International Journal of Applied Engineering Research
ISSN 0973-4562 Volume 10, Number 4 (2015) pp. 11145-11160

http://www.ripublication.com

Frontal Face Recognition Using Combination Of Feature Extraction And String Matching

K.M.Ponnmoli and Dr. S. Selvamuthukumaran

Research Scholar, PRIST University
Thanjavur, Tamilnadu, India
kmponnmoli@yahoo.in
Director, Department of Computer Applications
A.V.C.College of Engineering, Mannampandal, Tamilnadu, India
smksmk@gmail.com

Abstract—

The recognition of frontal human faces with changes in appearance, when compared with a single passport size photo from the data base, is a challenging task in image processing. A combinational approach of feature face using Chan-vese algorithm and string face matching is proposed for frontal face recognition to overcome the drawbacks of conventional techniques at minimal computation costs. Since the discomposure of shape retrieval is impossible to determine, we propose a Levenshtein distance analysis which implies the use of flexible shape formulations for string matching from feature faces by the symmetry of their appearance. The Levenshtein distance algorithm finds the minimum cost path, from the upper left corner to the lower right corner. The results show that the accuracy of determining the similarities and dissimilarities is improved to produce the best results

Index Terms— String Matching, Feature Extraction, Chan-vese algorithm, Levenshtein distance

I. INTRODUCTION

Face recognition is one of the biometrics methods to identify individuals by the features of the face. Though people are good at face identification, recognizing human face automatically by computer is very difficult. Face recognition has been widely applied in security system, credit-card verification, and criminal identifications, teleconference and so on. Face recognition is influenced by many complications, such as the differences of facial expression, the light directions of imaging, and the variety of posture, size and angle. Even to the same people, the images taken in different surroundings may be unlike. The problem is so complicated that the achievement in the field of automatic face recognition by computer is not as satisfied as the finger prints. Facial feature extraction has become an important issue in automatic recognition of human faces. Detecting the basic feature as eyes, nose and mouth exactly is necessary for most face recognition methods.

In feature extraction, we generally seek invariance properties so that the extraction process does not vary according to chosen (or specified) conditions. That is, techniques should find shapes reliably and robustly whatever the value of any parameter that can control the appearance of a shape. Flexible shape extraction implies knowledge of a model (mathematical or template) of the target shape (feature). The shape is fixed in that it is flexible only in terms of the parameters that define the shape, or the parameters that define a template's appearance. Flexible shape formulations can evolve to the target solution, or adapt their result to the data.

In this paper, we propose frontal face recognition by means of a combinational approach based on facial feature extraction using Chan-vese algorithm and string face matching from input images to produce the best results, and to improve the accuracy of face recognition through various environments.

II. FEATURE EXTRACTION AND STRING MATCHING

Feature extraction is the process of extracting relevant information from the images. Feature extraction involves dimensionality reduction, normalisation, segmentation and feature selection. Low-level features to be those basic features that can be extracted automatically from an image without any shape information (information about spatial relationships). High-level feature extraction concerns finding shapes in computer images. To be able to recognize faces automatically, for example, one approach is to extract the component features. This requires extraction of, say, the eyes, the ears and the nose, which are the major facial features. To find them, we can use their shape: the white part of the eyes is ellipsoidal; the mouth can appear as two lines, as do the eyebrows. Shape extraction implies finding their position, their orientation and their size. But before we can develop image analysis techniques, we need techniques to extract the shapes. Extraction is more complex than detection, since extraction implies that we have a description of a shape, such as its position and size, whereas detection of a shape merely implies knowledge of its existence within an image.

However, there are no clear-cut boundaries between these sub-tasks. The feature extraction step replaces an image with a set of features. All inputs have a corresponding feature set associated with them. These features are the inputs to the classifier [2]. However, the feature extraction and recognition are two subtasks which are very hard to isolate in most cases. They include PCA (Principal Component Analysis), which forms a linear mapping of the image to a subspace. Face images are projected into a feature space called face space. The face is encoded using a set of Eigen vectors called the Eigen faces. PCA is one of the first methods that were used for face feature extraction. Another approach called modular PCA was introduced later, where the image is divided into sub-regions and each region is projected into specific face sub-spaces. Modular PCA provided more accurate results than PCA [4].

Kernel PCA which is Eigen vector based forms a nonlinear mapping to a subspace. In simple terms, this is a nonlinear form of PCA. It uses Integral Operator kernel functions to compute the principal components in the higher dimensional space. The points in this space are related to the input space in a nonlinear manner. This nonlinear mapping forms a better feature when compared to the linear mapping provided by PCA [5]. Weighted PCA which is a weighted variant of PCA [6]. It offers a better recognition result than PCA.

LDA (Linear Discriminant Analysis) which is Eigen vector based that forms a supervised linear map [7]. It uses Fisher faces instead of Eigen faces for representing the face images in the subspace defined using LDA. Several modified variants of LDA have been introduced ever since first LDA was introduced. Most of the variants perform better than the conventional LDA.

Kernel LDA which is LDA with kernel methods [8]. This is similar to kernel PCA. Face image data distribution is highly complex. They form complex manifolds in higher dimensional space. To effectively represent these manifolds, a nonlinear mapping is required. The Kernel LDA is a variant of LDA that offers a nonlinear mapping. It is observed to perform better than LDA and Kernel PCA.

ICA (Independent Component Analysis) which forms a linear map [9]. This separates multivariate signals into components, based on assumption that non-Gaussian signals are statistical independent. Neural Network based methods that encompass a very large collection of algorithms that use neural networks [10]. They include the Multilayer Neural Networks based on error back propagation. Another popular architecture is Self Organizing Maps. The Nonlinear attractor is also a very promising Neural Network based nonlinear mapping architecture. They help model complex manifolds using Neural Networks. They are built based on the basic Hopfield Network architecture.

Active Shape Models that search boundaries using statistical methods [11]. This is a statistical model which involves two steps. They are the model construction step and the searching step. A set of feature points are identified in all images. The same feature points are marked on all the faces. The shape is represented using a shape vector. The mean shape and the new profiles are compared using Mahalanobis distance for recognition.

Graph Models that use a graphical structure as a feature for recognition task [12]. In some methods, probabilistic graphical models are used in conjunction with the feature set. The nodes in these graphs represent the conditional dependencies between the feature sets. These graphs are used to classify individuals.

SIFT features (Scale Invariant Feature Transform) which was introduced by David Lowe in 2004 for object recognition. The features are also useful for face recognition [13]. This method captures the Grey level features of the image by a scale space decomposition of the image.

SURF (Speeded Up Robust Features), which is another feature extraction technique used for face recognition [14] [15]. It relies on integral images. The detector is based on the Hessian matrix. SURF is partially inspired by SIFT. It is much faster than SIFT.

HoG (Histogram of Gradients) which is a method that was proposed in 2005 for object recognition [16]. The method describes shapes using the distribution of intensity gradients or edge directions. The method was found useful for face recognition later [17]. It performs better than many other conventional feature extraction techniques.

LBP (Local Binary Patterns) which is perhaps the most effective of all the available features so far [18]. The feature was initially defined for texture analysis. Later it was found that the same features were useful for face recognition and resulted in higher accuracy than other methods [19]. The LBP feature creates a binary number for each local neighborhood. LBP features are later classified using statistical measures such as Chi-Square distance or Log likelihood statistic.

These previous extraction techniques covered finding shapes by matching. This implies knowledge of a model (mathematical or template) of the target shape (feature). The shape is fixed in that it is flexible only in terms of the parameters that define the shape, or the parameters that define a template's appearance. Sometimes, however, it is not possible to model a shape with sufficient accuracy, or to provide a template of the target as needed for the frontal face recognition. It might be that the exact shape is unknown or that the perturbation of that shape is impossible to parameterize. For this, we propose a technique that can evolve to the target solution, or adapt their result to the data. This implies the use of flexible shape formulations for string matching from feature faces. We shall also look at determining a shape's skeleton, by distance analysis and by the symmetry of their appearance. This technique finds any symmetric shape by gathering evidence by considering features between pairs of points.

Two fundamental problems in visual retrieval and classification are the design of good data representations and the definition of a quantitative metric that measures the similarities or dissimilarities between each pair of visual data. The characteristics of a good representation include a high sensitivity to data that represent different concepts and invariability to data that are perceptually alike or belong to the same class. A good similarity measure appropriately quantizes the similarity and is robust to imperfect features, such as the presence of noise. For example, two images that contain the same object should have a small distance measurement even though the object is viewed at different angles or is posed in different backgrounds. Both tasks are challenging because of the well-known sensory and semantic gap [22]. With the recent development of robust local features [23, 24], representations that consist of parts that described by local descriptors have demonstrated impressive performance in various domains. For example, bag-of-features methods, which represent an image as a distribution of visual words' frequency, have shown promising results for recognizing natural scene categories and for string matching.

For such representations, string matching, which optimally determines corresponding points of two feature sets, is used to measure the similarity [24, 25, 26, 27]. However, the order of parts carries useful information as it preserves some spatial layout information of the features. In fact, many real-world objects might be better represented by a sequence of features.. Strings can be classified into two categories: symbolic strings and attributed strings. Attributed string matching is proposed for measuring the similarity of shapes by tracing the continuous contour of a shape (feature face), then converting them into a string. String matching has the ability to perform partial matching without considering the location and size of the missing data in the string. This is particularly very useful when recognizing human faces in real applications, because of occlusions can occur at any location, and shapes.

A person's face could be represented by patches, which are data sets grouped around different facial features such as nose, ears, and mouth. If we order the data from the top to the bottom, the eye patches should appear before the mouth patch. In this paper, we show that the descriptive ability of representations based on sets of features can be improved if their order is considered.

In this paper we propose a representation for visual data based on ordered features and a method to measure the similarity based on such a representation, we explore the use of approximate string matching method that addresses the problem of error toleration. The key characteristic of this method is that if one of the two strings-under-matching is likely an erroneous variant of the other. However, with the current state of technology, string matching only works on matching a single 1D sequentially ordered silhouette of the shape to another one. To our knowledge, there is no frontal face recognition using combinational approach of feature extraction and string matching techniques.

In the rest of the paper, we first review previous work in this area. We then describe our formulation of the representation and the matching method in section 3. Finally, we conclude the paper in section 4.

III. MATCHING APPROACH

The flow diagram of our proposed approach is shown in the fig.1 In our approach, we first get our facial feature extraction using chen- vese algorithm. Further we design to extract string face from this facial feature extraction in order to measure the similarities and dissimilarities accurately. Of course, these two techniques can able to identify the shape retrieval individually, we propose the combinational approach of feature face and string face to overcome the drawbacks of conventional techniques at minimal computation costs. The circle fitting algorithm is used for shape retrieval from the feature face. The Breadth first algorithm and Delaunay triangulation algorithm in addition with Levenshtein distance, are used to determine the strings of eyes, ears, nose and mouth for string face formulation. These string formulations will be stored in the data base in addition with feature face. After the enrolment, both the feature and string faces carryout the verification process for frontal face recognition

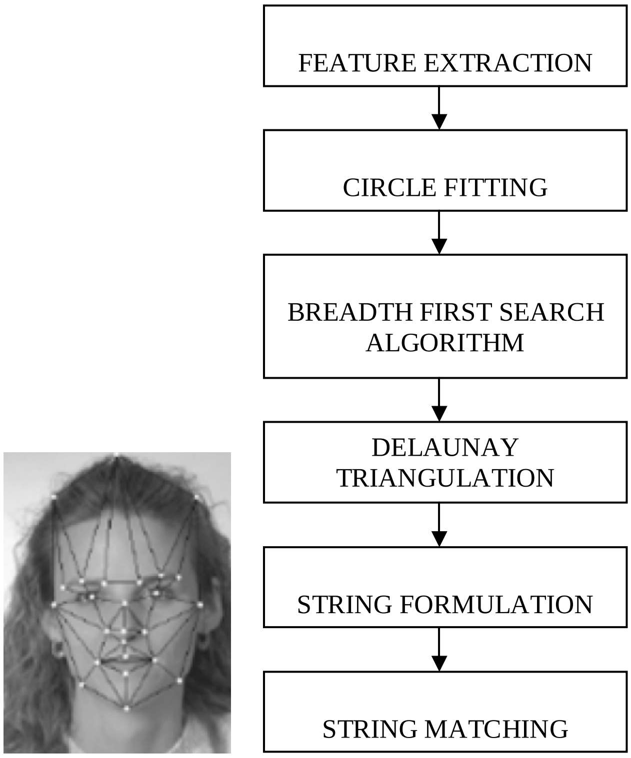


Fig. 1 Flow Diagram of Proposed Approach with example of face graph

In this study, we attempt to develop a new strategy for face recognition which can able to recognize faces with partial occlusions. It is single model based recognition, which also matches string faces and photo faces directly. We propose a novel string face representation and matching concept for face recognition with one image pattern per person. The string face is a synthesised pattern representation of a face image, which can be constructed from feature extraction of a single face image, without any training. A string is a chain of straight lines, connected together, representing an edge curve on the face. A string face is composed of many strings, which holds a description of image pattern about the identity of the face. Hereafter we discuss the algorithms implemented for our proposed approach in brief

a) Chan-Vese Algorithm for Feature Extraction:

The original energy function of CV is defined as

$E\_cv(C) = F1(C)+F2(C) = \text{Integrate<interior of C>} |U(x,y)-u1|^2 + \text{Integrate<exterior of C>} |U(x,y)-u2|^2$

where U(x,y) represents the pixel located at (x,y) in the 2D image U and u1 and u2 are the average for the interior region and the exterior region respectively.

Finally, to find a contour C such that E_cv is minimized

The above energy function can be understood as the OTSU Optimal Global Thresholding Scheme. The OTSU's method is applied directly on the histogram of an image, it searched the thresholding T, such that:

1. the histogram is separated into two groups, C_a (above T) and C_b (below T)
2. E_otsu(T) = W_a*Var_a+W_b*Var_b is minimized

Here W_a = sum[Pr(i>=T)], W_b = sum[Pr(i<T)] and Var_x is the variance of the group C_x

Although OTSU's method is derived for image histogram, it can be easily adapt to the image itself

Then we can say

E_otsu(T) = sum[(U(x,y) -u)^2] ,

where u = u_b is the mean of C_b, when U(x,y)<T

u = u_a is the mean of C_a, when U(x,y)>=T

Reader may noticed that both W_x and Var_x in the original E_otsu formula disappeared, this is because:

1. W_x is the weight of the group x and W_x = N [C_x]/ N[U], i.e. the ratio of the number of pixels in the group x to the number of pixels in image U
2. Var_x = (C_x - u_x^2) / N[C_x]

Therefore,

E_otsu(T) = SUM<x = a; x = b> { N[C_x]/ N[U^2] * SUM<U(p,q) is in the group of C_x>{ [(U(p,q) -u_x)^2]}/ N[C_x] }

E_otsu(T) = 1/N[U] SUM<x = a; x = b> { SUM<U(p,q) is in the group of C_x>{ [(U(p,q) -u_x)^2]} }

E_otsu(T) = SUM<in group Ca> |U(p,q)-u_a|^2 + SUM<in group Cb> |U(p,q)-u_b|^2

Finally, if one start to think the difference between the gray threshold T and the contour C, you will find that they are actually doing the same thing: classified the given image U into two parts, except the groups defined on C are called the interior region and the exterior region and the groups defined on the single threshold T are called the region above the threshold and the region below the threshold. In other words, for a given threshold T, there is always a contour C' such that T and C' have the identical partitions on the image U. The fig.2 shows the results of feature extraction from input images.

(a) Input Image **(b) Feature Extracted Image**

Fig. 2 Feature Extraction using Chan-Vese Algorithm

b) Circle Fitting by Full Least Squares Method

In general, suppose that we have a collection of $n \geq 3$ points in 2-space labeled $(x_1, y_1), (x_2, y_2, \ldots, (x_n, y_n)$.

Our basic problem is to _nd a circle that best represents the data in some sense. With our circle described by $(x-a)^2 + (y- b)^2 = r^2$, we need to determine values for the center (a; b) and the radius r for the best fitting circle.

A reasonable measure of the _t of the circle $(x-a)^2 + (y- b)^2 = r^2$ to the points $(x_1, y_1), (x_2, y_2, \ldots, (x_n, y_n)$. is given by summing the squares of the distances from the points to the circle. This measure is given by

$$SS(a,b, r) = \sum_{i=1}^{n} (r-\sqrt{(x_i- a)^2 + (y_i- b)^2})^2$$

[36] discusses numerical algorithms for the minimization SS over a,b, and r. Gander, Golub, and Strebel in [38] also discuss this problem.

For notational convenience, we make the following conventions:

$X_{ij} = xi - xj$
$X_{ijk} = X_{ij}X_{jk}X_{ki}$
$X^{(2)}{}_{ij} = x^2{}_i- x^2{}_j$
$Y_{ij} = y_j - y_j$
$\hat{y}_{ijk} = Y_{ij}Y_{jk}Y_{ki}$
$Y^{(2}{}_{)ij} = y^2{}_i - y^2{}_j$

An obvious approach is to choose a; b; and r to minimize SS. Differentiation of SS yields

$$\frac{\partial SS}{\partial r} = -2\sum_{i=1}^{n} (\sqrt{((x_i- a)^2 + (y_i- b)^2})^2 \quad (1)$$

$$\partial SS = 2r \sum^{n} (x_i-a)/\sqrt{((x_i- a)^2 + (y_i- b)^2})^2 \quad (2)$$

$$\overline{\partial a} \quad {}_{-2nx+2na}$$

$$\frac{\partial SS}{\partial b} = 2r \sum_{-2ny+2nb}^{n} (y_i - a)/\sqrt{((x_i - a)^2 + (y_i - b)^2)^2} \qquad (3)$$

Simultaneously equating these partials to zero does not produce closed form solutions for a, b and r. However, many software programs will numerically carry out this process quite efficiently. We shall refer to this method as the Full Least Squares method (FLS) with resulting values of a, b, and r labeled as a_f , b_f , and r_f .

c) Breadth First Search Algorithm

In graph theory, breadth-first search (BFS) is a strategy for searching in a graph when search is limited to essentially two operations: (a) visit and inspect a node of a graph (b) gain access to visit the nodes that neighbor the currently visited node.

The BFS begins at a root node and inspects all the neighboring nodes. Then for each of those neighbor nodes in turn, it inspects their neighbor nodes which were unvisited, and so on. Compare BFS with the equivalent, but more memory-efficient Iterative deepening depth-first search and contrast with depth-first search. The algorithm uses a queue data structure to store intermediate results as it traverses the graph, as follows:

1. Enqueue the root node
2. Dequeue a node and examine it
   If the element sought is found in this node, quit the search and return a result.
   Otherwise enqueue any successors (the direct child nodes) that have not yet been discovered.
3. If the queue is empty, every node on the graph has been examined – quit the search and return "not found"
4. If the queue is not empty, repeat from Step 2.

Breadth-first search can be used to solve many problems in graph theory, for example:

- Finding all nodes within one connected component
- Copying Collection, Cheney's algorithm
- Finding the shortest path between two nodes u and v (with path length measured by number of edges)
- Testing a graph for bipartiteness
- (Reverse) Cuthill–McKee mesh numbering
- Ford–Fulkerson method for computing the maximum flow in a flow network
- Serialization/Deserialization of a binary tree vs serialization in sorted order, allows the tree to be re-constructed in an efficient manner.

Finding connected components:

The set of nodes reached by a BFS (breadth-first search) form the connected component containing the starting node.

d) Delaunay Triangulation

In mathematics and computational geometry, a Delaunay triangulation for a set P of points in a plane is a triangulation DT(P) such that no point in P is inside the circumcircle of any triangle in DT(P). Delaunay triangulations maximize the minimum angle of all the angles of the triangles in the triangulation; they tend to avoid skinny triangles.

For a set of points on the same line there is no Delaunay. For four or more points on the same circle (e.g., the vertices of a rectangle) the Delaunay triangulation is not unique: each of the two possible triangulations that split the quadrangle into two triangles satisfies the "Delaunay condition", i.e., the requirement that the circumcircles of all triangles have empty interiors.

By considering circumscribed spheres, the notion of Delaunay triangulation extends to three and higher dimensions. Generalizations are possible to metrics other than Euclidean. However in these cases a Delaunay triangulation is not guaranteed to exist or be unique.

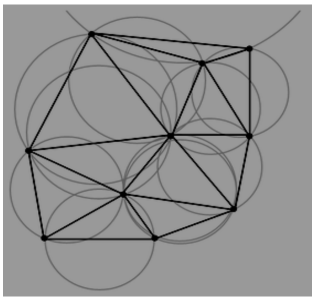


Fig. 3 A Delaunay triangulation in the plane with circum-circles

f) String Face Formulation

Let $X = [x_1, x_2,..., x_m]$ and $Y = [y_1,y_2,..., y_n]$ denote two sequences of feature vectors with sizes m and n respectively. The distance between X and Y is defined as the minimal cost of a sequence of operations that transforms X into Y.

$$d(X;Y) = \min \sum_i (c_i) \tag{4}$$

where c_i is the cost of an operation, denoted in the form of $\delta(. , .)$, which transforms a feature of X so that the resulting feature set is closer to Y, and the total cost is the sum of the costs of all operations that complete the transformation from X to Y.

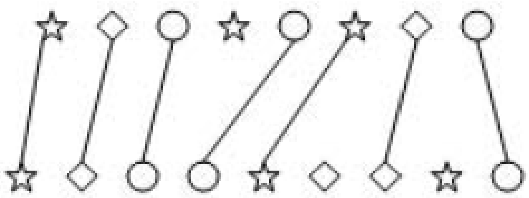


Fig. 4 A simple example of approximate string matching.

There are three types of feature vectors, which are indicated by stars, diamonds and circles. The best match occurs by deleting the 4th feature in the top sequence and the 6th and 8th features in the bottom sequence. Equivalently, the top string can be transformed to the bottom string by deleting the 4th feature and adding a diamond between the 6^{th} and 7th features, and a star between the 7th and 8^{th} features.

We use the Levenshtein distance, in which the set of operations is restricted to (1) insertion δ(□; a), i.e. inserting a feature vector a to X, (2) deletion *delta*(a; □), i.e. deleting a feature vector a from X, and (3) substitution *delta*(a,b), i.e. substituting a in X with b. This formulation can then be rephrased as "the minimal cost of insertions, deletions and substitutions to make two sequences of feature vectors, X and Y, equal." Fig. 4 shows a simple example of matching two sequences that consist of three feature types.

The determination of each operation's cost is application dependent; however, a logical solution would be to integrate a ground distance function into this framework. The ground distance function f(a; b) returns a non-negative real number that defines the dissimilarity between two feature vectors. For example, one may define the substitution cost as the ground distance function between the two features and assign a constant cost for the deletion and insertion operations. Note that d(X;Y) is symmetric if each operation cost is symmetric. That is, δ (a; ε) = δ(ε; a) and δ (a; b) = δ (b; a).

The Levenshtein distance can be easily computed using dynamic programming. While it may not be very efficient, the approach is adaptable to different cost operations. For more efficient computations, automata-based approaches might be applied [16]. We construct a matrix

$D_{[0,\ldots mi0,\ldots,n]}$. The matrix D is computed using the recurrent equation:

$$D(i; j) = \min\{D(i-1, j-1) + \delta(xi; yj), D(i-1; j) + \delta(xi; \varepsilon),$$

$$D(i; j-1) + \delta(\varepsilon; yj)\} \tag{5}$$

where $D_{i,j}$ is the minimum cost of operations needed to match $x_1,\ldots..x_i$ to $y_1,\ldots..,y_j$ and $D_{0,0} = 0$. We return $D_{m,n}$ as the Levenshtein distance. Fig. 5 shows a directed graph for an example with m = 2 and n = 3. In the graph, vertical and horizontal edges are assigned the cost of inserting a feature vector, while diagonal edges are assigned the substitution cost. Note that an insertion in one feature sequence is equivalent to a deletion in the other. Moreover, multiple paths might exist to achieve the minimal cost. The computational complexity is O(*mn*) and the space required is O(min(*m, n*)) since we can

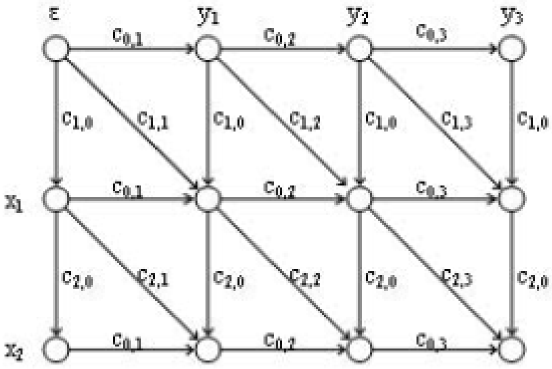
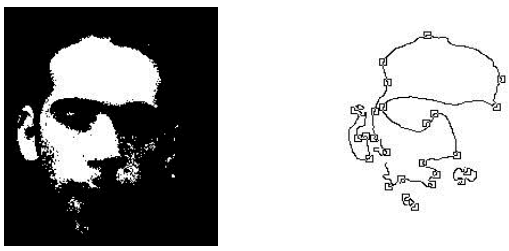


Fig. 5 A dynamic programming graph.

Each node records the total cost $D_{i,j}$ for comparing two feature subsets, and each edge represents the cost of an operation. Here $c_{i,j}$ is the output from the function $\delta(. , .)$ where both horizontal and vertical edges are costs for insertion and deletion, and diagonal edges are costs for substitution.

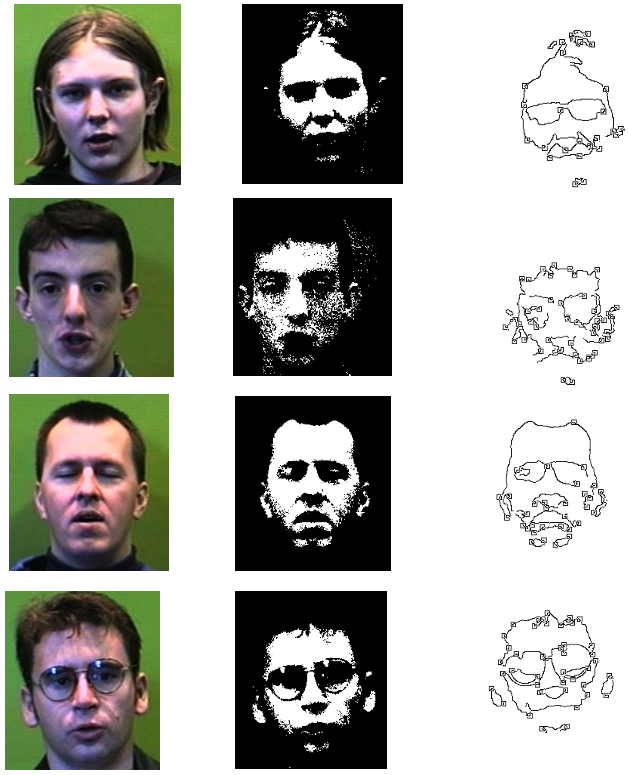
The sequences of operations performed to transform X into Y can be easily recovered from the matrix; but if we choose to do so, we need to store the entire matrix. We can extend this algorithm to search for an alignment between two feature sequences. That is, we want to search for a short sequence X in a long sequence Y. This is useful for comparing objects that are represented by a cyclic sequence of local descriptors. The algorithm is basically the same, but the difference is that we allow any position in Y to be the potential starting point of a match, which is achieved by setting $D_{0,j}$ to zero for all $j \in 0..n$. That is, the empty set matches with zero errors at any position in Y. Also, we return the minimum of $D_{m,j}$ for $j \in 1..n$, which is the cost of the best alignment found by the algorithm, as the Levenshtein distance.



(a)Feature Extracted Image **(b) String Face Image**

Fig. 6 String Formulation

IV. CASE STUDY



(a) Input Image Image **(b) Feature Extracted Image** **(c) String Face**

Fig. 7 Case Studies for Various Input Images

V. CONCLUSIONS

Face recognition influenced by light directions, occlusions, appearance changes, and images of same person taken from different surroundings are carried out and overcome efficiently through our combinational approach of feature face using Chan-vese algorithm and string face matching at low computation costs. The results show that the accuracy of determining the similarities and dissimilarities for frontal face recognition. This method needn't normalize the images before processing. It can also help to improve the accuracy of face recognition through various environments. The future work may relate to overcome the drawbacks of face recognition due to facial expressions and various postures of same person.

REFERENCES

[1] Weiping Chen and Yongsheng Gao, Senior Member, IEEE, “Face Recognition using Ensemble String Matching” IEEE Transactions on Image Processing, Vol. 22, No. 12, Dec. 2013.

[2] J. Rabia, A. R. Hamid, ―A Survey of Face Recognition Techniques, JIPS, Vol. 5, pp. 41-68, 2009.

[3] David G. Lowe, "Distinctive image features from scale-invariant keypoints," International Journal of Computer Vision, Vol. 60, pp. 91-110, 2004.

[4] H. A. Rowley, S. Baluja, and T. Kanade, ―Neural network-based face detection, IEEE trans. Pattern Analysis and Machine Intelligence, Vol. 20, pp. 23–38, 1998.

[5] R. Gottumukkal and V.K. Asari. "An improved face recognition technique based on modular PCA approach." Pattern Recognition Letters, Vol. 25, pp. 429-436, 2004.

[6] M. Sebastian, et al. "Kernel PCA and de-noising in feature spaces." Advances in neural information processing systems, Vol.11, pp. 536-542, 1999.

[7] Wang, Hui-Yuan, and Xiao-Juan Wu. "Weighted PCA space and its application in face recognition," Proceedings of IEEE International Conference on Machine Learning and Cybernetics, Vol. 7, pp. 4522-4527, 2005.

[8] H. Yu and J. Yang, "A direct LDA algorithm for high-dimensional data — with application to face recognition, Pattern Recognition, Vol. 34, pp. 2067–2069, 2001.

[9] Huang, Rui, et al. "Solving the small sample size problem of LDA, Proceedings of 16th IEEE International Conference on Pattern Recognition. Vol. 3, pp. 29-32, 2002.

[10] Bartlett, Marian Stewart, Javier R. Movellan, and Terrence J. Sejnowski. "Face recognition by independent component analysis," IEEE Transactions on Neural Networks, Vol. 13, pp. 1450-1464, 2002.

[11] Ming-Jung Seow, Ann Theja Alex, and Vijayan K. Asari, ―Learning embedded lines of attraction by self organization for pose and expression invariant face recognition, SPIE Journal of Optical Engineering, pp. 1-14, Vol. 51, 2012.

[12] Zheng, Zhong-Long, and Fan Yang. "Enhanced active shape model for facial feature localization", IEEE International Conference on Machine Learning and Cybernetics, Vol. 5, pp. 2841-2845, 2008.

[13] Chen, Yi, et al. "Discriminative Local Sparse Representations for Robust Face Recognition, pp. 1111-1947, 2011.

[14] Bicego, Manuele, et al. "On the use of SIFT features for face authentication," IEEE Computer Vision and Pattern Recognition Workshop, pp. 1-35, 2006.

[15] H. Bay, A. Ess, T. Tuytelaars, Luc Van Gool, "SURF: Speeded Up Robust Features", Computer Vision and Image Understanding (CVIU), Vol. 110, pp. 346-359, 2008.

[16] P. Dreuw, P. Steingrube, H. Hanselmann and H. Ney, ―SURF-Face: Face Recognition Under Viewpoint Consistency Constraints, In A. Cavallaro, S. Prince and D. Alexander, editors, Proceedings of the British Machine Conference, pp. 7.1-7.11, 2009.

[17] N. Dalal and B. Triggs, ―Proceedings of IEEE Conference Computer Vision and Pattern Recognition, pp. 886 - 893, 2005.

[18] O. Déniz, G. Bueno, J. Salido, and F. De la Torre, ―Face recognition using Histograms of Oriented Gradients, Pattern Recognition Letters, Vol. 32, pp. 1598-1603, 2011.

[19] T. Ojala, M. Pietikäinen, and D. Harwood, "Performance evaluation of texture measures with classification based on Kullback discrimination of distributions", Proceedings of the 12th IAPR International Conference on Pattern Recognition, Vol. 1, pp. 582 – 585, 1994.

[20] T. Ahonen, A. Hadid and M. Pietikäinen, ―Face description with local binary patterns : Application to Face Recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence , pp. 469-481, 2006.

[21] A. K. Jain, R. P. W. Duin, and J. Mao, ―Statistical Pattern Recognition: A Review, IEEE Trans. Pattern Anal. Mach. Intell, Vol. 22, pp. 4-37, 2000.

[22] A. W. M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain. Content-based image retrieval at the end of the early years. IEEE Transactions on Pattern Analysis and Machine Intelligence, 22:1349-1380, 2000.

[23] D. G. Lowe. Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision, 60:91-110, 2004.

[24] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. IEEE Transactions on Pattern Analysis and Machine Intelligence, 24:509-522, 2002.

[25] A. Berg, T. Berg, and J. Malik. Shape matching and object recognition using low distortion correspondences. In IEEE International Conference on Computer Vision and Pattern Recognition, pages 26-33, 2005.

[26] K. Grauman and T. Darrel. The pyramid match kernel: Discriminative classi_cation with sets of image features. In IEEE International Conference on Computer Vision, pages 1458-1465, 2005.

[27] K. Grauman and T. Darrell. Fast contour matching using approximate earth mover's distance. In IEEE International Conference on Computer Vision and Pattern Recognition, pages 220-227, 2004.

[28] S. C. Chan and A. K. C. Wong, "Synthesis and recognition ofsequences," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 13, no. 12,pp. 1245–1255, Dec. 1991.

[29] L. Rabiner and B.-H. Juang, Fundamentals of Speech Recognition. Englewood Cliffs, NJ, USA: Prentice-Hall, 1993.

[30] P. Heckel, "A technique for isolating differences between files," Commun.ACM, vol. 21, no. 4, pp. 264–268, 1978.

[31] L. Bahl and F. Jelinek, "Decoding for channels with insertions, deletions,and substitutions with applications to speech recognition," IEEE Trans.Inf. Theory, vol. 21, no. 4, pp. 404–411, Jul. 1975.

[32] W. H. Tsai and S. S. Yu, “Attributed string matching with mergingfor shape recognition,” IEEE Trans. Pattern Anal. Mach. Intell.,vol. PAMI-7, no. 4, pp. 453–462, Jul. 1985.

[33] Y. Gao and M. K. H. Leung, “Human face profile recognition usingattributed string,” Pattern Recognit., vol. 35, no. 2, pp. 353–360, 2002.

[34] S. W. Chen, S. T. Tung, C. Y. Fang, S. Cheng, and A. K. Jain, “Extendedattributed string matching for shape recognition,” Comput. Vis. Image Understand., vol. 70, no. 1, pp. 36–50, 1998.

[35] Mark S. Nixon, Alberto S. Aguado, Feature Extarction and Image Processing, Second Edition, Elsevier, 2008.

[36] I.D. Coope, “Circle fitting by linear and nonlinear least squares "Journal of Optimization Theory and Applications vol. 76, pp. 381-388, 1993.

[37] M.G. Cox and H.M. Jones, “An algorithm for least squares circle fitting to data with specified uncertainty ellipses" IMA Journal of Numerical Analysis vol. 9, pp. 285-298, 1989.

[38] W. Gander, G.H. Golub and R. Strebel, “Least squares fitting of circles and ellipses" Bulletin of the Belgian Mathematical Society vol. 3, pp. 63-84, 1996.