PERGAMON

Pattern Recognition 35 (2002) 2739–2747

PATTERN RECOGNITION
THE JOURNAL OF THE PATTERN RECOGNITION SOCIETY

www.elsevier.com/locate/patcog

Robust watermarking of fingerprint images

Bilge Gunsel[a,*], Umut Uludag[a], A. Murat Tekalp[a,b,c]

[a] *Multimedia Systems Group, Information Technologies Research Institute, TUBITAK, Marmara Research Center, P.O. Box 21, 41470 Gebze-Kocaeli, Turkey*

[b] *Department of Electrical Engineering, Koc University, Rumelifeneri Istanbul, Turkey*

[c] *Department of Electrical and Computer Engineering, University of Rochester, Rochester, NY 14627-0126, USA*

Accepted 31 October 2001

Abstract

This paper introduces two spatial methods in order to embed watermark data into fingerprint images, without corrupting their features. The first method inserts watermark data after feature extraction, thus preventing watermarking of regions used for fingerprint classification. The method utilizes an image adaptive strength adjustment technique which results in watermarks with low visibility. The second method introduces a feature adaptive watermarking technique for fingerprints, thus applicable before feature extraction. For both of the methods, decoding does not require original fingerprint image. Unlike most of the published spatial watermarking methods, the proposed methods provide high decoding accuracy for fingerprint images. High data hiding and decoding performance for color images is also observed. © 2002 Pattern Recognition Society. Published by Elsevier Science Ltd. All rights reserved.

Keywords: Adaptive watermarking; Data hiding; Digital security; Fingerprint watermarking

1. Introduction

Digital data can be duplicated very easily without introducing any quality degradations to the content. This brings out the problem of protection of Intellectual Property Rights (IPR). Watermarking has been a very active research area lately. By digital watermarking, information like origin, legal destination, and access rights, is embedded to the multimedia data without introducing any perceptible differences compared to the original. Generally, imperceptibility requirement is satisfied by utilizing some form of human sensory model (human visual system, human audible system) in watermark embedding [1–4].

Biometrics technology is essential for today's personal identification/verification systems. The security requirements of present electronic transactions necessitate utilization of reliable factors such as fingerprint features. Watermarking of fingerprint images can be used in applications like: (a) protecting the originality of fingerprint images stored in databases against intentional and unintentional attacks, (b) fraud detection in fingerprint images by means of fragile watermarks (which do not resist to any operations on the data and get lost, thus indicating possible tampering of the data), and (c) guaranteeing secure transmission of acquired fingerprint images from intelligence agencies to a central image database, by watermarking data prior to transmission and checking the watermark at the receiver site.

In the literature, there are a few published works for fingerprint image watermarking. Recently, Ratha et al. [5] introduced a data hiding algorithm for wavelet compressed fingerprint images. The method presented in Ref. [5] has the advantage of working in compressed domain. In our work, we introduce two fingerprint watermarking techniques in which gradient directions of the feature pixels or feature regions do not change with watermarking.

The organization of the paper is as follows. Section 2 outlines the proposed watermarking methods and their application to fingerprint image watermarking as well as color images. Section 3 presents experimental results. Conclusions are summarized in Section 4.

* Corresponding author. Tel.: +90-262-641-2300ext.4820; fax: +90-262-646-3187.

E-mail address: gunsel@btae.mam.gov.tr (B. Gunsel).

PII: S0031-3203(01)00250-3

2. Fingerprint image watermarking

Most common fingerprint verification methods are based on point patterns called ridge endings and bifurcations (minutiae) in fingerprints [6]. As a result of coarse level classification of point patterns, Wirbel (whorl and twin loop) and Lasso (arch, tented arch, right and left loop) classes can be specified. Thus, once these point patterns are extracted by directional images, they can be used to find out similarity (distance) between fingerprint patterns [7].

This paper introduces two fingerprint watermarking methods. Encoding and decoding rules of these methods are presented below.

2.1. Method 1

The first method inserts watermark data after feature extraction and thus prevents watermarking of regions used for fingerprint classification. Fig. 1 illustrates the functional block diagram of the system.

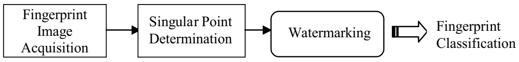

Fingerprint Image Acquisition
Singular Point Determination
Watermarking
Fingerprint Classification

Fig. 1. Watermark encoding after feature extraction.

2.1.1. Watermark encoding

The method utilizes an image adaptive strength adjustment technique which results in watermarks with low visibility. The watermark data are embedded onto gray scale fingerprint images according to the embedding rule given below:

$$P_{WM}(i,j) = P(i,j) + (2s-1)P(i,j)q\left(1+\frac{SD(i,j)}{A}\right) \times \left(1+\frac{GM(i,j)}{B}\right)\beta(i,j), \quad (1)$$

where $P_{WM}(i,j)$ and $P(i,j)$ are pixel values referring to watermarked and original pixels at watermark embedding location (i,j), respectively. The value of watermark bit is denoted as s. Watermark embedding strength is denoted as q. $SD(i,j)$ denotes the standard deviation of pixel values around a local neighborhood of pixel at (i,j), and $GM(i,j)$ denotes the gradient magnitude at (i,j). A and B are normalization factors for the standard deviation and gradient magnitude, respectively. $\beta(i,j)$ takes the value 0 if the pixel (i,j) under consideration belongs to a fingerprint feature region like delta or core areas (singular points); it has value 1 otherwise.

Every watermark bit with value s in Eq. (1) is embedded multiple times onto the fingerprint image pixels, whose locations are determined via the selected secret key. In addition to the real watermark data, two reference bits, 0 and 1, are embedded onto the image. These reference data provide an adaptive threshold in determining the watermark bit value in decoding.

In Eq. (1), standard deviation term $SD(i,j)$ can be computed as the standard deviation of the set containing the pixel values in a cross-shaped neighborhood of the watermark bit embedding location (i,j). Gradient magnitude term $GM(i,j)$ can be computed via a gradient operator, i.e., Sobel operator.

$SD(i,j)$ and $GM(i,j)$ terms adjust the strength of watermarking in an image adaptive way. At locations where either $SD(i,j)$ term is high (image regions with high variance) or $GM(i,j)$ term is high (edge regions), the watermark signal is added more strongly to the host image. This leads to more accurate decoding of embedded watermark data, especially for busy or textured images. Although watermark decoding accuracy is increased as a result of the image adaptive increase in embedding strength, due to the fact that human visual system is relatively less sensitive to pixel value changes in busy and edge image regions, the visibility of the watermark does not increase significantly.

When the host image is a fingerprint image, additional requirements arise which must be satisfied by the watermarking system. Watermark embedding process must not introduce any changes to the fingerprint image which may alter the features extracted from that image for personal authentication–verification purposes.

In Method 1, this requirement is satisfied. After extracting singular points from the fingerprint image and associated blocks corresponding to delta and core areas, watermark embedding is done according to Eq. (1). In this way, since $\beta(i,j)$ term is zero for those feature areas, watermarking does not change original pixel values and the singular points of the fingerprint image are preserved. As a result, the class of fingerprint image is not changed by watermarking.

In the case of color images, watermark data are embedded onto blue channel pixels of color image and Eq. (1) has been modified as

$$b_{WM}(i,j) = b(i,j) + (2s-1)L(i,j)q\left(1+\frac{SD(i,j)}{A}\right) \times \left(1+\frac{GM(i,j)}{B}\right), \quad (2)$$

where $b_{WM}(i,j)$ and $b(i,j)$ are gray values referring to watermarked and original blue channel pixels at watermark embedding location (i,j), respectively. $L(i,j)$ is the luminance value at (i,j) and can be calculated as

$$L(i,j) = 0.299R(i,j) + 0.587G(i,j) + 0.114B(i,j),$$

where $R(i,j)$, $G(i,j)$, $B(i,j)$ denote red, green and blue channel values at location (i,j). Note that multiplier $\beta(i,j)$ is eliminated in the color image case.

Every watermark bit with value s in Eq. (2) is embedded multiple times onto the blue channel pixels, whose locations are determined via the selected secret key. The reason for

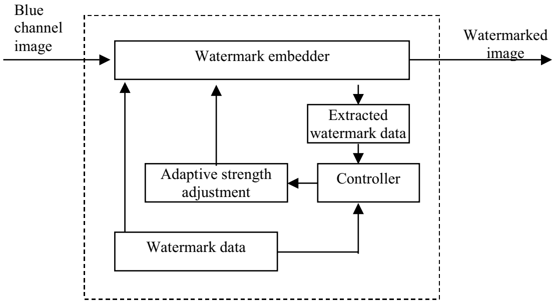


Fig. 2. Watermark encoder structure.

using blue channel for embedding is the lower sensitivity of human visual system to blue component of the color information.

2.1.2. Watermark decoding

Decoding starts with finding the watermark embedding locations on the watermarked image, via the secret key used in watermark encoding stage. For every bit embedding location, the value of the original pixel, $\hat{P}(i,j)$, is estimated as the linear combination of pixels in a cross-shaped neighborhood of the watermarked pixel as

$$\hat{P}(i,j) = \frac{1}{4c}\left(\sum_{k=-c}^{c} P_{WM}(i+k,j) + \sum_{k=-c}^{c} P_{WM}(i,j+k) - 2P_{WM}(i,j)\right), \quad (3)$$

where c is the neighborhood size. The difference between the estimated and current pixel values is calculated by

$$\delta = P_{WM}(i,j) - \hat{P}(i,j). \quad (4)$$

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 similarly for the reference bits, 0 and 1, as $\bar{\delta}_{R0}$ and $\bar{\delta}_{R1}$, respectively.

Then, the watermark bit value $\hat{s}$ is estimated as

$$\hat{s} = \begin{cases} 1, \text{ if } \bar{\delta} > \dfrac{\bar{\delta}_{R0} + \bar{\delta}_{R1}}{2}, \\ 0, \text{ otherwise.} \end{cases} \quad (5)$$

In the case of color image watermarking, to further increase the watermark decoding accuracy, in addition to using image-adaptive new terms for calculating watermark embedding strength, the watermark embedding strength can be controlled at the encoder site. The encoder performs the strength control by a simple feedback loop. The basic structure of the watermark encoder is given in Fig. 2. As shown in this figure, the accuracy of the watermark decoding is checked by controller, during watermark embedding. If this analysis yields the result that the watermark decoding will be incorrect, the encoder adaptively adjusts the watermark embedding strength until correct decoding is guaranteed. Note that the algorithm also considers the invisibility criterion.

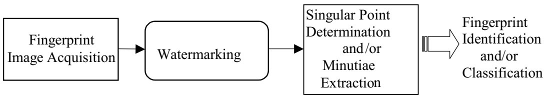

Fingerprint Image Acquisition
Watermarking
Singular Point Determination and/or Minutiae Extraction
Fingerprint Identification and/or Classification

Fig. 3. Adaptive watermark encoding with gradient direction analysis.

The idea behind the feedback loop is that the encoder changes the watermark strength q adaptively. For every bit embedding location, the term δ is calculated according to Eq. (4), by using the initial value of q. Then, if δ is calculated as a negative value for an embedded bit of "1", the value of q is increased until correct decoding is guaranteed, namely until δ is positive. Similarly, if the term δ is calculated as a positive value for an embedded bit of "0", the value of q is again increased until correct decoding is guaranteed, namely δ is negative. Furthermore, increasing the number of watermark embedding points, until the image capacity is reached, may improve the accuracy of watermark decoding, with the drawback of increased visibility.

2.2. Method 2

The second method introduces a feature adaptive watermarking technique for fingerprints that is applicable before feature extraction (Fig. 3).

2.2.1. Watermark encoding

Method 2 first utilizes an orientation analysis over the acquired fingerprint image. Then, the watermark embedding is performed by preserving the gradient orientations at and around watermark embedding locations specified by the secret key, within the quantization interval of the original data. Since the extraction of fingerprint features is based on gradient orientations, when watermark embedding does not change the quantized gradient orientation at considered pixel and its neighbors, the features of the fingerprint image are preserved. The same watermarking embedding technique is utilized for Methods 1 and 2. Note that unlike Method 1, the proposed watermark embedding scheme does not fix the actual gradient orientation at a pixel (i,j), but limits its change within the original orientation quantization interval. Hence, the performance of fingerprint verification–identification system is not affected by watermarking.

The quantization levels of orientation of fingerprint images, which are called Poincare index in literature [6], divide the 2π pixel gradient orientation circle into 16 bands, with each band covering $\pi/8$ rad, as shown in Fig. 4.

Let (i,j) denote the watermark embedding pixel specified by the secret key and $D(k,l)$ denote the eight neighbor pixels of (i,j). Watermark encoder first calculates gradients at all pixels belonging to $D(k,l)$. In our work, Sobel operator is used for gradient calculations. Let $\alpha(i-1,j-1)$ denote

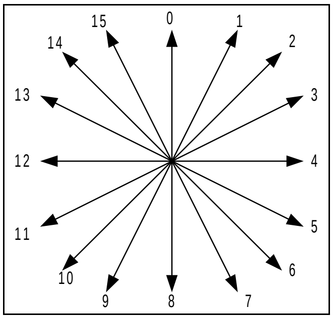


Fig. 4. Poincare index.

the gradient orientation at pixel $(i-1, j-1)$. α can be calculated by the following equation, where $G_y(i-1, j-1)$ and $G_x(i-1, j-1)$ refer to the gradients in y and x directions, respectively:

$$\alpha(i-1, j-1) = \arctan\left(\frac{G_y(i-1, j-1)}{G_x(i-1, j-1)}\right). \tag{6}$$

The encoder then verifies the inequality shown in Eq. (7) for all the pixels in $D(k, l)$, in order to specify the new gradient values that guarantees to preserve fingerprint features while embedding watermark data. In Eq. (7), α_q represents the Poincare index gradient direction associated with the pixel, and $G_{yn}(i-1, j-1)$ and $G_{xn}(i-1, j-1)$ are new values of vertical and horizontal gradients for pixel $(i-1, j-1)$. In fact, the embedding changes the actual gradient orientations, but preserves quantized gradient directions according to the Poincare index intervals. The value of the actual gradient at any pixel can change by at most $\pm\,\pi/8$ rad without altering the quantized Poincare index.

$$\left(\alpha_q - \frac{\pi}{16}\right) < \arctan\left(\frac{G_{yn}(i-1, j-1)}{G_{xn}(i-1, j-1)}\right) < \left(\alpha_q + \frac{\pi}{16}\right). \tag{7}$$

Note that, watermark encoder modifies the watermark embedding strength q of Eq. (1) to preserve the quantized gradient directions of these pixels.

2.2.2. Watermark decoding

Decoding is performed in a manner similar to that formulated for Method 1.

3. Experimental results

In order to evaluate the results obtained by proposed watermarking methods, fingerprint images shown in the left column in Fig. 5, are watermarked according to Methods 1 and 2. These images represent main fingerprint image classes: *tented arch*, *arch*, *right loop*, *left loop and whorl* [8].

The images watermarked by using Method 1 are presented in the middle column of Fig. 5. Watermarking parameters used for this experiment are: $q=0.2$, $A=100$, $B=1000$. The watermark data are the binary representation of the 22 character string *Fingerprint_watermark*. The embedded watermark data size is 156 bits. Thus, a maximum of 7.6% of the raw data pixels are modified during watermark embedding.

The last column of Fig. 5 shows fingerprint images which are watermarked by using Method 2, which utilizes gradient direction analysis. In this method, the number of pixels which satisfy the watermarking criteria (not altering the quantized gradient orientations of eight-neighbor pixels) is considerably small. Experiments showed that approximately 0.8% of image pixels are valid candidates for watermark embedding. Since the watermarking capacity is reduced, the watermark data size is decreased from 156 to 12 bits. In spite of this decrease in the capacity, the latter method has the advantage of not changing any of the fingerprint features which are used later in authentication.

For both of the watermarking methods, watermarked fingerprint images are decoded with 100% decoding accuracy since the watermarking does not change the fingerprint features of the original images. Visibility of the watermark data is kept low as a result of utilizing image adaptive embedding in watermark encoding.

In order to explore the performance of introduced adaptive watermarking method for color images, a number of experiments are performed on color images shown in Fig. 6. *New York and Baboon* images are examples for busy images; whereas *Lena and Sailboat* images are examples for relatively smooth images. Watermark data are embedded into blue channels of 512 × 512, 24 bit, color images. The embedded bits correspond to the binary representation of the text data given below and consists of 463 characters.

> Due to advantages like ease of production, distribution, editing and storing, multimedia data are mainly in digital form. Since digital data can be copied without any loss in fidelity, the protection of intellectual property rights poses a very big problem. Digital watermarks, which can be defined as codes imperceptibly embedded in the host multimedia data to carry information like origin, status or destination of the data, have gained considerable attention.

Watermark data decoding performance of the methods, measured as percentage of accurately decoded characters, is presented in Table 1. To better evaluate the efficiency of the proposed method, experiments are performed on images which are watermarked by (a) a well-known spatial method presented in Ref. [3] (column 1), (b) Method 1 including only standard deviation (SD) term in Eq. (2) (column 2), (c) Method 1 including only gradient magnitude (GM) term in Eq. (2) (column 3), (d) Method 1 including both of the image adaptation terms (column 4) and (e) Method 1 including both of the image adaptation terms and controller at encoder (column 5). Watermarking parameters used in the

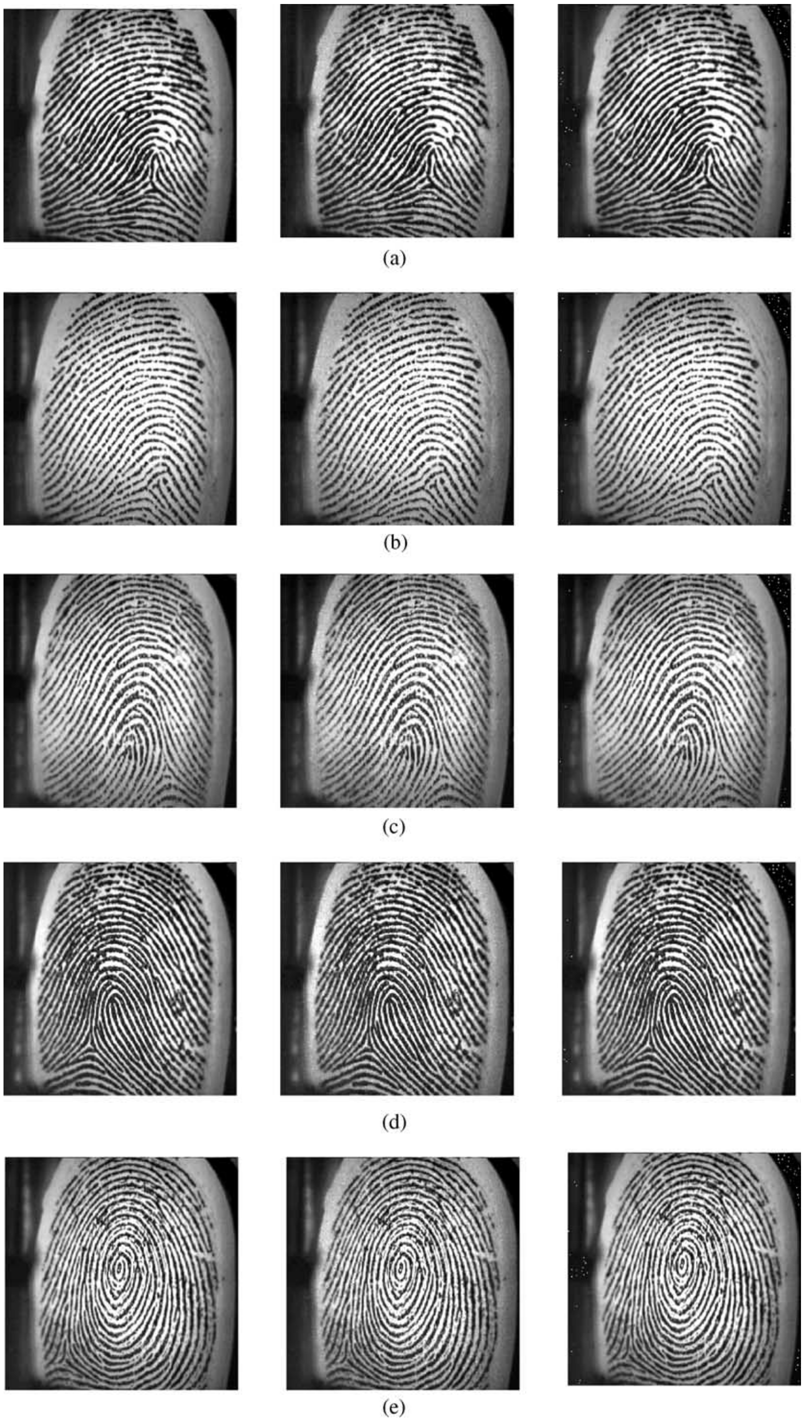
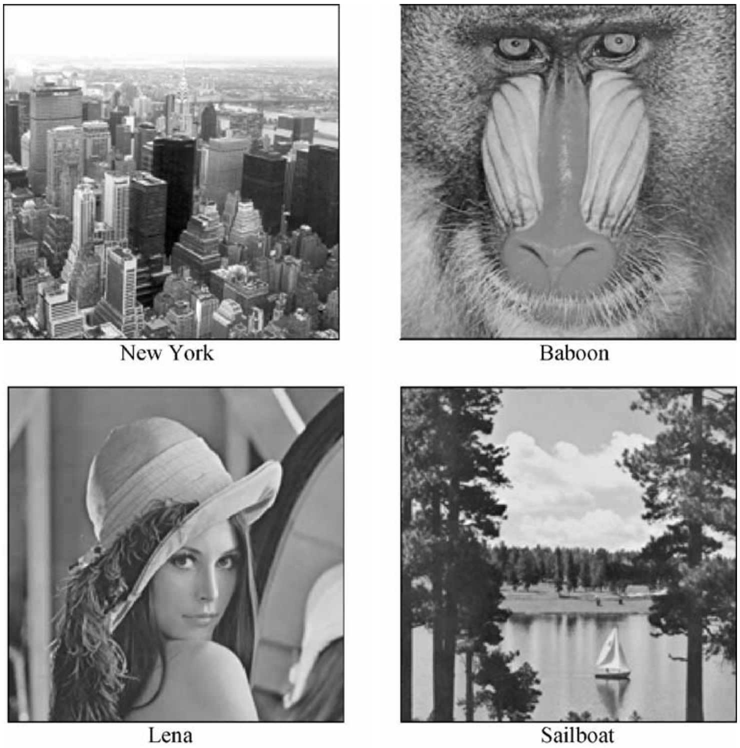


Fig. 5. Left column: Original fingerprint images: (a) Tented arch, (b) Arch, (c) Right loop, (d) Left loop, (e) Whorl. Middle column: Images watermarked by Method 1. Right column: Images watermarked by Method 2.



New York

Baboon

Lena

Sailboat

Fig. 6. Luminance of color images used in the experiments (New York image: Copyright photo courtesy of Patrick Loo, University of Cambridge; others are from USC-SIPI Database).

Table 1
The watermark data decoding performance (%)

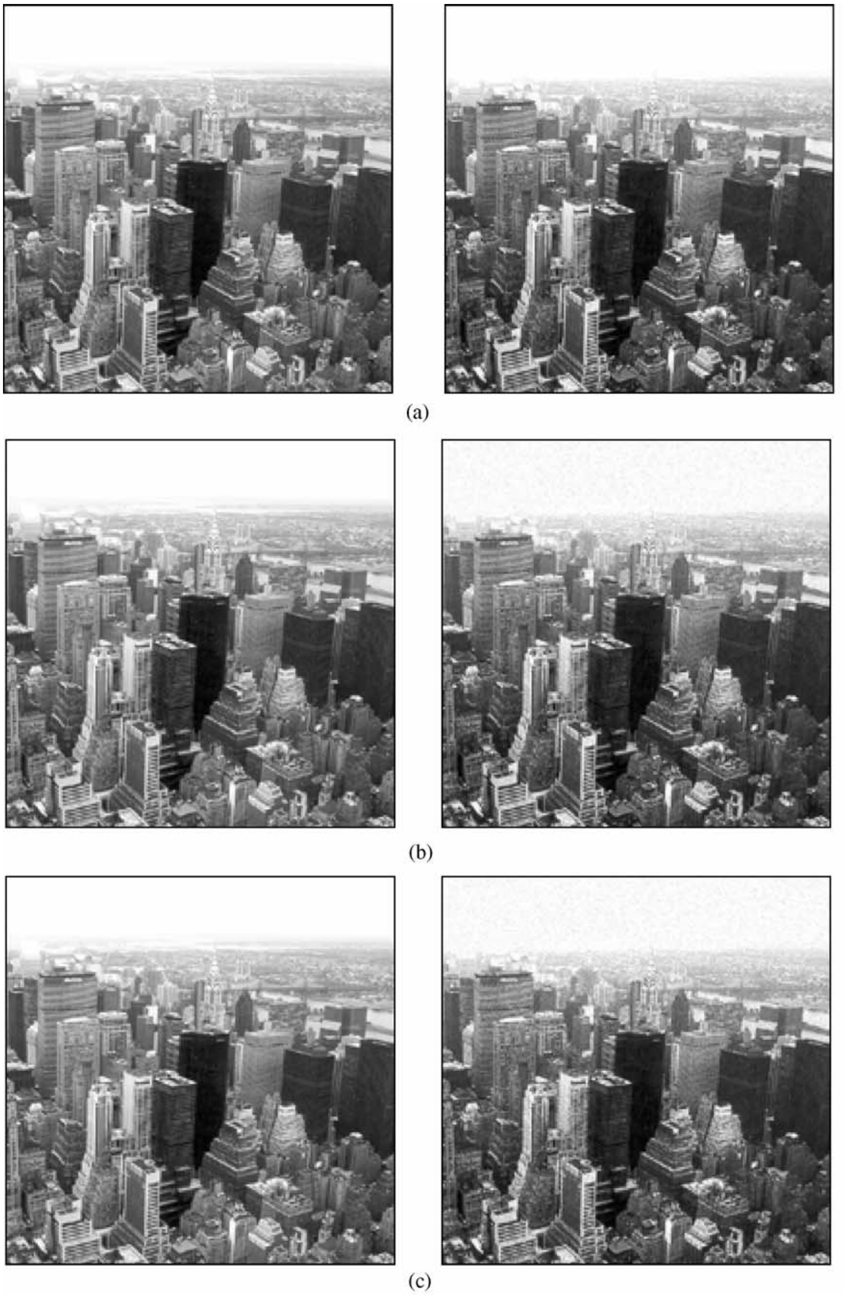
Image	Method in Ref. [1]	Only *SD*	Only *GM*	*SD*+ *GM*	*SD* + *GM*+ Controller
New York	73.87	90.5	83.59	94.82	99.78
Baboon	91.79	96.98	95.9	99.35	100
Lena	99.57	99.57	99.57	99.57	99.78
Sailboat	98.49	99.35	99.14	99.35	99.78

experiments are set to: initial $q = 0.1$, $A = 100$, $B = 1000$. Size of the embedded watermark data is 3243 bits. Totally, 40% of the pixels in the image are modified by watermark embedding.

As can be seen from Table 1, especially the decoding performance for busy images, i.e., *New York* and *Baboon*, increases considerably by using image adaptive and controlled watermark embedding. For relatively smooth images, i.e., *Lena* and *Sailboat*, this increase is small and the performance of the method introduced in Ref. [3] is nearly the same as that of our method. Using image adaptation terms individually, as shown in columns 2 and 3, also increases decoding performance but using both of them and controller leads to the highest performance. The normalization factors, A and B, control the effect of standard deviation and gradient magnitude on watermarking, respectively. Decreasing A or B increases effects of standard deviation and gradient magnitude, respectively; while increasing A or B decreases effects of standard deviation and gradient magnitude, respectively.

For evaluating the visibility of the embedded watermarks, Fig. 7 shows the luminance and blue channel images of (i) original *New York* image, (ii) image watermarked by the method presented in Ref. [3] and (iii) image watermarked by the proposed method, including both of the image adaptation terms and controller. Similarly, Fig. 8 shows the luminance and blue channel images of (i) original *Lena* image, (ii) image watermarked by the method presented in Ref. [3] and (iii) image watermarked by the proposed method, including both of the image adaptation terms and controller. The figures indicate that the watermarks are invisible for the proposed method and the method presented in Ref. [3].

Furthermore, we evaluated watermark data decoding performance of the adaptive watermarking method for cropped images. For this purpose, 50% of the watermarked images are cropped. The performance results obtained from the cropped images are presented in Table 2. Note that smaller the size of embedded data, higher the watermark decoding performance. Because when the size of the watermark data is small, each bit will be embedded into more locations and the effect of cropping on the performance will reduce.



(a)

(b)

(c)

Fig. 7. Luminance (left column) and blue channel (right column) images of (a) original *New York* image, (b) image watermarked by the method presented in Ref. [3], and (c) image watermarked by Method 1.

4. Conclusions

Two image adaptive watermarking methods are introduced for fingerprint image watermarking. The proposed methods preserve fingerprint feature regions either by isolating singular point regions during watermark embedding or adjusting the watermark embedding strength in order to guarantee that gradient directions remain within the analytically computed intervals. Thus, fingerprint images are watermarked without changing the features associated with them. Furthermore, image adaptive watermark embedding rule increases decoding accuracy and satisfies the invisibility criterion. Alternatively, by utilizing gradient direction analysis in watermark embedding,

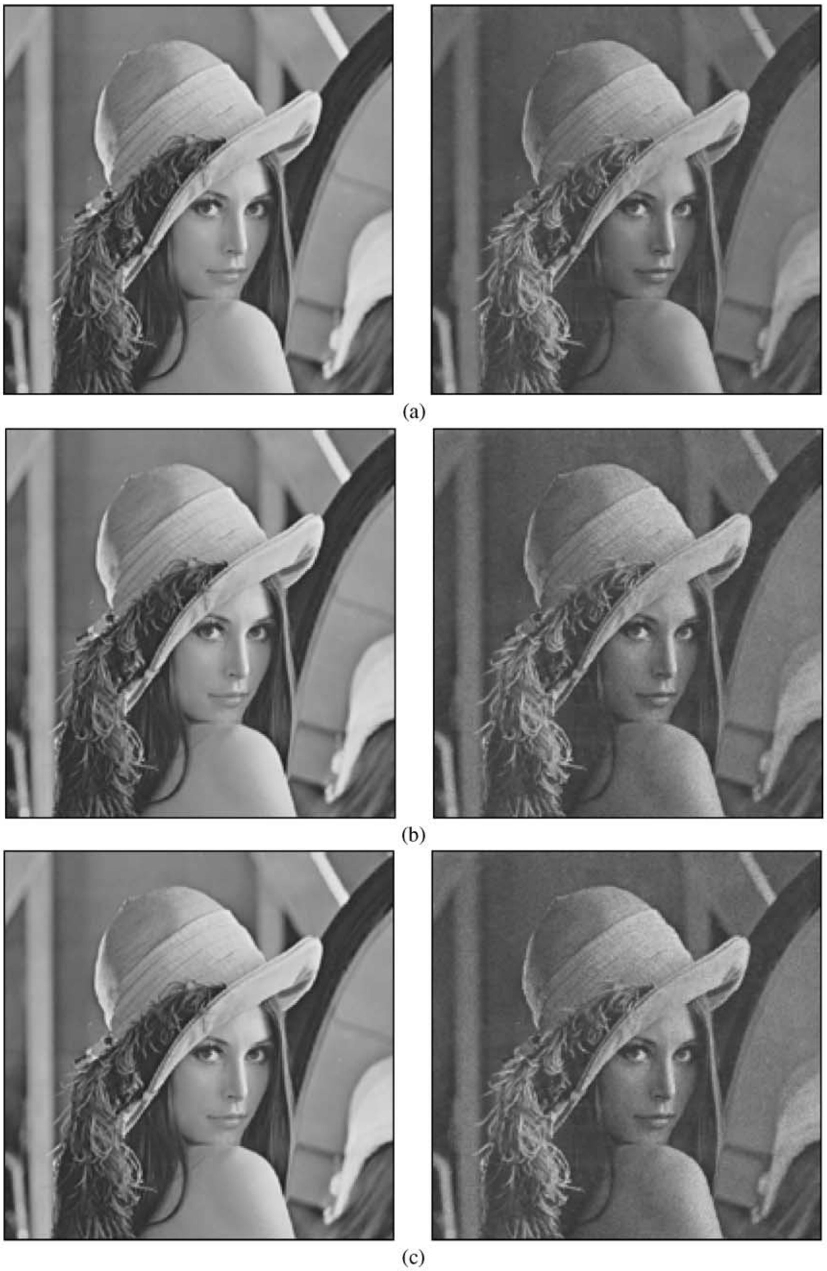


Fig. 8. Luminance (left column) and blue channel (right column) images of (a) original *Lena* image, (b) image watermarked by the method presented in Ref. [3], and (c) image watermarked by Method 1.

none of the fingerprint features used in authentication are altered.

One of the proposed methods, Method 1, is also applicable to watermarking of color images. In the case of color images, by utilizing standard deviation and gradient magnitude properties of the image regions in watermark embedding and by controlling watermark embedding process for correct decoding, the watermark data decoding performance has been increased, especially for textured or busy images. The embedded watermarks are invisible. It is shown that image cropping does not affect watermark data decoding performance considerably.

Table 2
The watermark data decoding performance (%) after cropping attack

Image	$SD + GM$ + Controller
New York	93.74
Baboon	95.03
Lena	93.52
Sailboat	84.02

Acknowledgements

Authors thank Meltem Ballan for her valuable contribution on the extraction of fingerprint feature images.

References

[1] U. Uludag, B. Gunsel, A.M. Tekalp, Robust watermarking of busy images, Proceedings of SPIE Electronic Imaging 2001 Conference, Security and Watermarking of Multimedia Contents III, Vol. 4314, California, USA, 2001, pp. 18–25.

[2] U. Uludag, B. Gunsel, M. Ballan, A spatial method for watermarking of fingerprint images, Proceedings of the First International Workshop on Pattern Recognition in Information Systems, PRIS 2001, Sebutal, Portugal, ICEIS Press, July 2001, pp. 26–33.

[3] F. Hartung, M. Kutter, Multimedia watermarking techniques, Proc. IEEE 87 (7) (1999) 1079–1107.

[4] M.D. Swanson, M. Kobayashi, A.H. Tewfik, Multimedia data embedding and watermarking technologies, Proc. IEEE 86 (6) (1998) 1064–1087.

[5] N.K. Ratha, J.H. Connell, R. Bolle, Secure data hiding in wavelet compressed fingerprint images, Proceedings of ACM Multimedia 2000 Workshops, California, USA, 2000, pp. 127–130.

[6] M. Kawagoe, A. Tojo, Fingerprint pattern classification, Pattern Recognition 17 (3) (1984) 295–303.

[7] A.K. Jain, L. Hong, S. Pankanti, R. Bolle, An identity-authentication system using fingerprints, Proc. IEEE 85 (9) (1997) 1365–1388.

[8] G.T. Candela, P.J. Grother, C.I. Watson, R.A. Wilkinson, C.L. Wilson, A pattern level classification automation system for fingerprints, NISTIR 5647, Technical Report and CD, NIST, 1995.

About the Author—BILGE GUNSEL received M.S. and Ph.D. degrees in Electronics and Communication Engineering from Istanbul Technical University, Istanbul, Turkey in 1988 and 1993, respectively. She is currently working at the Scientific and Technical Research Council of Turkey, Marmara Research Center, Information Technologies Research Institute, as senior researcher. She was a Research Associate in the Electrical Engineering Department at University of Rochester, Rochester, USA from 1995 to 1997. During 1994–1995, she was an Assistant Professor in the Electrical and Electronics Engineering Department at Istanbul Technical University. Her research interests include video/image content analysis and retrieval, stochastic image models and multimedia information systems. Dr. Gunsel is a member of IEEE.

About the Author—UMUT ULUDAG received B.S. and M.S. degrees in Electrical & Electronics Engineering from Bogazici University, Istanbul, Turkey in 1999 and 2001, respectively. He was a researcher in Marmara Research Center, Turkey, from 1999 to 2001. He is now a Ph.D. student in Computer Science and Engineering Department, Michigan State University, East Lansing, MI. His research interests include pattern recognition, multimedia, image processing, computer vision, watermarking and biometrics.

About the Author—A. MURAT TEKALP received B.S. degrees in Electrical Engineering and Mathematics from Bogazici University, Istanbul, Turkey in 1980, M.S. and Ph.D. degrees in Electrical, Computer and Systems Engineering from Rensselaer Polytechnic Institute (RPI), Troy, New York, in 1982 and 1984, respectively. He has worked in Eastman Kodak Company, Rochester, New York from December 1984 to August 1987, and at the Electrical and Computer Engineering Department of the University of Rochester, Rochester, New York since September 1987. He is now with the Koc University, Istanbul, Turkey. He has been a Visiting Assistant Professor at Rensselaer Polytechnic Institute (1987), Visiting Associate Professor at Bilkent University, Ankara, Turkey (1992–1993) and Visiting Professor at Sabanci University (1999–2000). His current research interests are in the area of digital image and video processing, including object-based video representations, motion tracking, image/video segmentation, detection of human subjects in a video scene and recognition of the motion of human subjects, video filtering and restoration, multimedia compression and multimedia content description.

Prof. Tekalp is a senior member of IEEE. He received the NSF Research Initiation Award in 1988, and named as Distinguished Lecturer by IEEE Signal Processing Society in 1998. He has chaired the IEEE Signal Processing Society Technical Committee on Image and Multidimensional Signal Processing (Jan. 1996–Dec. 1997) and is a founding member of the Technical Committee on Multimedia Signal Processing. He has served as an Associate Editor for the IEEE Transactions on Signal Processing (1990–1992), IEEE Transactions on Image Processing (1994–1996), and the Kluwer Journal Multidimensional Systems and Signal Processing (1994–1999). He was an area editor for the Academic Journal Graphical Models and Image Processing (1995–1998). He was also on the editorial board of the Academic Journal Visual Communication and Image Representation (1995–1999). He was appointed as the Technical Program Chair for the 1991 IEEE Signal Processing Society Workshop on Image and Multidimensional Signal Processing, the Special Sessions Chair for the 1995 IEEE International Conference on Image Processing, and Technical Program Chair for IEEE ICASSP 2000. He is the founder and first Chairman of the Rochester Chapter of the IEEE Signal Processing Society. He was elected as the Chair of the Rochester Section of IEEE in 1994–1995.

At present, he is the Editor-in-Chief of the EURASIP journal on Image Communication published by Elsevier Science. He is also the General Chair for IEEE ICIP 2002 to be held in Rochester, New York. He authored the Prentice-Hall book Digital Video Processing (1995). He holds five US patents. His group contributed technology to the ISO/IEC MPEG-4 and MPEG-7 standards.