Recognizing Partially Occluded Faces from a Single Sample Per Class Using String-Based Matching

Weiping Chen[1] and Yongsheng Gao[1,2]

[1] School of Engineering, Griffith University, Australia
[2] National ICT Australia, Queensland Research Lab
{luke.chen,yongsheng.gao}@griffith.edu.au

Abstract. Automatically recognizing human faces with partial occlusions is one of the most challenging problems in face analysis community. This paper presents a novel string-based face recognition approach to address the partial occlusion problem in face recognition. In this approach, a new face representation, Stringface, is constructed to integrate the relational organization of intermediate-level features (line segments) into a high-level global structure (string). The matching of two faces is done by matching two Stringfaces through a *string*-to-*string* matching scheme, which is able to efficiently find the most discriminative local parts (substrings) for recognition without making any assumption on the distributions of the deformed facial regions. The proposed approach is compared against the state-of-the-art algorithms using both the AR database and FRGC (Face Recognition Grand Challenge) ver2.0 database. Very encouraging experimental results demonstrate, for the first time, the feasibility and effectiveness of a high-level syntactic method in face recognition, showing a new strategy for face representation and recognition.

Keywords: Partial occlusion, Stringface.

1 Introduction

Face recognition has attracted much attention in both academic and industrial communities during the past few decades. A great deal of progress has been made to robustly identifying faces under controlled condition. However, recognizing faces under uncontrolled conditions remains challenging open problems in face recognition community. A face recognition system can be confront occluded faces in real world applications very often due to use of accessories, such as scarf and sunglasses. Hence, the face recognition system has to be robust to occlusion in order to guarantee reliable real-world applications. Recognizing partially occluded face has received considerable attention in recent years [1][11] [14][18].

Penev and Atick [14] proposed a Local Feature Analysis (LFA) technique by modifying PCA to solve the partial occlusion problem. LFA is a derivative of the eigenface method and utilizes specific facial features such as eyes, mouth

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part III, LNCS 6313, pp. 496–509, 2010.

and nose for identification instead of the entire representation of the face. These features are used as the basis for representation and comparison. Its performance is dependent on a relatively constant environment and the quality of the image. Bartlett et al. [1] presented an Independent Component Analysis (ICA) architecture to find a spatially local face representation. Conceptually, LFA also finds local basis images for face using the second-order statistics but its kernels are not sensitive to the higher than second-order dependencies in a face image. On the contrast, Independent Component Analysis (ICA) architecture I is sensitive to these high-order statistics. It treats the images as random variables and the pixels as outcomes to find a set of statistically independent basis images. Martinez [11] proposed a probabilistic face recognition approach that could compensate for the imprecise localization, partial occlusion, and extreme expressions with a single training sample. In his method, face images are analyzed locally in order to handle partial face occlusion. The face image is first divided into k local regions and for each region an eigenspace is constructed. If a region is occluded, it is automatically detected. Moreover, weighting of the local regions were also proposed in order to provide robustness against expression variations. Recently, Wright et al. [18] presented a partition Sparse Representation Classification (SRC) method which is inspired by the ideal of compressed sensing. In their method, a face is first partitioned into blocks and compute an independent sparse representation for each block. Then a general classification algorithm and a voting method are used to recognize face images.

In this paper, we propose a novel Stringface representation and matching concept for face recognition with one single model image per person under partial occlusions. Cognitive psychological studies [2][3] indicated that human beings recognize line drawings as quickly and almost as accurately as gray-level images since the line drawings preserve most important feature information. In addition, line segments are less sensitive to illumination changes and local variations as they integrate the inherent local structural characteristics with spatial information of a face image [5]. Based on these findings, we represent a face by an attributed string (Stringface), which groups the relational organization of intermediate-level features (line segments) into a high-level global structure representation. Because the Stringface represents not only the local structural information but also the global structure of a face, it improves upon the local characteristics of feature-based methods [5][7]. Furthermore, the Stringface can be constructed using only a single face image and without training stage involved in this approach. The matching of two frontal faces is done by matching two Stringfaces through a *string*-to-*string* matching scheme. The proposed attributed string matching concept is able to effectively find the most discriminative local parts (substrings) for recognition without making any assumption on the distributions of the deformed facial regions. This substring matching ability is used to address the occlusion problem. This is believed to be the first piece of work on frontal face analysis using a high-level syntactic matching method. The studies and experimental results in this paper are confined to human frontal face recognition. We deal with partial occlusion, but we do not explicitly account for other conditions, such as illumination, expressions and pose. We also assume the detection, cropping and normalization of the face have been performed prior to applying our algorithm.

The paper is organized as follows: Section 2 defines the Stringface representation and matching concept in detail. A feasibility investigation and performance evaluation of the proposed approach is given in Section 3. Finally, the paper concludes in Section 4.

2 Proposed Stringface Recognition Approach

String matching is a syntactic and structural method for similarity measurement between strings or vectors, which has been widely used for pattern search in molecular biology, speech recognition, and file comparison. Strings can be classified into two categories: symbolic strings and attributed strings. The symbolic string matching is widely used for shape recognition, in which shapes are described by string representation and primitives are described by symbols. However, symbols are discrete in nature while most problems of pattern recognition deal with attributes that are basically continuous in nature. It was found inadequate to use symbols as primitives for complex pattern recognition [6]. Hence, the attributed string matching [4][6] were proposed and the attributed string representation makes it easier to handle noise and distortion. One advantage of using variant attributes (location, length and orientation) is that segment merging becomes possible. However, string matching was believed a technique not suitable for frontal face recognition due to its highly ordered global representation and complex nature of a human face. The only most related work [6] is attempted on human face profile. Their method is based on Needleman-Wunsch algorithm [13], which performs a global alignment on two sequences of profile line segments, which fails to work when a face profile has large local shape deformations or occlusions. Obviously, this face representation is not able to describe frontal faces as it only can represent the continuous silhouette of a profile face, ignoring other important but unconnected distinctive features, such as the eyes, eyebrows, mouth and ears. In this study, we propose a novel string representation and matching concept to recognize frontal faces, an unattempted area, to address the challenging problem of face recognition with partial occlusions.

2.1 Stringface Representation

A novel syntactic face representation is proposed here to integrate the structure connectivity information of line segments in a face image. The basic primitives of our syntactic representation are line segments, which are generated by a polygonal line fitting process [9] from a face edge map. Each line segment, L, is represented as $L(l, \theta, x, y)$, where attributes l, θ, x and y are the length, direction and midpoint location of the line, respectively. The line direction θ is defined as the the minimum angle formed between the line segment and the reference line. The line between two eyes is used as the reference line in this study.

Definition 1. *A Stringface (SF) is defined as a syntactic representation of human face, which is viewed as being composed of a set of substrings S_i ($S_i \in SF$). Substrings are connected by null primitives ϕ linking the ith substring S_i and the $(i+1)$th substring S_{i+1} in SF.*

$$SF = S_1^{SF} \phi S_2^{SF} \phi \cdots \phi S_{n-1}^{SF} \phi S_n^{SF}, \tag{1}$$

where n is the number of substrings. The ith substring S_i^{SF} is given by

$$S_i^{SF} = L_j^{SF} L_{j+1}^{SF} \cdots L_{j+m_i}^{SF}, \tag{2}$$

where L_j^{SF} is the jth primitive in SF as well as the first primitive in substring S_i^{SF} and (m_i+1) is the number of primitives in S_i^{SF}, $i = 1, \ldots, n$.

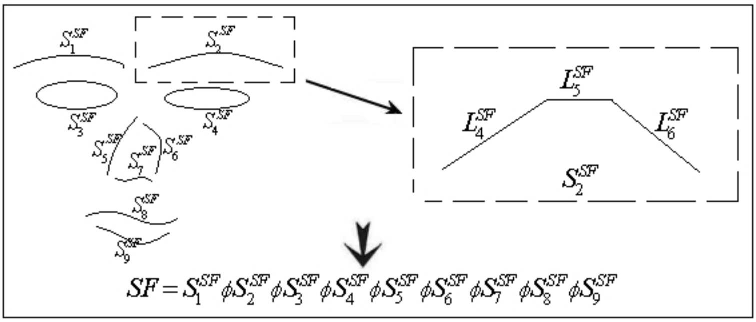

Fig. 1. An example Stringface representation

Fig. 1 illustrates a Stringface representation $SF = S_1^{SF} \phi S_2^{SF} \cdots \phi S_8^{SF} \phi S_9^{SF}$ generated from line segments, where ϕ is a null primitive connecting two substrings and each substring S_i^{SF} is a consecutive run of connected line segments $L_j^{S_i}$ (called primitives), $i = 1, \ldots, 9$. In Fig. 1, S_2^{SF} is composed of three line segments as

$$S_2^{SF} = L_4^{SF} L_5^{SF} L_6^{SF} \tag{3}$$

2.2 Cost Functions

The goal of string matching algorithms is to find a sequence of elementary edit operations which transform one sting into another at a minimal cost. The elementary operations for string matching are deletion, insertion, and substitution.

1. **Substitution (or Change)**: to replace a symbol or primitive (e.g. a in $S1$) with the other (e.g. b in $S2$), denoted as $a \rightarrow b$.
2. **Insert**: to insert a symbol or primitive (e.g. b) into a string (e.g. $S1$), denoted as $\phi \rightarrow b$, where ϕ is a symbol used to denote nothing (called null symbol).
3. **Delete**: to delete a symbol or primitive (e.g. a) from a string (e.g. $S1$), denoted as $a \rightarrow \phi$.

A new edit operation, **merge**, is introduced in attributed string matching, which can address the noise and distortion issues. The merge operation is used to combine any number of consecutive primitives in one string and match with those in the other string. An example of merge operation is illustrated in Fig. 2, where primitives L^{SF}_{i-k+1}, ... and L^{SF}_{i} are combined into a new primitive $L^{SF}_{i\rightarrow k}$.

We define new cost functions for edit operations of change, insert, delete and merge. Let SF_1 and SF_2 be the input and model Stringfaces, respectively. $L^{SF_1}_{i}$ and $L^{SF_2}_{j}$ are the ith and the jth primitives in SF_1 and SF_2 with attributes $(l_i, \theta_i, x_i, y_i)$ and $(l_j, \theta_j, x_j, y_j)$. Let $SF\langle i \rightarrow j\rangle$ specify the substring in SF from the ith to the jth primitives, that is $SF\langle i \rightarrow j\rangle = L^{SF}_{i} L^{SF}_{i+1} \cdots L^{SF}_{j}$. The cost functions of the proposed Stringface matching method are described as follows.

The cost function for change operation from $L^{SF_1}_{i}$ to $L^{SF_2}_{j}$ is defined as

$$Cost[Change(L^{SF_1}_{i}, L^{SF_2}_{j})] = |l_i - l_j| + f(\triangle(\theta_i, \theta_j)) + \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}, \tag{4}$$

where $\triangle(\theta_i, \theta_j)$ is the angle difference between two primitives (see Eq.(5)), and $f()$ is a non-linear penalty function to map the angle to a scalar using $f(x) = x^2/W$, and $W = 50$ is the weight to balance the angle and length.

$$\triangle(\theta_i, \theta_j) = \begin{cases} |\theta_i - \theta_j| & : |\theta_i - \theta_j| \leq 90^o, \\ 180^o - |\theta_i - \theta_j| & : 90^o < |\theta_i - \theta_j| \leq 180^o, \\ |\theta_i - \theta_j| - 180^o & : 180^o < |\theta_i - \theta_j| \leq 270^o, \\ 360^o - |\theta_i - \theta_j| & : 270^o < |\theta_i - \theta_j| \leq 360^o. \end{cases} \tag{5}$$

The costs of delete and insert operations can be derived from the above change cost function by introducing a null primitive ϕ with zero length and indefinite angle and location. The cost functions of these two operations are defined as

$$Cost[delete(L^{SF_1}_{i})] = f(K_\theta) + l_i + K_{loc}, \tag{6}$$

$$Cost[insert(L^{SF_2}_{j})] = f(K_\theta) + l_j + K_{loc}, \tag{7}$$

where K_θ and K_{loc} are constants to represent the indefinite orientation and location of the line segment. For the purpose of penalization, 90^o and the diagonal distance of the input image, which are the maximum angle difference and the maximum location difference, are used for K_θ and K_{loc}.

Next, we consider the merge operation. The merge operation is used to combine and match any number of consecutive primitives in one face with those in

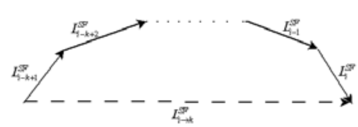

Fig. 2. An example of the merge operation

the other. Let $SF\langle i-k+1 \to i\rangle = L^{SF}_{i-k+1}L^{SF}_{i-k+2}\cdots L^{SF}_{i}$ be a substring with k primitives in Stringface SF to be merged, and $L^{SF}_{i\to k}$ be the merged primitive of these k primitives. An example of merge operation is illustrated in Fig. 2. The merge operation is denoted as $merge(SF\langle i-k+1 \to i\rangle, L^{SF}_{i\to k})$. If $k=1$, $L^{SF}_{i\to k} = L^{SF}_{i}$. This is the case without any merge operation. The merge cost is defined as

$$Cost[merge(SF\langle i-k+1 \to i\rangle, L^{SF}_{i\to k})] = f(\frac{k-1}{l^k}\sum_{p=i-k+1}^{i} \triangle(\theta^k, \theta_p) \times l_p), \quad (8)$$

where k is the number of merged primitives. l^k and θ^k are the length and the line direction of the merged primitive $L^{SF_1}_{i\to k}$, l_p and θ_p are the length and the line direction of primitive L^{SF}_{p} in $SF\langle i-k+1 \to i\rangle$ before merging. Now, by considering $L^{SF}_{i\to k}$ as a single primitive, the cost function for a change operation after merge can be rewritten as:

$$Cost[Change(L^{SF_1}_{i\to k}, L^{SF_2}_{j\to l})] = |l^k - l^l| + f(\triangle(\theta^k, \theta^l)) + \sqrt{(x^k-x^l)^2 + (y^k-y^l)^2}, \quad (9)$$

which is performed after the k primitives in $SF_1\langle i-k+1 \to i\rangle$ are merged as $L^{SF_1}_{i\to k}$ and the l primitives in $SF_2\langle j-l+1 \to j\rangle$ are merged as $L^{SF_2}_{j\to l}$. If $k=1$ and $l=1$, no merge is performed and the above change operation reduces to the conventional one-to-one change operation $Change(L^{SF_1}_{i}, L^{SF_2}_{j})$ (see Eq.4)

2.3 Dynamic Merge Limit Determination

The Stringface is composed of substrings and null primitives. The merge limit $merge\_limit^{SF}_{i}$ is used to ensure that the merge operation is restricted in the same substring, which means that primitives in substring S^{SF}_{i} cannot be merged with primitives in its neighboring substrings S^{SF}_{i-1} and S^{SF}_{i+1}. Let SF denote a Stringface:

$$SF = S^{SF}_1 \phi S^{SF}_2 \phi \cdots \phi S^{SF}_n = L^{SF}_1 L^{SF}_2 \cdots L^{SF}_N \quad (10)$$

where L^{SF}_i is the ith line primitive (including ϕ) in SF and S^{SF}_j is the jth substring (curve primitive) in SF, $j=1,\ldots,n$, $i=1,\ldots,N$, N and n are the number of line primitives plus the number of null primitives (ϕ) and the number of curve primitives in SF, respectively. Let $|S^{SF}_j|$ be the length of jth substring. For a primitive L^{SF}_i, if $L^{SF}_i \in S^{SF}_j$, then its $merge\_limit$ is defined as

$$merge\_limit^{SF}_i = i - \sum_{t=1}^{j-1}|S^{SF}_t| - j + 1 \quad (11)$$

2.4 Similarity Measure via Dynamic Programming

The similarity between the two faces can be characterized by the edit operation cost using Dynamic Programming (DP) between the two Stringfaces. Let

$SF_1 = L_1^{SF_1} \cdots L_{N_1}^{SF_1}$ and $SF_2 = L_1^{SF_2} \cdots L_{N_2}^{SF_2}$ be string representations of input face and model face, respectively, where N_1 and N_2 are numbers of primitives in SF_1 and SF_2. To find pairs of strings with high degrees of similarity, we set up a similarity matrix S. Let the input Stringface (i.e. SF_1) has N_1 primitives represented by the rows of the similarity matrix S, and let the model Stringface (i.e. SF_2) has N_2 primitives represented by the columns of the similarity matrix S. First we initialize

$$S(i,0) = S(0,j) = 0 \ (0 \leq i \leq N_1 \ , \ 0 \leq j \leq N_2). \tag{12}$$

$S(i,j)$ is the similarity of two strings ending at $L_i^{SF_1}$ and $L_j^{SF_2}$. If $L_i^{SF_1} = \phi$ or $L_j^{SF_2} = \phi$, $S(i,j) = 0$. If $L_i^{SF_1} \neq \phi$ and $L_j^{SF_2} \neq \phi$, $S(i,j)$ is defined as:

$$S(i,j) = \max \begin{cases} 0 \\ S(i,j-1) - Cost[\phi \rightarrow L_j^{SF_2}] \\ S(i-1,j) - Cost[L_i^{SF_1} \rightarrow \phi] \\ \max_{k,l}\{S(i-k,j-l) \\ \quad +c(SF_1\langle i-k+1 \rightarrow i\rangle, \\ \qquad SF_2\langle j-l+1 \rightarrow j\rangle)\} \end{cases} \tag{13}$$

where $Cost[\phi \rightarrow L_j^{SF_2}]$ and $Cost[L_i^{SF_1} \rightarrow \phi]$ are costs of insert and delete edit operations, respectively (see Eq.6). ϕ is a null primitive. $c(SF_1\langle i-k+1 \rightarrow i\rangle, SF_2\langle j-l+1 \rightarrow j\rangle)$ is defined as follows:

$$\begin{aligned} &c(SF_1\langle i-k+1 \rightarrow i\rangle, SF_2\langle j-l+1 \rightarrow j\rangle)\} = \lambda \\ &-Cost[SF_1\langle i-k+1 \rightarrow i\rangle, SF_2\langle j-l+1 \rightarrow j\rangle], \end{aligned} \tag{14}$$

where $Cost[SF_1\langle i-k+1 \rightarrow i\rangle, SF_2\langle j-l+1 \rightarrow j\rangle]$ is the cost of merge and change edit operations between substring $SF_1\langle i-k+1 \rightarrow i\rangle$ and $SF_2\langle j-l+1 \rightarrow j\rangle$ (see Eq. 8 and Eq.9). k and l are numbers of the merged primitives. In Eq.14, λ is used to decide the similarity between primitives $SF_1\langle i-k+1 \rightarrow i\rangle$ and $SF_2\langle j-l+1 \rightarrow j\rangle$. If the cost value $Cost[SF_1\langle i-k+1 \rightarrow i\rangle, SF_2\langle j-l+1 \rightarrow j\rangle]$ is less than λ, these primitives are considered as similar elements.

For two Stringfaces, SF_1 and SF_2, with primitives $L_i^{SF_1}(i = 1,2,\ldots,N_1)$ and $L_j^{SF_2}(j = 1,2,\ldots,N_2)$, we compute all the similarity costs between their primitives and obtain the similarity matrix S and edit operations matrix M:

$$S = \begin{pmatrix} S(0,0) & S(0,1) & \ldots & S(0,N_2) \\ \vdots & \vdots & \vdots & \vdots \\ S(N_1,0) & S(N_1,1) & \ldots & S(N_1,N_2) \end{pmatrix} \tag{15}$$

$$M = \begin{pmatrix} M(0,0) & M(0,1) & \ldots & M(0,N_2) \\ \vdots & \vdots & \vdots & \vdots \\ M(N_1,0) & M(N_1,1) & \ldots & M(N_1,N_2) \end{pmatrix} \tag{16}$$

The pair of substrings with maximum similarity is found by first locating the maximal element in S. The other matrix elements leading to this maximal value are then sequentially determined with a traceback procedure ending with an element of $S(i,j)$ equaling to zero. For example, $S(i_1, j_1) = v_1$ is the maximal element with maximal value v_1 in S. $M(i_1, j_1) = (r_1, c_1)$ is corresponding edit operations with value r_1 and c_1 in M. Then, the next element leading to $S(i_1, j_1)$ is then sequentially determined by r_1 and c_1. If $S(i_1 - r_1, j_1 - c_1) \neq 0$, then $S(i_2, j_2)$ is one of the element in the matched pair of substrings, where $i_2 = i_1 - r_1$ and $j_2 = j_1 - c_1$. The corresponding edit operations of $S(i_2, j_2)$ is $M(i_2, j_2) = (r_2, c_2)$ All elements of can be found using this procedure, until the element $S(i_k - r_k, j_l - c_l) = 0$, where $k \geq 1$ and $l \geq 1$. The pair of segments with the next best similarity is found by applying the traceback procedure to the second largest element in S not associated with the first traceback.

```
S(0,0) := 0
for i := 1 to N_1 do S(i,0) := 0;
for j := 1 to N_2 do S(0,j) := 0;
for i := 1 to N_1
 for j := 1 to N_2
   if L_i^{SF_1} = φ or L_j^{SF_2} = φ
         S(i,j) = 0;
   else
     m_1 := S(i,j-1) - Cost[insert(L_j^{SF_2})];
     m_2 := S(i-1,j) - Cost[delete(L_i^{SF_1})];
     for k := 1 to merge_limit_i^{SF_1}
       for l := 1 to merge_limit_j^{SF_2}
         T[k,l] := S(i-k, j-l) + λ
              -{Cost[merge(SF_1(i-k+1 → i), L_{i→k}^{SF_1})]
              + Cost[merge(SF_2(j-l+1 → i), L_{j→l}^{SF_2})]
              + Cost[change(L_{i→k}^{SF_1}, L_{j→l}^{SF_2})]}
     m_3 := max(T[k,l]);
     S(i,j) := max(0, m_1, m_2, m_3);
     if S(i,j) = m_3 , M(i,j) = argmax_{k,l}(T(k,l));
     if S(i,j) = m_1 , M(i,j) = (1,0);
     if S(i,j) = m_2 , M(i,j) = (0,1);
     if S(i,j) = 0 ,   M(i,j) = (0,0);
   end
```

Algorithm 1. Proposed Stringface matching.

String matching is conducted according to Algorithm 1, where $merg\_limit_i^{SF_1}$ and $merg\_limit_j^{SF_2}$ are controlling upper limits on the number of primitives to be merged into a new one in Stringfaces SF_1 and SF_2, respectively (as discussed in Section 2.3). The similarity of associating a group of segments from Stringface SF_1 with a group of segments from Stringface SF_2 is computed as

$$s(SF_1, SF_2) = \xi \times \sum_{i=1}^{f} S_i, \tag{17}$$

The term S_i is the similarity (the ith maximal element in S matrix table) of the ith best similar substrings between two Stringfaces and f is the number of best similar substrings. ξ is a weight term which emphasizes the importance of matching large parts from both Stringfaces in accordance to the way that humans pay more attention to large shape parts when judging the quality of matching [15]. The proportion of the matched substrings lengths with respect to their total length is used to define ξ:

$$\xi = \frac{length\ of\ matched\ SF_1 + length\ of\ matched\ SF_1}{length\ of\ SF_1 + length\ of\ SF_2} \tag{18}$$

3 Experimental Verification

In this section, we present a system performance investigation on publicly available databases, which covers human face recognition with real and synthetic occlusions.

3.1 Databases and Experimental Settings

In this study, two well-know face databases (AR [12] and FRGC ver2.0 [16]) were tested. The AR database contains faces with different conditions, including partially occluded condition, which can not be found in other latest databases. Hence, AR database is particularly suitable for our evaluation. The FRGC ver2.0 dataset is much larger than the AR database, and is used to test the performance of our proposed method with occlusion variations.

The AR database consists of over 4,000 frontal view images for 126 individuals (70 males and 56 females). Each person has 26 images captured in two different sessions (separated by two-week time interval). Each session contains 13 face images under different light conditions (right light, left light and both lights), different facial expressions (smiling, anger and screaming) and partial occlusions (sunglasses and scarf). Some images were found missing or corrupted for a few subjects. We chose a subset of the data set consisting of 50 male subjects and 50 female subjects for our experiments.

The FRGC ver2.0 dataset consists of 50,000 recordings divided into training and validation partitions. The training partition is designed for training algorithms. The training set consists of 12,776 images from 222 subjects, with 6,388 controlled still images and 6,388 uncontrolled still images and contains from 9 to 16 subject sessions per subject. The validation partition is for assessing performance of an approach in a laboratory setting. The validation set contains images from 466 subjects collected in 4,007 subject sessions. Each subject session consists of four controlled still images, two uncontrolled still images, and one three-dimensional image. The validation partition contains from 1 to 22 subject sessions per subject. In our experiment (Section 3.4), 410 subjects from the validation set with more than 2 subject sessions are used. Hence, The data set used in our experiments consists of 820 FRGC controlled frontal face images with neutral expressions corresponding to 410 subjects, with two images per subject (two sessions).

In all the experiments, the original images were first normalized (in scale and orientation). Then, the facial regions are cropped to the size of 160 x 160. In all experiments, there is only one single image per person used as the model of the person. We quantitatively compare our method to several popular techniques for face recognition in the vision literature. Partitioned SRC [18] (with tuned block size 4 x 2) is one of the latest partially occluded face recognition algorithms and achieved higher recognition rate. LocPb (local probabilistic approach) [11] is a well-known method to recognize partially occluded faces and widely used as a benchmark algorithm in many partial matching methods. ICA I [1], LNMF [10] and PCA [17] are three popular methods used as benchmarks in recent face recognition approaches under occlusions [18]. LEM (Line Edge Maps) method [5] which is one of the best illumination insensitive methods based on facial edges with only one training face image per individual. AWPPZMA (Adaptively Weighted Patch Pseudo Zernike Moment Array) [8] is one of the best moment-based face recognition techniques to address occlusion and illumination when only one exemplar image per person is available.

3.2 Determination of λ

In this section, we examine the parameter (λ) involved in the propose method (see Eq.14). To determine λ, an experimental investigation on recognition accuracy was conducted under controlled condition with different values of λ on AR face database. The neutral faces under controlled/ideal condition taken in the first session were selected as the gallery set and the neutral faces under controlled/ ideal condition taken in the second session were used as the probe set. Fig. 3 shows the curve of recognition rate against the values of λ. The horizontal axis indicates the value of λ used and the vertical axis represents the rate of correct face recognition, which is the rate that the best returned face is from the correct class. The recognition rate increases greatly from $\lambda = 2$ to $\lambda = 8$. Between $\lambda = 8$ and $\lambda = 14$, the rate remains stable. Then it decreases with further increase of λ. In the rest of the experiments, λ is set as 10.

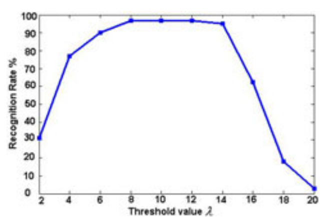

Fig. 3. The effect of λ on the recognition rate under controlled/idea condition

3.3 Face Recognition with Partial Occlusions

In this section, we test the performance of the proposed approach to cope with real partial occlusions using AR face database, which is the only database available that contains real images with disguise accessories. In the experiment, we

Fig. 4. Images of one subject in the AR database with different partial occlusions. (a) is a neutral facial image taken from the first session; (b-e) are images with partial occlusions taken from the first session and the second session, respectively.

Table 1. Performance comparison for sunglasses and scarf occluded faces

Methods	Session-1		Session-2	
	sunglasses	scarf	sunglasses	scarf
Stringface	88.0%	96.0%	76.0%	88.0%
SRC	86.0%	92.0%	64.0%	86.0%
LocPb	80.0%	82.0%	54.0%	48.0%
AWPPZMA	70.0%	72.0%	58.0%	60.0%
ICA I	54.0%	56.0%	38.0%	50.0%
LNMF	33.5%	24.0%	18.5%	9.6%

chose a subset of the data set consisting of 50 male subjects and 50 female subjects from AR face database. The neutral face images of the first session (see Fig. 4) were used as the galley set. Sunglasses and scarf occluded face images of the first and the second sessions (see Fig. 4 (b-e)) were used as the probes. The performance comparisons of the proposed approach with these benchmark methods are tabulated in Table 1, showing that the proposed approach archived the highest accuracies in both experiments.

3.4 Face Recognition with Random Block Occlusions

To further verify the performance of our method against various level of contiguous occlusions, we conducted a simulation experiment on the FRGC ver2.0 database . The data set used in our experiments consists of 820 FRGC controlled frontal face images with neutral expressions corresponding to 410 subjects, with two images per subject (two sessions). The data set is divided into gallery and probe sets. The gallery set consists of 410 images from 410 subjects. The rest of images are used as the probe set.

Occlusions are added to the probe images by using a black square of $s \times s$ with $s \in \{10, 20, \ldots, 100\}$ at a random location, as shown in Fig. 5. Note that

Fig. 5. Examples of FRGC ver2.0 face images with simulated occlusions. (a) images in the database; (b-k) the generated test images with random occluding blocks of sizes (10x10,20x20, ..., 100x100).

the $s \times s$ occlusion masks are randomly added to the images in probe sets. The graph in Fig. 6 shows the recognition rates of all four algorithms under varying degrees of occlusion. As can be seen, the Stringface method again outperformed the three benchmark methods for all levels of occlusion. Although there is only one sample image per person used as a template, SRC [18] still performed excellent as the proposed approach when the occlusion block size is small. The better performance of the Stringface approach against SRC becomes clear as the occlusion block size increases. Because of insufficient training samples, LNMF, ICA and PCA performed poorly in this single sample per class condition.

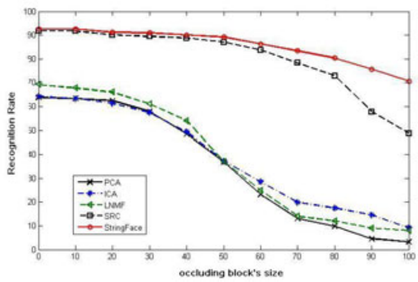

Fig. 6. Recognition under varying level of random occlusion (10x10, 20x20, ..., 100x100 of occluding blocks)

3.5 Preliminary Experiment under Varying Lighting and Expression Conditions

To evaluate the effects of different lighting conditions and facial expressions on the proposed approach, the preliminary experiment was designed using face images taken under different lighting conditions and facial expression from the AR database. In this experiment, we chose a subset of the data set consisting of 50 male subjects and 50 female subjects from AR database. The neutral face images taken in the first session were used as single models of the subjects. The face images under three different light conditions and facial expressions taken in the first session were used as probe images. The proposed approach is compared with the eigenface and LEM methods.

The experimental results on probe images with three lighting conditions and different facial expressions (smiling, angry and screaming) are illustrated in Table 2. In the three experiments under varying lighting conditions, the proposed Stringface method significantly outperformed the eigenface approach and also consistently performed better than the illumination-insensitive LEM approach[5]. The experimental results on faces with smile, anger and scream expressions show that the Stringface method achieved varying results compared to the LEM and eigenface methods.

Table 2. Preliminary results under varying lighting conditions and facial expressing changes

Conditions	Recognition rate(%)						
	Eigenface				LEM	AWPPZMA	Stringface
	k = 20	k = 60	k = 100	k = 100(w/o 1st 3)			
Left Light on	6.25%	9.82%	9.82%	26.79%	92.86%	74.36%	96.43%
Right Light on	4.46%	7.14%	7.14%	49.11%	91.07%	64.96%	95.53%
Both Light on	1.79%	2.68%	2.68%	64.29%	74.11%	42.74%	75.89%
Smiling	87.87%	94.64%	93.97%	82.04%	78.57%	96.58%	87.50%
Angry	78.57%	84.82%	87.50%	73.21%	92.86%	87.18%	87.50%
Screaming	34.82%	41.96%	45.54%	32.14%	31.25%	38.46%	25.89%

4 Conclusions

This paper proposes a novel Stringface approach for recognizing faces with partial occlusions from a single image per person. Stringface is a syntactic face representation, which integrates the local structural information with spatial information of a face image by grouping the relational organization of intermediate-level features (line segments) to a high-level global structure (a string). The proposed approach represents a face image as a string and enables it to define complex discontinuous features in a human frontal face. The matching of two frontal faces is achieved by matching two Stringfaces through a *string*-to-*string* matching, which was believed a technique not suitable for frontal face recognition due to its highly ordered global representation and complex nature of a human face. The performance of the proposed approach has been evaluated and compared with several state-of-the-art approaches. Experimental results demonstrated the feasibility and effectiveness of a high-level syntactic method in face recognition, showing a new way for face representation and recognition.

References

1. Bartlett, M., Movellan, J., Sejnowski, T.: Face recognition by independent component analysis. IEEE Transactions on Neural Networks 13(6), 1450–1464 (2002)
2. Biederman, I., Ju, G.: Surface versus edge-based determinants of visual recognition. Cognitive Psychology 20, 38–64 (1988)
3. Bruce, V., Hanna, E., Dench, N., Healey, P., Burton, M.: The importance of mass in line drawings of faces. Applied Congnitive Psychology 6(7), 619–628 (1992)
4. Chen, S.W., Tung, S.T., Fang, C.Y., Cheng, S., Jain, A.K.: Extended attributed string matching for shape recognition. Computer Vision and Image Understanding 70(1), 36–50 (1998)
5. Gao, Y., Leung, M.K.: Face recognition using line edge map. IEEE Transactions on Pattern Analysis and Machine Intelligence 24(6), 764–779 (2002)
6. Gao, Y., Leung, M.K.: Human face profile recognition using attributed string. Pattern Recognition 35, 353–360 (2002)
7. Gao, Y., Qi, Y.: Robust visual similarity retrieval in single model face databases. Pattern Recognition 38(7), 1009–1020 (2005)

8. Kanan, H.R., Faez, K., Gao, Y.: Face recognition using adaptively weighted patch pzm array from a single exemplar image per person. Pattern Recognition 41(12), 3799–3812 (2008)
9. Leung, M., Yang, Y.: Dynamic two-strip algorithm in curve fitting. Pattern Recognition 23(1-2), 69–79 (1990)
10. Li, S., Hou, X.W., Zhang, H.J., Cheng, Q.S.: Learning spatially localized, parts-based representation. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 1, pp. I-207–I-212 (2001)
11. Martínez, A.: Recognizing imprecisely localized, partially occluded, and expression variant faces from a single sample per class. IEEE Transactions on Pattern Analysis and Machine Intelligence 24(6), 748–763 (2002)
12. Martínez, A., Benavente, R.: The AR face database. CVC Technical Report 24 (1998)
13. Needleman, S.B., Wunsch, C.D.: A general method applicable to the search for similarities in the amino acid sequence of two proteins. J. Mol. Biol. 48(3), 443–453 (1970)
14. Penev, P.S., Atick, J.J.: Local feature analysis: A general statistical theory for object representation. Network: Computation in Neural Systems 7, 477–500 (1996)
15. Petrakis, E.G., Diplaros, A., Milios, E.: Matching and retrieval of distorted and occluded shapes using dynamic programming. IEEE Transactions on Pattern Analysis and Machine Intelligence 24(11), 1501–1516 (2002)
16. Phillips, P., Flynn, P., Scruggs, T., Bowyer, K., Chang, J., Hoffman, K., Marques, J., Min, J., Worek, W.: Overview of the face recognition grand challenge. In: Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 1, pp. 947–954 (2005)
17. Turk, M., Pentland, A.: Eigenfaces for recognition. Journal of Cognitive Neuroscience 3(1), 71–86 (1991)
18. Wright, J., Yang, A.Y., Ganesh, A., Sastry, S.S., Ma, Y.: Robust face recognition via sparse representation. Pattern Analysis and Machine Intelligence, IEEE Transactions on 31(2), 210–227 (2009)