Face Recognition using Ensemble String Matching

Author

Chen, Weiping, Gao, Yongsheng

Griffith Research Online

https://research-repository.griffith.edu.au

Face Recognition using Ensemble String Matching

Weiping CHEN and Yongsheng GAO, *Senior Member, IEEE*

Abstract— In this paper, we present a syntactic string matching approach to solve the frontal face recognition problem. String matching is a powerful partial matching technique, but is not suitable for frontal face recognition due to its requirement of globally sequential representation and the complex nature of human faces, containing discontinuous and non-sequential features. Here we build a compact syntactic Stringface representation, which is an ensemble of strings. A novel Ensemble String Matching approach that can perform non-sequential string matching between two Stringfaces is proposed. It is invariant to the sequential order of strings and the direction of each string. The embedded partial matching mechanism enables our method to automatically utilize every piece of non-occluded region, regardless of shape, in the recognition process. The encouraging results demonstrate the feasibility and effectiveness of using syntactic methods for face recognition from a single exemplar image per person, breaking the barrier that prevents string matching techniques from being utilized for addressing complex image recognition problems. The proposed method not only achieved significantly better performance in recognizing partially occluded faces, but also showed its ability to perform direct matching between sketch faces and photo faces.

Index Terms—Face recognition, ensemble string matching, Stringface, partial matching, occlusion, sketch recognition, syntactic.

I. INTRODUCTION

RECOGNIZING human faces with appearance changes from one exemplar photo per person is an important and challenging problem, both in theory and for practical applications. A recent question to computer vision researchers is whether we can recognize partially occluded faces in an unconstrained environment when only one example view per person is available, for example, if only a passport or identity card photograph [54] is available for each person in the database. It is clear that the locations, sizes and shapes of occlusions are not predictable in real applications and thus their effect cannot be removed by employing predefined regions. If, however, we can introduce a mechanism during the similarity calculation process to intelligently determine the occluded areas and exclude them from the similarity measurement, it seems possible to solve this problem.

Although there are studies [8],[49]-[52],[56],[57] that have attempted to solve the problem of recognising partially occluded faces, a number of important theoretical and practical issues in this task remain unsolved. For example, how to better determine the locations and shapes of the occluded regions and how to effectively exclude them from the similarity measurement are still unclear.

In this study, we attempt to develop a new strategy for face recognition that (1) can recognize faces with partial occlusions of arbitrary shapes and locations, (2) is suitable for single model based recognition, (3) works as a general face recognition method, and (4) matches sketch faces and photo faces directly without needing to know which image is a sketch.

We propose a novel Stringface representation and matching concept for face recognition with one exemplar image per person. The Stringface is a high-level syntactic representation of a face image, which can be constructed from a single face image, without any training. In our implementation, the low-level edge pixels used in early vision are considered as "letters" to be grouped into intermediate-level straight lines with additional local structural information. These straight lines are considered as "words" in a language. A string is a "sentence", which is a chain of straight lines, connected together, representing an edge curve on the face. A Stringface is a "body text" composed of many strings, which holds a syntactic description about the identity of the face.

A novel Ensemble String Matching scheme (ESM) is designed, which can handle the uncertainty problems regarding the direction of each string and the order of different strings when matching two Stringfaces. Different from conventional string matching techniques that provide a single matching score, the similarity between two Stringfaces is the aggregation of an ensemble of similarities between all matched string pairs. This is believed to be the first piece of work that uses a syntactic string representation and matching concept to address the frontal face recognition problem. Some of the ideas presented in this paper were initially reported in [53]. In this paper, we report the complete and updated formulation, along with an extensive experimental evaluation of our technique.

The rest of the paper is organized as follows: Section 2 gives a brief review of the related primitive-based recognition techniques and string matching methods. The Stringface representation and matching concepts are proposed in Section 3, with a detailed description of a novel Ensemble String Matching approach. In Section 4, we report the extensive experiments that have been conducted to validate the feasibility and effectiveness of the proposed method. Finally, the paper is concluded in Section 5.

II. BACKGROUND

In practical applications, faces often have to be recognized from a single exemplar image per person, for example, when just a passport or identity card photograph is available as the sample of a person. This presents a further challenge to the face recognition problem, due to the additional restrictions on

the training data that can be used and the recognition technique itself.

Primitive-based methods have been proven to be effective when dealing with the single sample problem [13]. According to the types of primitives, these approaches can be classified into three groups: low level, intermediate level, and high level primitive-based approaches. In low level primitive-based approaches, a face is represented by a group of identity related points whose spatial information (i.e., the locations of these low level primitives) is used for recognition [32], [55]. Low level primitive-based approaches rely only on the spatial information of an image, but lack the capacity to represent the local structure of a face. In order to deal with this problem, primitives with additional structural information are introduced. In [24], a face is described using a set of Directional Corner Points (DCPs) with three attributes, which integrate spatial features and additional structural information about the connectivity to their neighbors. The structural attributes of DCPs enhance the discriminative power of the descriptor, allowing it to achieve not only a noticeable accuracy increase over its low level edge map counterpart, but a lower computational cost and memory requirement, through the use of sparse points. Line Edge Map (LEM) [10] is another intermediate level primitive-based face representation and recognition approach, whose primitives are line segments grouped from the pixels of an edge map. The additional local structural attributes of orientation and length provide both better discriminative power, and greater robustness to noise than low level primitives.

In theory, a high level primitive-based approach that further considers the interrelationship between intermediate level primitives can achieve a higher level of performance by providing a meaningful complete interpretation of the image. An analogy in understanding a language can be drawn to illustrate the concept. The low level primitives (e.g., pixels) and structural primitives (e.g., straight lines) can be viewed as the letters and words of a language, while high level primitives (e.g., curves) are regarded as sentences of the language that are generated from the words according to a grammar. Hence, a high level primitive-based approach is sometimes called a syntactic approach.

String matching is a syntactic and structural method for similarity measurement between strings or vectors, and has been widely used for pattern matching in molecular biology [25], speech recognition [26], and file comparison [27],[28]. Strings can be classified into two categories: symbolic strings and attributed strings. The goal of string matching algorithms is to find a sequence of elementary edit operations that transform one string into another at a minimal cost. The elementary operations are often deletion, insertion, and substitution of string symbols. The most well-known application of symbolic string matching is the spelling checker algorithms built into word processing applications such as Microsoft Word. The use of symbols was found to be inadequate for achieving complex shape recognition [29] because symbols are discrete in nature while most problems of shape recognition deal with attributes that are basically continuous. To overcome this, attributed string matching [29]-[31] was proposed for measuring the similarity of shapes by tracing the contour of a shape using a series of primitives, then converting them into a string. One of the prominent advantages of string matching is its ability to perform partial matching without requiring prior knowledge about the location and size of the missing data in the string. Furthermore, it is also very robust (often invariant) to spurious outliers and noise. This is particularly appealing when recognizing complex objects such as human faces because in real applications, occlusions can occur at any location, and with arbitrary shapes and sizes. However, with the current state of technology, string matching cannot be used for recognizing natural images containing complex discontinuous features, as it only works on matching a single 1D sequentially ordered pattern (which only can represent a single continuous contour/silhouette of the shape) to another one. To our knowledge, there is no frontal face recognition using string matching techniques. The most related work is [30] for human face profile recognition. It encodes a continuous face profile silhouette into a chain of connected profile line segments to perform string-to-string matching, which outperformed the point-to-point [32] and line-to-line [10] matching methods. However, it can only represent the continuous silhouette of a profile face, and other important but unconnected distinctive features such as the eyes, eyebrows, mouth, and ears, cannot be used. Also, it performs a global alignment on two sequences of lines and fails to work when a face profile has occlusions.

III. ENSEMBLE STRING MATCHING OF STRINGFACES

In this study, we propose a novel high-level Stringface representation that describes the complex discontinuous facial features in a human face as an ensemble of attributed strings by integrating the structural connectivity information of a face image. A new Ensemble String Matching technique, which is invariant to the order of strings and is able to perform forward-backward string matching, is developed to address the discontinuity and order uncertainty problems in Stringface.

A. Stringface

The primitive string of a visual object is an abstract and high-level representation. However, conventional string matching techniques are limited to continuous (or connected) primitive representation and matching. For example, string matching [30] was used to measure the similarity between two continuous silhouettes of profile faces by ignoring other important but unconnected distinctive features, such as eyes, eyebrows, mouth, and ears. A human frontal face contains complex discontinuous features, and it was often assumed that string matching techniques were not suitable for frontal face recognition.

In this study, we define a syntactic representation of a human face, Stringface (SF), which is an ensemble of strings where each string describes a facial component. Strings in SF are separated by null primitives ϕ as

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

where n is the number of strings in SF. Each string S_p, $p = 1, \ldots, n$ is a sequence of connected primitives as

$$S_p = L_q L_{q+1} \cdots L_{q+m_p}, \tag{2}$$

where L_q is the *qth* primitive in SF and (m_p+1) is the number of primitives in S_p.

Cognitive psychological studies [33],[34] indicated that humans recognize line drawings as quickly and almost as accurately as gray-level images since the line drawings preserve most of the important feature information. In this study, line segments, which are obtained by applying polygonal line fitting [35] to the edge map of a face image, are employed as building units (i.e., "word" primitives) of a Stringface. Each primitive, $L(l, \theta, x, y)$, is associated with four attributes l, θ, x, and y, representing the length, orientation, and (x,y) midpoint location of the line, respectively. The line orientation θ is defined as the angle with respect to the horizontal. The Stringface represents not only the local structural information (by attributes of primitives) but also the global structure (by strings of primitives) of a face. Fig. 1 gives a visual illustration of an example Stringface $SF = S_1 \phi S_2 \phi \cdots \phi S_{26} \phi \cdots \phi S_{29} \phi \cdots \phi S_n$, where $S_{29} = L_{134} L_{135} L_{136} L_{137} L_{138} L_{139}$.

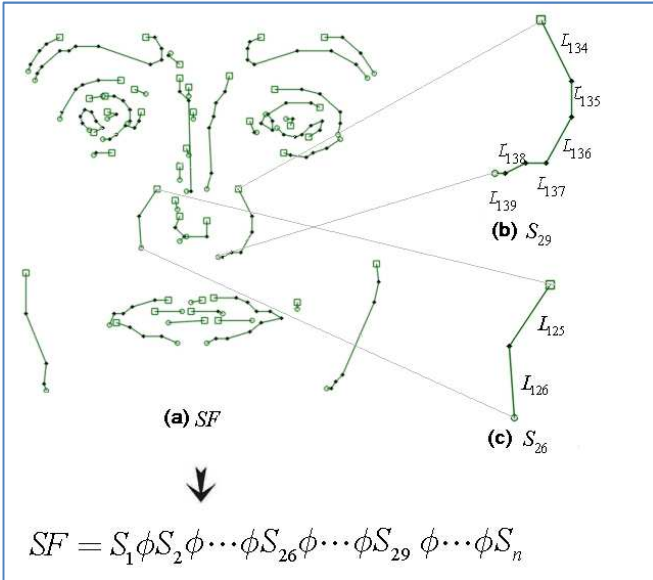$$SF = S_1 \phi S_2 \phi \cdots \phi S_{26} \phi \cdots \phi S_{29} \phi \cdots \phi S_n$$

Fig. 1.   Visual illustration of an example Stringface using line segments as primitives.

B. Ensemble String Matching

Let $SF_A = S_1^A \cdots S_{n_1}^A = L_1^A \cdots L_{N_1}^A$ and $SF_B = S_1^B \cdots S_{n_2}^B = L_1^B \cdots L_{N_2}^B$ be two Stringfaces representing the test face and the model face, respectively. n_1 and n_2 are the numbers of strings in SF_A and SF_B. N_1 and N_2 are the numbers of primitives including null primitives in SF_A and SF_B.

i. Cost functions

Attributed string matching (one string matched against another string) has been used in contour [29] and curve [30] matching. Their cost functions have proven to be effective for handling segmentation errors and the inconsistency problem in corner detection. In this study, we use the same cost functions for change and merge operations as in [30]. Here, we briefly summarise the two cost functions for the convenience of reading this paper. Readers can refer to [30] for more details.

Let $A^k < i >$ and $B^l < j >$ be two primitives merged from string segments $A\langle i-k+1:i\rangle = L_{i-k+1}^A L_{i-k+2}^A \cdots L_i^A$ (with k primitives) and $B\langle j-l+1:j\rangle = L_{j-l+1}^B L_{j-l+2}^B \cdots L_j^B$ (with l primitives) in Stringfaces SF_A and SF_B respectively. The cost function of a change operation from $A^k < i >$ to $B^l < j >$ is defined as

$$C(A^k < i > \rightarrow B^l < j >) = |l^k - l^l| + f(\Delta(\theta^k, \theta^l)) + \left\| (x,y)^k, (x,y)^l \right\| . \tag{3}$$

It measures the length differences ($|l^k - l^l|$), the intersecting angle of orientations ($\Delta(\theta^k, \theta^l)$), and the distance between the midpoints ($\left\| (x,y)^k, (x,y)^l \right\|$) of $A^k < i >$ and $B^l < j >$. When $k=1$ and $l=1$, no merge is performed and the above change operation reduces to the conventional one-to-one change operation $A < i > \rightarrow B < j >$.

The cost of merging a sequence of k primitives $A\langle i-k+1:i\rangle$ into one merged primitive $A^k < i >$ is defined as

$$C(A\langle i-k+1:i\rangle \rightarrow A^k < i >) = f(\frac{k-1}{l^k} \sum_{q=i-k+1}^{i} \Delta(\theta^k, \theta_q) \times l_q), \tag{4}$$

where l^k and θ^k are the length and the orientation of the merged primitive $A^k < i >$, l_q and θ_q are the length and the orientation of primitive L_q in $A\langle i-k+1:i\rangle$ before merging. Every $\Delta(\theta^k, \theta_q)$ is weighted by its normalized length l_q / l^k by assuming the contribution of angle difference of a primitive is proportional to its length. The number of merge operations, k-1, is also factored into the merge cost.

The insert and delete operations used in [29] and [30] are not needed due to the embedded partial matching mechanism in the proposed ESM (see Section *iii*).

ii. Bounding merge operation by null primitive

As a Stringface is composed of multiple strings delimited by null primitives, the merge operation has to be restricted in the same string, that is, the primitives in string S_p cannot be merged with primitives in its neighboring strings S_{p-1} and S_{p+1}. For a primitive $L_q \in S_p$, the merge operation has to be stopped by the first ϕ primitive it meets. Hence the maximum number of primitives that can be merged with L_q is bounded by

$$m\_bound_q = q - \sum_{t=1}^{p-1} |S_t| - (p-1), \tag{5}$$

where $|S_t|$ is the number of primitives in the tth string.

iii. S-M matrixes

The similarity matching between the two Stringfaces can be

characterized by the edit operation costs using dynamic programming, which generates an *S-M* matrix pair, i.e., a similarity matrix $\mathbf{S}$ and a merge step matrix $\mathbf{M}$. The *S-M* matrix pair has a size of $(N_1+1)\times(N_2+1)$ as follows.

$$\mathbf{S}=\begin{pmatrix} s(0,0) & s(0,1) & \dots & s(0,N_2) \\ s(1,0) & s(1,1) & \dots & s(1,N_2) \\ \vdots & \vdots & \vdots & \vdots \\ s(N_1,0) & s(N_1,1) & \dots & s(N_1,N_2) \end{pmatrix}, \quad (6)$$

where each element $s(i,j)$ records a similarity value between the primitives in SF_A and SF_B along the path determined by its corresponding element $m(i,j)$ in $\mathbf{M}$.

$$\mathbf{M}=\begin{pmatrix} m(0,0) & m(0,1) & \dots & m(0,N_2) \\ m(1,0) & m(1,1) & \dots & m(1,N_2) \\ \vdots & \vdots & \vdots & \vdots \\ m(N_1,0) & m(N_1,1) & \dots & m(N_1,N_2) \end{pmatrix}, \quad (7)$$

where each element $m(i,j)$ contains two merge step values k and l (see Eq. (10)).

The element $s(i,j)$ stores the maximum accumulated similarity between two segments (a segment is a portion of a string) of Stringfaces starting from L_1^A and L_1^B, and ending at L_i^A and L_j^B respectively, as defined in Eq. (8).

$$s(i,j)=\begin{cases} \max_{k,l}(T(k,l)) & \text{if } L_i^A\neq\phi \text{ and } L_j^B\neq\phi \\ 0 & \text{otherwise} \end{cases}, \quad (8)$$

where $T(k,l)$ is the accumulated similarity resulting from edit operations as defined below.

$$T(k,l)=\begin{cases} s(i-k,j-l)+S & \text{if } S\geq 0 \\ 0 & \text{otherwise} \end{cases} \quad (9)$$

The merge steps that produce the maximum $T(k,l)$ are recorded in $m(i,j)$ as

$$m(i,j)=\arg\max_{k,l}(T(k,l)). \quad (10)$$

If either L_i^A or L_j^B is a null primitive, $s(i,j)$ is reset to zero (see Eq. (8)). This is to ensure that $T(k,l)$ will not increase its value by edit operations on primitives beyond the current string.

To seamlessly include the partial matching mechanism in the design of ESM, the similarity parameter S in Eq. (9) is defined as

$$\begin{aligned} S=\lambda-[&C(A\langle i-k+1:i\rangle\rightarrow A^k<i>) \\ &+C(B\langle j-l+1:j\rangle\rightarrow B^l<j>), \\ &+C(A^k<i>\rightarrow B^l<j>)] \end{aligned} \quad (11)$$

where λ is a threshold to decide the strength of partial matching. If the combined cost of merging and changing primitives is greater than λ (which gives $S<0$), these primitives are considered as outliers and discarded from ESM. This is achieved by setting $T(k,l)=0$ when $S<0$ (see Eq.(9)). The detailed algorithm of creating the *S-M* matrix pair is given in Algorithm 1.

Inputs: $SF_A=L_1^A L_2^A\cdots L_{N_1}^A$, $SF_B=L_1^B L_2^B\cdots L_{N_2}^B$
Outputs: $\mathbf{S}(SF_A,SF_B)$, $\mathbf{M}(SF_A,SF_B)$
$s(0,0)=0$; $s(i,0)=0$; $s(0,j)=0$;
for $i=1$ to N_1, $j=1$ to N_2
  if $L_i^A==\phi$ or $L_j^B==\phi$
    $s(i,j)=0$;
  else
    for $k=1$ to $m\_bound_i^A$, $l=1$ to $m\_bound_j^B$
      $\begin{aligned}S=\lambda-[&C(A\langle i-k+1:i\rangle\rightarrow A^k<i>) \\ &+C(B\langle j-l+1:j\rangle\rightarrow B^l<j>) \\ &+C(A^k<i>\rightarrow B^l<j>)]\end{aligned}$;
      $T(k,l)=\begin{cases} s(i-k,j-l)+S & \text{if } S\geq 0 \\ 0 & \text{otherwise}\end{cases}$;
    end
    $s(i,j)=\max_{k,l}(T(k,l))$;
    $m(i,j)=\arg\max_{k,l}(T(k,l))$;
  end
end

Algorithm 1. The proposed **S-M** matrix pair generation in ESM.

iv. Similarity measurement

The similarity of associating a group of segments from Stringface SF_A with a group of segments from Stringface SF_B is computed as:

$$S(SF_A,SF_B)=\xi\times\left(\frac{1}{\sum_{i=1}^{f}\tau_i\lambda}\times\sum_{i=1}^{f}Score_i\right) \quad (12)$$

where $Score_i$ is the similarity score of the *ith* matched string segments between SF_A and SF_B, and f is the number of matched string segments (There may be more than one matched segment, or none, in a single string). λ is the thresholding value to decide the strength of a partial matching (see Eq. (11)). τ_i is the number of change operations for corresponding the *ith* matched segments in the two Stringfaces. ξ is a weight term taking the importance of matching a larger percentage of both Stringfaces in the similarity measurement design, according to the observation that humans consider the percentage of similar content between two objects when judging the quality of matching.

$$\xi=\frac{1}{2}\left(\frac{length\ of\ matched\ SF_A}{length\ of\ SF_A}+\frac{length\ of\ matched\ SF_B}{length\ of\ SF_B}\right) \quad (13)$$

The pairs of matched segments are found by first locating the maximal element in $\mathbf{S}$. Other matrix elements leading to this maximal value are then sequentially determined by $\mathbf{M}$ with a traceback procedure ending with an element of $\mathbf{S}$ equalling zero. The maximum value is recorded as the similarity score of this matched segment pair. Before searching for the next matched segment pair, all the elements in $\mathbf{S}$ related to the pervious matched primitives are set to zero to avoid matching a primitive in one Stringface to multiple

primitives in the other Stringface. The above traceback procedure is repeated until all the elements in $\mathbf{S}$ are 0 to find all $Score_i$ for Eq. (12). (see Algorithm 2).

v. Analysis and example

Different from a conventional string that guarantees the order of primitives is maintained in each instance, Stringface is actually an ensemble of strings in which both the order of these strings, and the start and end of each string are interchangeable. For example, an object X in three different pictures is encoded into SFs $SF_A = S_1^A \phi S_2^A = L_1 L_2 L_3 L_4 L_5 L_6 L_7 \phi L_9 L_{10} L_{11} (L_8 = \phi)$, $SF_B = S_2^B \phi S_1^B = L_9 L_{10} L_{11} \phi L_1 L_2 L_3 L_4 L_5 L_6 L_7$, $SF_C = \overleftarrow{S_1^C} \phi S_2^C = L_7 L_6 L_5 L_4 L_3 L_2 L_1 \phi L_9 L_{10} L_{11}$. Note that there is an order change of strings between SF_A and SF_B, and a direction change between the 1st string of SF_A and the 1st string of SF_C. The proposed ensemble string matching gives similarity values of $S(SF_B, SF_A) = 1$ ($Score_1 = 7\lambda$, $Score_2 = 3\lambda$, $\tau_1 = 7$, $\tau_2 = 3$, $f = 2$, $\xi = 1$) and $S(SF_C, SF_A) = 1$ ($Score_1 = 3\lambda$, $Score_i = \lambda$ ($i = 2,...,8$), $\tau_1 = 3$, $\tau_i = 1$ ($i = 2,...,8$), $f = 8$, $\xi = 1$), equalling the 100% self-match of $S(SF_A, SF_A) = 1$ ($Score_1 = 7\lambda$, $Score_2 = 3\lambda$, $\tau_1 = 7$, $\tau_2 = 3$, $f = 2$, $\xi = 1$). This provides the required string location and direction mutation invariances.

Inputs: $\mathbf{S}(SF_A, SF_B)$, $\mathbf{M}(SF_A, SF_B)$

Outputs: $Score(i)$, $i = 1, \cdots, f$

$i = 0$;

do

    // Find the maximum element in $\mathbf{S}$ and its location.

    $\max\_S = \max(\mathbf{S})$; $(r, c) = $ location of $\max(\mathbf{S})$;

    // Initialize Start & End of matched segment.

    $(rStart, cStart) = (r, c)$; $(rEnd, cEnd) = (r, c)$;

    // Trace back using k, l in $m(r, c)$.

    $r1 = rStart - k$; $c1 = cStart - l$;

    $t = 1$;

    while $s(r1, c1) > 0$

        $(rStart, cStart) = (r1, c1)$;

        $r1 = rStart - k$; $c1 = cStart - l$;

        $t = t + 1$;

    end

    if $s(rStart, cStart) \leq \lambda$

        $i = i + 1$; $Score(i) = \max\_S$; $\tau(i) = t - 1$;

    end

    // reset rows between rStart and rEnd in $\mathbf{S}$ to 0.

    $s(rStart : rEnd, 0 : N_1) = 0$;

    // reset columns between cStart and cEnd in $\mathbf{S}$ to 0.

    $s(0 : N_2, cStart : cEnd) = 0$;

while $\max(\mathbf{S}) > 0$

Algorithm 2. The proposed **S-M** matrix indexed searching of similarity scores in ESM.
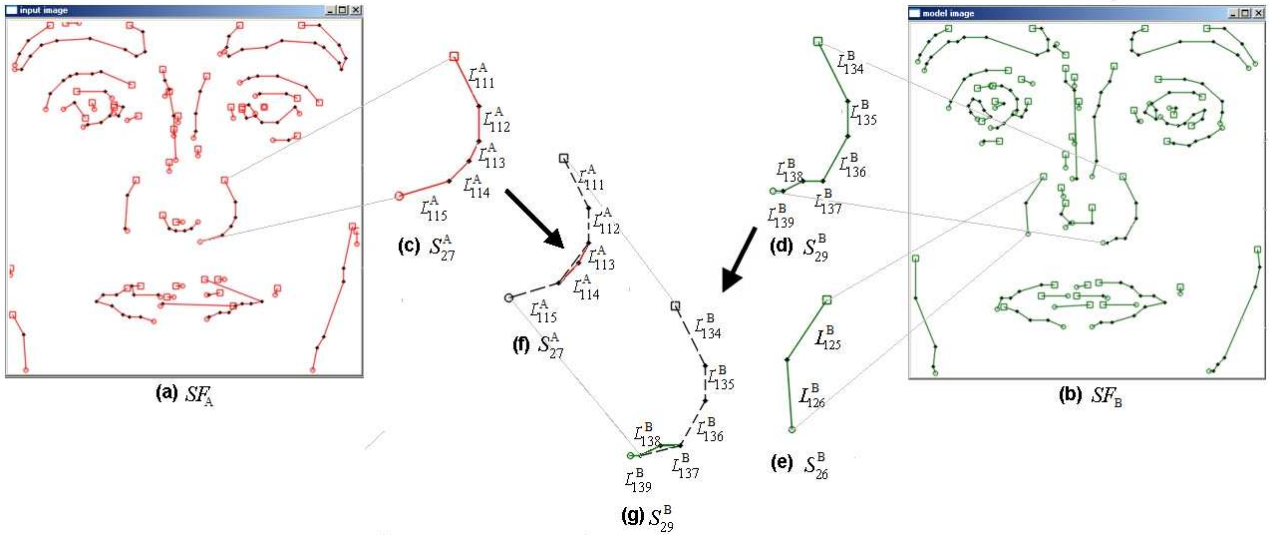
$$SF_A = S_1^A \cdots S_{26}^A L_{110}^A L_{111}^A L_{112}^A L_{113}^A L_{114}^A L_{115}^A L_{116}^A S_{28}^A \cdots S_{n_1}^A \qquad SF_B = S_1^B \cdots S_{25}^B L_{124}^B L_{125}^B L_{126}^B L_{127}^B \cdots S_{28}^B L_{133}^B L_{134}^B L_{135}^B L_{136}^B L_{137}^B L_{138}^B L_{139}^B L_{140}^B S_{30}^B \cdots S_{n_2}^B$$

Fig. 2. An example of the ensemble string matching between two Stringfaces. The squares and the circles represent the starts and the ends of strings, which decide the sequence directions of the strings.
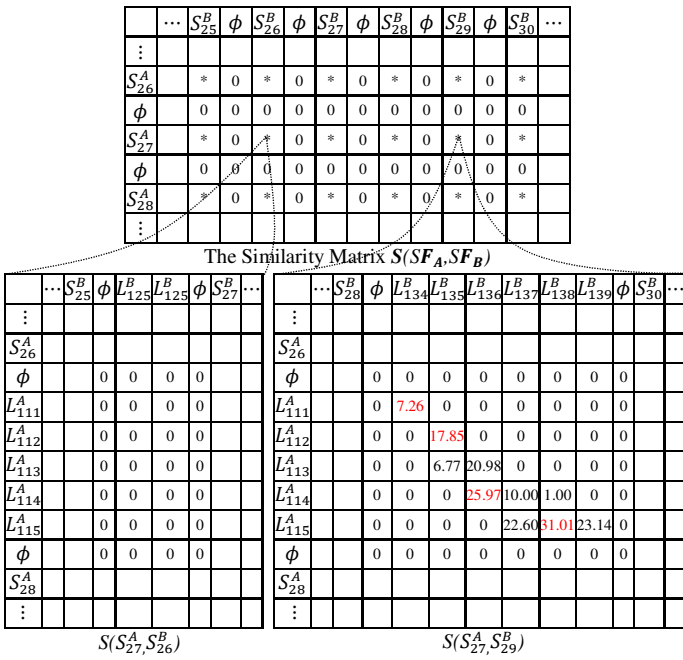
	...	S^B_{25}	ϕ	S^B_{26}	ϕ	S^B_{27}	ϕ	S^B_{28}	ϕ	S^B_{29}	ϕ	S^B_{30}	...
$\vdots$													
S^A_{26}		*	0	*	0	*	0	*	0	*	0	*	
ϕ		0	0	0	0	0	0	0	0	0	0	0	
S^A_{27}		*	0	*	0	*	0	*	0	*	0	*	
ϕ		0	0	0	0	0	0	0	0	0	0	0	
S^A_{28}		*	0	*	0	*	0	*	0	*	0	*	
$\vdots$													

The Similarity Matrix $S(SF_A, SF_B)$

$S(S^A_{27}, S^B_{26})$

	...	S^B_{25}	ϕ	L^B_{125}	L^B_{125}	ϕ	S^B_{27}	...
$\vdots$								
S^A_{26}								
ϕ			0	0	0	0		
L^A_{111}			0	0	0	0		
L^A_{112}			0	0	0	0		
L^A_{113}			0	0	0	0		
L^A_{114}			0	0	0	0		
L^A_{115}			0	0	0	0		
ϕ			0	0	0	0		
S^A_{28}								
$\vdots$								

$S(S^A_{27}, S^B_{29})$

	...	S^B_{28}	ϕ	L^B_{134}	L^B_{135}	L^B_{136}	L^B_{137}	L^B_{138}	L^B_{139}	ϕ	S^B_{30}	...
$\vdots$												
S^A_{26}												
ϕ			0	0	0	0	0	0	0	0		
L^A_{111}			0	7.26	0	0	0	0	0	0		
L^A_{112}			0	0	17.85	0	0	0	0	0		
L^A_{113}			0	0	6.77	20.98	0	0	0	0		
L^A_{114}			0	0	0	25.97	10.00	1.00	0	0		
L^A_{115}			0	0	0	0	22.60	31.01	23.14	0		
ϕ			0	0	0	0	0	0	0	0		
S^A_{28}												
$\vdots$												

Fig. 3.   Matrix **S** and traceback result.

	...	S^B_{25}	ϕ	S^B_{26}	ϕ	S^B_{27}	ϕ	S^B_{28}	ϕ	S^B_{29}	ϕ	S^B_{30}	...
$\vdots$													
S^A_{26}		*	0	*	0	*	0	*	0	*	0	*	
ϕ		0	0	0	0	0	0	0	0	0	0	0	
S^A_{27}		*	0	*	0	*	0	*	0	*	0	*	
ϕ		0	0	0	0	0	0	0	0	0	0	0	
S^A_{28}		*	0	*	0	*	0	*	0	*	0	*	
$\vdots$													

The Merge Step Matrix $\mathbf{M}(SF_A, SF_B)$

$M(S^A_{27}, S^B_{26})$

	...	S^B_{25}	ϕ	L^B_{125}	L^B_{125}	ϕ	S^B_{27}	...
$\vdots$								
S^A_{26}								
ϕ			×	×	×	×		
L^A_{111}			×	(1,1)	(1,1)	×		
L^A_{112}			×	(1,1)	(1,1)	×		
L^A_{113}			×	(1,1)	(2,1)	×		
L^A_{114}			×	(2,1)	(2,2)	×		
L^A_{115}			×	(2,1)	(2,2)	×		
ϕ			×	×	×	×		
S^A_{28}								
$\vdots$								

$M(S^A_{27}, S^B_{29})$

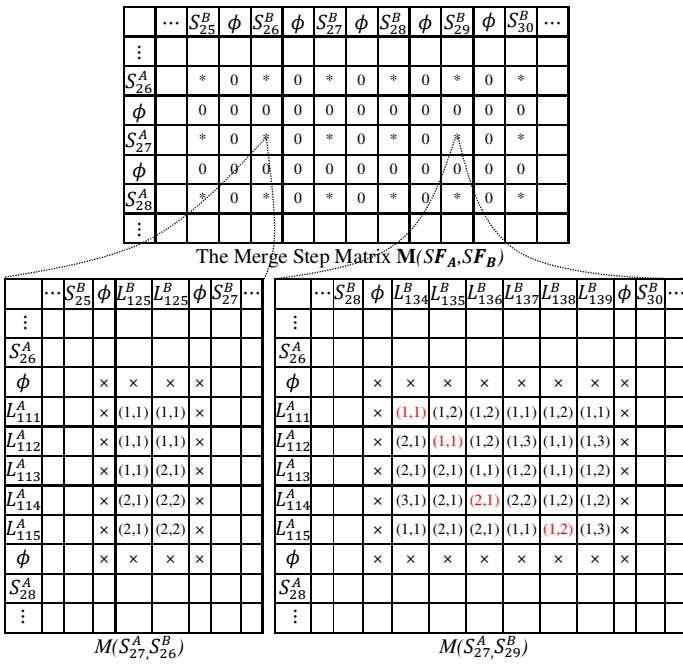|	...	S^B_{28}	ϕ	L^B_{134}	L^B_{135}	L^B_{136}	L^B_{137}	L^B_{138}	L^B_{139}	ϕ	S^B_{30}	...
$\vdots$												
S^A_{26}												
ϕ			×	×	×	×	×	×	×	×		
L^A_{111}			×	(1,1)	(1,2)	(1,2)	(1,1)	(1,2)	(1,1)	×		
L^A_{112}			×	(2,1)	(1,1)	(1,2)	(1,3)	(1,1)	(1,3)	×		
L^A_{113}			×	(2,1)	(2,1)	(1,1)	(1,2)	(1,1)	(1,2)	×		
L^A_{114}			×	(3,1)	(2,1)	(2,1)	(2,2)	(1,2)	(1,2)	×		
L^A_{115}			×	(1,1)	(2,1)	(2,1)	(1,1)	(1,2)	(1,3)	×		
ϕ			×	×	×	×	×	×	×	×		
S^A_{28}												
$\vdots$												

Fig. 4.   Matrix **M** and traceback result.

We now show a real example of the proposed Stringface matching approach. Fig. 2 (a) and (b) visualize Stringfaces SF_A and SF_B of two different face images from the same person in the AR face database. Fig. 2 (c) displays the 27th string of SF_A with 5 primitives ($L^A_{111} L^A_{112} L^A_{113} L^A_{114} L^A_{115}$). And Fig. 2 (d) and (e) are the 29th and the 26th strings of SF_B with 6 and 2 primitives, respectively. The **S-M** matrix pair calculated by Algorithm 1 is shown in Fig. 3 and Fig. 4. First, we locate the maximal element of matrix **S** which is $s(L^A_{115}, L^B_{138}) = 31.01$ in Fig. 3. The merge step of this entry is $m(L^A_{115}, L^B_{138}) = (1,2)$

(see Fig. 4), which means that L^A_{115} is not merged with any other primitives and primitives L^B_{138} and L^B_{137} are merged into a new primitive in the matching process. Then we traceback by one primitive in SF_A and two primitives in SF_B (see Algorithm 2) to entry (L^A_{114}, L^B_{136}), with similarity value $s(L^A_{114}, L^B_{136}) = 25.97$ and merge step $m(L^A_{114}, L^B_{136}) = (2,1)$. A whole path (marked in red in Fig. 3 and Fig. 4) can be traced by repeating the above process. The merged primitives in strings S^A_{27} and S^B_{29} during the matching process are shown in Fig. 2 (f) and (g), respectively. The similarity score between S^A_{27} and S^B_{29} is 31.01. Note that the similarity score between S^A_{27} and S^B_{26} is 0.

IV. EXPERIMENTAL RESULTS

Three publicly available databases (the AR face database [37], the FRGC ver2.0 face database [38], and the CUHK face sketch database [2]) were used to evaluate the effectiveness of the proposed approach. The AR database consists of over 4,000 frontal view images of 126 persons (70 male and 56 female). Each person is represented by 26 images captured in two different sessions with a two-week time interval. Each session contains 13 face images under different lighting conditions (right light on, left light on and both lights on), different facial expressions (smiling, angry, and screaming), and partial occlusions (sunglasses and scarf). Some images were found missing or corrupted for a few persons. The photos of 117 persons having complete sets of images (64 male and 53 female) covering all conditions were used in our experiments. Fig. 5 shows an example set of images from one person in the AR database used for our experiments.
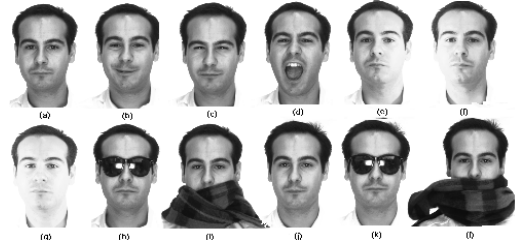

Fig. 5.   A sample set of AR face images from one person. (a) Neutral face under controlled/ideal conditions taken in the first session; (b-d) faces with different facial expressions taken in the first session; (e-g) faces under different lighting conditions taken in the first session; (h-i) faces with different partial occlusions taken in the first session; (j) neutral face under controlled/ideal conditions taken in the second session; (k-l) faces with different partial occlusions taken in the second session.

The FRGC ver2.0 database contains 50,000 recordings divided into training and validation sets. The training set consists of 12,776 images from 222 persons, which were taken in 9-16 sessions. The validation set contains images from 466 persons collected in 4,007 sessions. The number of sessions used for taking photos of each person is different, ranging from 1 to 22 sessions (410 persons were photographed in 2 or more sessions). In each session were recorded four controlled still images, two uncontrolled still images, and one 3D scan. In our experiments, all the 466 persons in the validation set were used for testing. Different from the AR database, there are multiple neutral expression images for each person in the first

session of FRGC. The first image in the first session was used as the single model of the person. One test image per person was randomly selected from the remaining neutral expression images (after having taken out the model image) of the person in all sessions.

Fig. 6. Example sketches for faces from (a) the AR database and (b) the CUHK student database.
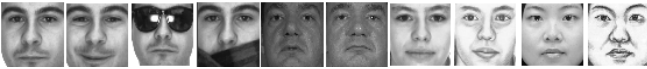

Fig. 7. Samples of the cropped faces used in our experiments.

The CUHK face sketch database (CUFS) contains sketches drawn by an artist based on 123 faces from the AR database [37], 188 faces from the Chinese University of Hong Kong (CUHK) student database, and 295 faces from the XM2VTS database [39]. All the 311 sketches for the faces in the publicly available AR and CUHK databases were used in our experiments (see examples in Fig. 6).

In all the experiments, a preprocess was applied to normalize (in scale and orientation) the original face images such that the two eyes were manually aligned roughly at the same position with a distance of 80 pixels. Then the facial regions were cropped to a size of 160 x 160 for recognition. Some example cropped faces are displayed in Fig. 7. Note that all the experiments employ a single model per person matching scheme.

The performances of the proposed method are compared with the major benchmark approaches that can handle appearance changes including partial occlusions and/or can perform face sketch recognition. They are: (1) partitioned Sparse Representation-based Classification (p-SRC) [40], Independent Component Analysis Architecture I (ICA-I) [41], Adaptively Weighted Patch Pseudo Zernike Moment Array (AWPPZMA) [42], Local Probabilistic approach (LocPb) [8], which achieved state-of-the-art performances in handling partial occlusions; (2) Eigenface [17], which is the most popular benchmark in performance comparison of face recognition approaches; (3) Line Edge Map (LEM) [10] and Directional Corner Point (DCP) [24] methods, which use descriptors based on the same source of edge information (but in an intermediate-level primitive manner) to benchmark the discriminative power of the proposed high-level representation and matching. Experimental results in all the tables are presented in groups according to the above three categories for easy comparison.

4.1. The effect of λ

The effect of λ in Eq.11 on the recognition accuracy was investigated on the AR face database, in which the neutral faces under controlled/ideal conditions taken in the first session (e.g., Fig. 5(a)) were selected as the model set, and the neutral faces under controlled/ideal conditions taken in the second session (e.g., Fig. 5(j)) were used as the test set. The

recognition rate is plotted against the values of λ in Fig. 8. It is found that Stringface with a low thresholding value (i.e. λ =2) performed badly. It improved quickly and reached the optimal value when λ ranged from 8 to 14. For all the other experiments in this study, λ was set as 10.
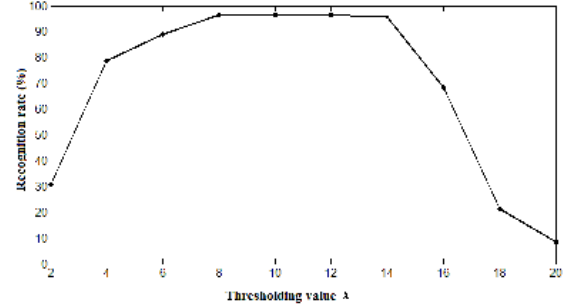

Fig. 8. The effect of λ on the recognition rate under ideal conditions.

4.2. Face recognition under controlled/ideal conditions

To have a complete performance evaluation across all recognition conditions, the system performance under controlled conditions was first evaluated on both AR and FRGC ver2.0 databases. The recognition rate of the proposed approach together with those of p-SRC, AWPPZMA, ICA-I, Eigenface, LEM, and DCP approaches are summarized in Table 1.

Table 1. Performance comparison under controlled/ideal conditions.

Method	Recognition Rate (%)	
	AR	FRGC
Stringface	96.58	82.62
p-SRC	97.43	81.76
AWPPZMA	92.31	72.75
ICA-I	82.05	65.45
Eigenface	76.07	64.38
LEM	96.58	66.95
DCP	94.87	69.53

From Table 1, it can be seen that the proposed approach performed similar to p-SRC, AWPPZMA, LEM, and DCP methods, and significantly outperformed ICA-I and Eigenface methods on the AR face database. On the large FRGC ver2.0 database, the Stringface and p-SRC methods achieved much better accuracies than the rest of the methods. This may be because the images of the same person under ideal conditions in the FRGC ver2.0 database have larger variations than those in the AR database, and AWPPZMA, LEM, and DCP methods are not as robust as Stringface and p-SRC to these variations. The Eigenface and ICA-I methods work well as long as the test image is "similar" to the ensemble of training samples and the training set should include multiple images of each person with some variations to obtain better performance. Here, all experiments employed a single model per person matching scheme. Hence, the single model based methods, i.e. Stringface, LEM, DCP, and AWPPZMA, achieved higher recognition rates than Eigenface and ICA-I. Partitioned SRC [40] is the state-of-the-art method that can recognize faces with varying expressions, varying illuminations, and occlusions. Although there is only one sample image per person, it still performed very well (97.43% and 81.76%), as

did the proposed approach (96.58% and 82.62%) on both databases.

4.3. Face recognition under partial occlusions

The proposed Stringface approach is, in theory, able to effectively find the most discriminative local parts (strings) for recognition without requiring any prior knowledge or assumption of the occlusions. The capacity of the proposed approach to recognize occluded faces was studied using two publicly available databases. The AR Face database was used to evaluate the system performance on real occluded faces, while the FRGC ver2.0 database was employed to investigate the sensitivity to the size of occlusion using the same strategy as in [43]-[46].

Table 2. Performance comparison for sunglasses and scarf occluded faces. AD: Accuracy decrease due to occlusion compared to that under ideal conditions in Table 1.

Methods	Session – 1 (%)				Session – 2 (%)				Average Accuracy (%)
	Sunglasses		Scarf		Sunglasses		Scarf		
	Accuracy	AD	Accuracy	AD	Accuracy	AD	Accuracy	AD	
Stringface	87.18	9.40	94.87	1.71	76.07	20.51	88.03	8.55	86.54
p-SRC	85.47	11.96	90.60	6.83	72.65	24.78	85.47	11.96	83.55
LocPb	80.0*	N/A*	82.0*	N/A*	54.0*	N/A*	48.0*	N/A*	66.0
AWPPZMA	70.0*	22.31*	72.0*	20.31*	58.0*	34.31*	60.0*	32.31*	65.0
ICA-I	54.70	27.35	56.41	25.64	39.31	42.74	50.43	31.62	50.21
Eigenface	12.82	63.25	8.55	67.52	8.55	67.52	5.98	70.07	8.97
LEM	58.12	38.46	29.91	66.67	43.59	52.99	22.22	74.36	38.46
DCP	28.20	66.67	6.84	88.03	14.53	80.34	5.98	88.89	13.89

* The values with asterisks are from [8] and [42] on images of 50 randomly selected persons.

1) Recognition with real data

In this experiment, the neutral face images in the first session (i.e., Fig. 5 (a)) were used as the models, while the occluded faces with sunglasses and scarfs in both the first and second sessions (i.e., Fig. 5 (h), (k), and Fig. 5 (i), (l)) were used as the tests respectively. It is observed from Table 2 that occlusions affected the performances of all methods with an accuracy drop ranging 9.40% - 66.67% and 1.71% - 88.03% for faces (Session 1) with sunglasses and scarfs, respectively. As expected, the proposed method and the baseline methods that can handle occlusions performed better than other benchmark methods. However, the Stringface (86.54% on average) and p-SRC (83.55% on average) performed significantly better than other methods including those robust to partial occlusions, and the proposed Stringface consistently obtained the highest accuracies under all four conditions (2 sessions and 2 occlusions). The sunglasses and scarfs (Session 1) only resulted in a small drop of accuracies (9.40% and 1.71% for Stringface, and 11.96% and 6.83% for p-SRC, respectively). It is interesting to note that the proposed method only has a 1.71% accuracy decrease when the faces are occluded by scarfs, showing its superior capacity in handling partial occlusions. The performance drops, due to the fact that occlusions in the session 2 images were bigger than those in the session 1 images, are understandable as the intra-class variations increase when taking the session 2 pictures after a two week time interval. As we expected, the results of all algorithms that can handle occlusions (Stringface, p-SRC, AWPPZMA, ICA-I, and LocPb) show that recognizing occluded faces with scarfs is easier than with sunglasses, because the eye areas carry more identity related information than the mouth and nose areas of a person's face, despite the

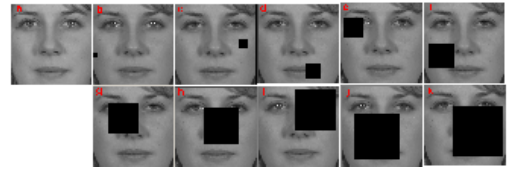fact that the scarfs covered a larger facial area than the sunglasses.

Fig. 9. Examples of model and occluded test faces. (a) Model image; (b-k) The corresponding test images with occluding blocks of sizes 10x10, 20x20, …, 100x100 respectively at a random location.

2) Sensitivity to occlusion size

Sensitivity to occlusion size is often used to quantitatively evaluate the robustness of an algorithm to occlusions. In this evaluation test, we employed the same experimental strategy as in [43]-[47] on the large FRGC ver2.0 database. The test images used in Section 4.2 were occluded by a square of $s \times s$ at a random location, while the model images remained unchanged without occlusion (see examples in Fig. 9). The size of the occlusion ranged from $s = 10$ (0.39% of the image) up to $s = 100$ (39.06% of the image) with a step size of 10.
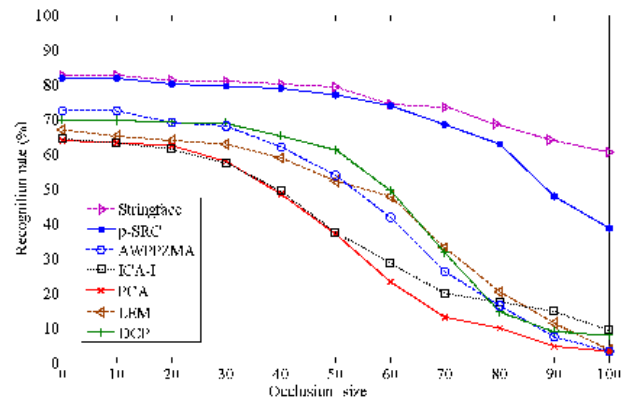

Fig. 10. Recognition under varying sizes of random occlusion (10x10, 20x20, …, 100x100 of occluding blocks).

The accuracies of the proposed method and the benchmark approaches are plotted against the occlusion size in Fig. 10. The proposed Stringface method demonstrated an ability superior to all other methods when recognizing partially occluded faces. For block sizes from 10x10 up to 50x50, the performances of Stringface and p-SRC were nearly invariant to occlusions, while all other benchmark algorithms experienced big drops in accuracy. For block sizes from 70x70 upwards, the performance gap between Stringface (73.39%) and p-SRC (68.45%) started to increase quickly, and recognition rates of the other methods dropped significantly, to below 35%. With a block size of 100x100, the Stringface approach can still achieve an impressive recognition rate of over 60%, compared to p-SRC's 38.41%. These results demonstrate the significant superiority of the proposed Stringface method in handling partial occlusions. The superior performance of the Stringface method is believed to be due to its content-based partial matching scheme (which can take every piece of non-occluded content, within regions of any shape, into the recognition process automatically), instead of the subregion-based matching scheme used by other methods (which cannot differentiate occluded and non-occluded areas

within a subregion).

4.4. *Face sketch recognition*

Computerized sketch-photo recognition has attracted much attention recently [2]-[7],[58]. This section examines how the proposed Stringface method performs for directly recognizing a face sketch with only a single sample photo per person. We randomly selected 100 persons from the AR database, and 150 from the CUHK student database. The neutral face photos from the first session (e.g., Fig. 5(a)) and the second session (e.g., Fig. 5(j)) of the AR database, and from the CUHK student database were used as the single samples of the persons, respectively. Their sketches were used as tests. Fig. 11 and Fig. 12 show the cumulative match score (CMS) of the proposed approach and the benchmark algorithms on faces from the AR database and the CUHK student database, respectively. Note that the recognition rates of p-SRC in Fig. 11 and Fig. 12 are only for rank = 1 as it can only perform rank 1 matching (In rank N matching, a correct match is counted only when the model face from the same person as the test face is among the best N matched model faces).

(a) Matching results between sketches and photos (same session)

(b) Matching results between sketches and photos (different session)
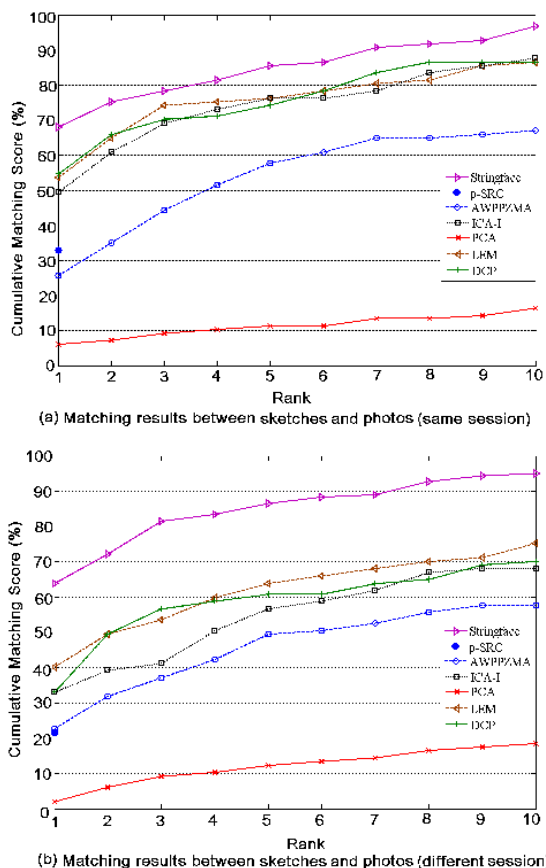
Fig. 11. Comparative performance of the proposed method and the benchmark algorithms for directly recognizing sketches of faces from the AR face database. (a) matching results between sketches and photos from the same session; (b) matching results between sketches and photos from different sessions.

Due to differences between sketches and photos and the unknown psychological mechanism of sketch generation [2], face sketch recognition is much harder than normal face recognition. It is interesting to observe from Fig. 11 and Fig.

12 that the proposed Stringface approach obtained very encouraging accuracies when directly applying it to sketch recognition, considering it as a general face recognition technique without using any sketch-photo synthesis techniques as used in [2]-[6]. Its better performance against benchmark methods may be attributed to the structural representation in Stringfaces and the flexible partial matching mechanism. A sketch drawn by an artist (1) depicts the structural characteristics of the person's face, and (2) contains some shadow textures to abstractly reflect its 3D shape information. A pencil can replicate the structural features of a face photo quite well. Stringfaces generated from facial structures in nature selectively take the reliable information of a sketch for recognition. This result, obtained from a computer vision system, seems in accordance with the findings of cognitive psychological studies [33],[34] that show that humans recognize line drawings as quickly and almost as accurately as gray level pictures. Some even conjecture that caricatures may be better representations than natural images [2] because caricatures contain some kind of "super-fidelity" due to its accentuated structure in addition to the essential minimum of identity information [48]. However, the experimental results in this section cannot be over-interpreted as supporting the above hypothesis because the benchmark approaches are those best at handling partial occlusions (plus the most popular baseline Eigenface and approaches based on similar source of features), instead of all face recognition algorithms, upon which a meaningful argument could be made.
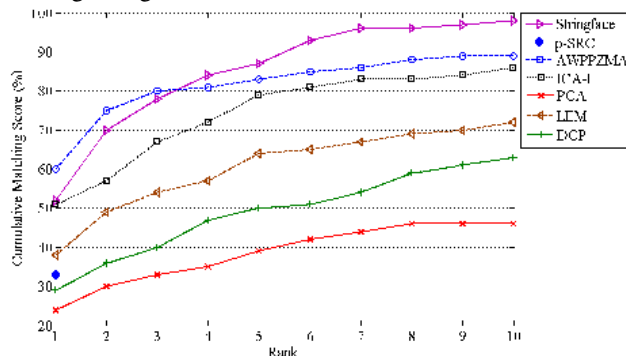

Fig. 12. Comparative performance of the proposed method and the benchmark algorithms for directly recognizing sketches of faces from the CUHK student database.

Although sketches reproduce the shape and structural features better than textures, they inevitably exaggerate some distinctive facial features like caricatures, which involve shape deformation [2]. For example, it is observed that most of the noses drawn in the sketches of CUHK faces are bigger than those in photos. The flexible partial matching mechanism in ESM can automatically exclude the largely distorted areas from the calculation of matching score, thus eliminating the negative effect of large distortions in a sketch. This is evidenced by the much higher performance of Stringface compared to non-partial matching LEM and DCP approaches, which use similar types of structural information at the intermediate level. The differences in accuracies in Fig. 12 are larger than those in Fig. 11 because the sketches of CUHK student faces have larger distortions from their photos than the sketches of AR faces (This also resulted in lower accuracies on the CUHK sketches compared to those on the AR

sketches). As the sketches of AR faces were drawn based on the photos taken in Session-1 [59], the recognition rates for matching the sketches and photos from the same session (Fig. 11(a)) are consistently higher than those from different sessions (Fig. 11(b)).

4.5. Face recognition under varying lighting conditions

The sensitivity to illumination changes is one of the important issues for the evaluation of face recognition systems. The proposed Stringface employs a high-level syntactic primitive representation derived from intermediate-level representation of lines grouped from low-level edge pixels. We performed an experimental study to evaluate how well Stringface performs for recognizing faces under varying lighting conditions. The neutral face images in the AR database taken in the first session (e.g., Fig. 5(a)) were used as the single models of the persons. The face images under three different lighting conditions (e.g., Fig. 5 (e), (f), and (g)) taken in the same session were used as test images. The results are summarized in Table 3.

Table 3. Experimental results under varying lighting conditions.

Methods	Recognition Rates (%)			
	Left light	Right light	Both lights	Average
Stringface	94.02	94.02	73.50	87.18
p-SRC	95.73	96.58	32.48	74.93
AWPPZMA	74.36	64.96	42.74	60.69
ICA-I	66.67	20.51	48.33	45.28
Eigenface	27.35	50.43	66.67	48.15
LEM	92.31	91.45	73.50	85.75
DCP	87.18	88.89	61.54	79.20

It can be seen from Table 3 that the proposed approach achieved an average accuracy improvement of 1.43% over the LEM method, which is one of the best illumination insensitive methods based on facial edges for single model image recognition. This is believed to be due to the fact that the LEM method employs an intermediate-level image representation derived from low-level edge map representation, while the proposed Stringface approach represents not only the local structural information but also their interrelationships within a face. Table 3 also shows that the proposed approach significantly outperformed DCP, AWPPZMA, ICA-I, and Eigenface methods under different lighting conditions.

4.6. Face recognition under varying facial expressions

Similar experiments were conducted to evaluate the effects of different facial expressions (smiling, angry, and screaming) on the system performance. The faces with neutral expression in the first session (e.g., Fig. 5 (a)) were used as the model images; the face images with three different expressions (e.g., Fig. 5 (b), (c), and (d)) taken in the same session were used as the test images.

The experimental results are summarized in Table 4. The smiling and the angry expressions caused the recognition rate of Stringface to drop by 11.11% and 9.40% compared to the neutral expression in Table 1. The large expression changes of screaming significantly affected the performance of Stringface. The Eigenface and the AWPPZMA methods are found the least sensitive to the facial expression changes.

Table 4. Experimental results under varying facial expression changes.

Methods	Recognition Rates (%)			
	Smiling	Anger	Scream	Average
Stringface	85.47	87.18	26.49	66.38
p-SRC	93.16	90.60	33.33	72.36
AWPPZMA	96.58	87.18	38.46	74.07
ICA-I	81.19	73.50	31.62	62.10
Eigenface	94.02	87.18	45.30	75.50
LEM	79.49	93.16	31.62	68.09
DCP	63.25	94.02	28.20	61.82

V. CONCLUSION

In this paper, we presented a new method for human frontal face recognition using a high-level string representation and matching strategy. A new Stringface representation is proposed, which groups low-level "letter" primitives into intermediate-level "word" primitives and then further integrates their relational organization into high-level "sentence" strings. The similarity measurement of two faces is performed by matching two Stringfaces through a new Ensemble String Matching scheme, which is able to effectively find the most similar segments between two Stringfaces, regardless of the sequential order of strings and the direction of each string in a Stringface. Its embedded partial matching mechanism can recognize faces with occlusions of arbitrary shapes and locations without requiring any prior knowledge or assumption of the missing regions. The proposed approach is compared against the benchmark algorithms using three databases (AR, FRGC ver2.0, and CUFS). The promising experimental results demonstrate the superiority of the proposed method in recognising partially occluded faces, and the feasibility of using a high-level syntactic method for face recognition, indicating a new strategy for face coding and matching.

REFERENCES

[1] G. U. J. Robert and N. d. V. Lobo, "Recognizing a facial image from a police sketch," *Proceedings of the Second IEEE Workshop on Applications of Computer Vision,* pp. 129-137, 5-7 December 1994.
[2] X. Wang and X. Tang, "Face photo-sketch synthesis and recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 31, pp. 1955-1967, Nov. 2009.
[3] P. C. Yuen and C. H. Man, "Human face image searching system using sketches," *IEEE Transactions on Systems, Man and Cybernetics - Part A: Systems and Humans,* vol. 37, pp. 493-504, July 2007.
[4] B. Xiao, X. Gao, D. Tao, and X. Li, "A new approach for face recognition by sketches in photos," *Signal Processing,* vol. 89, pp. 1576 - 1588, 2009.
[5] X. Gao, J. Zhong, D. Tao, and X. Li, "Local face sketch synthesis learning," *Neurocomputing,* vol. 71, pp. 1921--1930, 2008.
[6] B. Xiao, X. Gao, D. Tao, Y. Yuan, and J. Li, "Photo-sketch synthesis and recognition based on subspace learning," *Neurocomputing,* vol. 73, pp. 840 - 852, 2010.
[7] B. Klare and A. K. Jain, "Sketch to photo matching: a feature-based approach," *Proceedings of the SPIE,* vol. 7667, pp. 766702-766702-10, 2010.
[8] A. M. Martinez, "Recognizing imprecisely localized, partially occluded, and expression variant faces from a single sample per class," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 24, pp. 748-763, Jun 2002.
[9] X. Tang and X. Wang, "Face sketch recognition," *IEEE Transactions on Circuits and Systems for Video Technology,* vol. 14, pp. 50-57, 2004.
[10] Y. Gao and M. K. H. Leung, "Face recognition using line edge map," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 24, pp. 764-779, 2002.
[11] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs.

Fisherfaces: recognition using class specific linear projection," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 19, pp. 711-720, Jul 1997.

[12] I. Naseem, R. Togneri, and M. Bennamoun, "Linear regression for face recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 32, pp. 2106-2112, 2010.

[13] X. Tan, S. Chen, Z. Zhou, and F. Zhang, "Face recognition from a single image per person: a survey," *Pattern Recognition,* vol. 39, pp. 1725--1745, 2006.

[14] T. Kanade, "Picture processing system by computer complex and recognition of human faces," Dept. Information Science, Kyoto Univ. November 1973.

[15] R. Brunelli and T. Poggio, "Face recognition: features versus templates," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 15, pp. 1042-1052, 1993.

[16] M. Kirby and L. Sirovich, "Application of the Karhunen-Loeve procedure for the characterization of human faces," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 12, pp. 103--108, 1990.

[17] M. A. Turk and A. P. Pentland, "Eigenfaces for recognition," *Journal of Cognitive Neuroscience,* vol. 3, pp. 71-86, 1991.

[18] J. Wu and Z. H. Zhou, "Face recognition with one training image per person," *Pattern Recognition Letters,* vol. 23, pp. 1711-1719, 2002.

[19] J. P. Phillips, H. Moon, S. A. Rizvi, and P. J. Rauss, "The FERET evaluation methodology for face-recognition algorithms," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 22, pp. 1090--1104, 2000.

[20] S. C. Chen, D. Q. Zhang, and Z.-H. Zhou, "Enhanced (PC)2A for face recogniton with one training image per person," *Pattern Recognition Letters,* vol. 25, pp. 1173-1181, 2004.

[21] T. Ahonen, A. Hadid, and M. Pietikainen, "Face description with local binary patterns: application to face recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 28, pp. 2037-2041, 2006.

[22] B. Zhang, Y. Gao, S. Zhao, and J. Liu, "Local Derivative Pattern versus Local Binary Pattern: face recognition with high-order local pattern descriptor," *IEEE Transactions on Image Processing,* vol. 19, pp. 533-544, 2010.

[23] A. Pentland, B. Moghaddam and T. Starner, "View-based and modular eigenspaces for face recognition," *Proc. IEEE Computer Soc. Conf. On Computer Vision and Pattern Recognition*, pp. 84-91, 1994.

[24] Y. Gao and Y. Qi, "Robust visual similarity retrieval in single model face databases," *Pattern Recognition,* vol. 38, pp. 1009-1020, 2005.

[25] S. C. Chan and A. K. C. Wong, "Synthesis and recognition of sequences," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 13, pp. 1245--1255, 1991.

[26] L. Rabiner and B.-H. Juang, *Fundamentals of speech recognition.* Prentice-Hall, Inc., 1993.

[27] P. Heckel, "A technique for isolating differences between files," *Communications of ACM,* vol. 21, pp. 264-268, 1978.

[28] L. Bahl and F. Jelinek, "Decoding for channels with insertions, deletions, and substitutions with applications to speech recognition," *IEEE Transactions on Information Theory,* vol. 21, pp. 404-411, 1975.

[29] W. H. Tsai and S. S. Yu, "Attributed string matching with merging for shape recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 7(4), pp. 453-462, 1985.

[30] Y. Gao and M. K. H. Leung, "Human face profile recognition using attributed string," *Pattern Recognition,* vol. 35, pp. 353-360, 2002.

[31] S. W. Chen, S. T. Tung, C. Y. Fang, S. Cheng, and A. K. Jain, "Extended attributed string matching for shape recognition," *Computer Vision and Image Understanding,* vol. 70, pp. 36-50, 1998.

[32] B. Takács, "Comparing face images using the modified hausdorff distance," *Pattern Recognition,* vol. 31, pp. 1873-1881, 1998.

[33] I. Biederman and G. Ju, "Surface versus edge-based determinants of visual recognition," *Cognitive Psychology,* vol. 20, pp. 38-64, 1988.

[34] V. Bruce, E. Hanna, N. Dench, P. Healey, and M. Burton, "The importance of mass in line drawings of faces," *Applied Congnitive Psychology,* vol. 6, pp. 619-628, 1992.

[35] M. K. H. Leung and Y. H. Yang, "Dynamic two-strip algorithm in curve fitting," *Pattern Recognition,* vol. 23, pp. 69-79, 1990.

[36] A. Leonardis and H. Bischof, "Robust recognition using Eigenimages," *Computer Vision and Image Understanding*, Vol. 78, pp. 99-118, 2000.

[37] A. M. Martinez and R. Benavente, "The AR face database," *CVC Technical Report,* vol. 24, 1998.

[38] P. J. Phillips, P. J. Flynn, T. Scruggs, K. W. Bowyer, J. Chang, K. Hoffman, J. Marques, J. Min, and W. Worek, "Overview of the face recognition grand challenge," *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition,* vol. 1, pp. 947-954, 2005.

[39] K. Messer, J. Matas, J. Kittler, J. Luettin, and G. Maitre, "XM2VTSDB: the extended M2VTS database," *Proceedings of International Conference on Audio- and Video-Based Person Authentication,* pp. 72-77, 1999.

[40] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 31, pp. 210-227, 2009.

[41] M. S. Bartlett, J. R. Movellan, and T. J. Sejnowski, "Face recognition by independent component analysis," *IEEE Transactions on Neural Networks,* vol. 13, pp. 1450-1464, 2002.

[42] H. R. Kanan, K. Faez, and Y. Gao, "Face recognition using adaptively weighted patch pzm array from a single exemplar image per person," *Pattern Recognition,* vol. 41, pp. 3799-3812, 2008.

[43] J. Kim, J. Choi, J. Yi, and M. Turk, "Effective representation using ICA for face recognition robust to local distortion and partial occlusion," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 27, pp. 1977-1981, 2005.

[44] X. Tan, S. Chen, Z.-H. Zhou, and J. Liu, "Face recognition under occlusions and variant expressions with partial similarity," *IEEE Transactions on Information Forensics and Security,* vol. 4, pp. 217-230., 2009.

[45] H. J. Oh, K. M. Lee, and S. U. Lee, "Occlusion invariant face recognition using selective local non-negative matrix factorization basis images," *Image and Vision Computing,* vol. 26, pp. 1515-1523, 2008.

[46] H. Jia and A. M. Martinez, "Support vector machines in face recognition with occlusions," *IEEE Conference on Computer Vision and Pattern Recognition,* pp. 136 - 141, 2009.

[47] S. Z. Li, X. W. Hou, H. J. Zhang, and Q. S. Cheng, "Learning spatially localized, parts-based representation," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition,* vol. 1, pp. I-207-I-212 vol.1, 2001.

[48] J. Benson and D. I. Perrett, "Perception and recognition of photographic quality facial caricatures: implications for the recognition of natural images," *European Journal of Cognitive Psychology,* vol. 3, pp. 105-135, 1991.

[49] X. Tan, S. Chen, Z. Zhou and F. Zhang "Recognizing partially occluded, expression variant faces from single training image per person with SOM and soft k-NN ensemble," *IEEE Transactions on Neural Networks*, vol. 16, pp.875-886, 2005.

[50] K. Ohba and K. Ikeuchi, "Detectability, uniqueness, and reliability of eigen windows for stable verification of partially occluded objects," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, pp. 1043-1047, 1997.

[51] Y. Saito, Y. Kenmochi and K. Kotani, "Estimation of eyeglassless facial images using principal component analysis," *International Conference on Image Processing*, vol. 4, pp. 197-201, 1999.

[52] B. W. Hwang and S. W. Lee, "Reconstruction of partially damaged face images based on a morphable face model," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, pp. 365-372, 2003.

[53] W. Chen and Y. Gao, "Recognizing partially occluded faces from a single sample per class using string-based matching," *The 11th European Conference on Computer Vision*, vol. 3, pp. 496-509, 2010.

[54] T. Bourlai, A. Ross, and A. Jain, "On Matching Digital Face Images Against Scanned Passport Photos," *Proc. of First IEEE International Conference on Biometrics, Identity and Security*, pp. 1-10, 2009.

[55] M. Bicego, A. Lagorio, E. Grosso, and M. Tistarelli, "On the use of SIFT features for face authentication," *Proc. of the Conference on Computer Vision and Pattern Recognition Workshop*, 2006.

[56] X. Li and F. Da. "Efficient 3D Face Recognition Handling Facial Expression and Hair Occlusion." *Image and Vision Computing* , vol. 30, pp. 668–679, 2012.

[57] X. Li, X. Zhang, and C. Ren. "Structured Sparse Error Coding for Face Recognition with Occlusion." *IEEE transactions on image processing: a publication of the IEEE Signal Processing Society*, [Epub ahead of print] Jan. 2013.

[58] B. Klare, Z. Li, and A. K. Jain, "Matching Forensic Sketches to Mug Shot Photos," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, pp. 639-646, 2011.

[59] X. Tang and X. Wang, e-mails on CUHK Face Sketch Database, April 9, 2011.