Integrating the Face Verification Algorithm Into the Smart Card System

HYUNG-KEUN JEE, KYUNG-HEE LEE, YONG-WHA CHUNG
Biometrics Technology Research Team
Electronics and Telecommunications Research Institute
161 Gajung-Dong, Yusung-Gu, Daejeon
KOREA
hkjee@etri.re.kr http://www.etri.re.kr

Abstract: - Using a biometrics to authenticate a person's identity has several advantages over the present practices of Personal Identification Number stored in smart cards. However, there is an open issue of integrating biometrics into the smart cards. Typical verification algorithms by using biometrics may not be executed in real-time on the resource-constrained smart cards. In this paper, we propose a real-time automatic face verification system using Support Vector Machine(SVM) and Principal Component Analysis(PCA) to overcome this open issue. In our system, face detection and feature extraction steps which require relatively high computing power are performed in the host. In the card, not only the user's feature vector is stored but also the verification step is performed without any data leakage. Based on our performance analysis, the smart card can be designed such that the face verification algorithm can be executed in real-time.

Key-Words: - Face Verification, Face Detection, Smart Card

1 Introduction

Nowadays, many systems need a portable media to store some sensible data such as **smart card** [1]. The smart card generally refers to a credit card-sized plastic card with the chip that holds a microprocessor and a data-storage unit. Usually, the cards use Personal Identification Number(PIN) to protect the information inside the card. In the smart cards, the PIN is verified inside the card to be unable to be read from the outside of the card. However, PIN can be copied by inspecting the cardholder's movements as he or she enters his or her number in and also forgotten.

In recent years, there is an increasing trend of using **biometrics** information, which refers the human biological features used for user verification, such as fingerprint, iris, and face, to strengthen the security level of different electronic/embedded systems, including smart card systems.

However, most of these card systems just store the user's biometric template inside the card, and the verification step is performed outside of the card. In this case, there is a risk of duplication of critical user's template data when it is released into the external host from the card to be compared with the input template.

To prevent the risk, the smart card needs a capability of not only storing the user's template, but also performing the whole verification step inside the card without any data leaking out [2,3]. It is called as Match-On-Card. Unfortunately, the memory size and the processing power of the in-card processor are very limited. Thus, a light-weighted verification algorithm which requires small memory and processing power needs to be developed.

In this paper, we propose the face verification Match-On-Card system using Support Vector Machine (SVM) and Principal Component Analysis (PCA). We employ PCA method to represent a face image as a feature vector of reasonably low dimension and high discriminating power. The memory requirement can be decreased significantly by the method. Then, it is essential to develop robust and efficient algorithms detecting human faces. In fact, this is especially important in the holistic approach such as PCA method, in order to build a fully automated system that analyzes information of human faces. Thus, we apply Edge information and SVM to detect face accurately. The experimental results show that the face verification algorithm can be executed by the in-card processor in real-time.

The organization of the paper is as follows. Overview of the smart card system and the face recognition techniques are given in Section 2. In

Section 3, the proposed verification system is explained. Experimental results are shown in Section 4, and Section 5 concludes the paper.

2 Background

2.1 Smart Card System

A smart card resembles a credit card in terms of physical look and size with one or more semiconductor devices attached to a module embedded in the card. More specifically, the smart card is a portable, very secure, low cost, intelligent device, capable of manipulating and storing data. This intelligence is due to an in-card processor that is suitable for use in a wide range of applications.

Fig. 1 shows the smart card system we are developing [11]. The in-card processor is a 32-bit ARM7TDMI. The memory in the smart card consists of three different types, which are 64KB ROM, 8KB RAM and 34KB EEPROM.

The smart card also includes the Crypto-Coprocessor and the Random Number Generator (RNG) to perform cryptographic algorithms in real-time.

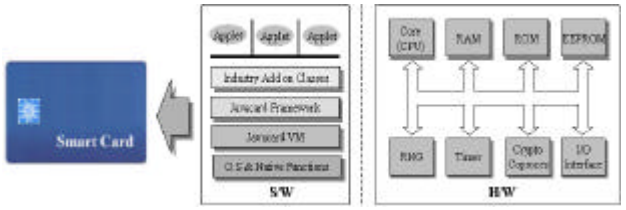

Fig. 1 Targeted Smart Card System

2.2 Face Recognition

Face is one of the most acceptable biometrics because it is one of the most common methods of identification which humans use in their visual interactions and acquiring face images is non-intrusive [4]. However, it is difficult to develop an automatic face recognition system, because face images can vary considerably in terms of facial expressions, 3D orientation, lighting conditions, hair styles, and so on.

Automatic recognition of human faces by computer has been approached in two ways: holistic and analytic [5]. The holistic approach treats a face as 2D pattern of intensity variation. The analytic approach recognizes a face using the geometrical measurements taken among facial features, such as eyes and mouth.

The representative method of holistic approach is Principal Component Analysis (PCA). The basic idea is to construct a new space that can represent a face image as a feature vector of reasonably low dimension and high discriminating power.

The face in the input image must be normalized to a standard size, location and orientation before it is matched. If both eyes on the face image are detected, the face can be easily normalized [5]. We detect both eyes using edge detection algorithm, and apply Support Vector Machine (SVM) for verification of candidate face region with accuracy.

3 Proposed Method

Our face verification system is composed of two part, host and smart card as shown in Fig. 2. The host program performs preprocessing, face detection, and feature extraction in the input image from the camera. The smart card executes registration and verification steps.

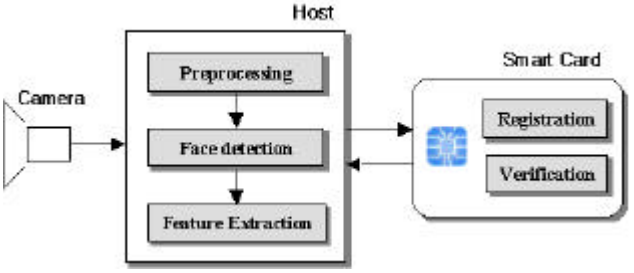

Fig. 2 Our Face Verification System using Smart Card

In case of registration, the host detects face and extracts feature vector from the input image. The extracted feature vector is stored in the smart card. When a user wants to authenticate himself, the same step is executed to extract the feature vector from the new image. Then, the smart card computes the similarity between this input feature vector and the reference feature vector stored in the smart card. Finally, similarity measure is compared to a decision threshold. The verification result -the person is accepted or not- is transmitted to the host. Fig. 3 shows the host program interface.

Fig. 3 Host program interface

3.1 Face Detection

If both eyes on the face image are detected, the size, location and the image-plane rotation of the face can be easily normalized using the positions of the both eyes [5]. Thus, we propose a face detection algorithm that detect both eyes in the input image, and then extract face region using them. First, we apply the Sobel edge detector and a labeling algorithm to the original intensity image in order to find isolated components. Then, the eye size rules are applied to detect several candidates of eye component, which will be verified by using the information of location. One eye pair location is decided using template matching.

The detected locations of both eyes, left and right, are used as reference points to the location of human face area, which will then be used as inputs to registration or verification. Using the center points of both eyes, this system automatically produces a standardized image by rotating and scaling such that only the important facial features are included. To eliminate problems due to variability in the background, a mask is applied to each of the standardized images.

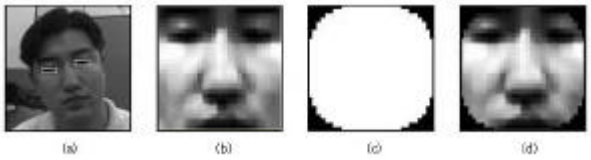

Fig. 4 Eye detection procedure (a) Eye detection in the input image, (b) Standardized face, (c) 32×32 mask, (d) Standardized face with mask

However, false face region can be extracted since eyebrow or hair can be recognized as eyes by mistake. Thus, we finally verify candidate of face region using SVM, which is widely used in pattern recognition field [6].

SVM is a binary classification method that finds the optimal linear decision surface based on the concept of structural risk minimization. The decision surface is a weighted combination of elements of the training set. These elements are called *support vectors* and characterize the boundary between the two classes [7].

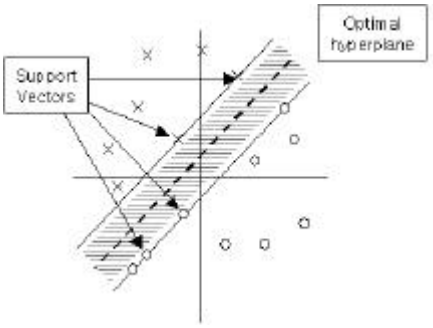

Fig. 5 Example of linear separable optimal hyperplane

The image data is preprocessed so that each pixel is normalized to a [-1, +1] range before training. Training the SVM for the face region verification task is challenging because of the difficulty in characterizing prototypical "non-face" images [8]. It is easy to get a representative sample of images which contains faces, but much harder to get a representative sample of those which do not.

We initially use images normalized by false detected eye pair as non-face images. Examples of non-face images used in SVM' training are shown in Fig. 6.

Fig. 6 Examples of non-face training set obtained by false detection of eyes

The images are collected during training by "bootstrap" manner [9]. The training set is consist of 150 face images and 150 non-face images of size 32 × 32, assigned to classes +1, and −1, respectively. The SVM uses Radial Basis Function(RBF) as kernel function. It is trained to produce an output of positive value if a face is present, and negative value otherwise.

3.2 Feature Extraction: Eigenfaces

Images of faces, being similar in overall configuration, will not be randomly distributed in this huge image space and thus can be described by a relatively low dimensional subspace. The main idea of the PCA is to find the vectors which best account for the distribution of the face images within the entire image space. PCA generates a new orthonormal basis vectors for the image space, where each component is not correlated with any other component. These vectors define the subspace of face images, which we call "face space" [10].

Let a face image be a N^2-dimensional feature vector of intensity values, where N^2 is the number of pixels of the image, and the train set of face images be $\Gamma_1, \Gamma_2, \Gamma_3, ..., \Gamma_M$. The average face of the set is defined by

$$\Psi = \frac{1}{M}\sum_{n=1}^{M}\Gamma_n \qquad (1)$$

and the average face is subtracted to all the vectors.

$$\Phi_i = \Gamma_i - \Psi \quad (I=1, 2, \ldots, M) \qquad (2)$$

The covariance matrix C is

$$C = \frac{1}{M}\sum_{n=1}^{M}\Phi_n\Phi_n^T = AA^T \qquad (3)$$

where the matrix $A = [\Phi_1\Phi_2 ... \Phi_M]$. The eigenvalus of the covariance matrix are computed, and k eigenvectors associated to the k ($k<<N^2$) greatest eigenvalues are selected. Each of these eigenvectors of the covariance matrix is called "eigenface", because it is face-like in appearance. The memory requirement can be decreased significantly by the method. When new face image is presented for registration or verification, its feature vector is extracted by projecting it into the "face space".

3.3 Face Verification

The verification step to determine whether a person should be granted admission or not should be performed in real time. For this purpose, a Euclidian distance between input feature vector and reference feature vector is computed because it is computationally simple. This distance measure is then compared to a decision threshold. If the distance is less than the threshold, the person is accepted. If the distance is greater than the threshold, the person is rejected.

Thus, the threshold acts as a tuning parameter that can be used to adjust the security level of the system. A low threshold will provide tighter security because it will require a closer match between the new image and the reference. However, it will also result in more false rejections.

4 Experimental Results

To evaluate the proposed method, two sets of experiments were conducted. First, we analyzed the performance of face detection. Then, we validated that the verification algorithm can be executed with the in-card processor in real-time.

In order to evaluate the accuracy of the face region verifier using SVM, we used the test set of 200 face images and 200 non-face images of a size of 32×32 made by ETRI. According to the result of the experiment, it was confirmed that the verifier reduced extraction error rate significantly.

Table 1. Results of face region verification using SVM

	FRR	FAR
SVM verifier	1.5%	1%

As we mentioned before, we used the simple Euclidian distance classifier for face verification. In order to analyze the memory requirement and time complexity, in detail we simulated the verification step with the iSAVE (in-System Algorithm Verification Engine)™ [12]. As shown in the Table 2 the result of simulation, the required working memory space of the verification algorithm is about 6KB, and the execution time on ARM7 is about 0.3 second. Therefore our proposed method makes it applicable to resource-constrained smart card systems.

Table 2. Resource requirements of our face verification system

Measurement Item	Results
Template Size	160 Byte
Verification Time	0.3 sec
Required Memory Size	6 KB

5 Concluding Remarks

Smart card is a model of very secure storage, and biometrics is the ultimate technology for authentication. These two can be combined in many applications to enhance the security further. Howerer, a careful design is required to integrate the biometrics into the smart card because the smart cards have very limited resources.

This paper proposed a real-time face verification system based on SVM and PCA for a smart card environment, which enhances security level as compared with the traditional PIN verification method currently being used. The system encapsulates the biometric data into the smart card and performs the whole verification process inside the card. Therefore, the user's template is never released out of the card, avoiding duplication of such a sensible data.

As the result of simulation, the required working memory space of our face verification algorithm is about 6KB, and the execution time on ARM7 is about 0.3 second. Therefore, our proposed method makes it applicable to the resource-constrained smart card systems.

To accomplish more reliable system, we will consider a multi-modal biometric identification system employing multiple biometrics into the smart card.

References:

[1] C. Mearns and D. Jones, *The Smart Card*. SJB Research, 1999.

[2] R. Sanchez-Reillo, A. Gonzalez-Marcos, Access Control System with Hand Geometry Verification and Smart Cards, *IEEE Aerospace and Electronics System Magazine*, Vol. 15, 2000, pp. 45-48.

[3] R. Sanchez-Reillo, Including Biometric Authentication in a Smart Card Operating System, *Proc. of AVBPA*, 2001, pp. 342-347.

[4] R. Chellappa, C. Wilson, and S. Sirohey, Human and Machine Recognition of Faces: A Survey, *Proc. of the IEEE*, Vol. 83, No. 5, 1995.

[5] R. Brunelli and T. Poggio, Face Recognition: Features versus Templates, *IEEE Trans. on PAMI*, Vol. 15, No. 10, 1993, pp. 1042-1052.

[6] V. Vapnik, *The Statistical Learning Theory*, John Wiley & Sons, New York, 1995.

[7] C. Cortes and V. Vapnik, Support Vector Networks, *Machine Learning*, Vol. 20, 1995, pp. 1-25.

[8] H. Rowley and T. Kanade, Neural Network- Based Face Detection, *IEEE Trans. on PAMI*, Vol. 20, No. 1, 1998.

[9] K. Sung, *Learning and Example Selection for Object and Pattern Detection*, PhD thesis, MIT AI Lab, 1994.

[10] M. Turk and A. Pentland, Face Recognition Using Eigenfaces, *Proc. of CVPR*, 1991, pp. 586-591.

[11] H. Kim, et al, Specification for the Next-Generation IC Card System(Korean), *Technical Report*, ETRI, 2000.

[12] iSAVE, http://www.dynalith.com.