

IrisCode Decompression Based on the Dependence between its Bit Pairs

Adams Wai Kin Kong, *IEEE Member*

Forensics and Security Laboratory, School of Computer Engineering, Nanyang Technological University,
Nanyang Avenue, Singapore, 639798 (Email: adamskong@ntu.edu.sg)

Abstract — IrisCode is an iris recognition algorithm developed in 1993 and continuously improved by Daugman. Understanding IrisCode’s properties is extremely important because over 60 million persons have been mathematically enrolled by the algorithm. In this paper, IrisCode is proved to be a compression algorithm, which is to say, its templates are compressed iris images. In our experiments, the compression ratio of these images is 1:655. An algorithm is designed to perform this decompression by exploiting a graph composed of the bit pairs in IrisCode, prior knowledge from iris image databases, and the theoretical results. To remove artifacts, two post-processing techniques that carry out optimization in the Fourier domain are developed. Decompressed iris images obtained from two public iris image databases are evaluated by visual comparison, two objective image quality assessment metrics and eight iris recognition methods. The experimental results show that the decompressed iris images retain iris texture, that their quality is roughly equivalent to a JPEG quality factor of ten and that the iris recognition methods can match the original images with the decompressed images. This paper also discusses the impacts of these theoretical and experimental findings on privacy and security.

Keywords: Biometrics, iris recognition, compression, Daugman algorithm, template protection

1. Introduction

¹IrisCode has drawn significant attention in the last two decades [1-3] because of its great success in biometric markets, and because of its computational advantages, including high matching speed, predictable false acceptance rates and robustness against local brightness and contrast variations. Many methods developed for iris and palmprint identification use the ²core of IrisCode to take advantage of these properties [6-20]. Researchers developing these methods generally replace the Gabor filters in

¹In this paper, IrisCode is used interchangeably to refer to both the method and features of iris recognition developed by Daugman. Recently, this method has also been called the Daugman algorithm.

²The core of IrisCode is used to refer to the integrals and the binarization operations in Eqs. 1-4.

IrisCode with different filters and transformations. IrisCode is also used for cancelable biometrics [21-22] to prohibit the recovery of original features from templates. For these reasons, a complete understanding of IrisCode is extremely important. Although numerous scientific papers regarding iris recognition have been published, few works are devoted to the analysis of IrisCode. In the original paper [1], Daugman put significant effort into investigating the imposter distribution. He concluded that it follows a binomial distribution, and bits “0” and “1” in IrisCodes are equally probable. In addition to Daugman, Yao et al. attempted to analyze the bandwidth of Gabor filters, which influence the distributions of the bits [23]. They claimed that the Gabor filters are not perfectly band-pass, and therefore, the “1” and “0” bits in IrisCodes are not equally probable, which contradicts Daugman’s result [1]. Yao et al.’s analysis was based on non-zero DC (direct current) Gabor filters, while Daugman removed the DC components.

³Hollingsworth et al. studied the stability of bits in their iris codes [43], and they identified the best bits. Most theoretical results have been derived by Kong and his coworkers [24-25]. They demonstrated that IrisCode is a clustering algorithm with four prototypes; that the locus of a Gabor function is a two-dimensional ellipse with respect to a phase parameter and can be approximated by a circle in many cases; that the Gabor function can be considered a phase-steerable filter; that the bitwise hamming distance can be regarded as a bitwise phase distance [24]; and that Gabor filters can be used as a Gabor atom detector, and the phase and magnitude of a target Gabor atom can be approximated by the phase and magnitude of the corresponding Gabor response [25]. Nevertheless, their analyses specifically focused on the intra-relationship of bit pairs in IrisCodes.

Many image compression and representation methods depend on Gabor analysis or phase information, which are two important components in IrisCode. Daugman demonstrated that Gabor filters are effective for image compression [26]. Lee extended Gabor filters to Gabor wavelets and proved that under suitable parameterization of Gabor filters, an image can be perfectly decomposed as

$$I = \frac{1}{Z} \sum_j \langle g_j, I \rangle g_j, \text{ where } I \text{ is an image, } Z \text{ is a normalization constant, } g_j \text{ is a Gabor filter and}$$

³ Hollingsworth et al. use 1D log-Gabor wavelets instead of 2D Gabor filters in their study. [43]

$\langle g_j, I \rangle$ is an inner product of g_j and I [27]. Behar et al. showed that images can be reconstructed from localized phase [28]. Furthermore, Kong et al. proved that IrisCode is a clustering algorithm [24]. It is well-known that clustering plays an important role in many compression methods. All these research results imply that IrisCode is a compression algorithm. Nevertheless, these implications have not been noted in the last two decades. If iris images are decompressed from IrisCodes, they can be used to link users enrolling in different iris recognition systems and to attack the systems directly. This deepens our concern for privacy and illegal matching [49-50]. In this paper, we prove that IrisCode is a compression algorithm. We then use a graph composed of the bit pairs in IrisCodes, prior knowledge from iris image databases, and the theoretical results to perform decompression.

We now give a brief computational summary for those who are not familiar with IrisCode generation. Two-dimensional Gabor filters with zero DC are applied to an iris image in a dimensionless polar coordinate system, $I_0(\rho, \phi)$. The complex Gabor response is encoded into two bits by using the following inequalities:

$$b_{jr} = 1 \quad \text{if} \quad \text{Re} \left(\int_{\rho} \int_{\phi} I_0(\rho, \phi) e^{-(r_{j0}-\rho)^2/\alpha_j^2} e^{-(\theta_{j0}-\phi)^2/\beta_j^2} e^{-i\omega_j(\theta_{j0}-\phi)} \rho d\rho d\phi \right) \geq 0, \quad (1)$$

$$b_{jr} = 0 \quad \text{if} \quad \text{Re} \left(\int_{\rho} \int_{\phi} I_0(\rho, \phi) e^{-(r_{j0}-\rho)^2/\alpha_j^2} e^{-(\theta_{j0}-\phi)^2/\beta_j^2} e^{-i\omega_j(\theta_{j0}-\phi)} \rho d\rho d\phi \right) < 0, \quad (2)$$

$$b_{ji} = 1 \quad \text{if} \quad \text{Im} \left(\int_{\rho} \int_{\phi} I_0(\rho, \phi) e^{-(r_{j0}-\rho)^2/\alpha_j^2} e^{-(\theta_{j0}-\phi)^2/\beta_j^2} e^{-i\omega_j(\theta_{j0}-\phi)} \rho d\rho d\phi \right) \geq 0, \quad (3)$$

$$b_{ji} = 0 \quad \text{if} \quad \text{Im} \left(\int_{\rho} \int_{\phi} I_0(\rho, \phi) e^{-(r_{j0}-\rho)^2/\alpha_j^2} e^{-(\theta_{j0}-\phi)^2/\beta_j^2} e^{-i\omega_j(\theta_{j0}-\phi)} \rho d\rho d\phi \right) < 0, \quad (4)$$

where (r_{j0}, θ_{j0}) is the center/location of the filter in the spatial domain, ω_j is the spatial frequency, and α_j and β_j control the shape of the Gaussian function [2]. Daugman uses 1024 Gabor filters with different parameters $(r_{j0}, \theta_{j0}, \omega_j, \alpha_j, \beta_j)$ to generate 1024 bit pairs (b_{jr}, b_{ji}) in IrisCodes and a mask to exclude the corrupted bits from the eyelashes, reflection, eyelids, and from a low signal-to-noise ratio [2]. These four

inequalities quantize phase information and represent it in binary format for high-speed matching. In this paper, we assume that the masked bits of IrisCode are retained.

The rest of this paper is organized as follows. Section 2 proves that IrisCode is a compression algorithm. Section 3 presents a decompression algorithm with two post-processing techniques. Section 4 reports the experimental results obtained from visual comparison, objective image quality metrics and iris recognition methods. Section 5 discusses the impacts of our theoretical and experimental findings.

2. IrisCode — A Compression Algorithm

For clear presentation, a clear set of notation is essential. The real and imaginary parts of a zero DC Gabor filter \mathbf{g}_j with the parameters $(r_{j0}, \theta_{j0}, \omega_j, \alpha_j, \beta_j)$, which generates a bit pair, $B_j = (b_{jr}, b_{ji})$ in an IrisCode, are represented by \mathbf{g}_{rj} and \mathbf{g}_{ij} , respectively. The entire IrisCode is composed of 1,024 bit pairs, i.e., $\{B_1, B_2, \dots, B_{1024}\}$. In Section 3, an additional subscript is employed for the bit pairs to emphasize that they are generated from a particular image (e.g., $B_{kj} = (b_{kjr}, b_{kji})$ is a bit pair generated from the image I_k). For convenience, we use \mathbf{I} to denote $I_0(\rho, \phi)\rho$ and $\langle \mathbf{g}_j, \mathbf{g}_k \rangle = \iint \mathbf{g}_j \times \mathbf{g}_k^* d\rho d\phi$ to denote the inner product of \mathbf{g}_j and \mathbf{g}_k , where * represents a complex conjugate. It is clear that \mathbf{I}_0 can be obtained from \mathbf{I} because ρ can never be zero. Bold font is used to indicate matrices, two-dimensional images and two-dimensional filters. For example, \mathbf{I} represents a two-dimensional image, while I represents the column vector form of the image and \mathbf{g}_{rj} is a real part of a two-dimensional Gabor filter, while g_{rj} is a column vector form of \mathbf{g}_{rj} . Using the vector representation, the inner product can be redefined as $\langle \mathbf{g}_j, \mathbf{g}_k \rangle = \mathbf{g}_k^H \mathbf{g}_j$, where H denotes a complex conjugate transpose. We use T to represent the transpose of a matrix or vector. Thus, the inner product of two real valued vectors (e.g., \mathbf{g}_{rj} and \mathbf{g}_{rk}) can be defined as $\langle \mathbf{g}_{rj}, \mathbf{g}_{rk} \rangle = \mathbf{g}_{rk}^T \mathbf{g}_{rj}$.

In the rest of this section, we prove that IrisCode is a compression algorithm. I can be decomposed as

$$I = \sum_{j=1}^n a_{rj} \mathbf{g}_{rj} + \sum_{j=1}^n a_{ij} \mathbf{g}_{ij} + \sum_{j=1}^m c_j \gamma_j + d\Phi, \quad (5)$$

where a_{rj} , a_{ij} , c_j and $d \in \Re$; $\Phi = [1 \ 1 \cdots 1]^T$; $n=1,024$; γ_j is a unit vector that is orthogonal to Φ and all \mathbf{g}_{rk} and \mathbf{g}_{ik} , i.e., $\langle \mathbf{g}_{r(i)k}, \gamma_j \rangle = 0$ and $\langle \Phi, \gamma_j \rangle = 0$, where $1 \leq k \leq n$, and m is a constant. Mathematically, the vectors $\gamma_1, \dots, \gamma_{m-1}$, and γ_m , which will be estimated through PCA (see Section 3.4), span a subspace in the orthogonal complement of the space spanned by $\mathbf{g}_{r1}, \dots, \mathbf{g}_{rm}, \mathbf{g}_{i1}, \dots, \mathbf{g}_{in}$ and Φ . Eq. 5 is always valid (see the proof in Appendix A). For the sake of convenience, the space spanned by $\mathbf{g}_{r1}, \dots, \mathbf{g}_{rm}, \mathbf{g}_{i1}, \dots, \mathbf{g}_{in}$ and Φ is denoted as Λ . Using Eq. 5, the inner product of $\mathbf{g}_{r(i)k}$ and I can be expressed as

$$\langle \mathbf{g}_{r(i)k}, I \rangle = \sum_{j=1}^n a_{rj} \langle \mathbf{g}_{r(i)k}, \mathbf{g}_{rj} \rangle + \sum_{j=1}^n a_{ij} \langle \mathbf{g}_{r(i)k}, \mathbf{g}_{ij} \rangle + \sum_{j=1}^m c_j \langle \mathbf{g}_{r(i)k}, \gamma_j \rangle + d \langle \mathbf{g}_{r(i)k}, \Phi \rangle. \quad (6)$$

Because $\mathbf{g}_{r(i)k}$ is orthogonal to Φ and all γ_j , Eq. 6 can be simplified as

$$\langle \mathbf{g}_{r(i)k}, I \rangle = \sum_{j=1}^n a_{rj} \langle \mathbf{g}_{r(i)k}, \mathbf{g}_{rj} \rangle + \sum_{j=1}^n a_{ij} \langle \mathbf{g}_{r(i)k}, \mathbf{g}_{ij} \rangle. \quad (7)$$

Using a matrix representation, we obtain $M = \mathbf{G}A$, where

$$M = [\langle \mathbf{g}_{r1}, I \rangle \cdots \langle \mathbf{g}_{rm}, I \rangle \langle \mathbf{g}_{i1}, I \rangle \cdots \langle \mathbf{g}_{in}, I \rangle]^T, \quad A = [a_{r1} \cdots a_{rm} \ a_{i1} \cdots a_{in}]^T \text{ and}$$

$$\mathbf{G} = \begin{bmatrix} \langle \mathbf{g}_{r1}, \mathbf{g}_{r1} \rangle & \cdots & \langle \mathbf{g}_{r1}, \mathbf{g}_{rm} \rangle & \langle \mathbf{g}_{r1}, \mathbf{g}_{i1} \rangle & \cdots & \langle \mathbf{g}_{r1}, \mathbf{g}_{in} \rangle \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \langle \mathbf{g}_{rm}, \mathbf{g}_{r1} \rangle & \cdots & \langle \mathbf{g}_{rm}, \mathbf{g}_{rm} \rangle & \langle \mathbf{g}_{rm}, \mathbf{g}_{i1} \rangle & \cdots & \langle \mathbf{g}_{rm}, \mathbf{g}_{in} \rangle \\ \langle \mathbf{g}_{i1}, \mathbf{g}_{r1} \rangle & \cdots & \langle \mathbf{g}_{i1}, \mathbf{g}_{rm} \rangle & \langle \mathbf{g}_{i1}, \mathbf{g}_{i1} \rangle & \cdots & \langle \mathbf{g}_{i1}, \mathbf{g}_{in} \rangle \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \langle \mathbf{g}_{in}, \mathbf{g}_{r1} \rangle & \cdots & \langle \mathbf{g}_{in}, \mathbf{g}_{rm} \rangle & \langle \mathbf{g}_{in}, \mathbf{g}_{i1} \rangle & \cdots & \langle \mathbf{g}_{in}, \mathbf{g}_{in} \rangle \end{bmatrix}. \quad (8)$$

If \mathbf{G} is invertible, we can substitute $A = \mathbf{G}^{-1}M$ into Eq. 5 and obtain

$$I = [\mathbf{g}_{r1} \cdots \mathbf{g}_{rm} \ \mathbf{g}_{i1} \cdots \mathbf{g}_{in}] \mathbf{G}^{-1}M + \sum_{j=1}^m c_j \gamma_j + d\Phi. \quad (9)$$

Let $[\xi_{r_1} \cdots \xi_{r_m} \xi_{i_1} \cdots \xi_{i_n}] = [\mathbf{g}_{r_1} \cdots \mathbf{g}_{r_m} \mathbf{g}_{i_1} \cdots \mathbf{g}_{i_n}] \mathbf{G}^{-1}$. Now, Eq. 9 can be rewritten as

$$I = \sum_{j=1}^n \langle \mathbf{g}_{r_j}, I \rangle \xi_{r_j} + \sum_{j=1}^n \langle \mathbf{g}_{i_j}, I \rangle \xi_{i_j} + \sum_{j=1}^m c_j \gamma_j + d \Phi. \quad (10)$$

Clearly, $\{\xi_{r_1} \cdots \xi_{r_m} \xi_{i_1} \cdots \xi_{i_n}\}$ and $\{\mathbf{g}_{r_1} \cdots \mathbf{g}_{r_m} \mathbf{g}_{i_1} \cdots \mathbf{g}_{i_n}\}$ form a biorthogonal system. Eq. 10 indicates that IrisCode is a compression algorithm. This compression algorithm uses one bit to store each coefficient of $\xi_{r(i)j}$, which is $\langle \mathbf{g}_{r(i)j}, I \rangle$, but erases all other coefficients. Eq. 10 also indicates that if all $\langle \mathbf{g}_{r(i)j}, I \rangle$ are known, an approximate iris image can be obtained, i.e.,

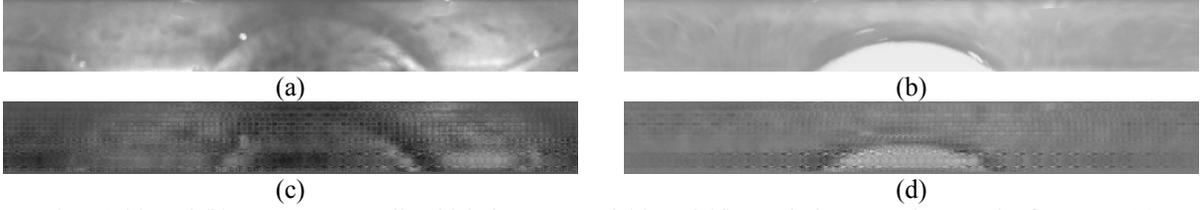
$$I \approx \sum_{j=1}^n \langle \mathbf{g}_{r_j}, I \rangle \xi_{r_j} + \sum_{j=1}^n \langle \mathbf{g}_{i_j}, I \rangle \xi_{i_j}. \quad (11)$$

Eq. 10 is called the IrisCode decompression equation. In this proof, we do not exploit any properties of Gabor filters, and therefore, this theoretical result is applicable to other methods that use the core of IrisCode for personal identification and template protection. These methods are, in fact, compression algorithms, and their templates are compressed biometric images. The only assumption required in the proof is that \mathbf{G} is invertible. Our experimental results show that the \mathbf{G} formed by the Gabor filters in IrisCode is invertible. The invertibility of \mathbf{G} s in other recognition methods should be tested. If they are non-invertible, other numerical methods (e.g., the least squares method) can be used to solve the system of linear equations $M = \mathbf{G} A$. This proof is only valid for the methods that directly use the core of IrisCode. The proof given in Appendix B demonstrates that other coding methods generalized from IrisCode, such as Competitive Code and precise phase representation [8, 24], are also compression algorithms.

Figure 1 shows two resultant images from Eq. 11 and their corresponding original images. The resultant images retain some key features, such as iris texture and eyelids. Because Eq. 11 does not include the brightness component, d , their average intensities are different. In addition, some artifacts are observable.

The compression ratio of IrisCodes is extremely high. Let the size of a normalized iris image be 64 by 512 pixels and one byte be used to store each pixel. The original size of IrisCodes is 256 bytes, and

therefore, the compression ratio is 1:128 ($512 \times 64 \times 1/256$). IrisCodes can be further compressed without a loss of information to about ⁴50 bytes [4], which is even smaller than the size of an 8-by-8 patch in an 8-bit image. Taking into account this significant compression, the compression ratio increases to 1:655 ($512 \times 64 \times 1/50$). Because of this enormously high compression ratio, decompressing IrisCode is challenging.



Figs. 1 (a) and (b) are two normalized iris images, and (c) and (d) are their respective results from Eq. 11.

3. A Decompression Algorithm

We now know that IrisCode is a compression algorithm and that its templates are highly compressed iris images. In this section, a decompression algorithm with two post-processing techniques are proposed. Before presenting this algorithm, it should be mentioned that we are not going to recover the coefficient d

(the brightness of I) and $\left\| \sum_{j=1}^n \langle g_{rj}, I \rangle \xi_{rj} + \sum_{j=1}^n \langle g_{ij}, I \rangle \xi_{ij} + \sum_{j=1}^m c_j \gamma_j \right\|$ (the contrast of I), which highly depend

on illumination environments, because IrisCodes do not store this lighting information, and there are no state-of-the-art iris recognition methods that use this information.

There are four major steps to decompress IrisCodes. First, graphs are used to model IrisCodes from training images (Section 3.1). These graphs will be used to estimate $\langle g_{rj}, I_q \rangle$ and $\langle g_{ij}, I_q \rangle$, where I_q is an image corresponding to an input IrisCode q (Section 3.2). Using Eq. 11, an approximate image can be obtained from the estimated $\langle g_{rj}, I_q \rangle$ and $\langle g_{ij}, I_q \rangle$. Finally, two post-processing techniques are used to remove interference artifacts (Section 3.3) and estimate coefficients c_j in Eq. 10 (Section 3.4). Fig. 2 illustrates the proposed decompression scheme.

⁴ The entropy of IrisCodes is not optimal and therefore, it can be further compressed.

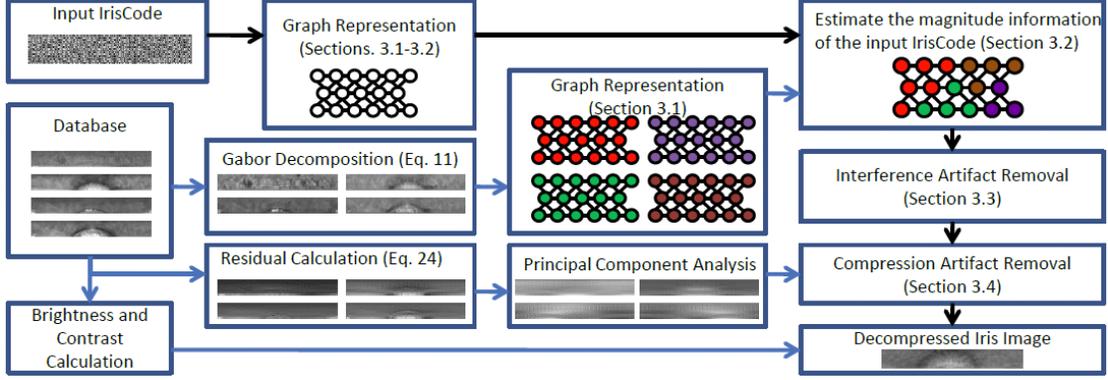


Fig. 2 A schematic diagram of the decompression scheme. The black and blue arrows indicate respectively the information flow of the decompression process and the information flow of the preparation of training images.

3.1. A Graph Formulation

To model the structure of IrisCodes for decompression, a graph, $\Delta = (V, \Theta)$, is used, where V is a set of nodes representing the 1,024 Gabor filters, and Θ is the unordered pairs of the vertices, which are called edges. A node j in the graph stores $S_{kj} = \langle g_j, I_k \rangle / X_k$, where I_k is a training image in a given database

and $X_k = \left\| \sum_{j=1}^n \langle g_{rj}, I_k \rangle \xi_{rj} + \sum_{j=1}^n \langle g_{ij}, I_k \rangle \xi_{ij} \right\|$. The corresponding bit pair, $B_{kj} = (b_{kjr}, b_{kji})$, is computed

by Eqs. 1-4. X_k is the norm of the Gabor components in the image, which depends on the image's contrast. The weight of an edge between nodes j and u is defined as the magnitude of $\langle g_j, g_u \rangle$, which is denoted as $|\langle g_j, g_u \rangle|$. $|\langle g_j, g_u \rangle|$ models the impact of node j on node u and vice versa. If $|\langle g_j, g_u \rangle| = 0$, the nodes are regarded as disconnected. We should remember that g_j and g_u are Gabor filters in the complex analytical form, and the structure of this graph is fixed for all training images because its edges do not depend on I_k . Figure 3 illustrates this graph. We define an edge weight matrix,

$$\mathbf{\Omega} = \begin{bmatrix} |\langle g_1, g_1 \rangle| & \cdots & |\langle g_1, g_n \rangle| \\ \vdots & \ddots & \vdots \\ |\langle g_n, g_1 \rangle| & \cdots & |\langle g_n, g_n \rangle| \end{bmatrix}, \quad (12)$$

which summarizes the connections of the nodes and the weights of the edges. To compare the impacts of different nodes on node j , we sort $|\langle g_j, g_1 \rangle|, \dots, |\langle g_j, g_n \rangle|$ in descending order such

that $|\langle g_j, g_{R(j,1)} \rangle| \geq |\langle g_j, g_{R(j,2)} \rangle| \geq \dots \geq |\langle g_j, g_{R(j,n)} \rangle|$, where R is an indexing function. Because all Gabor filters are normalized, i.e., $\|g_j\|=1$, then $R(j,1) = j$. Using this indexing function, a ranked weight matrix can be defined as

$$\Psi = \begin{bmatrix} |\langle g_1, g_{R(1,1)} \rangle| & \dots & |\langle g_1, g_{R(1,n)} \rangle| \\ \vdots & \ddots & \vdots \\ |\langle g_n, g_{R(n,1)} \rangle| & \dots & |\langle g_n, g_{R(n,n)} \rangle| \end{bmatrix}. \quad (13)$$

Given K training images, we have K graphs whose nodes store different bit pairs and S -values, though they share the same structure, i.e., Ω and Ψ , are the same for all graphs.

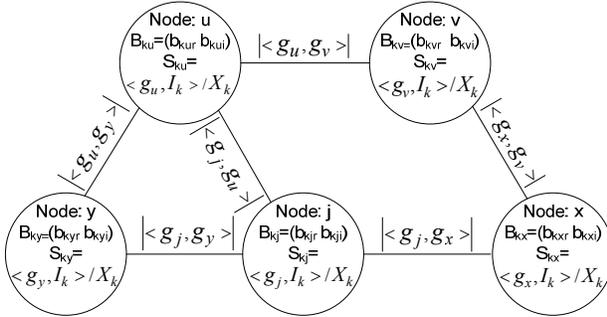


Fig. 3 Illustration of a graph generated from a training image I_k

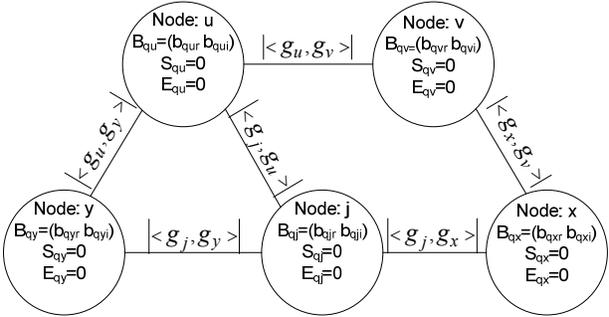


Fig. 4 Illustration of a graph generated from an input IrisCode, q , for decompression at the initial stage.

3.2. A Graph-based Estimation

The graph constructed in the previous subsection indicates that a node is influenced by its adjacent nodes.

This subsection will use these relationships to recover the magnitude of bit pairs ($|\langle g_{ij}, I \rangle|$ and $|\langle g_{ij}, I \rangle|$) in IrisCodes.

Given an IrisCode q consisting of 1024 bit pairs i.e., $\{B_{q1}, B_{q2}, \dots, B_{q1024}\}$, we use the previous notation to construct a graph whose nodes store its bit pairs. Figure 4 illustrates this graph. In the graph, all the S -values, which are Gabor responses in the nodes, and the variables E , which are counters, are set to zero.

Assume we have K graphs from K training images. We use this information to recover the S -values in the graph generated by the input IrisCode. Our proposed algorithm processes node by node.

Figure 5 illustrates this algorithm. Let us consider node j . The bit pairs in the input IrisCode are used as a searching criterion, which is defined as

$$(k_j, T_j) = \arg \max_k \left(\max_{T_k} \left(\sum_{t=1}^{T_k} f(B_{kR(j,t)}, B_{qR(j,t)}) \right) \right), \quad (14)$$

where

$$f(B_{kR(j,t)}, B_{qR(j,t)}) = \begin{cases} 1 & \text{if } b_{kR(j,t)r} = b_{qR(j,t)r} \text{ and } b_{kR(j,t)i} = b_{qR(j,t)i} \\ -\infty & \text{otherwise} \end{cases}, \quad (15)$$

k_j is an index of an optimal graph and T_j is the maximum number of matched bit pairs. This criterion is equivalent to searching a graph in the training database, which has the maximum consecutive bit pairs that match the corresponding bit pairs in the input IrisCode. Once the optimal graph k_j is found, the following updates are performed:

$$S_{qR(j,t)} = S_{qR(j,t)} + \Psi(j,t) \times S_{k_jR(j,t)}, \quad (16)$$

$$E_{qR(j,t)} = E_{qR(j,t)} + \Psi(j,t), \quad (17)$$

where $1 \leq t \leq T_j$ and $\Psi(j,t)$ is the element at row j and column t in Ψ . Eqs. 14-16 guarantee that $Q(\text{Re}(S_{qR(j,t)})) = b_{qR(j,t)r}$ and $Q(\text{Im}(S_{qR(j,t)})) = b_{qR(j,t)i}$, where Q represents the quantization process in Eqs. 1-4. When all 1,024 nodes are processed, all the S -values are normalized by

$$S_{qj} = S_{qj} / E_{qj}. \quad (18)$$

Separating the real and imaginary parts in the S -values, we can obtain the estimated $\langle g_{rj}, I_q \rangle$ and $\langle g_{ij}, I_q \rangle$, respectively denoted as $\langle g_{rj}, \hat{I}_q \rangle$ and $\langle g_{ij}, \hat{I}_q \rangle$, and by using Eq. 11, an approximate I_q can be obtained. The magnitude of the final S_{qj} may differ greatly from the magnitude of $\langle g_{rj}, I_q \rangle$ because the S -values in the database are normalized. To address this issue, we

compute $\hat{X}_q = \left\| \sum_{j=1}^n \langle g_{rj}, \hat{I}_q \rangle \xi_{rj} + \sum_{j=1}^n \langle g_{ij}, \hat{I}_q \rangle \xi_{ij} \right\|$ and rescale $\langle g_{rj}, \hat{I}_q \rangle$ and $\langle g_{ij}, \hat{I}_q \rangle$ by

$\langle g_{rj}, \hat{I}_{q, \text{rescale}} \rangle = \langle g_{rj}, \hat{I}_q \rangle \lambda / \hat{X}_q$ and $\langle g_{ij}, \hat{I}_{q, \text{rescale}} \rangle = \langle g_{ij}, \hat{I}_q \rangle \lambda / \hat{X}_q$, where λ is a given scale

controlling the contrast of the Gabor components in the image. If we know nothing about λ , the simplest

approach is to set $\lambda = \frac{1}{K} \sum_{k=1}^K X_k$. Figure 6 shows two results with their corresponding original images. We

use the DC and the contrast of the original images to display our results to avoid perception differences arising from these two factors. Further experimental results are given in Section 4. We can see that many features (e.g., eyelids and iris texture) are successfully recovered. However, we can also observe many artifacts. Some of these artifacts come from the updated process, while others are due to the missing γ_j in Eq. 10. In the following sub-sections, two post-processing techniques are proposed to remove these artifacts, and \hat{I} represents the resultant images in this subsection.

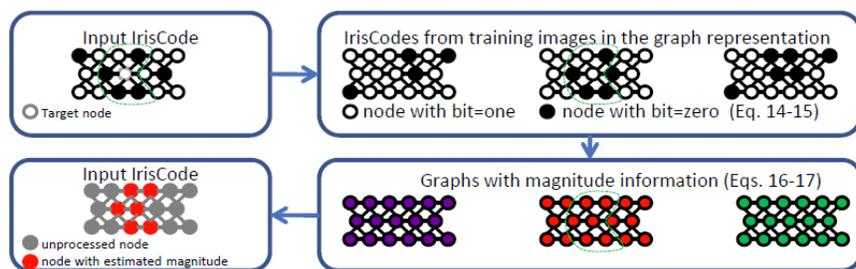


Fig. 5 Illustration of the graph-based estimation.

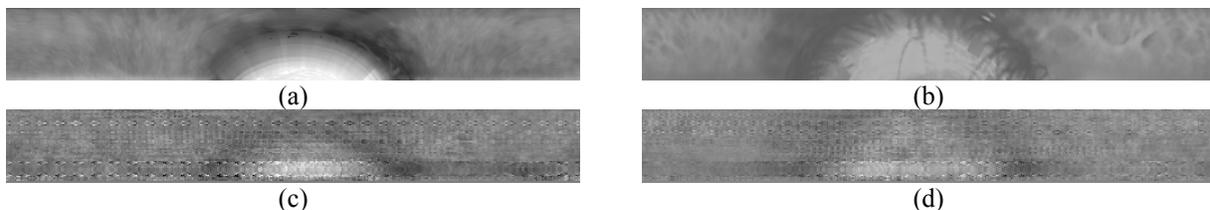


Fig. 6 Decompressed results. (a) and (b) are two original images. (c) and (d) are the respective results from the graph-based estimation algorithm.

In Section 2, we used g_{rj} and g_{ij} to derive the decomposition equation. In this subsection, we use the complex Gabor filters g_j to develop the algorithm. We do not exploit the complex Gabor filters to derive the decomposition equation so that the theoretical result will be applicable to other coding methods [7, 9-20]. However, the complex Gabor filters do add extra information to the algorithm because their phase and magnitude are the approximate phase and magnitude of the target Gabor atoms [25]. Thus, each $S_{k_j R(j,t)}$ can be regarded as one Gabor atom, which gives more precise information than either $\langle g_{rR(j,t)}, I_{k_j} \rangle$ or $\langle g_{iR(j,t)}, I_{k_j} \rangle$ alone.

It is worth mentioning that Eqs. 14-15 search *maximum consecutive instead of maximum* bit pairs that match the corresponding bit pairs in the input IrisCode. This scheme updates simultaneously a group of S -values in the nodes that are *spatially* close to each other to capture their spatial dependence. Figure 7 shows the spatial locations of 31 nodes (the locations of the nodes defined by (r_{j0}, θ_{j0})) in the corresponding Gabor filters). The large dot is the location of a target node, and the small dots are the locations of the top 30 nodes with the greatest impact on the target node. Their grey levels indicate the orders of their impact on the target node. The higher grey levels represent higher impact. Clearly, these orders imply their spatial relationship with the target node. When Eq. 16 updates a group of S -values, it is equivalent to spatially copying a ⁵patch, $U_{k_j} = \frac{1}{X_{k_j}} \sum_{t=1}^{T_j} (\langle g_{rR(j,t)}, I_{k_j} \rangle \xi_{rR(j,t)} + \langle g_{iR(j,t)}, I_{k_j} \rangle \xi_{iR(j,t)})$, from a training image I_{k_j} to \hat{I} . Patch-based methods commonly operate in the spatial domain for texture and face synthesis [44, 47-48], while the proposed algorithm operates in a graph and guarantees that the IrisCode generated from \hat{I} and the input IrisCode are the same.

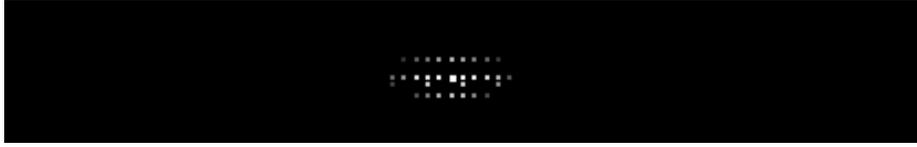


Fig. 7 Illustration of the design of Eqs. 14-16. The large dot is the location of a target node, and the small dots are the locations of the top 30 nodes with the greatest impact on the target node. Their grey levels indicate the orders of their impact on the target node. Some nodes have duplicated spatial locations. We slightly adjusted their locations for display purposes.

The IrisCode masks can be directly applied to Eq. 15 [2]. More precisely, the search constraint based on the masks, i.e., $mask_{kR(j,t)r} = mask_{qR(j,t)r}$ and $mask_{kR(j,t)i} = mask_{qR(j,t)i}$ can be combined with the original search constraint $b_{kR(j,t)r} = b_{qR(j,t)r}$ and $b_{kR(j,t)i} = b_{qR(j,t)i}$. This additional constraint would reduce T_j , the number of nodes updated in each search, and therefore less dependence between the nodes can be captured. To use the masks in this way, a large database is required. As mentioned before, we assume that

⁵ The patch here is denoted as U_{k_j} . It is not a rectangular image patch.

the masked bits are retained. If these are discarded, the masks and the remaining bits should be used to estimate the masked bits.

3.3. Interference Artifact Removal

In this and the next subsections, two post-processing techniques are presented. The first technique is to remove interference artifacts caused by Eq. 16, and the second is to remove the compression artifacts caused by the missing γ_j in Eq. 10. Both techniques are operated in the Fourier domain because these artifacts have a strong periodic behavior. Figure 8 gives the log power spectrums of an original image and the corresponding \hat{I} . We can observe that \hat{I} contains more energy in the high frequency spectrum, and it has 16 frequency bands with very high energy. Thus, post-processing techniques that minimize the high frequency energy are proposed to remove the artifacts.

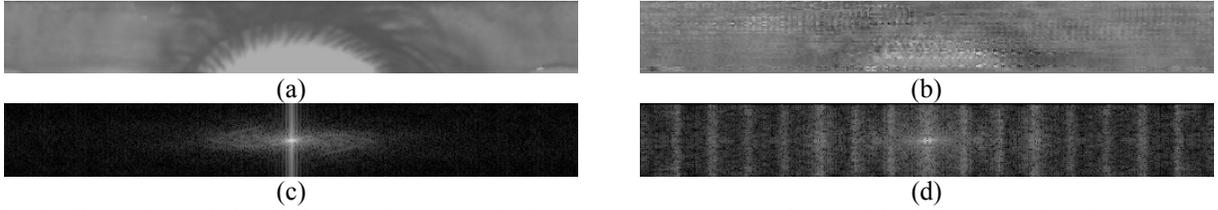


Fig. 8 Illustration of the difference between the log power spectrums of an original image and a decompressed image. (a) is an original image and (b) is the \hat{I} of (a). (c) and (d) are the respective log power spectrums of (a) and (b).

Let $\hat{I} = \sum_{j=1}^n \langle \mathbf{g}_{rj}, \hat{I} \rangle \xi_{rj} + \sum_{j=1}^n \langle \mathbf{g}_{ij}, \hat{I} \rangle \xi_{ij}$ be an estimated image from the previous subsection and

F_{2D} be the two-dimensional Fourier transform. Because F_{2D} is a linear operator,

$$F_{2D}(\hat{I}) = \sum_{j=1}^n \langle \mathbf{g}_{rj}, \hat{I} \rangle F_{2D}(\xi_{rj}) + \sum_{j=1}^n \langle \mathbf{g}_{ij}, \hat{I} \rangle F_{2D}(\xi_{ij}).$$

Note that we consider $F_{2D}(\hat{I})$ to be a vector although

F_{2D} is a two-dimensional transform. Using a matrix representation, we obtain

$$F_{2D}(\hat{I}) = \mathbf{Z}\hat{M}, \quad (19)$$

where $\mathbf{Z} = [F_{2D}(\xi_{r1}) \cdots F_{2D}(\xi_{in})]$ and $\hat{M} = [\langle \mathbf{g}_{r1}, \hat{I} \rangle \cdots \langle \mathbf{g}_{in}, \hat{I} \rangle]^T$. Therefore, the energy of \hat{I} is given

by

$$F_{2D}(\hat{I})^H F_{2D}(\hat{I}) = \hat{M}^H \mathbf{Z}^H \mathbf{Z} \hat{M}, \quad (20)$$

where H represents a complex conjugate transpose. Eq. 20 successfully connects the energy of \hat{I} to \hat{M} . \hat{M} is regarded as an independent variable vector in the minimization of the energy of \hat{I} . However, directly minimizing Eq. 20 is ineffective because it mixes together energies from different frequencies. The frequency is divided into 16 bands, as illustrated in Fig. 9, where only the even bands are labeled. The frequency between the two 16th bands is not considered because this low frequency information (e.g., DC) should not be artifacts.



Fig. 9 Illustration of the 16 frequency bands for artifact removal. The odd bands are not labeled.

The frequency information in each band can be computed by

$$\Gamma_e(F_{2D}(\hat{I})) = [\Gamma_e(F_{2D}(\xi_{r1})) \cdots \Gamma_e(F_{2D}(\xi_{in}))] \hat{M}, \quad (21)$$

where Γ_e is an operator that selects the elements of an input vector in the e^{th} band.

Let $\Gamma_e(\mathbf{Z}) = [\Gamma_e(F_{2D}(\xi_{r1})) \cdots \Gamma_e(F_{2D}(\xi_{in}))]$. The total energy in the e^{th} band is

$$\Gamma_e(F_{2D}(\hat{I}))^H \Gamma_e(F_{2D}(\hat{I})) = \hat{M}^H \Gamma_e(\mathbf{Z})^H \Gamma_e(\mathbf{Z}) \hat{M}. \quad (22)$$

Usually, low frequency components have more energy, but high frequency artifacts are more visually annoying. Thus, a logarithm function and a weighting function are introduced to form our objective function, which is defined as

$$\mu(\hat{M}) = \sum_{e=1}^{16} w(e) \log(\hat{M}^H \Gamma_e(\mathbf{Z})^H \Gamma_e(\mathbf{Z}) \hat{M}), \quad (23)$$

where w is a positive weighting function that emphasizes the minimization of energy in the high frequency bands. It is clear that $\hat{M} = [0 \cdots 0]^T$ is a global optimum, but it is not our target solution. We seek a local optimum that is close to the \hat{M} obtained from the previous subsection to retain the estimated structural information while removing the interference artifacts. μ is differentiable, and therefore, many existing methods are available for this optimization. In our experiments, we simply used the gradient descent method. To guarantee that the input IrisCode and the IrisCode generated from the resultant image

of this post-processing technique are the same, updates in each iteration are checked to ensure that $sign(\hat{M}_t) = sign(\hat{M}_{t-1})$, where t is an iteration index. For the sake of convenience, the resultant images in this subsection are denoted as \bar{I} , and the final \hat{M} is denoted as \tilde{M} . Figure 10 shows two resultant images. These images are smoother and more visually appealing. However, many compression artifacts still exist because of the missing γ_j .

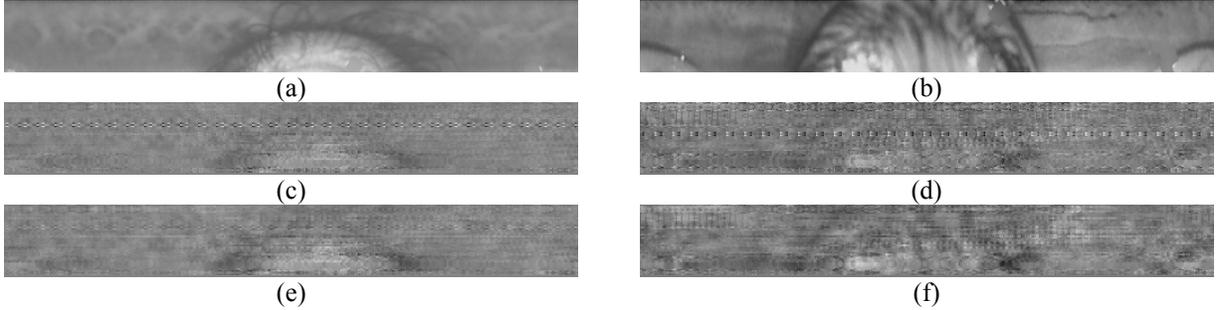


Fig. 10 Results from the post-processing technique for removing interference artifacts. (a) and (b) are original images; (c) and (d) are the corresponding results from Section 3.2; and (e) and (f) are the corresponding results from the post-processing technique in Section 3.3.

3.4. Compression Artifact Removal

In this subsection, $\{\gamma_1, \gamma_2, \dots, \gamma_m\}$ will be constructed, and the corresponding coefficients $\{c_1, c_2, \dots, c_m\}$ in Eq. 10 will be estimated. We should remember that $\gamma_1, \dots, \gamma_{m-1}$ and γ_m are orthogonal to Φ and all g_{rk} and g_{ik} , i.e., $\langle g_{r(i)k}, \gamma_j \rangle = 0$ and $\langle \Phi, \gamma_j \rangle = 0$, where $1 \leq k \leq n$ and $1 \leq j \leq m$. To fulfill these requirements, principal component analysis is used to derive $\{\gamma_1, \gamma_2, \dots, \gamma_m\}$. Given a training image I_k , we first remove all information in ${}^6 \Lambda$, i.e.,

$$\zeta_k = I_k - \sum_{j=1}^n \langle g_{rj}, I_k \rangle \xi_{rj} - \sum_{j=1}^n \langle g_{ij}, I_k \rangle \xi_{ij} - d\Phi. \quad (24)$$

The variances of $\{\zeta_1, \dots, \zeta_K\}$ along the directions of $g_{r1}, \dots, g_{rm}, g_{i1}, \dots, g_{in}$ and Φ are zero, and therefore, the principal components computed from $\{\zeta_1, \dots, \zeta_K\}$ are automatically orthogonal to $g_{r1}, \dots, g_{rm}, g_{i1}, \dots, g_{in}$, and Φ . The space spanned by these principal components is, in fact, within the intersection between the space spanned by the iris images and the orthogonal complement of Λ . The

⁶ We remember that Λ is a space spanned by $g_{r1}, \dots, g_{rm}, g_{i1}, \dots, g_{in}$ and Φ .

computational steps of principal component analysis are well known, so we do not repeat them here. However, we should remember that the computational trick that is always used in Eigenface calculation should be considered to properly handle the memory requirement [29].

Let γ_0 be the mean of $\{\varsigma_1, \dots, \varsigma_K\}$ and $\gamma_1, \gamma_2, \dots$ and $\gamma_{m'}$ be the m' most significant principal components. Then, $\varsigma_v \approx \sum_{j=1}^{m'} c'_{vj} \gamma_j + \gamma_0$, where c'_{vj} is the coefficient of γ_j and $m' < m$. Figure 11 shows the top three principal components and the mean estimated from 800 images and their log power spectrums. The figure clearly demonstrates that the principal components store important information for removing the compression artifacts.

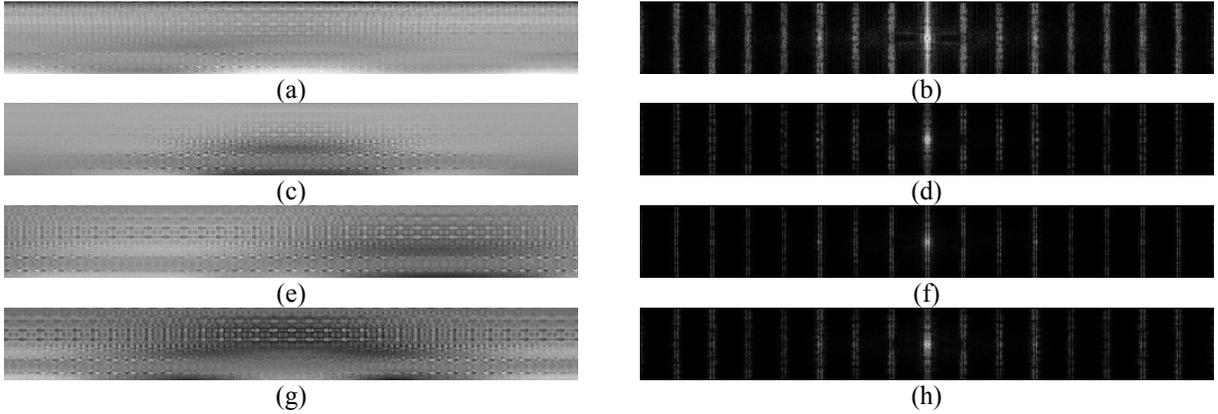


Fig. 11 (a) is the mean of $\{\varsigma_1, \dots, \varsigma_K\}$ and (c), (e) and (g) are the top three principal components; (b), (d), (f) and (h) are the log power spectrums of (a), (c), (e) and (g), respectively.

Using $\gamma_0, \gamma_1, \dots, \gamma_{m'}$ and \tilde{M} , an iris image can be approximated by

$$I \approx [\xi_{r1} \dots \xi_{rm} \xi_{i1} \dots \xi_{in}] \tilde{M} + \sum_{j=1}^{m'} c_j \gamma_j + \gamma_0 + d\Phi. \quad (25)$$

Let $\tilde{I} = [\xi_{r1} \dots \xi_{rm} \xi_{i1} \dots \xi_{in}] \tilde{M} + \gamma_0$, $\mathfrak{Z} = [\gamma_1 \gamma_2 \dots \gamma_{m'}]$ and $C = [c_1 c_2 \dots c_{m'}]^T$. Note that d is zero until the end of the decompression process. Eq. 25 can be rewritten as $I \approx \tilde{I} + \mathfrak{Z}C$. Because the energy of the compression artifacts is concentrated on the ⁷odd frequency bands (see Figs. 9 and 11), we minimize the

⁷The energy of the compression artifacts is concentrated on the odd frequency bands because of the parameters $(r_{j0}, \theta_{j0}, \omega_j, \alpha_j, \beta_j)$ of the Gabor filters. More precisely, the frequency information that cannot be captured by the Gabor filters would remain in the PCA components. If different filters are used, these compression artifacts are different.

energy in only these bands through modifying the coefficient vector C . As with the computational steps in the previous subsection, the energy of $\tilde{I} + \mathfrak{Z}C$ is

$$F_{2D}(\tilde{I} + \mathfrak{Z}C)^H F_{2D}(\tilde{I} + \mathfrak{Z}C) = (C^H \mathfrak{N}^H + F_{2D}(\tilde{I})^H)(\mathfrak{N}C + F_{2D}(\tilde{I})), \quad (26)$$

where $\mathfrak{N} = [F_{2D}(\gamma_1) \cdots F_{2D}(\gamma_m)]$, and the energy in e^{th} band is

$$\Gamma_e(F_{2D}(\tilde{I} + \mathfrak{Z}C))^H \Gamma_e(F_{2D}(\tilde{I} + \mathfrak{Z}C)) = (C^H \Gamma_e(\mathfrak{N})^H + \Gamma_e(F_{2D}(\tilde{I}))^H)(\Gamma_e(\mathfrak{N})C + \Gamma_e(F_{2D}(\tilde{I}))), \quad (27)$$

where $\Gamma_e(\mathfrak{N}) = [\Gamma_e(F_{2D}(\gamma_1)) \cdots \Gamma_e(F_{2D}(\gamma_m))]$. Finally,

$$\mu_1(C) = \sum_{e=1}^8 \log(C^H \Gamma_{2e-1}(\mathfrak{N})^H + \Gamma_{2e-1}(F_{2D}(\tilde{I}))^H)(\Gamma_{2e-1}(\mathfrak{N})C + \Gamma_{2e-1}(F_{2D}(\tilde{I}))), \quad (28)$$

is used as an objective function to minimize the compression artifacts. The weighting function w in Eq. 23 is not included in Eq. 28 because the compression artifacts in different odd bands are equally important. The weighting function was, in fact, examined for Eq. 28. However, its contribution is limited. As with the objective function in Eq. 23, this objective function is also differentiable, but the target solution of Eq. 28 is a near global optimal. Figure 12 shows two final resultant images. These images demonstrate that the proposed post-processing technique effectively weakens the compression artifacts. Further experimental results are given in Section 4. For the sake of convenience, the resultant images in this subsection are denoted as I_f .

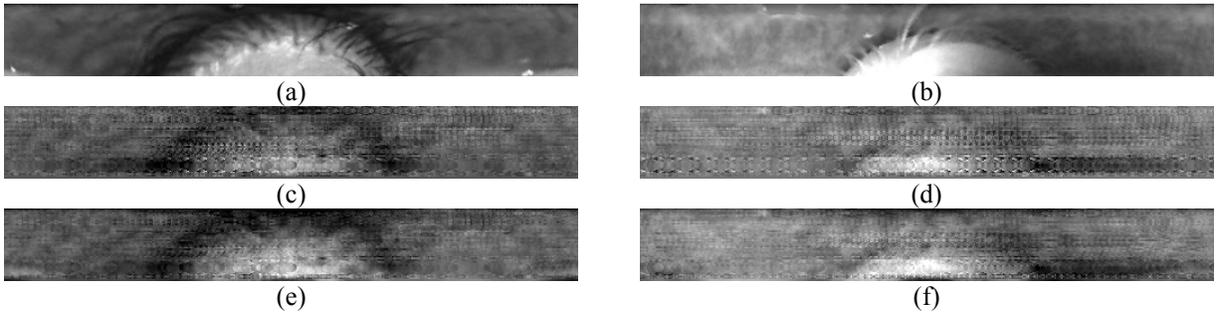


Fig. 12 Final resultant images. (a) and (b) are original images; (c) and (d) are the respective results from the post-processing technique in Section 3.3; and (e) and (f) are the final results.

4. Experimental Results

Two public iris databases, the West Virginia University (WVU) iris database and the UBIRIS.v1 database [30-31], were used to examine the proposed algorithms. ⁸The WVU iris database contains 3,099 iris images from 472 irises, and the UBIRIS.v1 database contains 1,877 images from 241 irises. All the images in the WVU iris database were employed in the experiments. However, 48 images from the UBIRIS.v1 database were automatically removed because of their poor quality (some images did not even have irises). Figure 13 gives examples of the removed iris images. Even though some extremely low quality images were removed, many challenging iris images remained for evaluation (see Fig. 14). The WVU iris images were captured under an infrared lighting environment, while the UBIRIS.v1 iris images were captured under a visible lighting environment. The original images in the UBIRIS.v1 database are color images. We only employed their red component for evaluation because iris texture in this channel is the clearest (see Fig. 15).

In the experiments, we decompressed IrisCodes from these two databases. Given an IrisCode q , we used all images from other irises in the same database to estimate $\langle g_{rj}, I_q \rangle$ and $\langle g_{ij}, I_q \rangle$ and eight hundred images from other irises in the same database to compute the principal components. The top 30 principal components were used as $\gamma_1, \gamma_2 \dots$, and γ_m . It should be emphasized that we did not exploit any information from the same eye to perform the decompression. We employed a leave-one-out cross validation scheme. The decidability index was used as an objective function to optimize the parameters of the Gabor filters [24].

To validate our results, we computed the IrisCodes from the original images and the corresponding resultant images, and we performed a bit-by-bit comparison. As the theoretical predication, we found that the IrisCodes were completely identical, without any bit error.

⁸ Some mislabeled images were corrected.



Fig. 13 Examples of the removed iris images in the UBIRIS.v1 database



Fig. 14 Examples of low quality iris images for evaluation

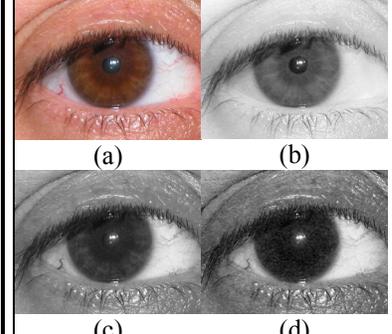


Fig. 15 Iris texture in different channels. (a) is a color image; (b)-(d) are the R, G and B components of (a), respectively

4.1. Evaluation of the Post-Processing Techniques

Some results have already been given in Section 3 for the subjective evaluation of the effectiveness of the post-processing techniques. In this subsection, an objective evaluation is reported. The error norm ratio was employed as an objective measure. To compare the results with the original images, the original contrast and DC were employed in this experiment. Figure 16 shows the distributions of $\|I - \hat{I}\| / \|I_{DC}\|$,

$\|I - \bar{I}\| / \|I_{DC}\|$, $\|I - I_f\| / \|I_{DC}\|$ and $\|I - I_M\| / \|I_{DC}\|$, where I is an original image, $I_{DC} = I - d\Phi$, and

$I_M = \sum_{j=1}^n \langle g_{rj}, I \rangle \xi_{rj} + \sum_{j=1}^n \langle g_{ij}, I \rangle \xi_{ij} + d\Phi$. We remember that \hat{I} , \bar{I} and I_f are the results of Sections

3.2-3.4, respectively. $\|I_{DC}\|$ was used as a normalizer because it measures non-DC energy. Floating point

variables were utilized to calculate $\langle g_{rj}, I \rangle$ and $\langle g_{ij}, I \rangle$ in I_M . Figure 16 clearly demonstrates the

effectiveness of the post-processing techniques. These techniques successfully reduced the error norm

ratios. Figure 16 also shows that for the UBIRIS.v1 database, many of our final results are even better

than I_M , while for the WVU iris database, a good portion of our final results are comparable with I_M .

Some I_f have greater error norm ratios because the low frequency information (e.g., that in the 16th

frequency band (see Fig. 9)) in the original images is not well matched with the low frequency

information in the principal components. The proposed post-processing algorithm in Section 3.4

minimizes the compression artifacts but does not recover the low frequency information in the original

images. It should be emphasized that I_M was computed from 2,048 floating point numbers, while our results were obtained from IrisCodes, which have an optimal size of only about 50 bytes [4].

4.2. Visual Quality Assessment

Figure 17 gives more results for visual comparison. As with the previous figures, the DC and the contrast of the original images are used to display the results. Once again, they show the effectiveness of the proposed algorithms. Features such as iris texture and eyelids can be observed.

In addition to subjective visual comparison, two state-of-the-art image quality metrics — Visual Information Fidelity (VIF) [32] and Information Fidelity Criterion (IFC) [33] — were employed to objectively quantify the visual quality of our results. VIF and IFC both use a statistical model and conditional mutual information to measure differences between a reference image and a distorted image. VIF is based on a human visual system model. IFC is parameter-free. These two metrics were designed for full-reference evaluation, meaning that a complete reference image is assumed to be known. In our evaluation, the original iris images were considered as reference images, and our results were considered as distorted images. Iris images compressed by the JPEG algorithm with quality factors (QF) of 20, 10 and 5 were employed for comparison. The VIF and IFC values of each image in these two iris databases were calculated. Their means, summarized in Table 1, indicate that for the UBIRIS.v1 database, the quality of our results is between the JPEG QFs of 10 and 5, while for the WVU iris database, the quality of our results is between the JPEG QFs of 20 and 10. In this experiment, the file sizes of the JPEG images with QFs of 10 and 5 were about 1,140 and 890 bytes, respectively. Our results were decompressed from IrisCodes that have an optimal size of about 50 bytes, which is even smaller than an 8-by-8 patch in an 8-bit image. The VIF and IFC values and the file sizes clearly indicate that the density of iris image information in IrisCodes is extremely high — $81.92 (512 \times 64 / (50 \times 8))$ pixels per bit in our experiments.

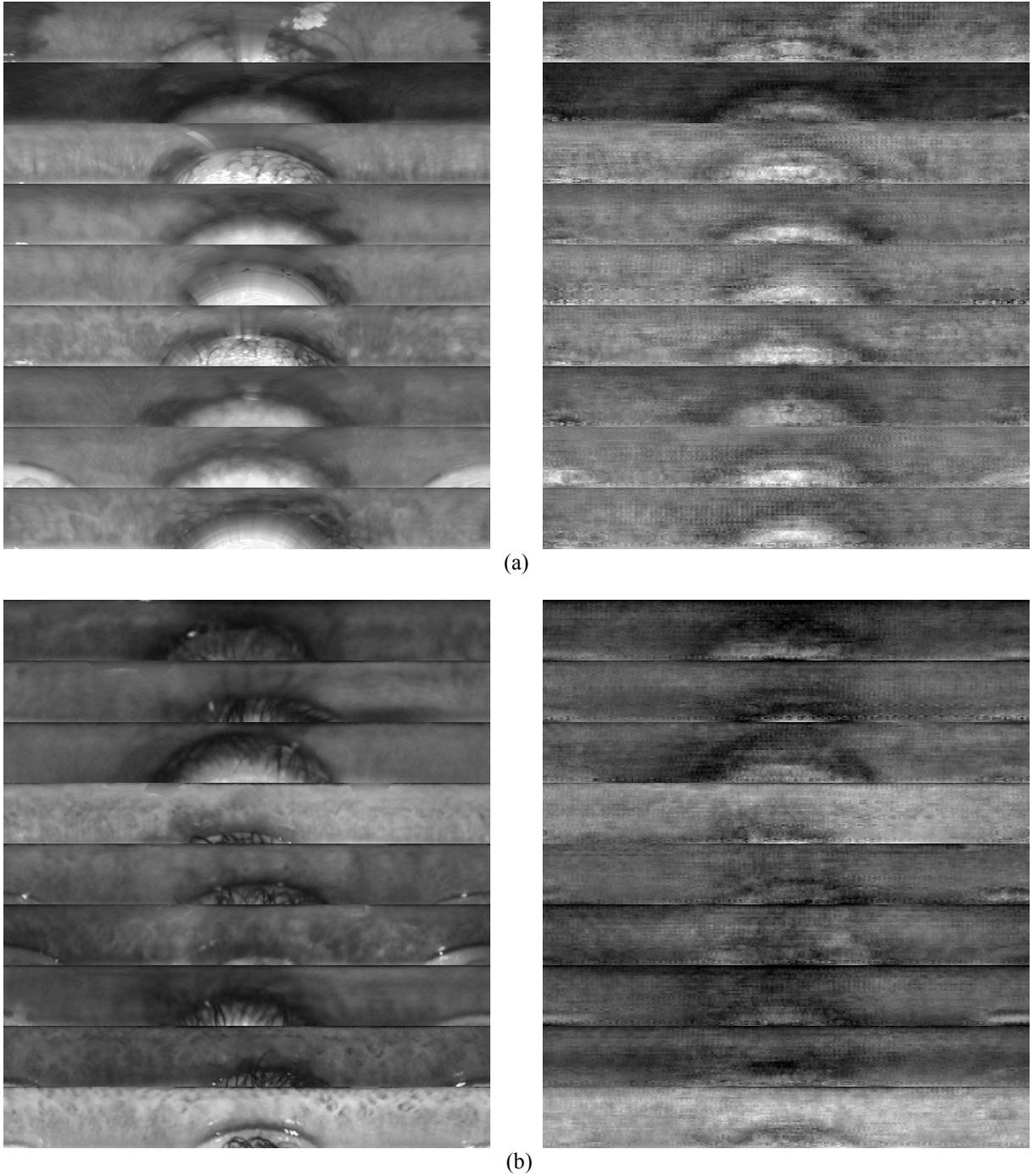


Fig. 17 Results for visual comparison. The first column presents the original images, and the second column presents the results from the proposed algorithms. (a) results from the UBIRIS.v1 database and (b) results from the WVU iris database

Table 1 Image Quality Assessment based on VIF and IFC

Database	Quality metrics	JPEG (QF=20)	JPEG (QF=10)	JPEG (QF=5)	Result
UBIRIS.v1	VIF	0.0826	0.0707	0.0548	0.0650
	IFC	0.2861	0.2423	0.1871	0.2200
WVU	VIF	0.0564	0.0447	0.0399	0.0566
	IFC	0.2593	0.2021	0.1708	0.2322

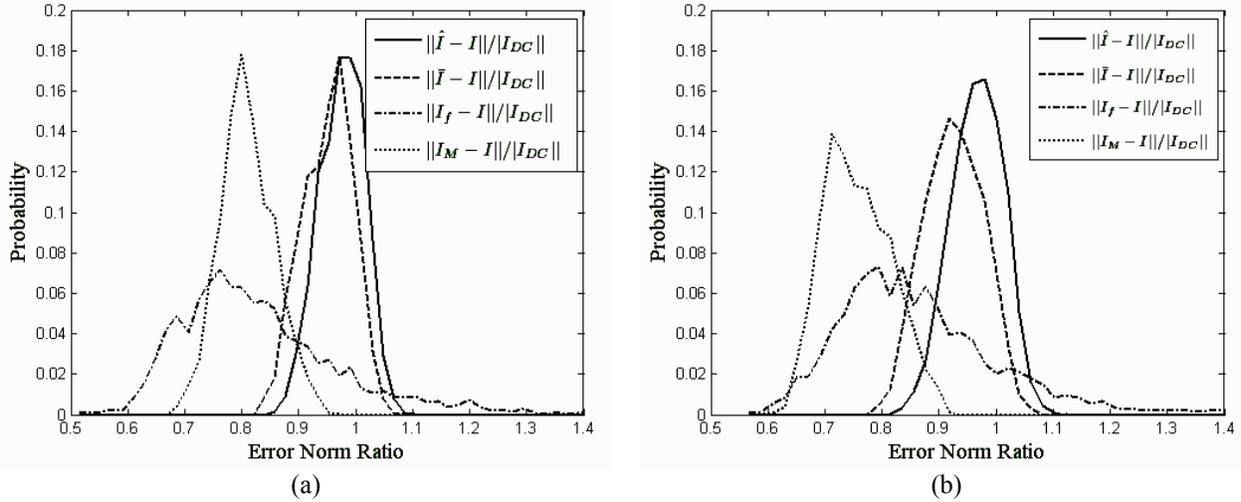


Fig. 16 Error norm ratio distributions from (a) the UBIRIS.v1 database and (b) the WVU iris database

4.3. Evaluation by Other Iris Recognition Methods

In this subsection, eight iris recognition methods were re-implemented to evaluate our results in terms of Receiver Operating Characteristic (ROC) curves: SVM [36]; multiple signature [37]; Ordinal Code (Dilobe $d=5$ and $d=9$ and Tri-lobe $d=9$ and $d=17$) [20]; zero-crossing [35]; and DCT-based [34] methods. Masks that denoted noise pixels, such as eyelids and eyelashes, were used to exclude corrupted bits in the Ordinal Code [20] and zero crossing methods [35]. Their raw hamming distances were rescaled to those in IrisCode [24]. The masks were also used in the SVM method, and 90 images from each database were used to train the SVM with a Gaussian kernel [36]. Two hundred images were employed to estimate preliminary threshold sets for the multiple signature method, and the step size for the exhaustive search was 0.01 [37]. The optimal threshold sets were then selected from these preliminary threshold sets. The testing images were resized for the methods that required different sizes of normalized iris images. The original contrast and DC were not used to decompress IrisCodes in these experiments.

In each set of experiments, cross-matching between all original images was first performed, and the corresponding genuine and imposter distributions were estimated. Then, our results were matched with their parent images. We also matched our results with all other iris images from the same iris, and we obtained the corresponding genuine distribution. As with the visual quality assessment, iris images compressed by the JPEG algorithm with QFs of 20, 10 and 5 were employed for comparison. We

matched the JPEG images with the original images from the same iris, but we did not allow them to match their parent images. In total, for each method and each database, three genuine distributions were obtained from matching the JPEG images. The imposter distributions from the cross-matching between the original images and all the genuine distributions were used to plot ROC curves. It should be emphasized that for the same database and the same method, all the ROC curves were generated from the same imposter distribution that determines the false acceptance rate of a system when a threshold is given. For the sake of convenience, the ROC curves from the cross-matching between the original images are called original ROC curves; the ROC curves from matching our results with their parent images are called parent ROC curves; the ROC curves from matching our results with the original images are called resultant ROC curves; and the ROC curves from matching the JPEG images with a quality factor of X with the original images are called JPEG (QF=X) ROC curves.

Figures 18-22 show, respectively, the ROC curves of the SVM [36], multiple signature [37], Ordinal Code [20], zero-crossing [35], and DCT-based [34] methods. The parent ROC curves help us note the differences between our results and the original images in terms of recognition errors. All the parent ROC curves are higher than the corresponding original ROC curves. The GARs (genuine acceptance rates) of the 14 parent ROC curves are very close to 100% when their FARs (false acceptance rates) are 0.001, indicating that the errors caused by the decompression process are insignificant compared with the errors from the original images. The resultant ROC curves indicate the cumulative errors from the original images and the decompression process. Three of the resultant ROC curves are above or very close to the JPEG (QF=20) ROC curves; seven are between the JPEG (QF=20) ROC curves and the JPEG (QF=10) ROC curves; and only one resultant ROC curve is below the JPEG (QF=5) ROC curve; the rest are between the JPEG (QF=10) ROC curves and the JPEG (QF=5) ROC curves. These results suggest that, in terms of ROC curves, the quality of the decompressed images is a JPEG quality factor of approximately 10.

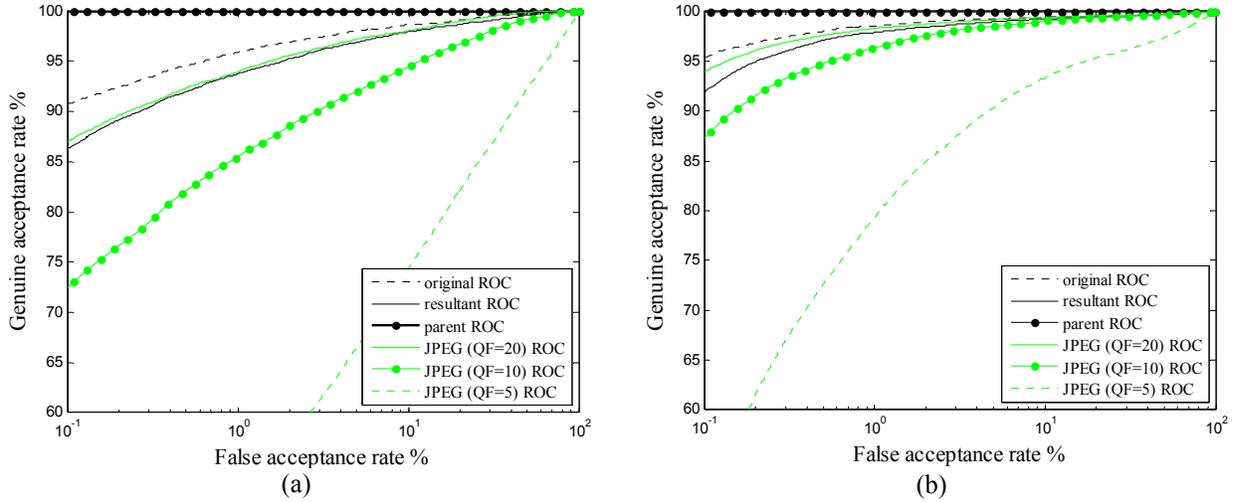


Fig. 18 ROC curves of the SVM method. (a) presents the results from the UBIRIS.v1 database and (b) presents the results from the WVU iris database. (color figure)

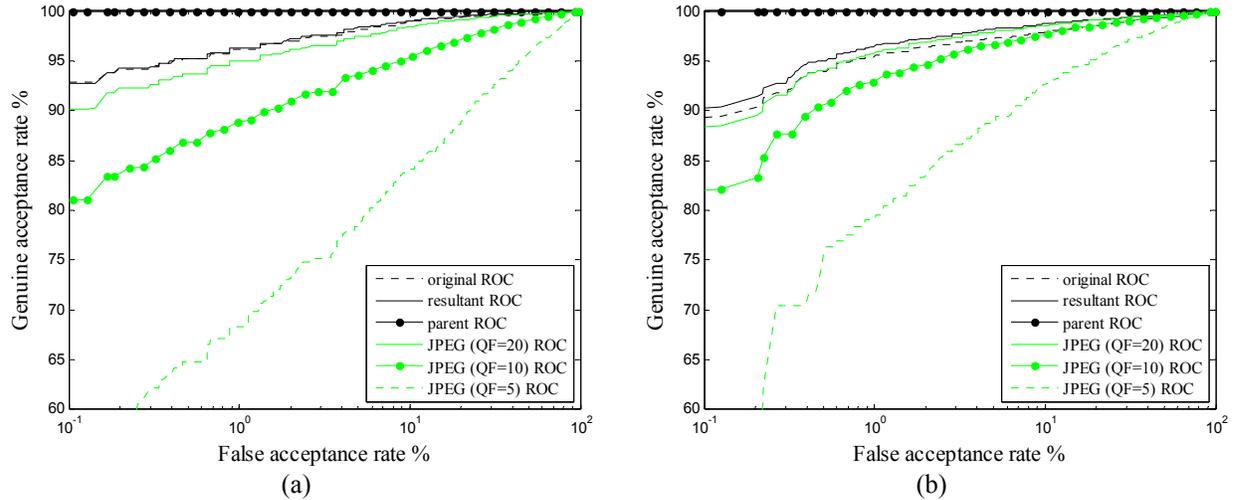
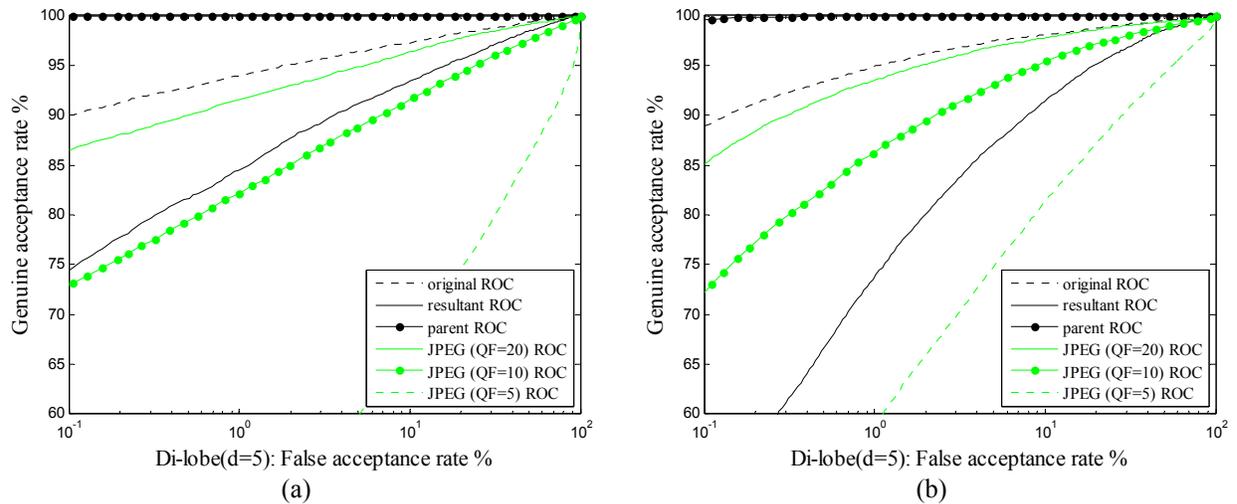


Fig. 19 ROC curves of the multiple signature method. (a) presents the results from the UBIRIS.v1 database and (b) presents the results from the WVU iris database. (color figure)



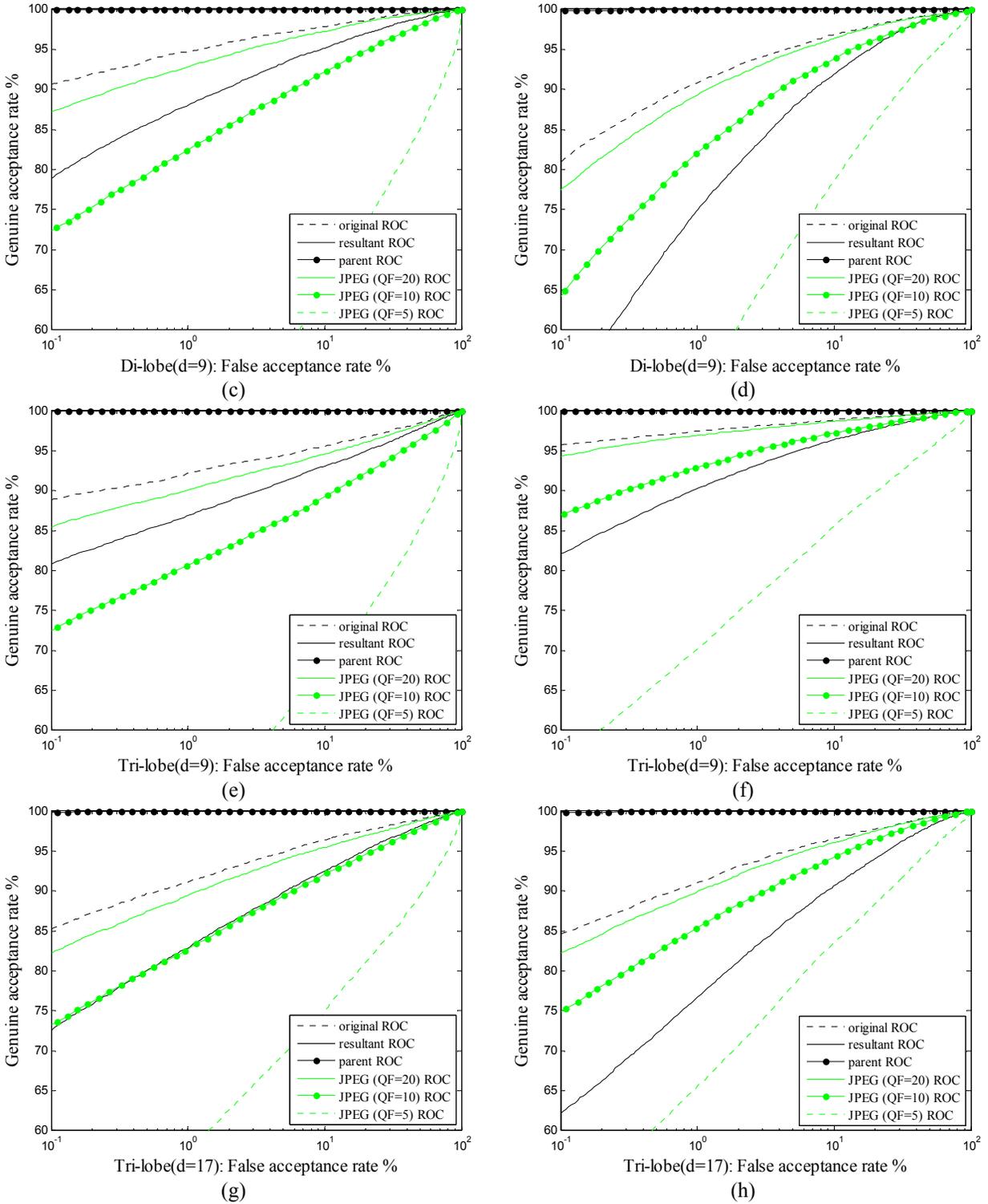


Fig. 20 ROC curves of the Ordinal Code method. The first column presents the results from the UBIRIS.v1 database, and the second column presents the results from the WVU iris database. Rows 1-4 are the results of di-lobe ($d=5$), di-lobe ($d=9$), tri-lobe ($d=7$) and tri-lobe ($d=13$), respectively. (color figure)

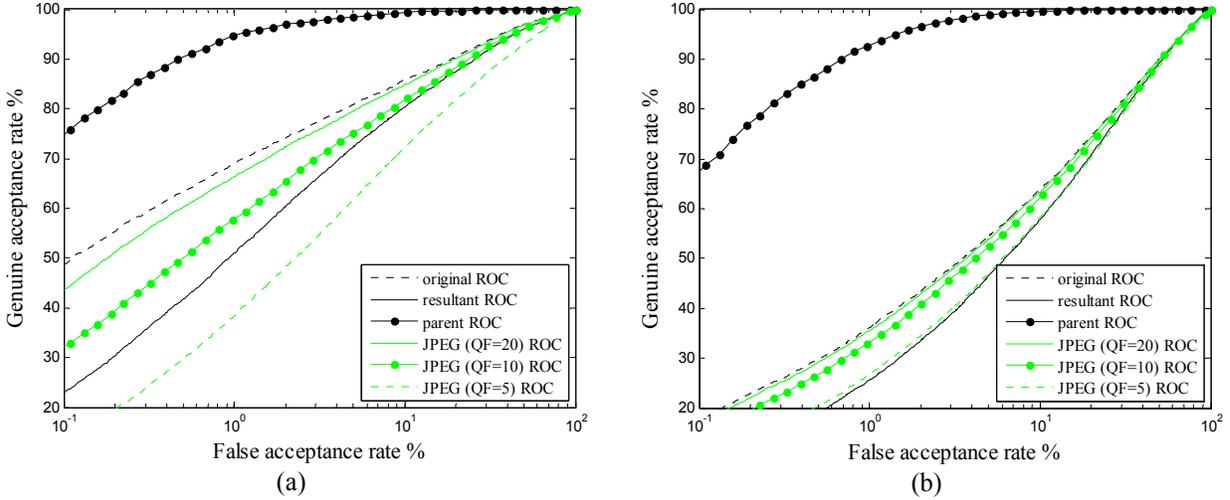


Fig. 21 ROC curves of the zero-crossing method. (a) presents the results from the UBIRIS.v1 database and (b) presents the results from the WVU iris database. (color figure)

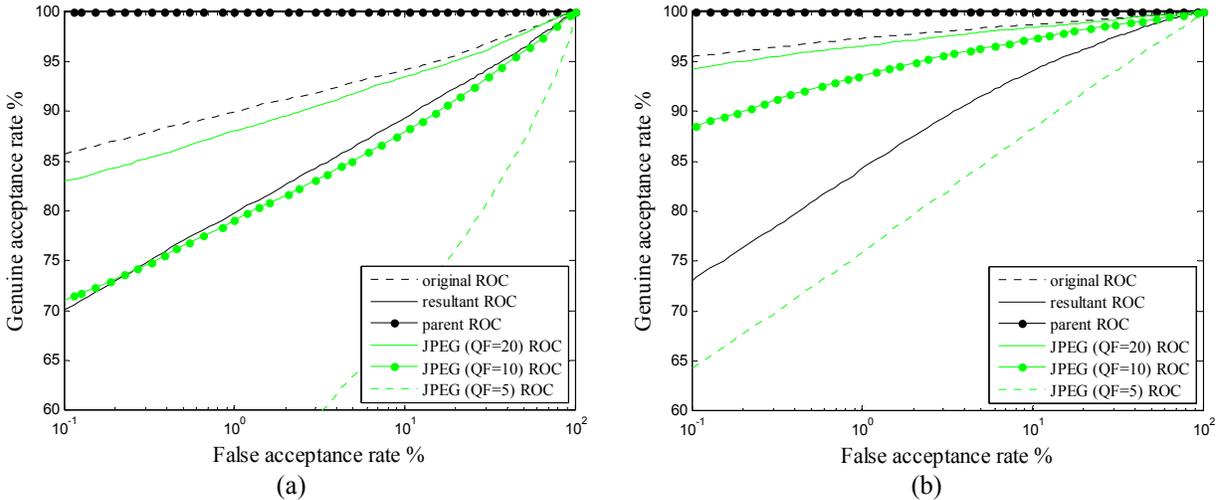


Fig. 22 ROC curves of the DCT-based method. (a) presents the results from the UBIRIS.v1 database and (b) presents the results from the WVU iris database. (color figure)

5. Discussion and Future Work

The primary aim of this paper is to provide a deeper theoretical understanding of IrisCode, but it does impact various aspects of iris recognition. Current commercial iris recognition systems randomly flip or place the bits in IrisCode to form cancelable IrisCodes [39] for privacy protection — templates of the same user registered in different systems cannot be matched — and for system security — compromised templates (stolen templates) cannot be matched with re-issued templates. These cancelable transforms are invertible, if their keys are known, which is not a significant problem, even if attackers obtain the keys, because the re-issued templates will match neither their compromised templates nor their original

templates. Thus, the systems are still protected. However, our results show that once attackers obtain the original IrisCodes and the ⁹Gabor parameters, they can obtain the corresponding iris images. These images can be recognized by IrisCode and even other methods, which implies that they can be used to attack systems running IrisCode and different iris recognition methods at various levels [38] (e.g., the sensor level) if they are not well protected. Furthermore, these images can be used to link up users enrolling in different iris recognition systems. As a result, privacy can be highly compromised. According to the theoretical and experimental results, we recommend that biometric templates, including IrisCodes and all templates generated by similar methods, should be protected at the same security level as enrollment images.

This work not only shows that IrisCode is a compression algorithm, but it also helps us further understanding the relationship between IrisCode and the related methods. According to Eq. 10, Kong et al.'s precise phase representation [24] enhances the performance of IrisCode by allocating more bits to each $\langle g_{r(i)j}, I \rangle$, which is equivalent to extracting more information from the subspace spanned by $g_{r1}, \dots, g_{rm}, g_{i1}, \dots, g_{in}$. Other methods that replace the Gabor filters in IrisCode with other linear filters either extract information from different subspaces or use different bases to represent the same subspace. Designing new linear filters to replace the Gabor filters in IrisCode is equivalent to searching an optimal basis.

Synthesizing iris images is an important topic in iris recognition [5, 40-42]. Synthesizing saves time and effort used to collect a large number of iris images for algorithm evaluation and development. Once an IrisCode is synthesized, we can use the algorithm to obtain an iris image with a predefined IrisCode. However, decompressing IrisCodes and synthesizing iris images are two different problems. The bits in an IrisCode and its mask are the only information available for decompressing it, while we can generate IrisCodes, masks and other information inside and outside Λ for synthesizing iris images.

⁹ The Gabor parameters are not a secret, because one can estimate them from public iris databases or use a hill climbing approach to obtain them if they have a system running IrisCode.

Developing algorithms to synthesize IrisCodes, masks and other iris information to generate high quality iris images is a potential research direction.

For administrative reasons, compressed enrollment images are always stored [45-46]. If the quality of our results can be further improved, for example, to a JPEG quality factor of 50, the necessity of storing these images should be discussed. Using the masks of IrisCodes and other statistical approaches to infer information inside and outside Λ should be considered for further improvement.

The theoretical results reported in this paper are applicable to biometric methods that use the core of IrisCode. More clearly, all the coding methods using the feature extractor generalized from IrisCode are compression algorithms [24]. However, the post-processing techniques must be modified because they have different interference and compression artifacts. For example, the methods that encode the orientation fields of palmprints as features definitely have different artifacts [8-9]. The quality of decompressed images from these methods depends highly on the information stored in their templates. Further research is required on decompressing these templates.

Acknowledgements

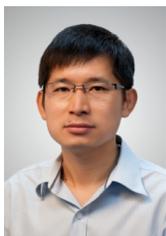
We would like to thank the University of West Virginia and the University of Beira Interior for sharing their databases. We also thank Dr. Naif Alajlan for his comments. This work is partially supported by a consultant contract provided by the King Saud University, Kingdom of Saudi Arabia.

References

- [1] J.G. Daugman, "High confidence visual recognition of persons by a test of statistical independence", *TPAMI*, vol. 15, no. 11, pp. 1148-1161, 1993.
- [2] J. Daugman, "How iris recognition works", *IEEE TCSVT*, vol. 14, no. 1, pp. 21-30, 2004.
- [3] J. Daugman, "New methods in iris recognition", *IEEE TSMC Part B*, vol. 37, No. 5, pp. 1167-1175, 2007.
- [4] J. Daugman, "The importance of being random: statistical principles of iris recognition", *Pattern Recogn*, vol. 36, pp. 279-291, 2003.
- [5] S. Shad and A. Ross, "Generating synthetic irises by feature agglomeration", *ICIP*, pp. 317-320, 2006.
- [6] A. Kong, D. Zhang and M. Kamel, "Palmprint identification using feature-level fusion", *Pattern Recogn*, pp. 478-487, 2006.

- [7] D. Zhang, W.K. Kong, J. You and M. Wong, "On-line palmprint identification", *TPAMI*, vol. 25, no. 9, pp. 1041-1050, 2003.
- [8] A.W.K. Kong and D. Zhang, "Competitive coding scheme for palmprint verification", in *Proc. ICPR*, vol. 1, pp. 520-523, 2004.
- [9] Z. Sun, T. Tan, Y. Wang and S.Z. Li, "Ordinal palmprint representation for personal identification", in *Proceedings of IEEE CVPR*, vol. 1, pp. 279-284, 2005.
- [10] L. Ma, T. Tan, Y. Wang, D. Zhang, "Efficient iris recognition by characterizing key local variations", *IEEE TIP*, vol. 13, no. 6, pp. 739-750, 2004.
- [11] Z. Sun, T. Tan and Y. Wang, "Iris recognition based on non-local comparisons", *LNCS*, Springer, vol. 3338, pp. 491-497, 2004.
- [12] E. Krichen, M.A. Mellakh, S. Garcia-Salicetti, and B. Dorizzi, "Iris identification using wavelet packets", in *Proceedings of ICPR*, vol. 4, pp. 226-338, 2004.
- [13] S.I. Noh, K. Bae, Y. Park and J. Kim, "A novel method to extract features for iris recognition system", *LNCS*, Springer, vol. 2688, pp. 861-868, 2003.
- [14] K. Bea, S. Noh and J. Kim, "Iris feature extraction using independent component analysis", *LNCS*, Springer, vol. 2688, pp. 838-844, 2003.
- [15] P.F. Zhang, D.S. Li and Q. Wang, "A novel iris recognition method based on feature fusion", in *Proc. of the 3rd International Conference on Machine Learning and Cybernetics*, pp. 26-29, 2004.
- [16] T. Ea, A. Valentian, F. Rossant, F. Amiel and A. Amara, "Algorithm implementation for iris identification", in *Proc. of 48th Midwest Symposium on Circuits and Systems*, pp. 1207-1210, 2005.
- [17] C.H. Park, J.J. Lee, S.K. Oh, Y.C. Song, D.H. Choi and K.H. Park, "Iris feature extraction and matching based on multiscale and directional image representation", *LNCS, Springer*, vol. 2695, pp. 576-583, 2004.
- [18] E. Rydgren, T.E.A.F. Amiel, F. Rossant and A. Amara, "Iris features extraction using wavelet packets", in *Proc. of ICIP*, vol. 2, pp. 861-864, 2004.
- [19] L. Masek, Recognition of Human Iris Patterns for Biometric Identification, *Bachelor thesis*, The University of Western Australia.
- [20] Z. Sun and T. Tan, "Ordinal measures for iris recognition", *TPAMI*, vol. 31, no. 12, pp. 2211-2226, 2009.
- [21] A.T.B. Jin, D.N.C. Ling and A. Goh, "Biohashing: two factor authentication featuring fingerprint data and tokenized random number", *Pattern Recogn*, vol. 37, pp. 2245-2255, 2004.
- [22] A. Kong, K.H. Cheung, D. Zhang, M. Kamel and J. You, "An analysis of Biohashing and its variants", *Pattern Recogn*, vol. 39, no. 7, pp. 1359-1368, 2006.
- [23] P. Yao, J. Li, X. Ye, Z. Zhuang and B. Li, "An analysis and improvement of an iris identification algorithm", in *Proceeding of the 18th ICPR*, vol. 4, pp. 362-365, 2006.
- [24] A.W.K Kong, D. Zhang and M. Kamel, "An analysis of IrisCode", *TIP*, vol. 19. No. 2, pp. 522-532, 2010
- [25] A. Kong, "An analysis of Gabor detection", in *Proc. of ICIAR*, pp 64-72, 2009
- [26] J.G. Daugman, "Complete discrete 2-D Gabor transforms by neural networks for image analysis and compression", *IEEE TASSP*, vol. 36, no. 7, pp. 1169-1179, 1988
- [27] T.S. Lee, "Image representation using 2D Gabor wavelets", *TPAMI*, vol. 18, no. 10, pp. 959-971, 1996
- [28] J. Behar, M. Porat and Y.Y. Zeevi, "Image reconstruction from localized phase", *IEEE TSP*, vol. 40, no. 4, pp. 736-743, 1992.
- [29] M. Turk and A. Pentland, "Eigenfaces for recognition", *J Cognitive Neurosci*.vol. 3, no. 1, pp. 71-86, 1991
- [30] H. Proenca and L.A. Alexandre, "UBIRIS: a noisy iris image database" in *Pro. of the 13th ICIAP*, vol. 1, pp. 790-977, 2005.
- [31] A. Ross and S. Shah, "Segmenting non-ideal irises using geodesic active contours", in *Proc. of Biometrics Symposium*, pp. 1-6, 2006

- [32] H. R. Sheikh and A. C. Bovik, “Image information and visual quality”, *TIP*, vol. 15, no. 2, pp. 430-444, 2006.
- [33] H. R. Sheikh, A. C. Bovik and G. de Veciana, “An information fidelity criterion for image quality assessment using natural scene statistics”, *IEEE TIP*, vol. 14, no. 12, pp. 2117-2128, 2005.
- [34] D.M. Monro, S. Rakshit and D. Zhang, “DCT-based iris recognition”, *TPAMI*, vol. 29, no. 4, pp. 586-595, 2007.
- [35] C. Sanchez-Avila and R. Sanchez-Reillo, “Two different approaches for iris recognition using Gabor filters and multiscale zero-crossing representation”, *Pattern Recogn*, vol. 38, pp. 231-240, 2005.
- [36] H.A Park and K.R. Park, “Iris recognition based on score level fusion by using SVM”, *Pattern Recogn Lett*, vol. 28, pp. 2019-2028, 2007.
- [37] H. Proença and L.A. Alexandre, “Toward noncooperative iris recognition: a classification approach using multiple signatures”, *TPAMI*, vol. 29, no. 4, pp. 607- 612, 2007
- [38] N.K. Ratha, J.H. Connell and R.M. Bolle, “Biometrics break-ins and band-aids”, *Pattern Recogn Lett*, vol. 24, no. 13, pp. 2105-2113, 2003.
- [39] M. Braithwaite, U.C. von Seelen, J. Cambier, J. Daugman, R. Class, R. Moore and I. Scott, “Applications-specific biometric template”, in *Proceeding of IEEE Workshop on Automatic Identification Advanced Technologies*, pp. 167-171, 2002.
- [40] S. Makthall and A. Ross, “Synthesis of iris images using Markov random fields”, in *Proceeding of 13th European Signal Processing Conference*, 2005
- [41] J. Cui, Y. Wang, J. Huang, T. Tan, and Z. Sun, “An iris image synthesis method based on PCA and super-resolution”, in *Proc. 17th ICPR*, vol. 4, pp. 471-474, 2004.
- [42] J. Zuo, N.A. Schmid and Z. Chen, “On generation and analysis of synthetic iris images”, *IEEE TIFS*, vol. 2, no. 1, pp. 77- 90, 2007.
- [43] K.P. Hollingsworth, K.W. Bowyer and P.J. Flynn, “The best bits in an iris code”, *TPAMI*, vol. 31, no. 6, pp. 964-973, 2009
- [44] L. Liang, C. Liu, Y.Q. Xu, B. Guo and H.Y. Shum. “Real-time texture synthesis by patch-based sampling”, *ACM T Graphics*, vol. 20, no. 3, pp. 127–150, 2001.
- [45] S. Rakshit and D. M. Monro, “An evaluation of image sampling and compression for human iris recognition”, *IEEE TIFS*, vol. 2, no. 3, pp. 605–612, Sep. 2007.
- [46] J. Daugman and C. Downing, “Effect of severe image compression on iris recognition performance”, *IEEE TIFS*, vol. 3, no 1, pp. 52-61, 2008
- [47] X. Wang and X. Tang, “Face photo-sketch synthesis and recognition”, *TPAMI*, vol. 31, no. 11, pp. 1955-1967, 2009.
- [48] C. Liu, H.Y. Shum and W.T. Freeman, “Face hallucination: theory and practice”, *IJCV*, vol. 75, no. 1, pp. 115-134, 2007.
- [49] J.D. Woodward, “Biometrics: privacy’s foe or privacy’s friend?”, *Proc. of IEEE*, vol. 85, no 9, 1997.
- [50] B. Schneier, “Security and function creep”, *IEEE Security & Privacy*, vol. 8, no .1, pp. 88, 2010
- [51] W. Cheney and D. Kincaid, *Linear algebra theory and applications*, Jones and Bartlett Publishers, Sudbury, Massachusetts, 2009.



Adams Wai-Kin Kong received his PhD from the University of Waterloo, Canada. Currently, he is an assistant professor at the Nanyang Technological University, Singapore. His research interests include biometrics, forensics, image processing, and pattern recognition.

Appendix

A.

This appendix shows that Eq. 5 is always true. Let $\wp = [g_{r1} \cdots g_{rn} \ g_{i1} \cdots g_{in} \ \Phi]$, $A_d = [a_{r1} \cdots a_{rn} \ a_{i1} \cdots a_{in} \ d]^T$, $\mathfrak{I}_m = [\gamma_1 \cdots \gamma_m]$ and $C_m = [c_1 \ c_2 \ \cdots \ c_m]^T$. Using a matrix representation, Eq. 5 can be rewritten as $I = \wp A_d + \mathfrak{I}_m C_m$ and using the least square method, i.e., $\min_{A_d} \|I - \wp A_d\|$ to compute A_d , we have $A_d = (\wp^T \wp)^{-1} \wp^T I$ if $(\wp^T \wp)^{-1}$ exists. Let the residual of I be $\zeta = I - \wp A_d = I - \wp (\wp^T \wp)^{-1} \wp^T I$. Clearly, $\wp^T \zeta = 0$ implying that ζ is orthogonal to Φ and all g_{rk} and g_{ik} . If $\wp^T \wp$ is not invertible, $\wp^T \zeta = 0$ is still true because the solutions of $\min_{A_d} \|I - \wp A_d\|$ must satisfy the equation $(\wp^T \wp) A_d = \wp^T I$. Note that the 0 represents a zero vector.

It is worth to mention that if \mathbf{G} defined in Eq. 8 is invertible, $\wp^T \wp$ is also invertible. Note that $\wp = [\wp_g \ \Phi]$, where $\wp_g = [g_{r1} \cdots g_{rn} \ g_{i1} \cdots g_{in}]$. Using a block matrix representation, $(\wp^T \wp)^{-1} = \begin{bmatrix} \wp_g^T \wp_g & \wp_g^T \Phi \\ \Phi^T \wp_g & \Phi^T \Phi \end{bmatrix}^{-1}$. This inverse exists if $\wp_g^T \wp_g$ and $\Phi^T \Phi - \Phi^T \wp_g (\wp_g^T \wp_g)^{-1} \wp_g^T \Phi$ are invertible [51]. $\Phi^T \Phi - \Phi^T \wp_g (\wp_g^T \wp_g)^{-1} \wp_g^T \Phi$ is a positive number and equal to $\Phi^T \Phi$ because g_{rk} and g_{ik} are zero DC filters, i.e., $\wp_g^T \Phi = 0$. Consequently, the inverse of $\Phi^T \Phi - \Phi^T \wp_g (\wp_g^T \wp_g)^{-1} \wp_g^T \Phi$ always exists. We should remember that $\wp_g^T \wp_g = \mathbf{G}$. Our experimental results show that the \mathbf{G} formed by the Gabor filters in IrisCode is invertible, implying that $\wp^T \wp$ is also invertible.

Now, we consider a set of images, $\{I_1, \cdots, I_K\}$ and define their residuals as $\zeta_k = I_k - \wp A_{kd}$. Using the result $\wp^T \zeta_k = 0$, we can easily prove that $\wp^T (\frac{1}{K} \sum_{k=1}^K \zeta_k) = 0$. In other words, the mean of ζ_k is orthogonal to Φ and all g_{rk} and g_{ik} . The sample covariance matrix of ζ_k can be computed through

$\frac{1}{K} \left(\sum_{k=1}^K \zeta_k \zeta_k^T - \mu_\zeta \mu_\zeta^T \right)$, where $\mu_\zeta = \frac{1}{K} \sum_{k=1}^K \zeta_k$. Its principal components γ_j have to satisfy

$\frac{1}{K} \left(\sum_{k=1}^K \zeta_k \zeta_k^T - \mu_\zeta \mu_\zeta^T \right) \gamma_j = \lambda_j \gamma_j$, where λ_j is the corresponding non-zero eigenvalue. Using

$\frac{1}{K} \left(\sum_{k=1}^K \zeta_k \zeta_k^T - \mu_\zeta \mu_\zeta^T \right) \gamma_j = \lambda_j \gamma_j$, $\phi^T \zeta_k = 0$ and $\phi^T \mu_\zeta = 0$, we have

$$\phi^T \gamma_j = \frac{1}{\lambda_j K} \phi^T \left(\sum_{k=1}^K \zeta_k \zeta_k^T - \mu_\zeta \mu_\zeta^T \right) \gamma_j = \frac{1}{\lambda_j K} \left(\sum_{k=1}^K \phi^T \zeta_k \zeta_k^T - \phi^T \mu_\zeta \mu_\zeta^T \right) \gamma_j = 0. \quad (29)$$

Thus, I can be decomposed as $I = \sum_{j=1}^n a_{rj} g_{rj} + \sum_{j=1}^n a_{ij} g_{ij} + \sum_{j=1}^m c_j \gamma_j + d\Phi$, where $\gamma_m = \mu_\zeta$ and γ_j is orthogonal

to Φ and all g_{rk} and g_{ik} , i.e., $\langle g_{r(i)k}, \gamma_j \rangle = 0$ and $\langle \Phi, \gamma_j \rangle = 0$.

B.

This appendix shows that the coding methods generalized from IrisCode such as Competitive Code and precise phase representation [8, 24] are also compression algorithms. Their feature extractors can be defined as $\tau_j = \arg \max_{0 \leq k < s} (\langle h_{j,k}, I \rangle)$, where τ_j is a feature value to be encoded, $h_{j,k}$ is a filter, and s is the total number of the filters. The relationship between $\tau_j = \arg \max_{0 \leq k < s} (\langle h_{j,k}, I \rangle)$ and Eqs. 1-4 can be found in [24]. Each feature value is computed from a group of filters, $h_{j,0}, \dots, h_{j,s-1}$. For example, Competitive Code uses six filters. Each template from these methods is composed of J feature values, which can be represented in a binary format for high speed matching [24]. The inequalities, $\langle h_{j,k}, I \rangle \leq \langle h_{j,\tau_j}, I \rangle$, where $\tau_j \neq k$, can be derived from the feature extractor. For the sake of convenience, let $\tau_j = 0$. Rewriting these inequalities, we obtain $0 \leq \langle \hat{h}_{j,k}, I \rangle$, where $\hat{h}_{j,k} = h_{j,0} - h_{j,k}$.

Let $I = \sum_{j=1}^J \sum_{k=1}^{s-1} a_{j,k} \hat{h}_{j,k} + \sum_j c_j \eta_j$, where $a_{j,k}, c_j \in \mathfrak{R}$ and η_j is orthogonal to all $\hat{h}_{e,k}$. As in the proof in

Section 2, we can obtain

$$I = \sum_{j=1}^J \sum_{k=1}^{s-1} \langle \hat{h}_{j,k}, I \rangle v_{j,k} + \sum_{j=1}^J c_j \eta_j, \quad (30)$$

where $[v_{1,1} \cdots v_{J,s-1}] = [\hat{h}_{1,1} \cdots \hat{h}_{J,s-1}] \begin{bmatrix} \langle \hat{h}_{1,1}, \hat{h}_{1,1} \rangle & \cdots & \langle \hat{h}_{1,1}, \hat{h}_{J,s-1} \rangle \\ \vdots & \ddots & \vdots \\ \langle \hat{h}_{J,s-1}, \hat{h}_{1,1} \rangle & \cdots & \langle \hat{h}_{J,s-1}, \hat{h}_{J,s-1} \rangle \end{bmatrix}^{-1}$. Here, we assume that the

inverse exists. Eq. 30 indicates that these coding methods generalized from IrisCode are also compression algorithms. They use one bit to store each coefficient of $v_{j,k}$, which is $\langle \hat{h}_{j,k}, I \rangle$, but erase all other coefficients. Note that different templates, even from the same methods, have different $\{v_{1,1} \cdots v_{J,s-1}\}$ because they have different feature values.