Short Papers

Hiding Biometric Data

Anil K. Jain, *Fellow*, *IEEE*, and
Umut Uludag, *Student Member*, *IEEE*

Abstract—With the wide spread utilization of biometric identification systems, establishing the authenticity of biometric data itself has emerged as an important research issue. The fact that biometric data is not replaceable and is not secret, combined with the existence of several types of attacks that are possible in a biometric system, make the issue of security/integrity of biometric data extremely critical. We introduce two applications of an amplitude modulation-based watermarking method, in which we hide a user's biometric data in a variety of images. This method has the ability to increase the security of both the hidden biometric data (e.g., eigen-face coefficients) and host images (e.g., fingerprints). Image adaptive data embedding methods used in our scheme lead to low visibility of the embedded signal. Feature analysis of host images guarantees high verification accuracy on watermarked (e.g., fingerprint) images.

Index Terms—Biometrics, data hiding, face, fingerprint, minutiae, steganography, watermarking.

1 Introduction

Biometrics-based personal identification techniques that use physiological or behavioral characteristics are becoming increasingly popular compared to traditional token-based or knowledge-based techniques such as identification cards (ID), passwords, etc. One of the main reasons for this popularity is the ability of the biometrics technology to differentiate between an authorized person and an impostor who fraudulently acquires the access privilege of an authorized person [1]. Among various commercially available biometric techniques such as face, voice, fingerprint, iris, etc., fingerprint-based techniques are the most extensively studied and the most frequently deployed.

While biometric techniques have inherent advantages over traditional personal identification techniques, the problem of ensuring the security and integrity of the biometric data is critical. For example, if a person's biometric data (e.g., her fingerprint image) is stolen, it is not possible to replace it unlike replacing a stolen credit card, ID, or password. Schneier [2] points out that a biometrics-based verification system works properly only if the verifier system can guarantee that the biometric data came from the legitimate person at the time of enrollment. Furthermore, while biometric data provide uniqueness, they do not provide secrecy. For example, a person leaves fingerprints on every surface she touches and face images can be surreptitiously observed anywhere that person looks. Ratha et al. [3] identify eight basic sources of attacks that are possible in a generic biometric system (Fig. 1). In the first type of attack, a fake biometric (such as a fake finger) is presented at the sensor. Resubmission of digitally stored biometric data constitutes the second type of attack. In the third type of attack, the feature detector could be forced to produce feature values chosen by the attacker, instead of the actual values generated from the data obtained from the sensor. In the fourth type of attack, the features extracted using the data obtained from the sensor are replaced with a synthetic feature set. In the fifth type of attack, the matcher component could be attacked to produce high or low matching scores, regardless of the input feature set. Attack on the templates stored in databases is the sixth type of attack. In the seventh type of attack, the channel between the database and matcher could be compromised to alter transferred template information. The final type of attack includes altering the matching result itself. All of these attacks have the possibility to decrease the credibility of a biometric system. As a solution to the second type of attacks, called replay attacks, Ratha et al. [4] proposed a challenge/response-based system. In a related context, Janbandhu and Siyal [5] proposed using biometric data in the generation of digital signatures in both symmetric and asymmetric systems.

In order to promote the wide spread utilization of biometric techniques, an increased security of the biometric data, especially fingerprints, is necessary. Encryption, watermarking, and steganography are possible techniques to achieve this. Steganography, derived from the Greek language and meaning secret communication, involves hiding critical information in unsuspected carrier data. While cryptography focuses on methods to make encrypted information meaningless to unauthorized parties, steganography is based on concealing the information itself. As a result, steganography-based techniques can be suitable for transferring critical biometric information, such as minutiae data, from a client to a server. Steganographic techniques reduce the chances of biometric data being intercepted by a pirate, hence reducing the chances of illegal modification of the biometric data. Digital watermarking techniques can be used to embed proprietary information, such as company logo, in the host data to protect the intellectual property rights of that data [6]. They are also used for multimedia data authentication. Encryption can be applied to the biometric templates for increasing security; the templates (that can reside in either 1) a central database, 2) a token such as smart card, or 3) a biometric-enabled device such as a cellular phone with fingerprint sensor) can be encrypted after enrollment. Then, during authentication, these encrypted templates can be decrypted and used for generating the matching result with the biometric data obtained online. As a result, the encrypted templates are secured since they cannot be utilized or modified without decrypting them with the correct key, which is typically secret. But, one problem associated with this system is that encryption does not provide security once the data is decrypted. Namely, if there is a possibility that the decrypted data can be intercepted, encryption does not address the overall security of the biometric data. On the other hand, since watermarking involves embedding information into the host data itself, it can provide security even after decryption. The watermark, which resides in the biometric data itself and is not related to encryption-decryption operations, provides another line of defense against illegal utilization of the biometric data. For example, it can provide a tracking mechanism for identifying the origin of the biometric data. Also, searching for the correct decoded watermark information during authentication can render the modification of the data by a pirate useless, assuming that the watermark embedding-decoding system is secure. Furthermore, encryption can be applied to the watermarked data, combining the advantages of watermarking and encryption into a single system. In the context of our work, the security of the biometric data should be thought of as the means to eliminate at least some of the attacks shown in Fig. 1.

2 Watermarking Techniques

Digital watermarking, or simply watermarking, which is defined as embedding information such as origin, destination, access level, etc. of multimedia data (e.g., image, video, audio, etc.) in the host data, has been a very active research area in recent years [6]. General image watermarking methods can be divided into two groups according to the domain of application of watermarking. In spatial domain methods (e.g., [7]), the pixel values in the image channel(s) are changed. In spectral-transform domain methods, a watermark

- *The authors are with the Department of Computer Science and Engineering, Michigan State University, 3115 Engineering Building, East Lansing, MI 48824. E-mail: {jain, uludagum}@cse.msu.edu.*

Manuscript received 7 May 2002; revised 3 Feb. 2003; accepted 13 Mar. 2003. Recommended for acceptance by V. Govindaraju.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number 116497.

0162-8828/03/$17.00 © 2003 IEEE Published by the IEEE Computer Society

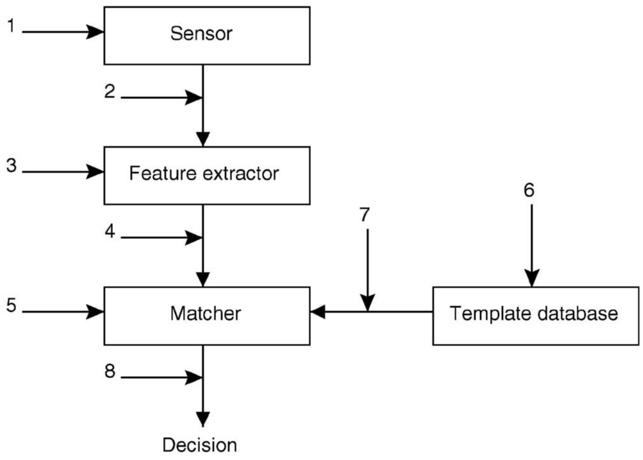

Fig. 1. Eight different attack points in a biometric authentication system (adapted from [3]).

signal is added to the host image in a transform domain such as the full-frame DCT domain [8].

There have been only a few published papers on watermarking of fingerprint images. Ratha et al. [9] proposed a data hiding method, which is applicable to fingerprint images compressed with WSQ wavelet-based scheme. The discrete wavelet transform coefficients are changed during WSQ encoding, by taking into consideration possible image degradation. Pankanti and Yeung [10] proposed a fragile watermarking method for fingerprint image verification. A spatial watermark image is embedded in the spatial domain of a fingerprint image by utilizing a verification key. The proposed method can localize any region of image that has been tampered. Pankanti and Yeung conclude that their watermarking technique does not lead to a significant performance loss in fingerprint verification. A semiunique key based on local block averages is used by Jain [11] to detect tampering of host images, including fingerprints and faces. Gunsel et al. [12] described two spatial domain watermarking methods for fingerprint images. The first method utilizes gradient orientation analysis in watermark embedding, so the watermarking process alters none of the features extracted using gradient information. The second method preserves the singular points in the fingerprint image, so the classification of the watermarked fingerprint image (e.g., into arch, left loop, etc.) is not affected.

3 Hiding Biometric Data

In this paper, we consider two application scenarios. The basic data hiding method is the same in both of the scenarios, but it differs in the characteristics of the embedded data, host image carrying that data, and medium of data transfer. While we are using fingerprint and face feature vectors as the embedded data, other information such as user name or user identification number can also be hidden into the images. We have selected to use one type of biometric data to secure another type of biometric data to increase the overall security of the system.

3.1 Application Scenarios

The first scenario involves a steganography-based application (Fig. 2a): The biometric data (fingerprint minutiae) that need to be transmitted (possibly via a nonsecure communication channel) is hidden in a host (also called cover and carrier) image, whose only function is to carry the data. For example, the fingerprint minutiae may need to be transmitted from a law enforcement agency to a template database, or vice versa. In this scenario, the security of the system is based on the secrecy of the communication. The host image is not related to the hidden data in any way. As a result, the host image can be any image available to the encoder. In our application, we consider three different types of cover images: a synthetic fingerprint image, a face image and an arbitrary image (Fig. 3). The synthetic fingerprint image (360 x 280) is obtained after a postprocessing of the image generated using the algorithm described by Cappelli et al. [13]. Using such a synthetic fingerprint image to carry actual fingerprint minutiae data provides an increased level of security since the person who intercepts the communication channel and obtains the carrier image is likely to treat this synthetic image as a

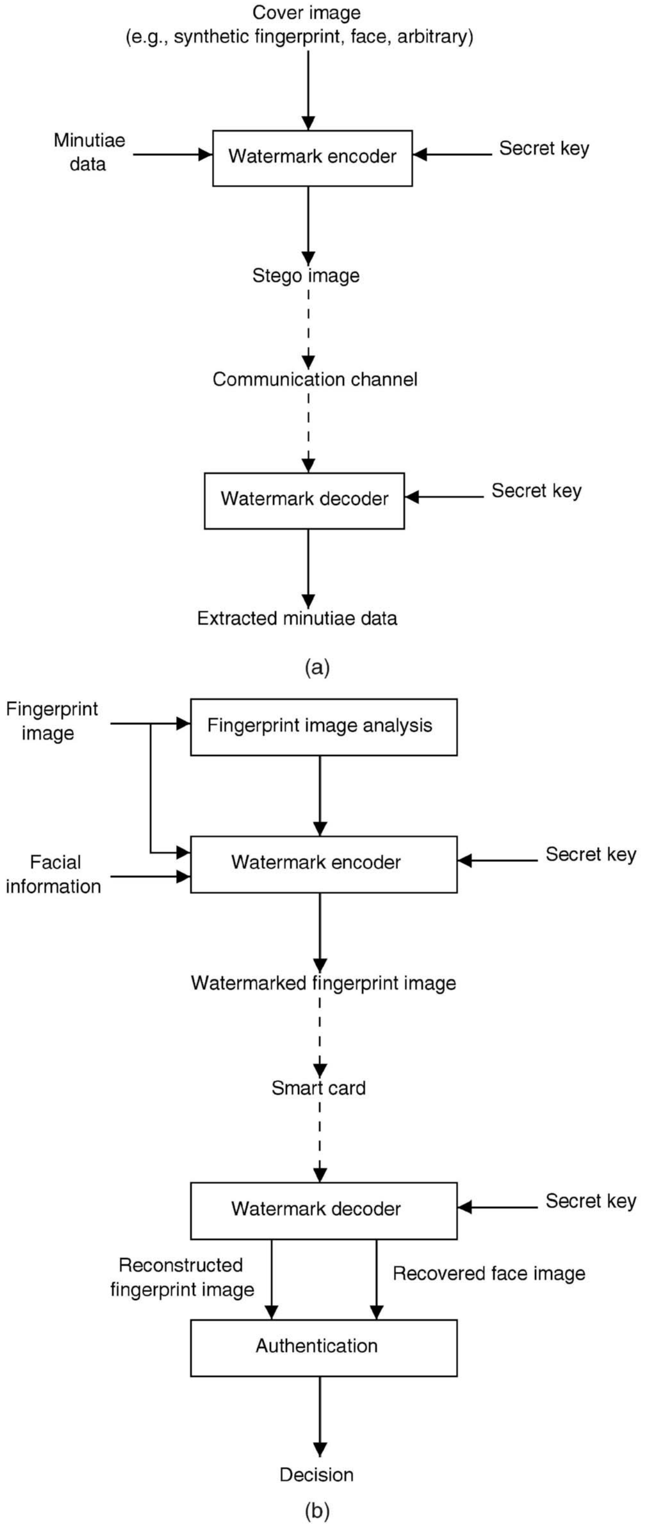

Fig. 2. Diagrams of application scenarios: (a) scenario 1 and (b) scenario 2.

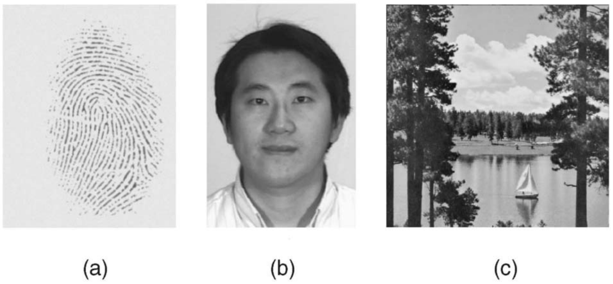

Fig. 3. Sample cover images: (a) synthetic fingerprint, (b) face, and (c) "Sailboat."

real fingerprint image! The face image (384 x 256) was captured in our laboratory. The "Sailboat" image (512 x 512) is taken from the USC-SIPI database [14].

This application can be used to counter the seventh type of attack (on the communication channel between the database and the fingerprint matcher) depicted in Fig. 1. An attacker will most likely not suspect that a cover image is carrying the minutiae information. Furthermore, the security of the transmission can be further increased by encrypting the stego image before transmission. Here, symmetric or asymmetric key encryption [15] can be utilized, depending on the requirements of the application such as key management, coding-decoding time (much higher with asymmetric key cryptography), etc. The position and orientation attributes of fingerprint minutiae constitute the data to be hidden in the host images. The fingerprint images (300 x 300) used in this work were captured by a solid-state sensor manufactured by Veridicom. The minutiae are extracted using the method outlined in [16]. A secret key is utilized in encoding to increase the security of the hidden data. The image with embedded data (called *stego* image) is sent through the channel that may be subject to interceptions. At the decoding site, using the same key that was used by the encoder (which can be delivered to the decoder using a secure channel prior to stego image transfer), the hidden data is recovered from the stego image. The keys can be different for every transmission, or several parameters such as receiver, sender, and fingerprinted subject identities can be used in determining the key assignment.

The second scenario is based on hiding facial information (e.g., eigen-face coefficients) into fingerprint images. In this scenario, the marked fingerprint image of a person can be stored in a smart card issued to that person (Fig. 2b). At an access control site, for example, the fingerprint of the person possessing the card will be sensed and it will be compared to the fingerprint stored on the smart card. Along with this fingerprint matching, our proposed scheme will extract the face information hidden in the fingerprint image. The recovered face will be used as a second source of authenticity, either automatically or by a human in a supervised biometric application. In this scenario, an additional biometric (e.g., face) is embedded into another biometric (e.g., fingerprint), in order to increase the security of the latter.

3.2 Data Hiding Method

The amplitude modulation-based watermarking method described here is an extension of the blue channel watermarking method of Kutter et al. [7]. The proposed method includes image adaptivity, watermark strength controller, and host image feature analysis along with the basic method in [7]. An earlier version of the method is presented in [12], in which the increase in data decoding accuracy related to these extensions is analyzed. In the first step, the data to be hidden into the host image is converted to a binary stream. In the first scenario, where fingerprint minutiae data are hidden, every field of individual minutia is converted to a 9-bit binary representation. Such a representation can code integers between [0, 511] and this range is adequate for x-coordinate ([0, N-1]), y-coordinate ([0, M-1]), and orientation ([0, 359]) of a minutia, where N and M are the number of rows and number of columns in the fingerprint image, respectively. In the second scenario, eigen-face coefficients are converted to a binary stream using four bytes per coefficient. A random number generator initialized with the secret key generates locations of the host image pixels to be watermarked. The details of this procedure are as follows: First, a sequence of random numbers between 0 and 1 is generated using uniform distribution. Then, every number with odd indices is linearly mapped to [0, X-1], and every number with even indices is linearly mapped to [0, Y-1], where X and Y are the number of rows and columns of the host image, respectively. Every pair comprised of one number with odd indices and one number with even indices indicates the location of a candidate pixel to be marked. During watermark embedding, a pixel is not changed more than once, as this can lead to incorrect bit decoding. Also, the pixels where $\beta(i,j)$ (marked pixel map, explained below) is zero are not marked. If at any step in embedding the candidate pixel cannot be marked due to one of these situations, the next pixel location is considered.

The $(i,j)\mathrm{th}$ pixel is changed according to the following equation

$$\begin{aligned} P_{WM}(i,j) &= P(i,j) + (2s-1)P_{AV}(i,j)q \\ &\quad * \left(1 + \frac{P_{SD}(i,j)}{A}\right)\left(1 + \frac{P_{GM}(i,j)}{B}\right)\beta(i,j), \end{aligned} \tag{1}$$

where $P_{WM}(i,j)$ and $P(i,j)$ are values of the watermarked and original pixels at location (i,j), respectively. The value of watermark bit is denoted as s and watermark embedding strength is denoted as q, $s \in [0,1]$, $q > 0$. $P_{AV}(i,j)$ and $P_{SD}(i,j)$ denote the average and standard deviation of pixel values in the neighborhood of pixel (i,j) and $P_{GM}(i,j)$ denotes the gradient magnitude at (i,j). The parameters A and B are weights for the standard deviation and gradient magnitude, respectively, and they modulate the effect of these two terms; increasing either of them decreases the overall modulation effect on the amount of change in pixel intensity, while decreasing them has the opposite effect. The minimum values for $P_{SD}(i,j)$ and $P_{GM}(i,j)$ are both 0, for neighborhoods with constant gray level; the maximum value for $P_{SD}(i,j)$ is around 127, for a checkerboard pixel pattern composed of just 0 and 255 gray levels. The maximum value for $P_{GM}(i,j)$ is around 1,082 for a maximum magnitude diagonal edge (e.g., intersection of gray levels 0 and 255). The $\beta(i,j)$ term guarantees that image pixels, called marked pixels, whose alteration may affect the performance of an algorithm using the watermarked image (e.g., fingerprint verification in the case of watermarked fingerprint images) are unchanged; $\beta(i,j)$ takes the value 0 if the pixel (i,j) is a marked pixel and takes the value 1, otherwise. These three parameters ($P_{SD}(i,j)$, $P_{GM}(i,j)$, and $\beta(i,j)$) modulate the amount of change in pixel values made due to marking, and it is a significant modification of the basic marking method given in [7].

In the second scenario, the marked pixels are defined by either minutiae analysis or ridge analysis of the fingerprint image. In our experiments, P_{AV} is calculated in a 5 x 5 square neighborhood and P_{SD} is calculated in a 5 x 5 cross-shaped neighborhood. The gradient magnitude is computed via the 3 x 3 Sobel operator.

The image adaptivity terms discussed above adjust the magnitude of watermarking, by utilizing several properties of the human visual system (HVS). Using $P_{AV}(i,j)$ in modulating watermark magnitude conforms to amplitude nonlinearity of HVS. As (1) shows, the magnitude of the change in the value of pixel (i,j) caused by watermarking is higher when the $P_{AV}(i,j)$ value is high. Standard deviation and gradient magnitude terms utilize contrast/texture masking properties of HVS. These image adaptivity terms increase the magnitude of watermarking in image areas where such an increase does not become very visible to a human observer.

Every watermark bit with value s in (1) is embedded at multiple locations in the host image. This redundancy increases the correct decoding rate of the embedded information. The amount of this redundancy is limited by image capacity (size) and visibility of the changes in pixel values. Furthermore, the $\beta(i,j)$ mask preserves critical features of the host fingerprint image. In addition to the binary watermark data, two reference bits, 0 and 1, are also embedded in the host image. These reference bits help in calculating an adaptive threshold in determining the watermark bit values during decoding. Decoding starts with finding the data embedding locations in the watermarked image, via the secret key used during the watermark encoding stage. Note that an original, nonwatermarked image is not used in decoding, just the watermarked image is used. For every bit embedding location (i, j), its value during decoding is estimated as the linear combination of pixel values in a 5 x 5 cross-shaped neighborhood of the watermarked pixels as given by (2).

$$\begin{aligned}\hat{P}(i,j) =& \frac{1}{8}\left(\sum_{k=-2}^{2} P_{WM}(i+k,j)\right.\\ &\left. + \sum_{k=-2}^{2} P_{WM}(i,j+k) - 2P_{WM}(i,j)\right).\end{aligned} \tag{2}$$

The difference between the estimated and watermarked pixel values is calculated as

$$\delta = P_{WM}(i,j) - \hat{P}(i,j). \tag{3}$$

These differences are averaged over all the embedding locations associated with the same bit, to yield $\bar{\delta}$. For finding an adaptive threshold, these averages are calculated separately for the reference bits, 0 and 1, as $\bar{\delta}_{R0}$ and $\bar{\delta}_{R1}$, respectively. Finally, the watermark bit value $\hat{s}$ is estimated as

$$\hat{s} = \begin{cases} 1 & \text{if } \bar{\delta} > \frac{\bar{\delta}_{R0}+\bar{\delta}_{R1}}{2}, \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

Equation (4) essentially indicates that, if $\bar{\delta}$ for a specific bit is closer to $\bar{\delta}_{R1}$, that bit is declared as "1;" if it is closer to $\bar{\delta}_{R0}$, that bit is declared as "0." The watermark decoding process can produce erroneous bits since decoding is based on an estimation procedure, which may fail to find the exact original pixel values. This may lead to switched bits in decoding. Also, in the context of this work, we want every one of the embedded bits to be decoded correctly (i.e., 0 percent error rate), since we are embedding critical information where even one bit change can decrease the usability of the data (e.g., minutiae data change, eigen-face coefficient change due to switched bits). In order to increase the decoding accuracy, the encoder uses a controller block. This block adjusts the strength of watermarking, q, on a pixel-by-pixel basis, if there is a possibility of incorrect bit decoding. Effectively, given the parameters such as A, B, and q, the encoder checks whether the decoding will be correct or not. In the former case, the controller moves on to analyze the next bit embedding location; in the latter case, q is increased to the point where the bit can be correctly decoded.

From decoded watermark bits, the data hidden in the host image (minutiae data or eigen-face coefficients) is extracted. Using the recovered eigen-face coefficients and the eigen-faces stored in the watermark decoding site, the hidden face image is reconstructed. In the second application scenario, an estimate of the original host fingerprint image is also found via replacing the watermarked pixel values with the $\hat{P}(i, j)$ calculated by (2).

4 Experimental Results

In this section, experimental results for the two application scenarios explained in the previous section will be presented. Factors such as decoding accuracy and matching performance will be highlighted. For the first scenario, nearly 17 percent of the stego image pixels are changed during minutiae data hiding for all the three cover images shown in Fig. 3. The key used in generating the locations of the pixels to be watermarked is selected as the integer 1,000. However, the exact value of key does not affect the performance of the method. In our implementation, this key is used as the seed for the C++ random number generator. The generated random numbers are used as explained in the previous section. Other random number generators can be used without affecting the performance of the proposed method. Remaining watermarking parameters are set to: $q = 0.1$, $A = 100$, $B = 1000$. A higher q value increases the visibility of the hidden data. Increasing A or B decreases the effect of standard deviation and gradient magnitude in modulating watermark embedding strength, respectively. The size of the hidden data here is approximately 85 bytes. The extracted minutiae data from all of the three cover images is found to be exactly the same as the hidden data. Furthermore, the performance of the proposed algorithm was determined as follows: 15 images (five synthetic fingerprint, five face, five arbitrary) were watermarked with five different sets of minutiae data, and by using five different keys. As a result, 375 different watermarked images were produced. Characteristics and sources of the host images and watermarking parameters are the same as given previously. Individual minutiae data sets contained between 23 to 28 points, with an average of 25 points. From all of these 375 watermarked images, we were able to extract the embedded minutiae information with 100 percent accuracy.

For the second application scenario, the fingerprint image (300 x 300) shown in Fig. 4a is watermarked using the input face image (150 x 130) shown in Fig. 4b. The watermark information occupies 56 bytes, corresponding to the 14 eigen-face coefficients (four bytes per coefficient). These 14 eigen-face coefficients generate the 150 x 130 watermark face image of Fig. 4c [17]. Note that 14 eigen-face coefficients are sufficient for a high fidelity reconstruction of input face. A small face image database, which consists of 40 images, with four images for each of the 10 subjects, was used to generate the eigen-faces and coefficients.

Figs. 4d and 4e correspond to minutiae-based data hiding. The input image in Fig. 4a is watermarked without changing the pixels shown in black (16 percent of the total image pixels) in Fig. 4d. This minutiae-based feature image, which represents the $\beta(i, j)$ term in (1), is obtained by drawing 23 x 23 square blocks around every minutiae of the input fingerprint image. Fig. 4e shows the image reconstructed during watermark decoding. Nearly 15 percent of all the image pixels are modified during watermark encoding. This marking ratio is determined experimentally by requiring 100 percent correct decoding of the embedded data. Figs. 4f and 4g correspond to ridge-based data hiding. The input image in Fig. 4a is watermarked without changing the pixels shown in black (31 percent of the total number of image pixels) in Fig. 4f. This ridge-based feature image is obtained from the thinned ridge image of the input fingerprint via dilation with a 3 x 3 square structuring element comprised of all nine pixels. Fig. 4g shows the image reconstructed during watermark decoding. Nearly 15 percent of all the image pixels are modified during watermark encoding. This embedding ratio is the same as the one used for minutiae-based embedding; fixing this parameter allows us to compare the two methods based on their $\beta(i, j)$ mask characteristics. Effectively, the images in Figs. 4d and 4f denote the binary $\beta(i, j)$ maps.

In both of these cases, the key used in generating the locations of the pixels to be watermarked is selected as the integer 1,000. However, as mentioned earlier, the exact value of key does not affect the performance of the method. Other watermarking parameters are set to the same values used previously, namely: $q = 0.1$, $A = 100$, $B = 1000$. The watermark data are decoded correctly in the decoding phase in both of the cases; the recovered faces are exactly the same as the watermark face image in Fig. 4c.

In order to assess the effect of watermarking on fingerprint verification accuracy, ROC (Receiver Operating Characteristics) curves for original images and images that are recovered after watermark decoding are computed. A total of 640 fingerprint images are used in our experiments. These images come from 160 users, with four impressions each of the right index finger captured using a Veridicom sensor. Three ROC curves given in Fig. 5 correspond to fingerprint verification 1) without data hiding, 2) with minutiae-based data hiding, and 3) with ridge-based data hiding.

The proximity of the three curves in Fig. 5 indicates that both minutiae-based and ridge-based watermarking methods do not introduce any significant degradation in fingerprint verification accuracy, though it is observed that ridge-based watermarking leads to less degradation. Furthermore, in both of the cases, the embedded

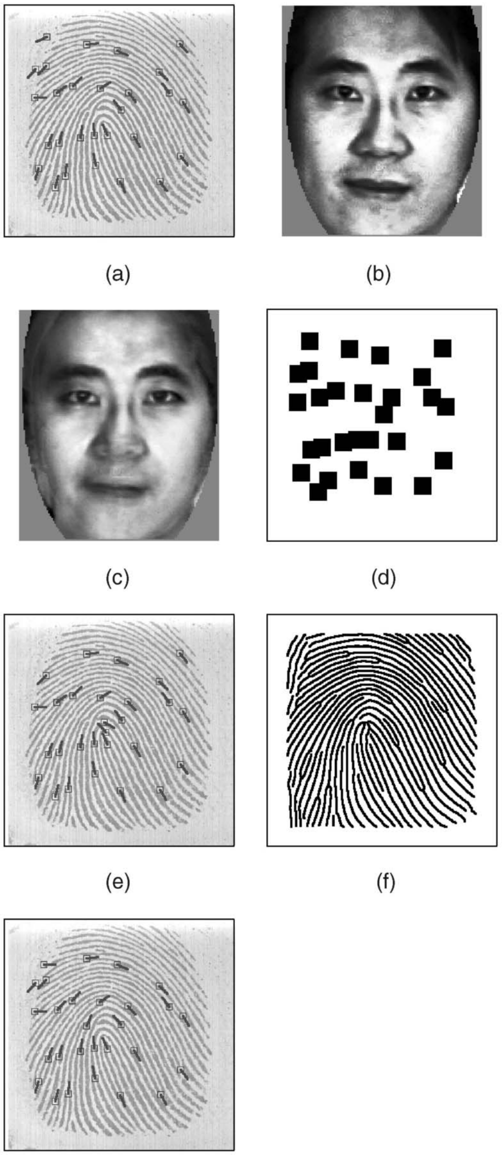

Fig. 4. Facial information embedding and decoding: (a) input fingerprint image with overlaid minutiae, (b) input face image, (c) watermark face image, (d) fingerprint feature image based on the minutiae, (e) reconstructed fingerprint image with overlaid minutiae, where watermarking did not change the pixels shown in black in (d), (f) fingerprint feature image based on the ridges, (i) reconstructed fingerprint image with overlaid minutiae, where watermarking did not change the pixels shown in black in (f).

information (i.e., 14 eigen-face coefficients) was decoded with 100 percent accuracy from all of the 640 watermarked images.

5 Conclusions

The ability of biometrics-based personal identification techniques to differentiate between an authorized person and an impostor who fraudulently acquires the access privilege of an authorized person is one of the main reasons for their popularity compared to traditional identification techniques. However, the security and integrity of the biometric data itself are important issues. Encryption, watermarking, and steganography are possible techniques to secure biometric data. In this paper, two applications of watermarking to secure that data are presented. In addition to watermarking, encryption can also be used to further increase the security of biometric data. The first application is related to increasing the security of biometric data exchange, which is based on steganography. In the second application, we embed facial information in fingerprint images. In this application, the data is hidden in such a way that the features that are used in fingerprint matching are not significantly changed during encoding/decoding. As a consequence, the verification accuracy based on decoded watermarked images is very similar to that with original images.

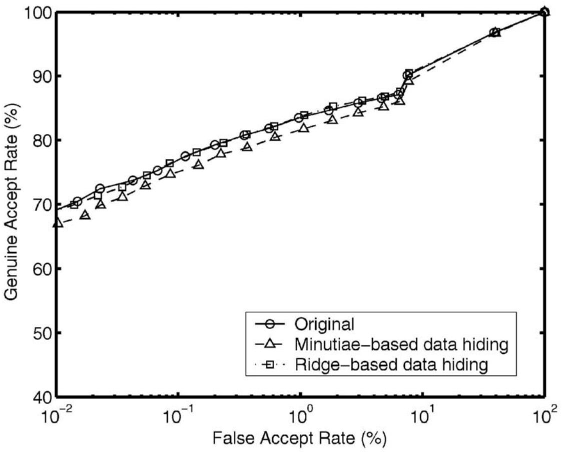

Fig. 5. ROC curves.

The proposed method utilizes several properties of the human visual system to keep the visibility of the changes made to the host image low. We are currently working on increasing the data hiding capacity of the host images. Another topic for future research is to investigate how different (e.g., robust and fragile) watermarking schemes can be combined.

References

[1] *BIOMETRICS: Personal Identification in Networked Society,* A. Jain, S. Pankanti, and R. Bolle, eds., Kluwer, 1999.

[2] B. Schneier, "The Uses and Abuses of Biometrics," *Comm. ACM,* vol. 42, no. 8, p. 136, Aug. 1999.

[3] N.K. Ratha, J.H. Connell, and R.M. Bolle, "An Analysis of Minutiae Matching Strength," *Proc. Third Int'l. Conf. Audio- and Video-Based Biometric Person Authentication,* pp. 223-228, June 2001.

[4] N.K. Ratha, J.H. Connell, and R.M. Bolle, "A Biometrics-Based Secure Authentication System," *Proc. IEEE Workshop Automatic Identification Advanced Technologies,* pp. 70-73, Oct. 1999.

[5] P.K. Janbandhu and M.Y. Siyal, "Novel Biometric Digital Signatures for Internet-Based Applications," *Information Management and Computer Security,* vol. 9, no. 5, pp. 205-212, 2001.

[6] F. Hartung and M. Kutter, "Multimedia Watermarking Techniques," *Proc. IEEE,* vol. 87, no. 7, pp. 1079-1107, July 1999.

[7] M. Kutter, F. Jordan, and F. Bossen, "Digital Signature of Color Images Using Amplitude Modulation," *Proc. SPIE,* vol. 3022, pp. 518-526, 1997.

[8] M. Barni, F. Bartolini, V. Cappellini, and A. Piva, "A DCT Domain System for Robust Image Watermarking," *Signal Processing,* vol. 66, no. 3, pp. 357-372, May 1998.

[9] N.K. Ratha, J.H. Connell, and R.M. Bolle, "Secure Data Hiding in Wavelet Compressed Fingerprint Images," *Proc. ACM Multimedia,* pp. 127-130, Oct. 2000.

[10] S. Pankanti and M.M. Yeung, "Verification Watermarks on Fingerprint Recognition and Retrieval," *Proc. SPIE,* vol. 3657, pp. 66-78, 1999.

[11] S. Jain, "Digital Watermarking Techniques: A Case Study in Fingerprints & Faces," *Proc. Indian Conf. Computer Vision, Graphics, and Image Processing,* pp. 139-144, Dec. 2000.

[12] B. Gunsel, U. Uludag, and A.M. Tekalp, "Robust Watermarking of Fingerprint Images," *Pattern Recognition,* vol. 35, no. 12, pp. 2739-2747, Dec. 2002.

[13] R. Cappelli, A. Erol, D. Maio, and D. Maltoni, "Synthetic Fingerprint Image Generation," *Proc. 15th Int'l Conf. Pattern Recognition,* vol. 3, pp. 475-478, Sept. 2000.

[14] The USC-SIPI Image Database. http://sipi.usc.edu/services/database/Database.html. 2003.

[15] B. Schneier, *Applied Cryptography,* second ed., John-Wiley, 1996.

[16] A.K. Jain, L. Hong, S. Pankanti, and R. Bolle, "An Identity-Authentication System Using Fingerprints," *Proc. IEEE,* vol. 85, no. 9, pp. 1365-1388, Sept. 1997.

[17] Evaluation of Face Recognition Algorithms. http://www.cs.colostate.edu/evalfacerec/index.html. 2003.