in 2007-2008. From August 1993 to October 1997, he worked at AT&T Bell Laboratories, Holmdel, New Jersey. He received the M.S. and Ph.D. degrees in electrical engineering from the California Institute of Technology, Pasadena, California, in 1990 and 1993, respectively. He received the B.S. degree in electrical engineering from the National Taiwan University in 1987. Tsuhan received the Charles Wilts Prize at the California Institute of Technology in 1993. He was a recipient of the National Science Foundation CAREER Award, from 2000 to 2003. He received the Benjamin Richard Teare Teaching Award in 2006, and theEta Kappa Nu Award for Outstanding Faculty Teaching in 2007. He was elected to the Board of Governors, IEEE Signal Processing Society, 2007-2009, and a Distinguished Lecturer, IEEE Signal Processing Society, 2007-2008. He is a Fellow of IEEE.