IJRIT
Always Searching for Research

International Journal of Research in Information Technology (IJRIT)

www.ijrit.com **ISSN 2001-5569**

Face Authentication /Recognition System For Forensic Application Using Sketch Based On The Sift Features Approach

Poonam A. Katre
Department of Electronics Engineering KITS, RTMNU Nagpur University, India
Poonamkatre415@gmail.com

ABSTRACT

Recently, various methods have been proposed to address the problem of face sketch recognition by matching face photos and sketches, which are of different modalities. However, their performance is strongly affected by the modality difference between sketches and photos. Faces are highly deformable objects which may easily change their appearance over time. It is concerned with the problem of correctly identifying face images and assigning them to persons in a database. Scale invariant feature transform (SIFT) proposed by Lowe has been widely and successfully applied to object detection and recognition . However, the representation ability of SIFT features in face recognition has rarely been investigated systematically.

Keywords: *GUI, SIFT, keypoint detector, SIFT descriptor, face recognition, Feature Selection, SIFT Features.*

I. INTRODUCTION

Face recognition is extensively used in a wide range of commercial and law enforcement applications. Face recognition has attracted much attention [1] in last decade because of its wide applications. However, face recognition is still an unsolved problem as human face is not rigid object and it can be transformed easily under different situations .Therefore , how to represent the intrinsic attributes of a human face effectively becomes much more important to increase the accuracy of face recognition systems .

Scale invariant feature transform [SIFT][2] proposed by Lowe becomes one of the research interests for pattern recognition because of its excellent performance on object recognition . the SIFT method first detects the local key-points that are notable and stable for images in different resolutions and uses scale and rotation invariant descriptors to represent the key-points. An investigation of SIFT features on face representation has ever done as the first attempt to analyze the SIFT approach in face analysis context[3].

II. FACE RECOGNITION/AUTHENTICATION: STRUCTURE AND PROCEDURE

In this report , we focus on image-based face recognition. Given a picture taken from a digital camera, we'd like to know if there is any person inside, where his/her face locates at, and who he/she is . Toward this goal, we generally separate the face recognition procedure into three steps: face detection, feature extraction, and face recognition (shown at fig 1).

Figure 1. Configuration of a general face recognition structure

A . Face Detection

The main function of this step is to determine (1) whether human faces appear in a given image , and (2) where these faces are located at. The expected outputs of this step are patches containing each face in the output image . In order to make further face recognition system more robust and easy to design, face alignment are performed to justify the scales and orientations of these patches.

B. Feature Extraction

After the face detection step , human-face patches are extracted from images. Directly using these patches for face recognition have some disadvantages, first, each patch usually contains over 1000 pixels, which are too large to build a robust recognition system [4]. Second , face patches may be taken from different camera alignments ,with different face expressions, illuminations, and may suffer from occlusion and clutter . To overcome these drawbacks, feature extractions are perfomed to do information packing, dimension reduction , salience extraction, and noise cleaning.

C. Face Recognition

After formulizing the representation of each face, the last step is to recognize the face. There are two general applications of face recognition , one is called identification and another is called verification. Face identification means given a face image, we want the system to tell who he/she is or the most probable identification ; while in face verification , given a face image and a guess of the identification , we want the system to tell true or false about the guess.

III. RELATED WORK

Authenticate a face, the SIFT features computed in the test image should be matched with the SIFT features of the template. In this section different matching methodologies are investigated. The different methodologies[1] could employ all or only apart of the whole information included in the sift feature.

A. Scale Invariant Feature Transform

SIFT is an algorithm that has the ability to detect and describe local features in images. It was describe by David Lowe in [9]. The features are invariant to image scaling, translation and rotation. It was originally developed for matching an object in images with different views of object . The first step of the detection is determination of extrema in the image filtered by the Difference Of Guassian (DoG) filter . The filtering is performed in several scales. After this step, the best "points" are identified . Only points with high enough contrast are used. An orientation is assigned to each of these points. The resulting set of points is then use for creation of feature vectors (descriptors). Each descriptor contains a vector of the length 128 and also its coordinates .

B. SIFT Feature Extraction

The SIFT algorithm has basically four steps : extrema detection ,removal of key-points with low contrast, orientation assignment and descriptor calculation [5] .

To determine the key-point location, an image pyramid with re-sampling between each level is created. It ensures the scale invariance. Each pixel is compared with its neighbours . Neighbors in its level as well as in the two neighbouring (lower and higher) level are examined. If the pixel is maximum or minimum of all the neighbouring pixels, it is considered to be a potential key-points.

For the resulting set of key-points their stability is determined Location with low contrast and unstable location along edges are discarded. Further, the orientation of each key-point is computed. The computation is based upon gradiant orientations in the neighbourhood of the pixel. The values are weighted by the magnitudes of the gradiant.

The final step is the creation of the descriptors. The computation involves the 16×16 neighbourhood of the pixel. Gradiant magnitudes and orientations are computed in each point of the neighbourhood . their values are weighted by a Gaussian . for each sub-region of size 4×4 (16 regions), the orientation histograms are created. Finally, a vector containing $128\ (16 \times 8)$ value is created. The algorithm is described in [6,7] and [5].

IV. DETAILS OF OUR METHOD

Our method mainly consist of the following three steps:

A Choose N "candidates" based on the Euclidean distance.

We define $X = \{x_{11}, \cdots, x_{ij}, \cdots\}$ as a test sample , $Y = \{y_{11}, \cdots y_{ij}, \cdots\}$ as a training sample, where x_{ij} and y_{ij} are the pixels at position (i,j) of the test and training samples respectively. Let $B = \{Y_1, Y_2, \ldots\}$ denote the training set. Let $D(X, Y_k)$ be the Euclidean distance between a test sample X and a training sample Y_k.

$$D(X,Y_k)\sqrt{\sum_{i=1,j=1}^{i=H,j=W}(x_{ij}-y_{ij})^2}$$

(1)

H and W are the height and width of a test sample respectively. The training samples should have the same size as the test sample . We sort the whole training samples in the ascending order of Euclidean distance , and then choose N training samples with smaller distance as "candidates" set C.

$$C = \{(Y_{1\ldots}Y_N) | D(X,Y_1) \leq D(X,Y_2) \leq \ldots \leq D(X,Y_N) \leq D(X,Y_{N+1})\}$$

(2) Where $N \leq n$, n is the number of training samples in training samples in training set B.

B. Choose P new "candidates" based on SIFT features.

In this step, we choose P new "candidates" from C based on the number of well matched pairs of SIFT features. First of all, we define the criterion of well matched pair of SIFT features. We build a KD-tree [8] using the descriptors of sift features in a training sample. And then, for each descriptor a in the test sample , we employ best-bin –first search algorithm to find out k nearest nodes $b_1, b_2, \ldots b_k$ in the KD-tree (usually k=2), which are sorted in descending order. Let d_1, d_2 respectively be the distances between a and b_1, a and b_2. we then calculate the ratio of d_1, d_2:

$$ratio = {d_1}/{d_2}$$

(3)

If ratio < threshold (defined manually), we define *a* and b_1 are well matched pair of SIFT features. Fig 1.1 shows the effect of *threshold* on the recognition accuracy. When threshold is below a certain value, the recognition accuracy increases rapidly and reaches the highest while *threshold* is 0.5. Thus, we fix it as 0.5 in our method.

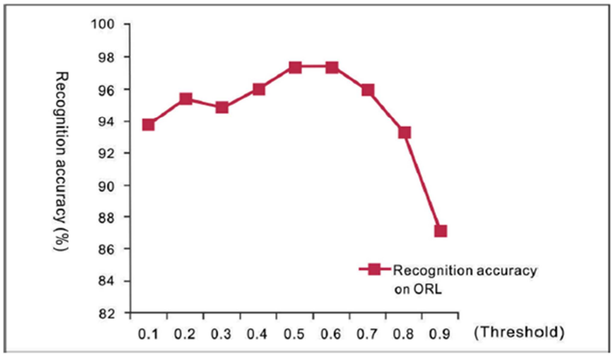


Figure 1.1: Recognition accuracy with respect to varying threshold.

We take some images of a subject from FERET face database for database for example to show the.Well matched pairs of two images , as shown in fig. 1.2.

In this step, we count the number of well matched pairs of SIFT features between the test sample and each "candidate" in *C*, and sort the candidate samples in descending order. Then we choose P samples from C with the greater number of well matched pairs as the new "candidates" set $C' = \{C'_1, C'_2, \cdots, C'_P\}$.

Figure 1.2 : An example of matched pairs of SIFT features .

C. Calculate the average similarity between classes.

Let M_i be a class with the same label as new "candidates" $c'_i \in C'$, e_{ij} the angle of SIFT feature descriptors between a test sample and the j-th training sample in M_i, and $\bar{e}_i$ be the average angle between the test sample and each class M_i. we calculate e_{ij} is:

$$e_{ij} = \cos^{-1}\left(\frac{f_{test} \cdot f_{ij}}{\|f_{test}\| \cdot \|f_{ij}\|}\right) \quad (4)$$

$$\bar{e}_i = \frac{\sum_{j=1}^{L} e_{ij}}{L} \quad (5)$$

Where f_{test} and f_{ij} are the descriptors of the test sample and the j-th training samtple in M_i respectively, and L is the number of training samples in M_i. In our method , we describe the similarity of a test sample and the class M_i by using $\bar{e}_i$ and choose the class with the minimum angle as the recognition result .

To get high recognition accuracy, we adjust some parameters involved in our method by using a small validation set. The parameters mainly include the numbers *N,P* of "candidates" chosen in the first two steps and the *threshold* used to count the number of well matched pair of SIFT.

V. PARAMETERS

A. *Mean Squared Error (MSE)*

PSNR is most easily defined via the mean squared error (*MSE*). Given a noise-free $m \times n$ monochrome image I and its noisy approximation K, *MSE* is defined as:

$$MSE = \frac{1}{m\,n}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[I(i,j) - K(i,j)]^2$$

B. *Peak Signal to Noise Ratio (PSNR)*

Peak signal to noise ratio is the ratio between the maximum possible power of a signal and the corrupting noise signal. It is given by Peak signal-to-noise ratio, often abbreviated PSNR, is an engineering term for the ratio between the maximum possible power of a signal and the power of corrupting noise that affects the fidelity of its representation. Because many signals have a very wide dynamic range, PSNR is usually expressed in terms of the logarithmic decibel scale.

PSNR is most commonly used to measure the quality of reconstruction of lossy compression codecs. The signal in this case is the original data, and the noise is the error introduced by compression. When comparing compression codecs, PSNR is an approximation to human perception of reconstruction quality. Although a higher PSNR generally indicates that the reconstruction is of higher quality, in some cases it may not. One has to be extremely careful with the range of validity of this metric; it is only conclusively valid when it is used to compare results from the same codec (or codec type) and same content.

The PSNR is defined as:

$$\begin{aligned} PSNR &= 10 \cdot \log_{10}\left(\frac{MAX_I^2}{MSE}\right) \\ &= 20 \cdot \log_{10}\left(\frac{MAX_I}{\sqrt{MSE}}\right) \\ &= 20 \cdot \log_{10}(MAX_I) - 10 \cdot \log_{10}(MSE) \end{aligned}$$

C. *Signal to Noise Ratio (SNR)*

Signal to noise ratio is the ratio between original signal and noise. It is given by,

$$SNR = \frac{10\log_{10}\sum_{i-1}^{M}\left(S(i,j)^2\right)}{\sum_{i-1}^{M}\sum_{j-1}^{N}\left[S(i,j) - F(i,j)\right]^2}$$

D. *Entropy (EN)*

Entropy evaluated the information quality which is contained an image. If the value of entropy becomes grater after the fusion process, it indicates that information of image increases and performance of the fusion is improved. It is given by,

$$ENTROPY = -\sum_{l-0}^{L-1} P_l \log_2 P_l$$

VI. GRAPHIC USER INTERFACE (GUI)

Graphic User Interface (GUI) is a program interface item that allows people to interact with the programs in more ways than just typing commands. It offers graphical icons, and a visual indicator, as opposed to text-based interfaces, typed command labels, or text navigation, to fully represent the information and actions available to users. The GUI is introduced in reaction to the steep learning curve of command-line interfaces.

During analysis of face recognition we used a modified software Face Recognition System 2.1 [13] working in MATLAB environment. This program uses an algorithm based on PCA (principal component analysis), called also eigenfaces (eigenvectors determined by PCA are called eigenfaces, when the PCA is used to analyze the face image) [9, 10]. Face recognition is based on the distance from the nearest class, according to the numbering assigned at the beginning to individual photographs (indicating a person in the class).

Our software is equipped with GUI (*graphical user interface*) (Fig. given below) and allows for operation in two modes: continuous and batch processing [12]. In the first mode (continuous processing) we can acquire the image from an IP wireless webcam (e.g. D-Link DSC-930L [11]) or a standard USB camera and recognize face belonging to the person, which is in front of the camera.

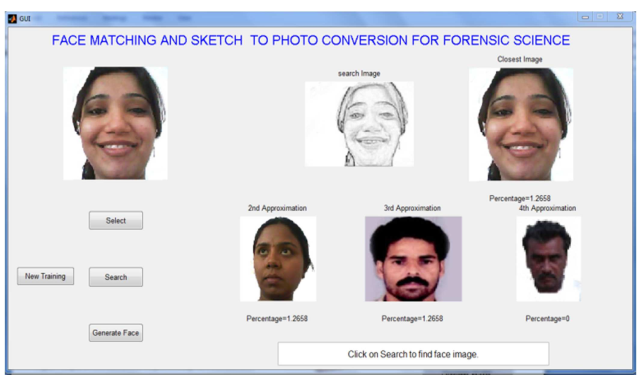


Figure 1: Graphical User Interface developed for image face authentication application(in the case of original image).

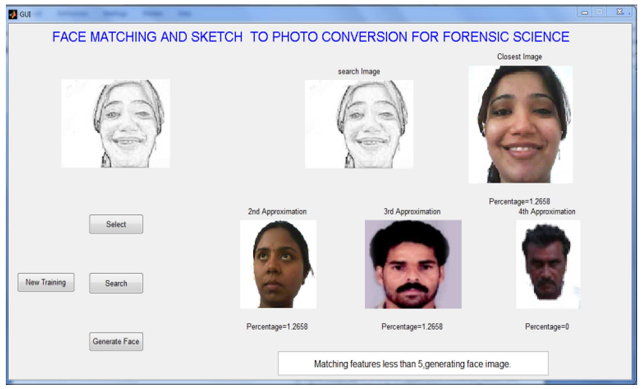


Figure 2: Graphical User Interface developed for image face authentication application(in the case of sketch image).

VII. RESULT AND ANALYSIS

The procedure of our method can be shown as:

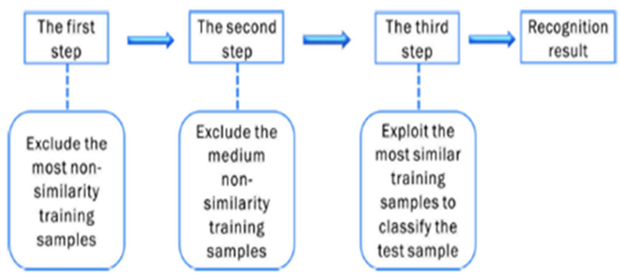


Figure 1: The General Procedure Of Our Method.

In this section, we present our experiments and our inferences based on the results we obtain. Here face recognition algorithms are applied. The first experiment is face detection algorithm based.

The source image and the result by different algorithm are shown from fig.2 to fig.5. Fig. 2(a), fig 3(a) and fig.4(a) are the original test image. Whereas, fig.5(a) is the original test sketch image. Fig. 2(b) shows the detected face image, fig .3(b) shows the extracted face image feature vector, whereas fig. 4(b) and fig.5(b)shows the generated face.

2(a). Input Image 2(b). Output Image

Fig.2 Face Detection

Fig 3(a). Input Image

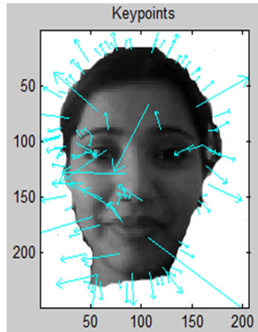


Fig 3(b). Output Image

Fig.3 Feature Extraction

Fig 4(a). Input Image

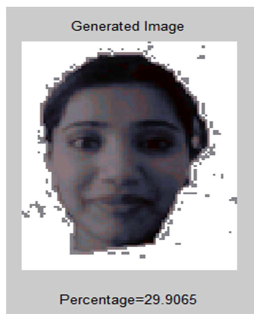


Fig 4(b). Output Image

Fig 4. Face Generation(original image)

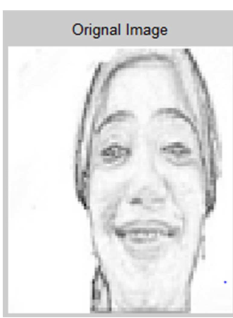


Fig 5(a). Input Image

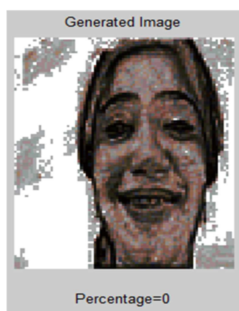


Fig 5(b). Output Image

Fig 5. Face Generation(sketch image)

The SIFT features have been computed with Lowe's c ode1. Both the three matching methodologies have been tested: accuracies of authentication are proposed in Table 1. In particular, Prior Equal Error Rates for G1 and G2 are presented in Table 1 (the corresponding ROC curves are shown in Fig. 5.

. The error rate was computed using the following procedure.

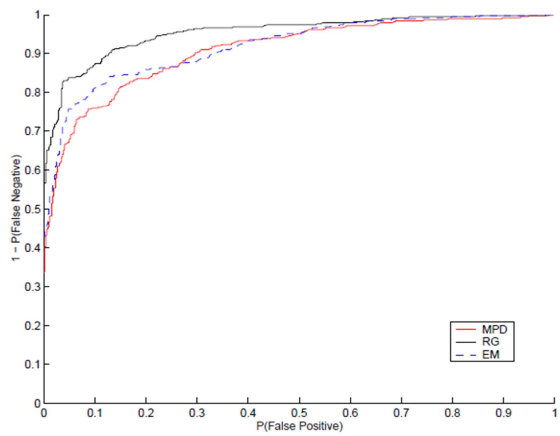
1. Perform the experiment on G1, getting G1 scores
2. Perform the experiment on G2, getting G2 scores

3. Compute the ROC curve using G1 scores, determine the Prior Equal Error Rate and the corresponding threshold θ_{G1}.

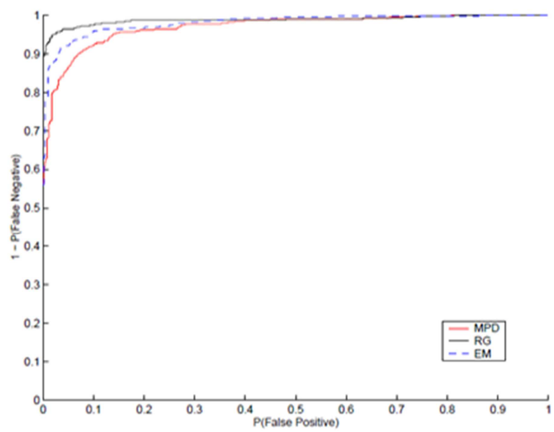
	MPD	EM	RG
Prior EER on G1	17.15%	15.38%	11.31%
Prior EER on G2	8.69%	6.38%	3.85%
Average	12.92%	10.88%	7.58%

TABLE I. PRIOR EER ON G1 AND G2 FOR THE THREE METHODS ;

'MPD'STAND FOR MINIMUM PAIR DISTANCE, 'EM' FOR EYES AND MOUTH , 'RG'FOR REGULAR GRID



(G1)



(G2)

Fig 5. Face Generation(sketch image) ROC curves for G1 and G2 .

VIII. CONCLUSION

In the paper, after giving the definitions for face recognition approach; a SIFT algorithm has been suggested to select the most useful features of the face. Recently, SIFT features and their distances have been used in [2] for the face recognition problem. For our future work, we are planning to apply the genetic algorithm on a number of interest points of some faces and determine the best features for face. Then using only these selected features, same tests as in [2] will be done for performance and accuracy analysis. There will be a scope for a problem of face authentication in multifaces image.

IX. REFERENCES

[1] W. Y. Zhao , R. Chellappa , P.J. Philips , and A. Rosenfeld," Face Recognition: A Literature Survey ," *ACM Computing Survey* ,PP. 399-458, 2003.

[2] David G. Lowe, Distinctive image features from scale-invariant keypoints , *International Journal of Computer Vision,* 60, 2 (2004), pp. 91-110.

[3] M. Bicego, A. Lasario, E. Grosso, and M. Tistarelli, "On the use of SIFT features for face authentication ," *Pr. of IEEE Int Workshop on Biometrics, in association with CVPR*, NY, 2006.

[4] R. Chellappa, C. L. Wilson, and S. Sirohey , "Human and machine recognition of faces : a survey ," PROC. IEEE, vol. 83, no. 5, pp. 705-740, 1995.

[5] Krizaj, J., Struc, V ., Pavesic , N .: Adaptation of sift features for robust face recognition (2010).

[6] Lowe, D.G.: Object recognition from local scale-invariant features. In : International Conference on Computer Vision (1999).

[7] Lowe, D.G.: Dinstinctive image features from scale-invariant keypoints. International Journal of Computer Vision 2 (2004).

[8] Rob Hess, 2011, http ://blogs . oregonstate . edu/hess/.

[9] Agarwal M, Agrawal H, Jain N, Kumar M (2010) Face Recognition using principle component analysis, eigenface and neural network. In: International conference on signal acquisition and processing, 2010. ICSAP '10, pp 310–314.

[10] Belhumeur PN, Hespanha JP, Kriegman DJ (1997) Eigenfaces vs. fisherfaces: recognition using class specific linear projection. IEEE Trans Pattern Anal Mach Intell 19(7):711–720.

[11] Georghiades AS, Belhumeur PN, Kriegman DJ (2001) From few to many: illumination cone models for face recognition under variable lighting and pose. IEEE Trans Pattern Anal Mach Intell 23(6):643–660.

[12] Rzepecki Sz (2011) Real-time localization and identification of faces in video sequences, (M.Sc. Thesis), Supervisor: Marciniak T, Poznan University of Technology.

[13] Rosa L (2013) Face Recognition System 2.1. http://www.advancedsourcecode.com//face.asp.