A Vein Map Biometric System

Felix Fuentes
Probado Technologies Corporation
5350 S. Staples, Suite 103
Corpus Christi, TX 78411, USA

and

Dulal C. Kar
Department of Computing Sciences
Texas A&M University-Corpus Christi
6300 Ocean Dr.
Corpus Christi, TX 78412, USA

ABSTRACT

There is increasing demand world-wide, from government agencies and the private sector for cutting-edge biometric security technology that is difficult to breach but user-friendly at the same time. Some of the older tools, such as fingerprint, retina and iris scanning, and facial recognition software have all been found to have flaws and often viewed negatively because of many cultural and hygienic issues associated with them. Comparatively, mapping veins as a human barcode, a new technology, has many advantages over older technologies. Specifically, reproducing a three-dimensional model of a human vein system is impossible to replicate. Vein map technology is distinctive because of its state-of-the-art sensors are only able to recognize vein patterns if hemoglobin is actively flowing through the person's veins. Additionally, each individual's vein map is unique, even in the case of identical twins. The combinations of these factors provide vein map authentication an edge over existing biometric identification products. In this work, we present a vein-map based biometric authentication system using Fujitsu's newly released vein map scanner. The prototype system has been successfully developed and tested for Microsoft Windows environment.

Keywords: Biometric authentication, vein mapping, fingerprint identification, Rijndael algorithm, retina scanning.

1. INTRODUCTION

Biometric methods are used to recognize a person based on a physiological or behavioral characteristic of the person. Among the features measured are: face, fingerprints, hand geometry, handwriting, iris, retina, vein, and voice [1]-[3],[8],[9]. Biometric technologies are becoming the foundation of an extensive array of highly secure identification and personnel verification solutions. As the level of security breaches and transaction fraud increases, the need for highly secure identification and personnel verification technologies is becoming apparent.

Many companies now employ biometrics. As opposed to using badges, sign-ins or other ways of tracking employees, a biometric time clock assures that no employee can punch in for another, eliminating time fraud and reducing payroll costs. Because every person's biometric characteristic is unique, a biometric time clock provides a quick, accurate, and reliable way to record in- and out-punches for each employee. To date, popular biometric authentication systems include, fingerprint identification, retina and iris scan, face recognition, and voice analysis [9]. The problem with each system is that most can be breached easily or intrude upon the rights of individuals or both [6], [8].

Fingerprint identification is one of the most well-known and publicized biometrics. However, experiments performed over time have revealed that fingerprint scanning could be easily tricked [6]. Also, gathering of fingerprints is associated with criminal behavior in the minds of people and is rejected by many. Iris recognition is the process of recognizing a person by analyzing the random pattern of the iris [2]. Such a complex system as the iris has also been proven to fail under experimentation [6]. One such experiment authenticated an unauthorized person into the system by simply holding an ink-jet print-out over their eye. The page was a printout of an authentic iris with a small hole cut into the page through which the pupil of the imposter was visible to the camera [6]. People often use faces to recognize individuals and advancements in computing capability over the past few decades now enable similar recognitions automatically. Face recognition can be used for both verification and identification. Like fingerprint scanning and iris recognition previously mentioned, facial recognition methods have also been proven to have some flaws under experimentation [6]. These methods are sensitive to whether a subject is wearing eyeglasses or not, the lighting condition is proper or not , the subject is complete still or not, or the angle of the facial image relative to the original image is within some limit or not, and so on [5], [7], [10]. Also, there are some issues in biometrics used today that range from hygiene, cultural, social, and religious issues, depending on where the biometric technology is being deployed.

On the contrary to biometric authentication technology available today, a system based on the vein pattern in the palm of a human hand has been developed. The system is essentially a sensor, much like a fingerprint scanner where the user is required to hold their finger over the scanner. However, with this newly developed vein map authentication technology, the user never actually makes contact with the scanner. The sensor can only recognize the vein map if blood is actively flowing the individual's veins. Processing is not affected by race, skin discoloration, hair, age, or time. As veins are internal to the body it is extremely difficult to forge the vein pattern of someone else to gain authentication into the system, thereby enabling a very high level of security. The technology is relatively inexpensive, highly accurate, good in response time and small in size. It does not have any apparent hygienic issue and it is friendly to use because of its contact-less and non-intrusive technology.

In this work, we present a biometric authentication system using vein maps and attempts to persuade the audience of its significance in the biometric world and highlight key points that give vein map authentication an edge over other biometric systems on the market. We accomplish this by implementing a rich user interface using the Visual Basic programming language interacting with Fujitsu's C++ library to control a vein map scanner prototype. The scope of the work focuses more on the ability of Fujitsu's scanner and API (Application Programming Interface) to extract the vein map pattern from various individuals' palms and the retrieval of the vein map data from a data repository for authentication, as opposed to analyzing the algorithms used for extraction and identification in the C++ library.

In section 2, we describe the vein map scanner manufactured by Fujitsu Corporation. In sections 3, 4, and 5 we present the system architecture and its development and processing details. Section 6 deals with discussions on testing and security of the system. Finally, section 7 concludes the paper with some direction to future work.

2. THE FUJITSU VEIN MAP SCANNER

The Fujitsu scanner works by capturing a person's vein pattern image while radiating it with near infrared rays. The deoxidized hemoglobin in the palm vein absorbs these rays, thereby reducing the reflection rate and causing the veins to appear as a black pattern as shown in Fig. 1. The vein pattern is then verified against a pre-registered pattern to authenticate an individual. Fujitsu's proprietary algorithm takes into account identifying features such as the number of veins, their position, and the points in which they cross. Internal research by Fujitsu resulted in a false acceptance rate of less than 0.00007% and a false rejection rate of only 0.00004%. False acceptance rate is a rate at which someone other than the actual person is falsely recognized. False rejection rate is a rate at which the actual person is not recognized accurately. [4]

3. SYSTEM ARCHITECTURE

The hardware components used for this work include a vein map scanner, a camera, and an electromechanical lock that can be opened or closed through a hardware interface to a computer. The vein map authentication software application uses a number of dynamically linked library files owned by Fujitsu Corporation. These DLLs are invoked from the proposed Visual Basic application to control a prototype of the vein map scanner provided by Fujitsu Corporation. The application is written using the latest version of Microsoft's Visual Basic release; therefore the production machine requires Visual Studio Enterprise Edition to be installed on it. After the vein map pattern is extracted from the palm, the DLLs return an array of bytes un-encrypted. The application then encrypts the returned data using one of the encryption classes within Visual Studio environment. The encrypted data is then stored in a Microsoft SQL Server database. Therefore, SQL Server 2000 or any latest version of SQL server is required to be installed on the production machine. The application also uses a PC camera for a facial snapshot during registration. The camera is installed with all its drivers for the application to make use of the imaging device. Fig. 2 shows the system architecture of the application. Text marked in red indicates software and hardware provided by Fujitsu Corporation. The electromechanical lock and its associated interface hardware and software components are not shown on the diagram.

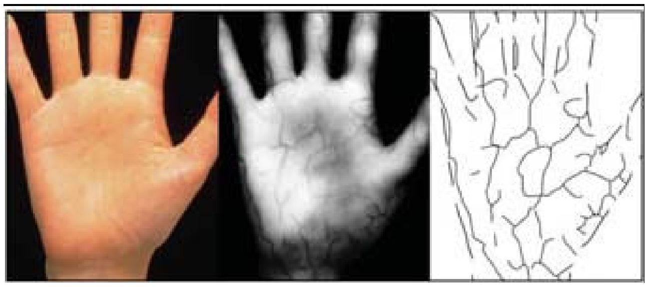


Fig. 1. Vein map after near infrared light penetration [Fujitsu Corporation].

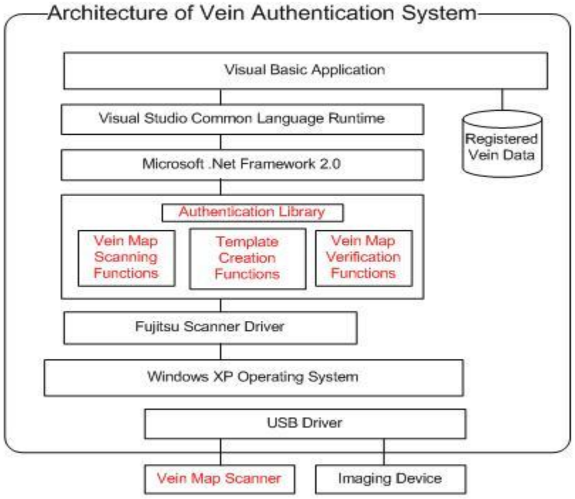
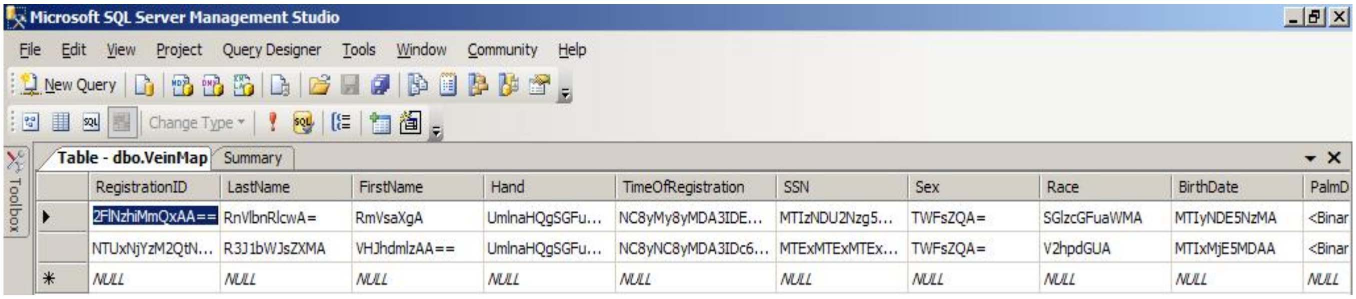

Architecture of Vein Authentication System
Visual Basic Application
Visual Studio Common Language Runtime
Registered Vein Data
Microsoft .Net Framework 2.0
Authentication Library
Vein Map Scanning Functions
Template Creation Functions
Vein Map Verification Functions
Fujitsu Scanner Driver
Windows XP Operating System
USB Driver
Vein Map Scanner
Imaging Device

Fig. 2. System architecture.



	RegistrationID	LastName	FirstName	Hand	TimeOfRegistration	SSN	Sex	Race	BirthDate	PalmD
▶	2FlNzhiMmQxAA==	RnVlbnRlcwA=	RmVsaXgA	UmlnaHQgSGFu...	NC8yMy8yMDA3IDE...	MTIzNDU2Nzg5...	TWFsZQA=	SGlzcGFuaWMA	MTIyNDE5NzMA	<Binar
	NTUxNjYzM2QtN...	R3J1bWJsZXMA	VHJhdmlzAA==	UmlnaHQgSGFu...	NC8yNC8yMDA3IDc6...	MTExMTExMTEx...	TWFsZQA=	V2hpdGUA	MTIxMjE5MDAA	<Binar
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Fig. 3. Sample vein map table in SQL server.

The database contains personal and biometric records of all people who use the system for authenticated access. The information that are stored in the database are registration ID, first and last names, hand identifying information (left hand or right hand), time of registration, SSN, sex, race, birth date, palm data, and photo. The palm data stored in the database is essentially an encrypted version of the vein map data. The registration ID is a unique identifier that is essentially a Global Unique Identifier (GUID) generated using the Visual Studio's class library. For the purpose of encryption, MD5 hashing algorithm is used to derive a private key from a GUID, which is used to encrypt the rest of the personal information for a person. The photograph is essentially the facial snap-shot of the individual whose vein map is registered. Fig. 3 shows a sample vein map table that contains personal information.

To run the application on any machine, the application needs: Microsoft's .Net Framework 2.0 redistributable package, all Fujitsu DLLs installed and registered within the PC registry, the Fujitsu scanner, an imaging device such as a PC camera, and depending on if the application will be run in standalone or client/server mode, the machine also needs to have a version of SQL Server installed for the storage of the vein map data.

4. CLASSES AND METHODS

The proposed application makes use of classes and methods available in the Fujitsu Vein Map (FVM) library to communicate with the vein map scanner, and also uses some additional classes and methods to communicate and control the camera and the model door lock.

FVM Class

The FVM is essentially the header file that wraps Fujitsu's C++ API into the .Net Framework. The class marshals all memory needed to communicate with the DLLs provided by Fujitsu Corporation. It is the class where all methods are defined along with all the structures and enumerations that are used to communicate with the vein map scanner. The methods included in the class can be categorized as: 1) Basic methods, 2) Sequence methods, and 3) Callback methods. The basic methods are used to open or close the vein authentication library and control transactions with the library from the application. For example, `FVM_BeginTransaction` is used to begin the transaction of registration or verification process. Similarly, `Fvm_EndTransaction` ends the transaction of registration or verification process. Sequence methods are used to create registration templates, verify vein data, cancel the execution, etc. For example, `Fvm_CreateTemplateSequence` method is used to create vein data for verification or registration purpose. Similarly, `Fvm_VerifyMatchSequence` is used to verify vein data for authentication. Callback methods are used to guide the user where to navigate or place their hand on the scanner. Fig. 4 shows interactions of the application with the class library for authentication.

iCam Class

The iCam class is used to control any imaging device installed on a system. Its main function in this application is to capture a snapshot (frame) of the screen given a particular *x*, *y* coordinate in pixels. The class uses, user32.dll, GDI32.dll, and avicap32.dll which are shipped and installed with Windows XP. The class includes methods to setup, initialize, and position the camera and capture frames.

VMF Class

The VMF class is used to allocate memory dynamically based on the number of vein map templates in the data repository. The VMF class initializes three arrays: `KeyArray()`, which holds all the registration keys in the data repository, `TemplateArray`, which is a two dimensional array that holds in one colum, the location of the vein map data in the data repository (acts like a pointer), and in another column the actual vein map data, and `PointerArray`, which is an array of pointers which points to the pointers that point to the template data. There is only on method in this class named `BuildTemplateList`.

Its purpose is to connect to the veinmap database, given a connection string read from an application configuration file, get the count of all the templates to compare agianst, redim (re-initialize) the mentioned arrays based on the count received, and load the arrays with the template data.

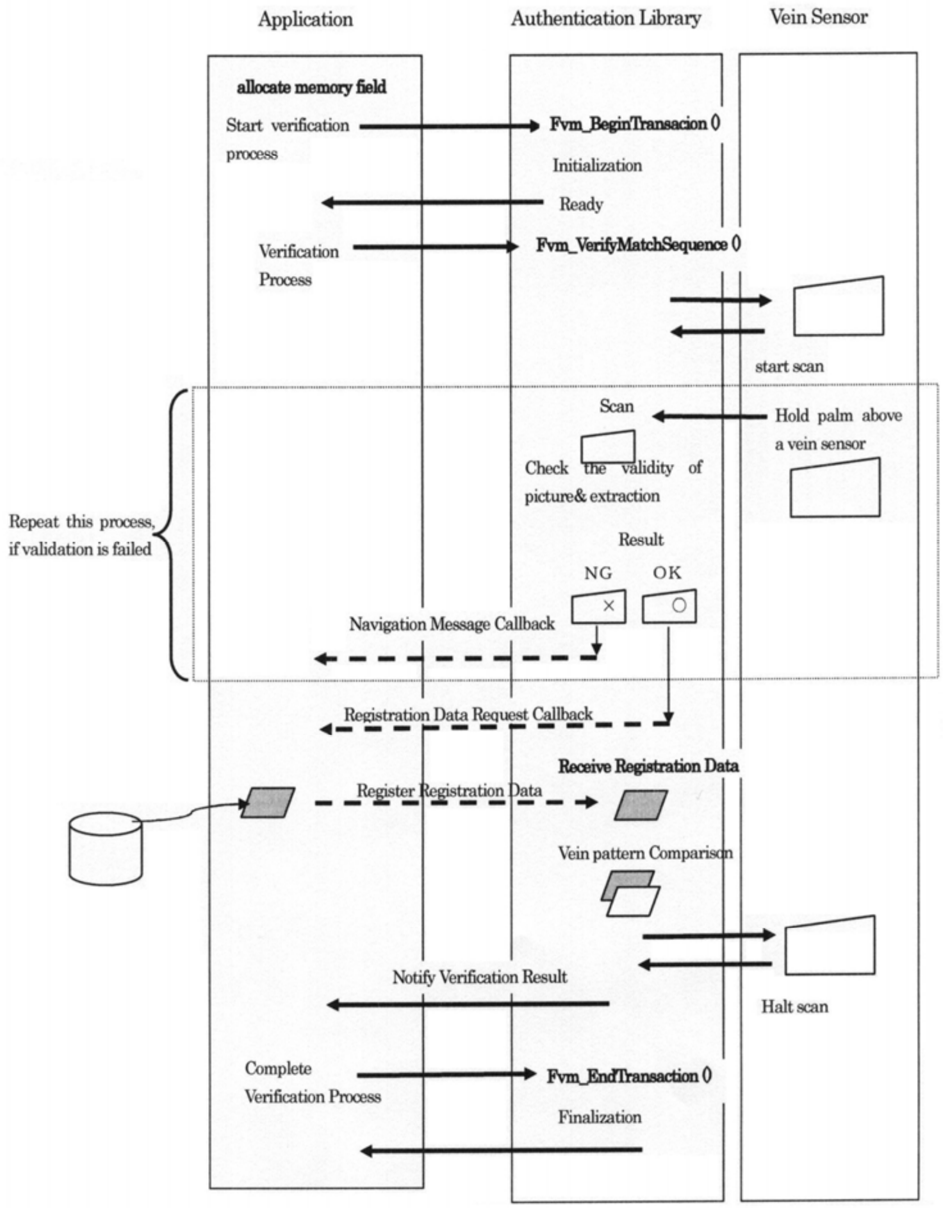
SEAIO Class

The SEAIO Class interfaces (wraps) the API provided by Sealevel Systems Inc. to interact with their I/O device which locks and unlocks a model door when a person's vein map is authenticated. The list of methods the application interacts with includes: `SeaIo_OpenDevice`, `SeaIo_CloseDevice`, `SeaIo_GetAdapterInfo`, `SeaIo_GetAdapterState`, and `SeaIo_WriteBit`.

5. INFORMATION PROCESSING

The application handles three distinct processes: registration, authentication, and then door lock control. The registration process is necessary to introduce the vein data into the vein map system for comparison during the authentication process.

During registration, a person needs to place his or her palm over the scanner while the operator fills a form as shown in Fig. 5 for the person. Also, a photograph of the person is taken. During this process, several scans of the palm are taken by the system to generate a vein map of the palm, which is eventually stored in the database after encryption. Rijndael's symmetric key encryption algorithm is used for encryption of the palm vein data. All other pertinent data items are also stored in the database.



Application
Authentication Library
Vein Sensor
allocate memory field
Start verification process
Fvm_BeginTransacion ()
Initialization
Ready
Verification Process
Fvm_VerifyMatchSequence ()
start scan
Scan
Hold palm above a vein sensor
Check the validity of picture& extraction
Result
NG
OK
Repeat this process, if validation is failed
Navigation Message Callback
Registration Data Request Callback
Receive Registration Data
Register Registration Data
Vein pattern Comparison
Notify Verification Result
Halt scan
Complete Verification Process
Fvm_EndTransaction ()
Finalization

Fig. 4. Authentication data flow.

The authentication process extracts the vein map of the individual whose palm is over the scanner and compares the extracted vein map to the registered vein map data in the vein map repository. To initiate the authentication process, the operator enters the person’s full name, date of birth, social security number, ethnicity, hand, and sex. The application then retrieves a facial snapshot stored with the demographic information and the vein map data. During authentication, the person’s vein map is taken as input into the system, after the vein map is extracted and stored in memory, each row in the database containing registered vein maps are decrypted and compared until either a match is found or the end of the registered data is reached.

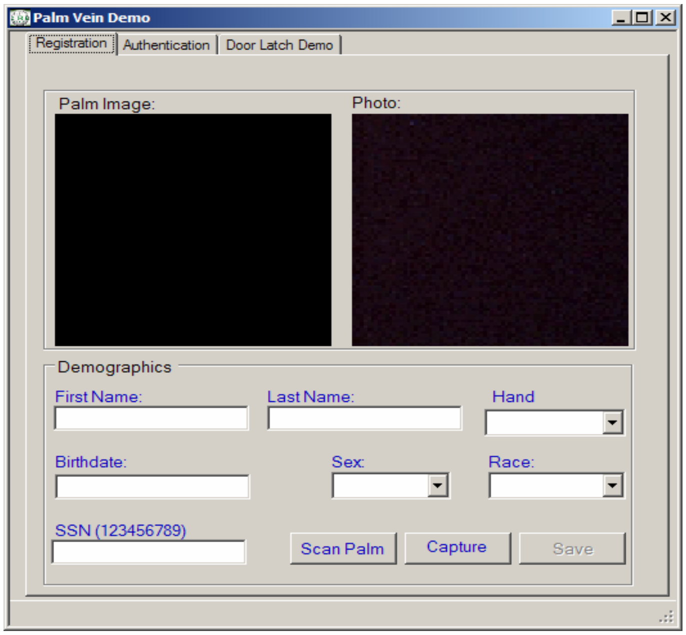


Fig. 5. Registration tab page.

At that point, the system either presents the person’s demographic information along with their facial snapshot if the vein map of the person being authenticated found a match in the database; otherwise a message appears stating that authentication has failed. Fig. 6 shows the authentication page.

Upon successful authentication, the application signals to unlock a door latch by applying voltage to a USB digital I/O device that controls the lock. A separate function in the application monitors the I/O device for changing occurrences in voltage and sends current through a terminal strip to an electronic door latch, thus locking and unlocking the door.

6. DISCUSSIONS

The proposed system is a product that has been developed as proof of concept and is exclusively used for demonstration purposes. Like any typical software development project, this application development went through component as well as system level testing at various stages of the project.

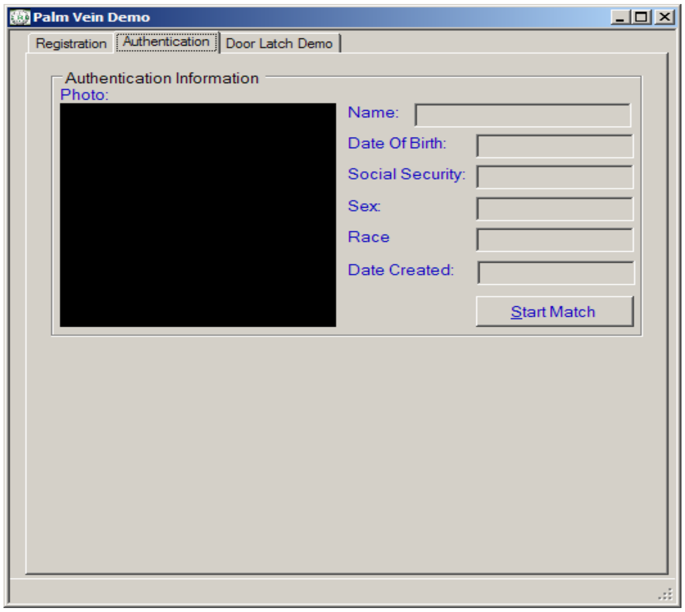


Fig. 6. Authentication tab page.

As part of unit testing, each of the four main classes (FVM Class, iCam Class, VMF Class, and SEAIO Class) that are used in the application were first developed as stand alone programs and rigorously tested before being integrated into the overall product. Tests were also performed after integration as well. Security testing ensures that protection mechanisms built into the system protects the application from improper penetration. The system is very secure and robust. All the vein map data previously registered are stored in a SQL Server database which is password protected to gain access. The data within the repository are encrypted using the RijndaelManaged crytography class provided by Visual Studio, a salt value read from an application configuratin file, and an MD5 hash of the resgistration ID used as a private key. Steganography was considered as a mechanism to hide the private key, but opted against due to image distortion.

7. CONCLUSION

There are various authentication systems available on the market these days, however, all have been found to have flaws or are not widely accepted due to various cultural, religious, psychological, and hygienic issues. As an alternative, in this work, vein map technology is introduced as a robust authentication system for today’s security need. The efficacy of the system has been tested in a prototype implementation. Future works using vein map technology should include research and possible development of algorithms and protocols on how to extract the vein map from the palm and then send the authentication data via IP, e.g., the data repository could be located in Langley, Virginia, while the person under authentication could be in Baghdad, Iraq. There are various “USB over IP” devices available on the market today which could make this readily possible.

7. ACKNOWLEDGEMENTS

This work was partially supported by the National Science Foundation under grant number EIA-0330822.

8. REFERENCES

[1] D. Bolme, M. Teixeira, and B. Draper, "The CSU Face Identification Evaluation System: Its Purpose, Features and Structure," **International Conference on Vision Systems**, April 1-3, 2003.

[2] J. Daugman, "Probing the uniqueness and randomness of IrisCodes: Results from 200 billion iris pair comparisons." **Proceedings of the IEEE**, Vol. 94, No. 11, 2006, pp 1927-1935.

[3] Feature Article Archive. Available from http://www.findbiometrics.com/Pages/feature_archive.html (visited March 15, 2008).

[4] PalmSecure: Palm Vein Authentication System http://www.fujitsu.com/us/services/biometrics/palm-vein/#footnote0 (visited March 15, 2008).

[5] J. Lu, K. N. Plataniotis, and A. N. Venetsanopoulos, "Regularized Discriminant Analysis for the Small Sample Size Problem in Face Recognition," **Pattern Recognition Letters**, Vol. 24, No. 16, 2003, pp. 3079-3087.

[6] P. Masons, " Major flaws in biometric security products," Available from http://www.out-law.com/page-2624 (visited October 2, 2006).

[7] L. Sirovich and M. Kirby, "A Low-Dimensional Procedure for the Characterization of Human Faces," **J. Optical Soc. Am. A**, Vol. 4, No. 3, 1987, pp. 519-524.

[8] Unknown. Biometric authentication: what method works best? Available from http://www.technovelgy.com/ct/Technology-Article.asp?ArtNum=16 (visited March 15, 2008).

[9] A. Osborn, "Biometrics history - looking at biometric technologies from the past to the present," Available from http://www.video-surveillance-guide.com/biometrics-history.htm (visited March 15, 2008).

[10] L. Wiskott, "Face recognition by elastic bunch graph matching," available from http://www.neuroinformatik.ruhr-uni-bochum.de/ini/VDM/research/computerVision/graphMatching/identification/faceRecognition/contents.html (visited March 15, 2008).