Modeling IrisCode and its Variants as Convex Polyhedral Cones and its Security Implications

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, developed by Daugman in 1993, is the most influential iris recognition algorithm. A thorough understanding of IrisCode is essential, because over 100 million persons have been enrolled by this algorithm and many biometric personal identification and template protection methods have been developed based on IrisCode. This paper indicates that a template produced by IrisCode or its variants is a convex polyhedral cone in a hyperspace. Its central ray, being a rough representation of the original biometric signal, can be computed by a simple algorithm, which can often be implemented in one MATLAB command line. The central ray is an expected ray and also an optimal ray of an objective function on a group of distributions. This algorithm is derived from geometric properties of a convex polyhedral cone but does not rely on any prior knowledge (e.g., iris images). The experimental results show that biometric templates, including iris and palmprint templates, produced by different recognition methods can be matched through the central rays in their convex polyhedral cones and that templates protected by a method extended from IrisCode can be broken into. These experimental results indicate that, without a thorough security analysis, convex polyhedral cone templates cannot be assumed secure. Additionally, the simplicity of the algorithm implies that even junior hackers without knowledge of advanced image processing and biometric databases can still break into protected templates and reveal relationships among templates produced by different recognition methods.

Keywords: Biometrics, iris recognition, template protection, palmprint recognition.

1. Introduction

IrisCode[1] has over 100 million users from approximately 170 countries [1]. As of 4 January 2012, the total number of people just in India who have had their iris patterns enrolled by IrisCode is 103 million. The Unique Identification Authority of India is enrolling about one million persons per day, at 40,000 stations, and they plan to have the entire population of 1.2 billion people enrolled within 3 years [33]. The extraordinary market success of IrisCode relies heavily on its computational advantages, including extremely high matching speed for large-scale identification and automatic threshold adjustment based on image quality (e.g., number of effective bits) and database size [1-3]. In the last two decades, this algorithm influenced many researchers [4-15]. Many methods modified from IrisCode were proposed for iris, palmprint, and finger-knuckle recognition. This paper refers to these methods as generalized IrisCodes (GIrisCode) [4]. The simplest modification replaced the Gabor filters in IrisCode with other linear filters or transforms. A more complex modification used a clustering scheme to perform feature extraction and a special coding table to perform feature encoding [4-5]. With these modifications, feature value precision could be increased, and many IrisCode computational advantages could be retained. Another modification replaced the Gabor filters in IrisCode with random vectors to construct cancelable biometrics for template protection [16-17]. A complete understanding of IrisCode is thus necessary. Though many research papers regarding iris recognition have been published, our understanding of this important algorithm remains incomplete. Daugman indicated that the imposter distribution of IrisCode follows a binomial distribution, and the bits "0" and "1" in IrisCode are equally probable [1]. Hollingsworth et al.[2] analyzed bit stability in their iris codes and detected the best bits for enhancing recognition performance [18]. Kong and his coworkers theoretically derived the following points: IrisCode is a clustering algorithm with four prototypes and a compression algorithm; the Gabor function can be regarded as a phase-steerable filter; the locus of a Gabor function is a two-dimensional ellipse with respect to a phase parameter and can often be approximated by a circle; the bitwise hamming distance can be considered a bitwise phase distance; and Gabor filters can be utilized as a Gabor atom detector, and the magnitude and phase of a target Gabor atom can be approximated by the magnitude and phase of the corresponding Gabor

[1] In this paper, IrisCode is used interchangeably to refer to both the method and features of iris recognition developed by Daugman.
[2] Hollingsworth et al. used 1D log-Gabor wavelets instead of 2D Gabor filters in their study [18].

response [4, 19-20]. Using these theoretical results and information from iris image databases, Kong designed an algorithm to reconstruct iris images from IrisCodes [20]. Nevertheless, the geometric structure of IrisCodes has never been studied. This paper primarily aims to provide a deeper understanding of the geometric structures of IrisCode and its variants and secondarily seeks to analyze the potential security and privacy risks from this geometric information.

Though the IrisCode computational procedures are well-known, we present a brief computational summary for notation consistency. Two-dimensional Gabor filters with zero DC are used to extract phase information from an iris image in a dimensionless polar coordinate system, $\boldsymbol{I}_0(\rho,\phi)$. The complex Gabor response is quantized into two bits by the following inequalities:

$$b_{jr}=1 \quad if \quad \mathrm{Re}\left(\int_{\rho}\int_{\varphi}\boldsymbol{I}_0(\rho,\varphi)e^{-(r_{j0}-\rho)^2/\alpha_j^2}e^{-(\theta_{j0}-\varphi)^2/\beta_j^2}e^{-i\omega_j(\theta_{j0}-\varphi)}\rho d\rho d\varphi\right)\geq 0\,, \tag{1}$$

$$b_{jr}=0 \quad if \quad \mathrm{Re}\left(\int_{\rho}\int_{\varphi}\boldsymbol{I}_0(\rho,\varphi)e^{-(r_{j0}-\rho)^2/\alpha_j^2}e^{-(\theta_{j0}-\varphi)^2/\beta_j^2}e^{-i\omega_j(\theta_{j0}-\varphi)}\rho d\rho d\varphi\right)<0\,, \tag{2}$$

$$b_{ji}=1 \quad if \quad \mathrm{Im}\left(\int_{\rho}\int_{\varphi}\boldsymbol{I}_0(\rho,\varphi)e^{-(r_{j0}-\rho)^2/\alpha_j^2}e^{-(\theta_{j0}-\varphi)^2/\beta_j^2}e^{-i\omega_j(\theta_{j0}-\varphi)}\rho d\rho d\varphi\right)\geq 0\,, \tag{3}$$

$$b_{ji}=0 \quad if \quad \mathrm{Im}\left(\int_{\rho}\int_{\varphi}\boldsymbol{I}_0(\rho,\varphi)e^{-(r_{j0}-\rho)^2/\alpha_j^2}e^{-(\theta_{j0}-\varphi)^2/\beta_j^2}e^{-i\omega_j(\theta_{j0}-\varphi)}\rho d\rho d\varphi\right)<0\,, \tag{4}$$

where ω_j is the spatial frequency, α_j and β_j control the shape of the Gaussian function and (r_{j0},θ_{j0}) is the center/location of the filter in the spatial domain [2]. One thousand and twenty-four Gabor filters with different parameters $(r_{j0},\theta_{j0},\omega_j,\alpha_j,\beta_j)$ generate 1024 bit pairs (b_{jr},b_{ji}) in an IrisCode, and a mask excludes the corrupted bits from the eyelashes, eyelids, reflection, and a low signal-to-noise ratio [2]. Bitwise hamming distance is employed for high speed matching. Although Eqs. 1-4 are given as integrations over a filter kernel, in the rest of this paper, two dimensional functions are expressed in discrete form as matrices and that these matrices are expressed as vectors after lexicographic ordering. Therefore, all filtering operations will be expressed as inner products.

The rest of this paper is organized as follows: Section 2 gives an introduction to convex polyhedral cone, presents an algorithm to estimate its central ray and reveals the relationships among the central ray, the expected ray and the optimal ray of an objective function on a group of distributions. Section 3 shows that a template produced by IrisCode or GIrisCode is a convex polyhedral cone in a hyperspace. Section 4 reports the experimental results obtained from iris recognition methods, palmprint recognition methods, and a security analysis of a template protection method. Section 5 discusses the impact of our theoretical and experimental findings.

2. Estimating the Central Ray of a Convex Polyhedral Cone

For clear presentation, a set of notations is necessary. Given a zero DC Gabor filter $\boldsymbol{g}_j$ with 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, $\boldsymbol{g}_{jr}$ and $\boldsymbol{g}_{ji}$ represent its real and imaginary parts. $\boldsymbol{I}$ is used to denote $\boldsymbol{I}_0(\rho,\phi)\rho$. $\boldsymbol{I}_0$ can be computed from $\boldsymbol{I}$, because ρ can never be zero. For other biometric methods, $\boldsymbol{I}$ represents an original image and biometric signal. Bold font is used to denote matrices and both two-dimensional filters and images. For example, $\boldsymbol{g}_{ji}$ is an imaginary part of a two-dimensional Gabor filter, while g_{ji} is a column vector form of $\boldsymbol{g}_{ji}$, and $\boldsymbol{I}$ is a two-dimensional image, while I is a column vector form of the image. T represents the transpose of a matrix or vector. Therefore, the inner product of two real valued vectors (e.g., g_{jr} and g_{kr}) can be defined as $< g_{jr}, g_{kr} >= g_{kr}{}^T g_{jr}$.

2.1. Fundamental Knowledge of a Convex Polyhedral Cone

Convex polytope, a branch of mathematics, has strong connections with constraint optimization (e.g., linear programming) [21-22]. A convex polyhedral cone is a special type of convex polytope. In this subsection, we introduce the convex polyhedral cone for readers without background knowledge.

A convex polytope can be defined in several different ways. This paper uses the half-space representation, which defines a convex polytope through the intersection of closed half-spaces. Let $C \subseteq \Re^d$ be a convex polytope. Using the half-space representation, $C = \{x \in \Re^d : \boldsymbol{Q}x \le z\}$, where $\boldsymbol{Q} \in \Re^{m \times d}$ and $z \in \Re^m$. Each row in the inequality system describes a half-space, i.e., $q_k x \le z_k$ [21], where q_k is the k^{th} row in $\boldsymbol{Q}$, and z_k is the k^{th} element in z. Note that $q_k x = z_k$ is a hyperplane. When z is a zero vector, C becomes a convex polyhedral cone with apex zero [22]. This paper considers only convex polyhedral cones with apex zero; $C = \{x \in \Re^d : \boldsymbol{Q}x \le 0\}$ is thus used as a definition of the convex polyhedral cone. Three properties can be derived: (1) C is a convex set; (2) the intersection of convex polyhedral cones is also a convex polyhedral cone; and (3) if $x \in C$, then $\lambda x \in C$, $\forall \lambda \ge 0$, where λx is called a ray. The following discussion uses these three properties. Fig. 1 illustrates convex polyhedral cones and other cones. Fig. 2 illustrates that the intersection of convex polyhedral cones is also a convex polyhedral cone.

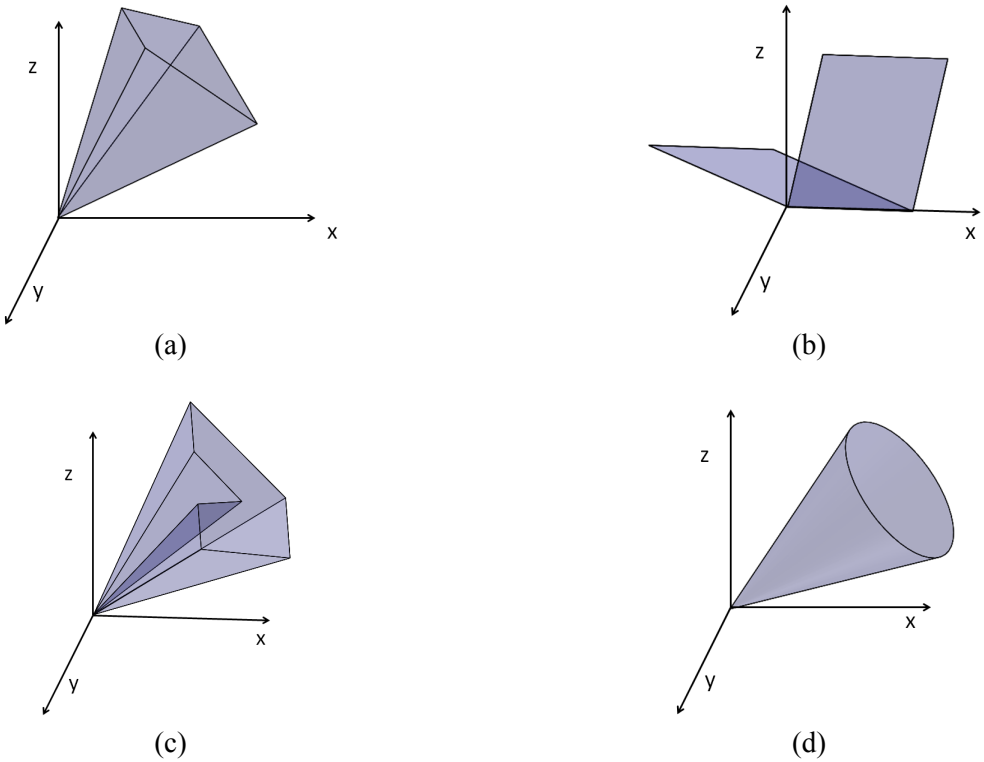


Fig. 1 Illustration of convex polyhedral cones and other cones. (a) and (b) convex polyhedral cones, (c) a non-convex polyhedral cone and (d) a circular cone.

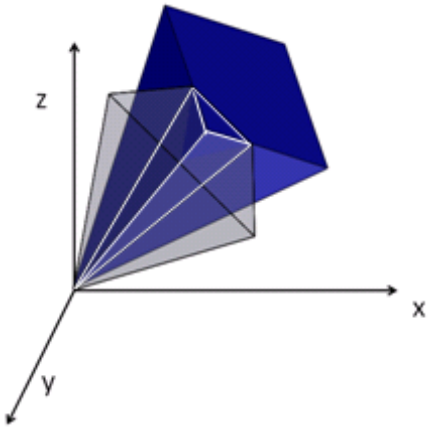


Fig. 2 The intersection of convex polyhedral cones is also a convex polyhedral cone.

The half-space representation is not unique. In other words, there exists more than one inequality system representing the same convex polyhedral cone. For example,

$$C_1 = \left\{ \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \in \Re^2 : \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \le 0 \right\}, \tag{5}$$

and

$$C_2 = \left\{ \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \in \Re^2 : \begin{bmatrix} -1 & 0 \\ 0 & -1 \\ -1 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \le 0 \right\}, \tag{6}$$

represent the same convex polyhedral cone (Fig. 3), but their inequality systems differ. The inequality $-x_1 - x_2 \le 0$ in Eq. 6 is a redundant constraint that can be removed. Note that the number of inequalities in the minimal representation is unique [24]. Several ways exist to remove redundant constraints in a linear inequality system. A simple one is based on linear programming. Let us consider an inequality system $\begin{bmatrix} \boldsymbol{Q} \\ q \end{bmatrix} x \le \begin{bmatrix} z \\ z_q \end{bmatrix}$, where $\boldsymbol{Q} \in \Re^{m \times d}$, $x \in \Re^d$, $z \in \Re^m$, $q \in \Re^{1 \times d}$, and $z_q \in \Re$. If $qx \le z_q$ is a redundant constraint in the linear inequality system, then $\tau \le z_q$, where τ is defined as

$$\tau = \max qx \quad \text{subject to} \quad \begin{bmatrix} \boldsymbol{Q} \\ q \end{bmatrix} x \le \begin{bmatrix} z \\ z_q + 1 \end{bmatrix}. \tag{7}$$

Eq. 7 can be solved by linear programming. Convex hull algorithms can also remove redundant constraints [23].

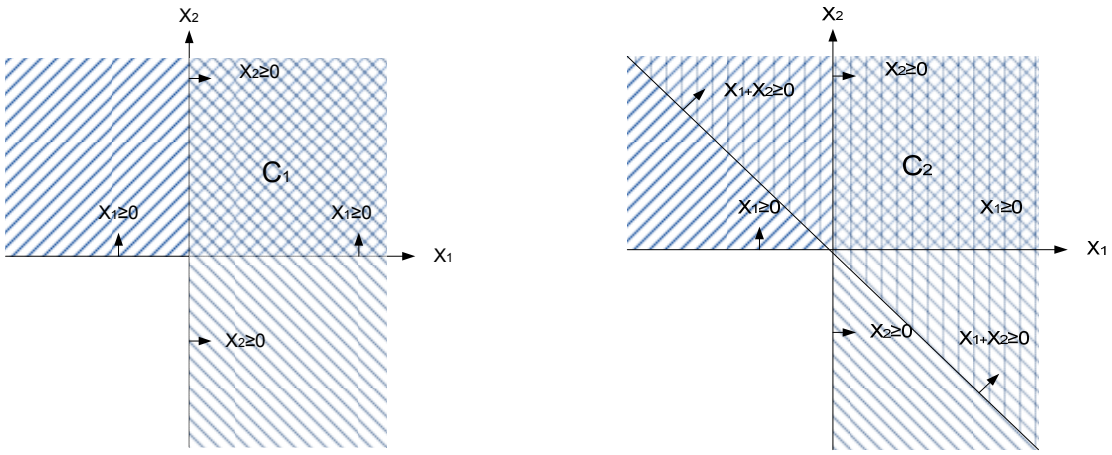


Fig. 3 Illustration of a non-unique representation of a convex polyhedral cone and a redundant constraint. (a) the convex polyhedral cone C_1 defined by Eq. 5 and (b) the convex polyhedral cone C_2 defined by Eq. 6.

2.2. A Simple Geometric Algorithm

In this subsection, a simple geometric algorithm[3] is derived to estimate the central ray of a convex polyhedral cone. This algorithm is the realization of the idea that a ball is put into a convex polyhedral cone. If the ball touches all planes of the cone, it cannot be moved closer to the apex (Fig. 4). By locating the centers of balls with different radii, the central ray of the cone can be constructed. Let us mathematically formulate this idea. Given a convex polyhedral cone $C = \{x \in \Re^d : \boldsymbol{Q}x \le 0\}$, a set of hyperplanes $q_1 x = 0, \cdots$ and $q_m x = 0$, where q_k is the k^{th} row in $\boldsymbol{Q}$, can be obtained. Here, we assume that $\boldsymbol{Q}x \le 0$ is a minimum representation [24]. Given a point $y \in \Re^d$, the displacement between y and a hyperplane $q_k x = 0$ can be calculated by

$$\varepsilon_k = q_k y / \|q_k\| . \tag{8}$$

[3] Although the geometric algorithm was completely designed by the author without help from other materials, he cannot guarantee that there is no other published algorithm for computing the central ray of a convex polyhedral cone because geometry is a large field with many years of history and many other fields use it.

Note that ε_k is displacement, not distance, and $\varepsilon_k \le 0$ when $y \in C$. Rewriting Eq. 8 gives $q_k y = \varepsilon_k \|q_k\|$. Let us seek a point with the same displacement to all hyperplanes, i.e., $\varepsilon = \varepsilon_1 = \cdots = \varepsilon_m$. This point is the center of a hypersphere with radius $|\varepsilon|$, which touches all hyperplanes. Using the matrix representation, we obtain

$$\boldsymbol{Q}y = \varepsilon \begin{bmatrix} \|q_1\| \\ \vdots \\ \|q_m\| \end{bmatrix}, \tag{9}$$

where $\varepsilon < 0$. As with other linear systems, Eq. 9 can have unique, multiple, or no solutions. Eq. 9 has a unique solution, meaning that there exists only one position in the cone such that the hypersphere can touch all hyperplanes. Eq. 9 has multiple solutions, meaning that there exists more than one position in the cone such that the hypersphere can touch all hyperplanes. Eq. 9 has no solution, meaning that no such position exists in the cone. Fig. 5 illustrates these three cases.

We use the center of the hypersphere to determine the central ray of the convex polyhedral cone. Let *y* be a solution of Eq. 9 for a hypersphere with radius $|\varepsilon|$. It is easily proven that λy, where $\lambda > 0$, is also a solution for a hypersphere with radius $\lambda|\varepsilon|$. Each solution of Eq. 9 produces one central ray, i.e., λy. Any point on the ray is the center of a hypersphere, which touches all hyperplanes. If Eq. 9 has no solution, the least square method can compute *y*.

One may suggest generalizing skeletonization algorithms, which were designed for digital images, from two-dimensional to *d*-dimensional domains to find the central ray. It is an ineffective approach because these algorithms have to discretize the *d*-dimensional solution space and label all the *d*-dimensional voxels in it. Note that the solution space is unbounded, which makes it impossible to label all the voxels in the solution space. In other words, only a subset of voxels can be labeled. If they are not selected appropriately, the central ray cannot be obtained, even for over-constrained systems. For under-constrained systems, this problem is more obvious.

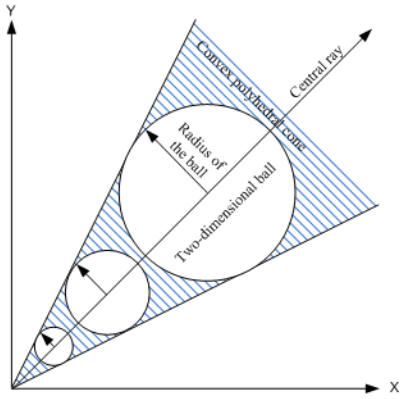


Fig. 4 Illustration of the geometric algorithm.

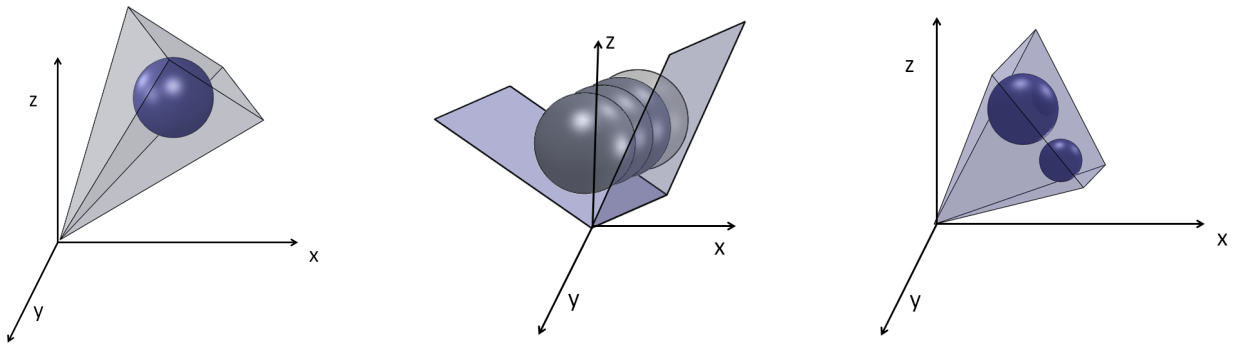


Fig. 5 Geometric explanation of Eq. 9. (a) Unique solution, (b) multiple solutions, and (c) no solution.

2.3 The Relationships among the Central Ray, Expected Ray, and Optimal Ray

The geometric algorithm to compute central rays has been presented. From the algorithmic point of view, the relationship between the central ray and the expected ray[4] of a convex polyhedral cone should be studied because they are anticipated to be the same ray. From the application point of view, the relationship between the central ray and the corresponding optimal ray should also be investigated because, if the central ray is far from the optimal ray, it implies that there is some space for improving the algorithm. This subsection reveals the relationships among the central ray, the expected ray and the optimal ray.

[4] The term expected ray refers to the ray generated from the mathematical expectation.

To study their relationships, a distribution of rays has to be defined. Let C be a convex polyhedral cone, $C = \{x \in \Re^d : \boldsymbol{Q}x \le 0\}$, where each row of $\boldsymbol{Q}$ is normalized, i.e., $\|q_1\| = \cdots \|q_d\| = 1$ and $\boldsymbol{Q}$ is an invertible d by d matrix. Thus, its central ray[5] R_c can be calculated by $R_c = -\boldsymbol{Q}^{-1}\Phi$, where $\Phi = [1\ 1 \cdots 1]^T$. Let R_r be a random ray generated by

$$R_r = -\boldsymbol{Q}^{-1}\Gamma, \tag{10}$$

where $\Gamma^T = [\varphi_1 \cdots \varphi_d]$ and $\varphi_1, \cdots, \varphi_d$ are independent and identically distributed random variables following a distribution with a probability density function f_φ whose mean is μ_φ and support is $[a\ b]$. Thus, the probability density of Γ is $\prod_{i=1}^{d} f_\varphi(\varphi_i)$. Note that $a \ge 0$. All random rays generated by Eq. 10 are in the convex polyhedral cone $\boldsymbol{C}$. φ_i represents the distance between the point that is used to construct a random ray and the hyperplane $q_i x = 0$.

Let R_t be a target ray for $\boldsymbol{C}$. In the following sections, R_t represents the original features and images. Note that R_t is an unknown. Without other prior knowledge, each ray in $\boldsymbol{C}$ can be the target ray. Assuming that R_t follows the distribution generated by Eq. 10, an optimal ray R_o is defined

$$R_o = \arg\max_{R_\varphi} \int {R_\varphi}^T R_t p(\Gamma) d\Gamma, \text{ subject to } \|R_\varphi\| = 1, \tag{11}$$

where $p(\Gamma)$ represents the probability density of Γ. The expected inner product is used as an objective function because many biometric methods, including the variants of IrisCode, use the inner product to measure the similarity between an input biometric signal (e.g., iris image) and a filter (e.g., Gabor filter) as a feature extractor. Thus, Eq. 11 uses the inner product to measure the similarity between R_φ and R_t. If the norm of R_φ is not constrained, $\max \int {R_\varphi}^T R_t p(\Gamma) d\Gamma$ approaches infinity. Rewriting Eq. 11,

$$R_o = \arg\max_{R_\varphi} {R_\varphi}^T \int R_t p(\Gamma) d\Gamma, \text{ subject to } \|R_\varphi\| = 1, \tag{12}$$

[5] In the subsection, the term ray also refers to the point that generates the ray.

can be achieved. Clearly, $\int R_t p(\Gamma) d\Gamma$ is the expected ray, and the optimal ray of this objective function is

$$R_o = \frac{\int R_t p(\Gamma) d\Gamma}{\left\| \int R_t p(\Gamma) d\Gamma \right\|}. \tag{13}$$

Thus, the optimal ray and the expected ray are the same ray.

Now, we show that the central ray is the expected ray as well as the optimal ray. Substituting Eq. 10 into $\int R_t p(\Gamma) d\Gamma$, we can obtain

$$\int -\boldsymbol{Q}^{-1} \Gamma p(\Gamma) d\Gamma . \tag{14}$$

Simplifying Eq. 14, $\int -\boldsymbol{Q}^{-1} \Gamma p(\Gamma) d\Gamma = -\mu_\varphi \boldsymbol{Q}^{-1} \Phi$, which is the central ray, is derived. Thus, the central ray, the expected ray, and the optimal ray of the objective function on the distribution are the same ray. In this derivation, we do not specify the probability density function f_φ, so a group of probability density functions, including a uniform distribution can be employed as f_φ.

For over-constrained systems, the previous random model is not well defined because it is not necessary to have an R_t such that $\tilde{\boldsymbol{Q}} R_t = -\Gamma$, where $\tilde{\boldsymbol{Q}}$ is an over-constrained matrix. Thus, the random model has to be revised. Using the least square method, $\tilde{R}_t = -\left(\tilde{\boldsymbol{Q}}^T \tilde{\boldsymbol{Q}} \right)^{-1} \tilde{\boldsymbol{Q}}^T \Gamma$ can be achieved. For the sake of convenience, let $\boldsymbol{Q}_{L2} = \left(\tilde{\boldsymbol{Q}}^T \tilde{\boldsymbol{Q}} \right)^{-1} \tilde{\boldsymbol{Q}}^T$. Replacing R_t in Eq. 11 with $\widetilde{R}_t$, we can obtain an optimal ray $-\mu_\varphi \boldsymbol{Q}_{L2} \Phi / \left\| \mu_\varphi \boldsymbol{Q}_{L2} \Phi \right\|$, which is also the expected ray, for the objective function,

$$R_o = \arg\max_{R_\varphi} \int {R_\varphi}^T \tilde{R}_t p(\Gamma) d\Gamma , \text{ subject to } \left\| R_\varphi \right\| = 1 . \tag{15}$$

For under-constrained systems, there is more than one R_t satisfying $\hat{\boldsymbol{Q}} R_t = -\Gamma$, where $\hat{\boldsymbol{Q}}$ is an under-constrained matrix. Let $\boldsymbol{Q}^\dagger$ be the Moore-Penrose pseudoinverse of $\hat{\boldsymbol{Q}}$ and $\hat{R}_t = -\boldsymbol{Q}^\dagger \Gamma$. In the following experiments, the Moore-Penrose pseudoinverse was also used for under-constrained systems.

Replacing R_t in Eq. 11 with $\hat{R}_t$, we can derive that the optimal ray $-\mu_\varphi \boldsymbol{Q}^\dagger \Phi / \left\| \mu_\varphi \boldsymbol{Q}^\dagger \Phi \right\|$, which is also the expected ray, for the objective function,

$$R_o = \arg\max_{R_\varphi} \int {R_\varphi}^T \hat{R}_t \, p(\Gamma) d\Gamma \,, \text{ subject to } \left\| R_\varphi \right\| = 1 \,. \tag{16}$$

Thus, for all the cases, the central ray is the expected ray as well as the optimal ray.

3. Convex Polyhedral Cones in Different Biometrics Templates

This section shows that templates produced by IrisCode and GIrisCodes are convex polyhedral cones in different hyperspaces and provides an algorithm to project these cones into a lower-dimensional space to find a unique solution for Eq. 9 for iris and palmprint templates. Additionally, an algorithm based on the intersection of convex polyhedral cones is also offered to demonstrate that protected templates are vulnerable.

3.1. IrisCode and its Low-Order Generalization

Let us consider IrisCode first. Using the notations provided in Section 2, we can derive the inequalities $-\hat{b}_{jr} g_{jr}^T I \le 0$ and $-\hat{b}_{ji} g_{ji}^T I \le 0$, where $\hat{b}_{jr} = 2(b_{jr} - 0.5)$ and $\hat{b}_{ji} = 2(b_{ji} - 0.5)$ from Eqs. 1-4. $\hat{b}_{jr(i)} = 1$ when $b_{jr(i)} = 1$ and $\hat{b}_{jr(i)} = -1$ when $b_{jr(i)} = 0$. The notation $r(i)$ signifies either real or imaginary part from the Gabor filter. Each bit in an IrisCode generates one inequality, which associates with a hyperplane, i.e., $-\hat{b}_{jr(i)} g_{jr(i)}^T x = 0$. Putting these inequalities into a matrix form, we obtain $-{\boldsymbol{\Psi}_{\hat{b}}}^T I \le 0$, where $\boldsymbol{\Psi}_{\hat{b}} = [\hat{b}_{1r} g_{1r}, \cdots, \hat{b}_{nr} g_{nr}, \hat{b}_{1i} \, g_{1i}, \cdots, \hat{b}_{ni} g_{ni}]$ and n=1024. An IrisCode is clearly a convex polyhedral cone. $\boldsymbol{\Psi}_{\hat{b}}$ is an N by 2048 matrix, where N equals the total number of pixels in a normalized iris image. The central ray of this convex polyhedral cone can be obtained by solving the linear system ${-\boldsymbol{\Psi}_{\hat{b}}}^T I = \varepsilon \Phi$, where $\Phi = [1 \, 1 \cdots 1]^T$ and $\varepsilon < 0$, because all $g_{jr(i)}$s have been normalized, i.e., $\left\| g_{jr(i)} \right\| = 1$. It is clearly an

under-constrained system, because $N \gg 2048$. In our experiments, N is 32,768. The result in Section 2 pinpoints that if y is a solution of the linear system for a hypersphere with radius $|\varepsilon|$, λy, where $\lambda > 0$, is also a solution of the linear system for the hypersphere with radius $\lambda|\varepsilon|$. We can thus set $\varepsilon = -1$. Rewriting the linear system, we obtain

$$\boldsymbol{\Psi}^T I = \hat{B}, \tag{17}$$

where $\hat{B} = [\hat{b}_{1r}, \cdots, \hat{b}_{nr}, \hat{b}_{1i}, \cdots, \hat{b}_{ni}]^T$ and $\boldsymbol{\Psi} = [g_{1r}, \cdots, g_{nr}, g_{1i}, \cdots, g_{ni}]$ [6]. Eq. 17 has multiple solutions, because it is still an under-constrained linear system.

We now seek a particular solution for Eq. 17. Using the results in [20], I can be decomposed as

$$I = \sum_{j=1}^{n} a_{jr} g_{jr} + \sum_{j=1}^{n} a_{ji} g_{ji} + \sum_{j=1}^{N-2n-1} c_j \gamma_j + t\Phi_N, \tag{18}$$

where a_{jr}, a_{ji}, c_j and $t \in \Re$, $\Phi_N = [1\ 1 \cdots 1]^T$, γ_j is a unit vector orthogonal to Φ_N and all g_{kr} and g_{ki}, i.e., ${\gamma_j}^T g_{kr(i)} = 0$ and ${\gamma_j}^T \Phi_N = 0$, where $1 \le k \le n$. Note that Φ_N and Φ are different in size. Eq. 18 can be simplified as

$$I = \boldsymbol{\Psi} A + \sum_{j=1}^{N-2n-1} c_j \gamma_j + t\Phi_N, \tag{19}$$

where $A = [a_{1r} \cdots a_{nr}\, a_{1i} \cdots a_{ni}]^T$. Note that $\boldsymbol{\Psi}^T \gamma_j = 0$ and $\boldsymbol{\Psi}^T \Phi_N = 0$, where 0s are zero vectors. Substituting Eq. 19 and $\hat{B} = 2(B - 0.5\Phi)$, where $B = [b_{1r}, \cdots b_{nr}, b_{1i}, \cdots, b_{ni}]^T$, into Eq. 17, we obtain

$$\boldsymbol{\Psi}^T \boldsymbol{\Psi} A = 2(B - 0.5\Phi), \tag{20}$$

where

[6] The matrix $\boldsymbol{\Psi}$ can be downloaded from http://goo.gl/mv2EF.

$$\boldsymbol{\Psi}^T\boldsymbol{\Psi}=\begin{bmatrix} g_{1r}^T g_{1r} & \cdots & g_{1r}^T g_{nr} & g_{1r}^T g_{1i} & \cdots & g_{1r}^T g_{ni} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ g_{nr}^T g_{1r} & \cdots & g_{nr}^T g_{nr} & g_{nr}^T g_{1i} & \cdots & g_{nr}^T g_{ni} \\ g_{1i}^T g_{1r} & \cdots & g_{1i}^T g_{nr} & g_{1i}^T g_{1i} & \cdots & g_{1i}^T g_{ni} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ g_{ni}^T g_{1r} & \cdots & g_{ni}^T g_{nr} & g_{ni}^T g_{1i} & \cdots & g_{ni}^T g_{ni} \end{bmatrix}. \tag{21}$$

Let $\hat{I}=\boldsymbol{\Psi}A$. If $\boldsymbol{\Psi}^T\boldsymbol{\Psi}$ is invertible, we obtain

$$\hat{I}=\boldsymbol{\Psi}\left(\boldsymbol{\Psi}^T\boldsymbol{\Psi}\right)^{-1}2(B-0.5\Phi). \tag{22}$$

Clearly, $\hat{I}$ in Eq. 22 is a solution of Eq. 17. Substituting Eq. 19 into the original convex polyhedral cone $-\boldsymbol{\Psi}_{\hat{b}}{}^T I\le 0$, we obtain another convex polyhedral cone $-\boldsymbol{\Psi}_{\hat{b}}{}^T\boldsymbol{\Psi}A\le 0$ with a lower dimension. The original cone is projected into the lower-dimensional space because our previous results indicate that the space spanned by the Gabor filters $\langle g_{1r},\cdots,g_{nr},g_{1i},\cdots,g_{ni}\rangle$ stores rich iris information [20]. Another two points should be mentioned. This derivation uses neither prior knowledge from iris images nor any special IrisCode structure. Templates produced by other biometric methods that use

$$\tilde{b}_j=1 \quad if \quad \int_{\rho}\int_{\varphi} f_j I d\rho d\varphi\ge 0, \tag{23}$$

$$\tilde{b}_j=0 \quad if \quad \int_{\rho}\int_{\varphi} f_j I d\rho d\varphi< 0, \tag{24}$$

where $\tilde{b}_j$ is a bit in their feature codes and f_j represents their filter, as a feature extractor are also convex polyhedral cones, and Eq. 22 can compute their central rays. Note that I in Eqs. 23-24 is a two dimensional image. Column vectors in the corresponding $\boldsymbol{\Psi}$ should be normalized. Eq. 22 can be implemented in one MATLAB command line, if memory is enough. If other methods can recognize $\hat{I}$, the simplicity of Eq. 22 significantly deepens our concern for privacy, because it implies that junior hackers with neither advanced image processing knowledge nor biometric databases can still reveal relationships among templates produced by different recognition methods. $\boldsymbol{\Psi}$ is not a secret for many methods, because researchers disclosed their $\boldsymbol{\Psi}$ s in their papers.

3.2. The High-Order Generalization of IrisCode

The previous subsection has pinpointed that templates produced by IrisCode and other biometric methods that use Eqs. 23 and 24 as feature extractors are convex polyhedral cones in different hyperspaces. This subsection shows that templates produced by methods that are high-order IrisCode generalizations are also convex polyhedral cones, including Competitive Code and its variants for palmprint, vein and finger-knuckle recognition and precise phase representation [4-5, 10].

The feature extractor of these algorithms can be summarized as

$$\tau_j = \arg\max_{0 \le k < S} \left(h_{j,k}^T I \right), \tag{25}$$

where τ_j is a feature value to be encoded, $h_{j,k}$ is a filter, and S is the total number of filters used to compute one feature value. τ_j can be encoded by a specially designed code table such that it can be represented in a binary format and hamming distance can be used for matching [4]. Each feature value is computed from a group of filters, i.e., $h_{j,0}, \cdots, h_{j,S-1}$. Competitive Code uses six real parts of Gabor filters with different orientations [5]. In other words, for Competitive Code, S is equal to 6 and $h_{j,i}$ is a real part of a Gabor filter. S-1 inequalities, $h_{j,k}^T I \le h_{j,\tau_j}^T I$, where $\tau_j \ne k$, can be derived from Eq. 25 for each feature value. Rewriting the inequalities, we obtain $(h_{j,k} - h_{j,\tau_j})^T I \le 0$. For convenience, let $\tau_j = 0$ and $(h_{j,k} - h_{j,\tau_j}) = \tilde{h}_{j,k}$. Using a matrix presentation, a linear inequality system $\mathbf{H}^T I \le 0$, where $\mathbf{H} = [\tilde{h}_{1,1} \cdots \tilde{h}_{J,S-1}]$, can be obtained. Templates produced by Eq. 25 are clearly also convex polyhedral cones. If $\tilde{h}_{j,k}^T I \le 0$, then $(\tilde{h}_{k,j}^T I) / \left\| \tilde{h}_{j,k} \right\| \le 0$. Using this fact, another inequality system $\hat{\mathbf{H}}^T I \le 0$, where $\hat{\mathbf{H}} = \left[\tilde{h}_{1,1} / \left\| \tilde{h}_{1,1} \right\| \cdots \tilde{h}_{J,S-1} / \left\| \tilde{h}_{J,S-1} \right\| \right]$, can be derived.

The central ray of this convex polyhedral cone can be determined by solving the matrix equation $\hat{\mathbf{H}}^T I = -\Phi$. As in the previous subsection, let $I = \sum_{j=1}^{J} \sum_{s=1}^{S-1} u_{j,s} \tilde{h}_{j,s} / \left\| \tilde{h}_{j,s} \right\| + \sum_j w_j \eta_j$, where $u_{j,s} \in \Re$ and

$w_j \in \Re$ and $\tilde{h}_{j,s}^T \eta_k = 0,\ \forall j,\ s$ and k. Substituting $I = \sum_{j=1}^{J}\sum_{s=1}^{S-1} u_{j,s}\tilde{h}_{j,s} / \left\|\tilde{h}_{j,s}\right\| + \sum_j w_j \eta_j$ into $\hat{\mathbf{H}}^T I = -\Phi$, we obtain $\hat{\mathbf{H}}^T \hat{\mathbf{H}} U = -\Phi$, where $U = [u_{1,1} \cdots u_{J,S-1}]^T$. If $\left(\hat{\mathbf{H}}^T \hat{\mathbf{H}}\right)^{-1}$ exists, $U = -(\hat{\mathbf{H}}^T \mathbf{H})^{-1}\Phi$. Let $\tilde{I} = \hat{\mathbf{H}} U$, which is an approximation of *I*. Substituting $U = -(\hat{\mathbf{H}}^T \hat{\mathbf{H}})^{-1}\Phi$ into $\tilde{I} = \hat{\mathbf{H}} U$, we obtain

$$\tilde{I} = -\hat{\mathbf{H}}(\hat{\mathbf{H}}^T \hat{\mathbf{H}})^{-1}\Phi . \tag{26}$$

After constructing $\hat{\mathbf{H}}$, Eq. 26 can also be implemented in one MATLAB command line.

3.3. Using the Intersection of Convex Polyhedral Cones to Break into Protected Templates

Biohashing, a cancelable biometric method modified from IrisCode, was designed for template protection [16-17]. Once a template is compromised, Biohashing attempts to reissue a new template. It is expected that Biohashing can issue new templates without limit, compromised templates cannot be matched with newly issued templates, and the original biometric signal cannot be revealed from compromised templates. In this subsection, we first show that templates produced by Biohashing are also convex polyhedral cones. We then provide an algorithm estimating the central ray in the intersection of convex polyhedral cones from compromised templates to reveal the protected biometric signal.

Biohashing uses the same computational step as IrisCode to produce protected feature vectors. Given an original feature vector *v*, Biohashing uses a random matrix, $\mathbf{\Omega} = [\zeta_1, \cdots, \zeta_{m_\Omega}]$ with size n_Ω by m_Ω and $m_\Omega < n_\Omega$, whose $\zeta_k^T \zeta_j = 0$, $\forall k \neq j$, $\|\zeta_k\| = 1$ and $\zeta_k^T \zeta_k = 1 \ \ \forall k$, to project *v* into a lower-dimensional random subspace, i.e., $W_c = \mathbf{\Omega}^T v$ and further binarizes W_c as with Eqs. 1-4. In summary, each bit in a protected feature vector can be computed through

$$w_j = \begin{cases} 0 & if \quad v^T \zeta_j \le \varphi \\ 1 & if \quad v^T \zeta_j > \varphi \end{cases}, \tag{27}$$

where φ was set to zero in the experiments [16]. If one template is compromised, Biohashing replaces the random matrix $\mathbf{\Omega}$ with another random matrix. It is expected that the original feature vector in a

protected template cannot be revealed, even if attackers obtain $\mathbf{\Omega}$. Some researchers attempted to break into the protected templates [25-26]. Our experimental results show that if the system threshold is high enough, Biohashing can defend their attacks. Using the result in Section 3.1, we can easily derive the convex polyhedral cone of Biohashing, i.e., $V_\Omega = \{x \in \Re^{m_\Omega} \mid \hat{\Omega}^T x \le 0\}$, where $\hat{\mathbf{\Omega}} = [-\hat{w}_1\zeta_1, \cdots, -\hat{w}_{m_\Omega}\zeta_{m_\Omega}]$ and $\hat{w}_j = 2(w_j - 0.5)$. As ζ_k has been normalized, i.e., $\|\zeta_k\| = 1$, the central ray of this convex polyhedral cone can be determined by solving the linear system

$$\mathbf{\Omega}^T v = 2(W - 0.5\Phi), \tag{28}$$

where $W = [w_1, \cdots, w_{m_\Omega}]^T$. The derivation of Eq. 28 is same as Eq. 17. Clearly, it is still an under-constrained linear system. However, if an attacker obtains *e* compromised templates generated from the same biometric signal and corresponding random matrixes $\mathbf{\Omega}_1, \cdots \mathbf{\Omega}_{e-1}$ and $\mathbf{\Omega}_e$, they can simultaneously reveal the original biometric signal and finally break into the system. Let the convex polyhedral cone associated with $\mathbf{\Omega}_k$ be $V_{\Omega_k} = \left\{x \in \Re^{m_\Omega} : \hat{\mathbf{\Omega}}_k^T x \le 0\right\}$. All convex polyhedral cones must contain the original biometric signal. In other words, the original biometric signal must be in the intersection of all convex polyhedral cones. The intersection is also a convex polyhedral cone defined as

$$V_{\Omega_1} \cap \cdots \cap V_{\Omega_e} = \left\{ x \in \Re^{m_\Omega} : \begin{bmatrix} \hat{\mathbf{\Omega}}_1^T \\ \vdots \\ \hat{\mathbf{\Omega}}_e^T \end{bmatrix} x \le 0 \right\}. \tag{29}$$

Clearly,

$$v \in (V_{\Omega_1} \cap \cdots \cap V_{\Omega_e}) \subseteq (V_{\Omega_1} \cap \cdots \cap V_{\Omega_{e-1}}) \subseteq \cdots \subseteq (V_{\Omega_1} \cap V_{\Omega_2}) \subseteq V_{\Omega_1}. \tag{30}$$

Eq. 30 implies that when *e* increases, the size of the convex polyhedral cone $V_{\Omega_1} \cap \cdots \cap V_{\Omega_e}$ decreases. If the convex polyhedral cone is small enough, we can use its central ray to approximate the original

biometric signal. If $\begin{bmatrix} \hat{\Omega}_1^T \\ \vdots \\ \hat{\Omega}_e^T \end{bmatrix} x \le 0$ is a minimum representation, the central ray of the intersection can be obtained by solving the linear system

$$\begin{bmatrix} \Omega_1^T \\ \vdots \\ \Omega_e^T \end{bmatrix} v = 2\left(\begin{bmatrix} W_{\Omega_1} \\ \vdots \\ W_{\Omega_e} \end{bmatrix} - 0.5 \begin{bmatrix} \Phi \\ \vdots \\ \Phi \end{bmatrix} \right). \tag{31}$$

In the experiments, we used the Moore-Penrose pseudoinverse to solve Eq. 31. To keep the simplicity of the algorithm and simulate junior hackers breaking into Biohashing, we did not use any methods to remove redundant constraints.

4. Experimental Results

Two iris databases, the UBIRIS.v1 and West Virginia University (WVU) iris databases [27-28], and one palmprint database were used to test the proposed algorithms. The UBIRIS.v1 database contains 1,877 images from 241 irises, and the WVU[7] iris database contains 3,099 iris images from 472 irises. All images in the WVU iris database were tested in the experiments. However, 48 images from the UBIRIS.v1 database were automatically removed, because of their poor quality (Fig. 6). Though some extremely low-quality images were discarded, many challenging iris images were retained for evaluation. Fig. 7 gives some examples of the challenging images. The UBIRIS.v1 iris images were taken under a visible lighting environment, while the WVU iris images were taken under an infrared lighting environment. The original images in the UBIRIS.v1 database are color images. We only used their red component for evaluation, because the iris texture in this component is the clearest (Fig. 8). The palmprint database contains 7,500 images from the right and left palms of 250 persons. Each palm has 15 images in the database. Ten images were collected in the first session, and the others were collected in the second session. The left palm images were flipped, so all images in this database could be regarded as right

[7] Some mislabeled images were corrected.

palms. The central parts of these palmprints were extracted [29]. Fig. 9 gives some examples of the preprocessed palmprints.

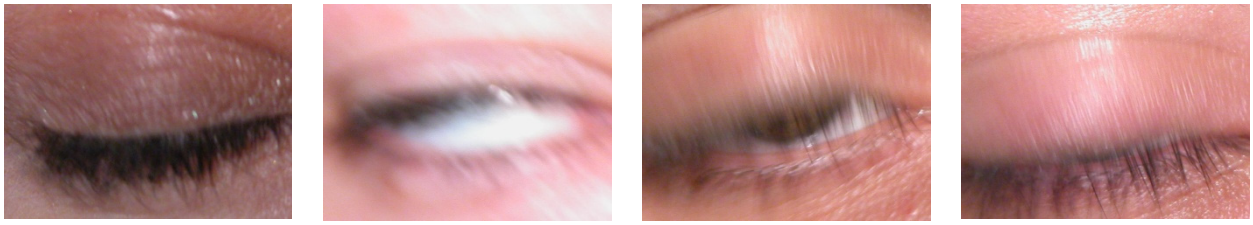
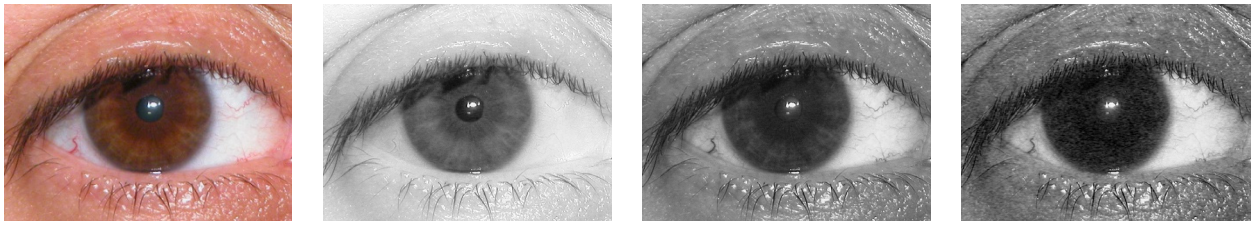


Fig. 6 Examples of the discarded iris images in the UBIRIS.v1 database.

Fig. 7 Examples of low-quality iris images for evaluation.



(a) (b) (c) (d)

Fig. 8 Iris texture in different components. (a) is a color image; (b)-(d) are the R, G and B components of (a), respectively.

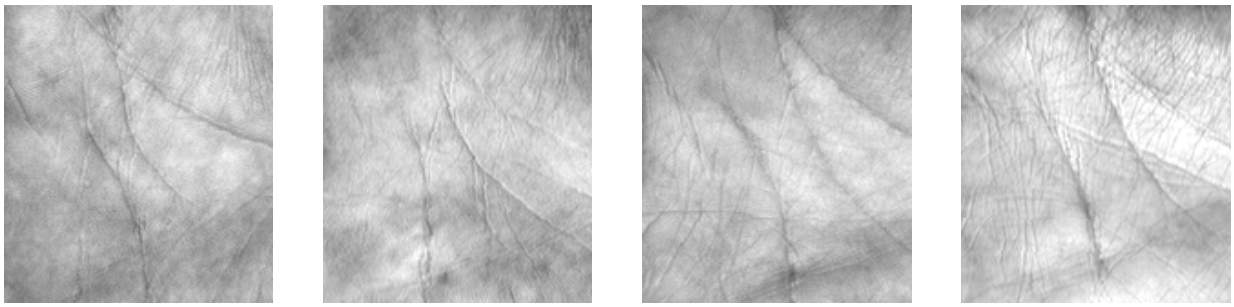


Fig. 9 Examples of preprocessed palmprints.

4.1. Evaluation of Iris Recognition Methods

In this section, six iris recognition methods, IrisCode [2], SVM [11], and Ordinal Code (Di-lobe d=5 and d=9 and Tri-lobe d=9 and d=17) [9], were re-implemented to evaluate the proposed algorithm derived in Section 3.1 in terms of Receiver Operating Characteristic (ROC) curves. Masks that denote noise pixels,

such as eyelids and eyelashes, were used to exclude corrupted bits in all templates generated by these methods. The raw hamming distances of the Ordinal Code were rescaled to those in IrisCode [4]. From each database, 90 images were used to train the SVM with a Gaussian kernel [11]. The proposed algorithm estimated the central rays of IrisCode and Ordinal Code. Ordinal Code is a low-order generalization of IrisCode. For the central rays of IrisCode, the SVM approach was used as a tester. We matched the central rays of IrisCode with the original images using the SVM approach. The central rays of Ordinal Code used IrisCode as a tester. We matched the central rays of Ordinal Code with the original images using IrisCode. In each set of experiments, cross-matching between all original images was first performed, and the corresponding genuine and imposter distributions were estimated. The central rays were then matched with their parent[8] images and all other iris images from the same eye. Two genuine distributions were obtained. The imposter distributions from the cross-matching between the original images and all genuine distributions were used to plot ROC curves. For the same database and method, all ROC curves were generated from the same imposter distribution that determines the false acceptance rate of a system when a threshold is given. This experimental setting studies the risk of using the central rays to break into the systems. For example, attackers can submit a central ray of a compromised template into the data link between a biometric sensor and feature extractor, and if they can stop the liveness detector and quality checker, they can use the central ray to perform a sensor-level break-in [32]. Note that the theory given in this paper can guarantee that an IrisCode computed from an original image and an IrisCode computed from the corresponding central ray are no different. Thus, it is impossible to distinguish them at the matcher level. For convenience, the ROC curves from the cross-matching between the original images are called original ROC curves, the ROC curves from matching the central rays with their parent images are called parent ROC curves, and the ROC curves from matching the central rays with the original images are called resultant ROC curves. Fig. 10 illustrates the matching processes. Figs. 11 and 12 show the ROC curves of the SVM approach and IrisCode, respectively. All parent ROC curves are above the corresponding original ROC curves, indicating that the errors introduced by the central rays

[8] A parent image of a central ray is an image producing a biometric template that is used to compute the central ray.

are smaller than those in the original images. If a compromised template and template in a database are generated from the same iris image, the central ray of the compromised template performs even better than other iris images from the same eye. If the central ray and template are not generated from the same image, there is a 70% to 85% chance to break into the systems when the thresholds are set at false acceptance rate of 0.001. Using Eq. 22, it can be proven easily that if a central ray is obtained from an IrisCode (Ordinal Code) and IrisCode (Ordinal Code) is used as a matcher, the central ray does not introduce any additional error bits. The central ray can break into an iris recognition system running IrisCode (Ordinal Code) if attackers can stop its liveness detector and quality checker or submit the central ray into the data link between its sensor and feature extractor.

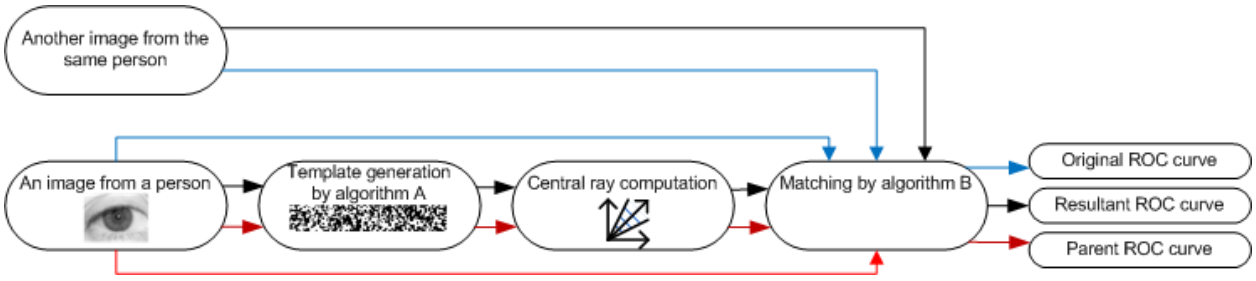


Fig. 10 Illustration of three different matching processes. The blue, black and red arrows indicate respectively the processes of computing genuine matching scores for original, resultant and parent ROC curves.

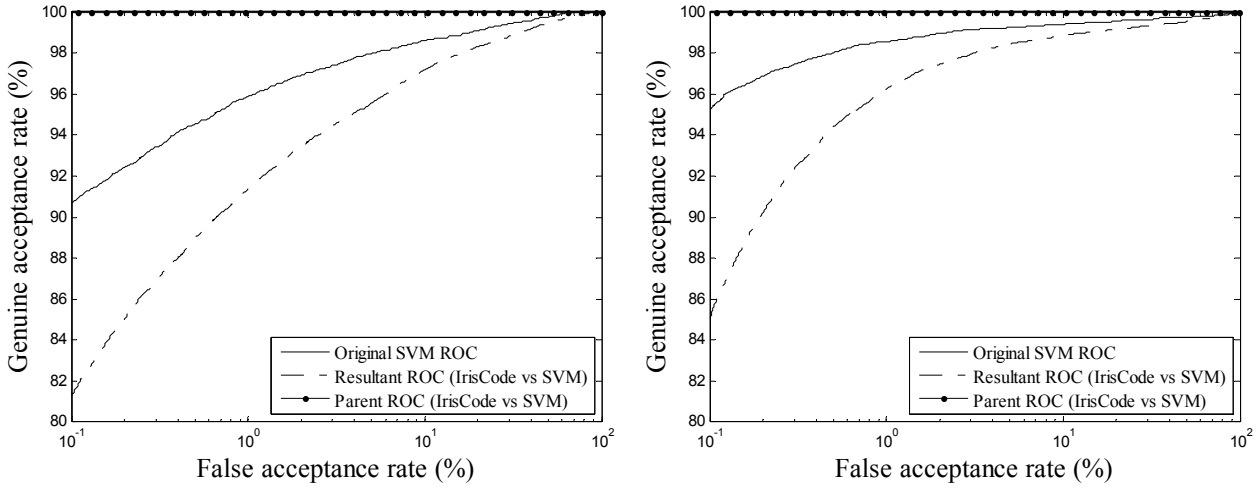
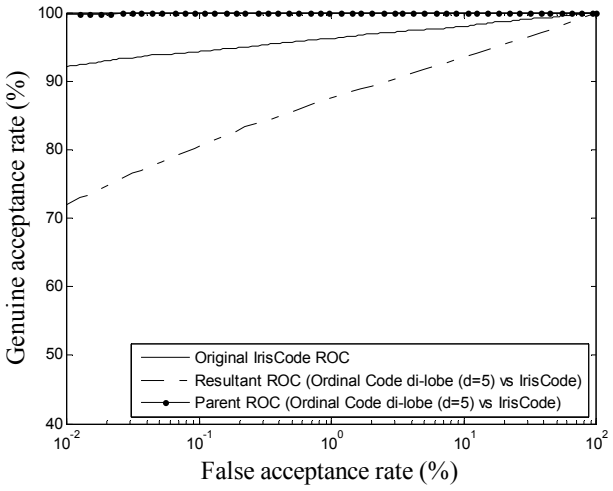
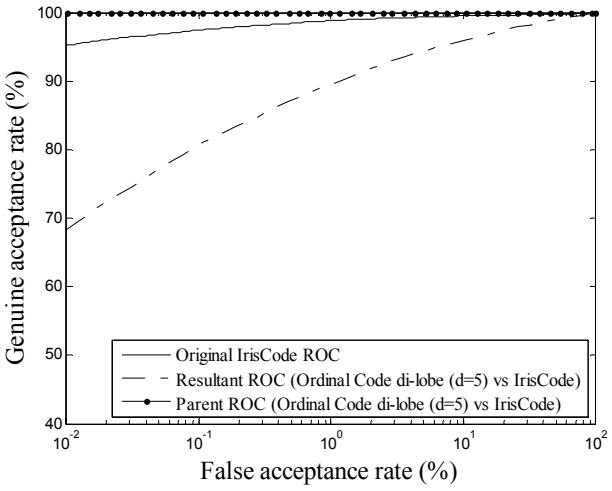


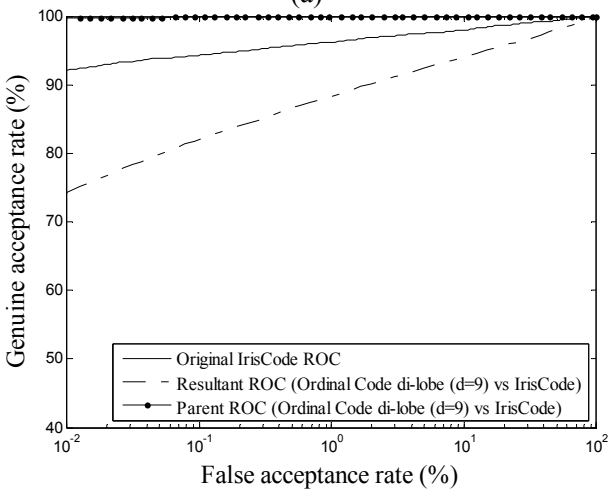
Fig. 11 ROC curves of the SVM method. The iris images were estimated from the convex polyhedral cones of IrisCode. (a) The results from the UBIRIS.v1 database and (b) the results from the WVU iris database.


Genuine acceptance rate (%)
False acceptance rate (%)
Original IrisCode ROC
Resultant ROC (Ordinal Code di-lobe (d=5) vs IrisCode)
Parent ROC (Ordinal Code di-lobe (d=5) vs IrisCode)

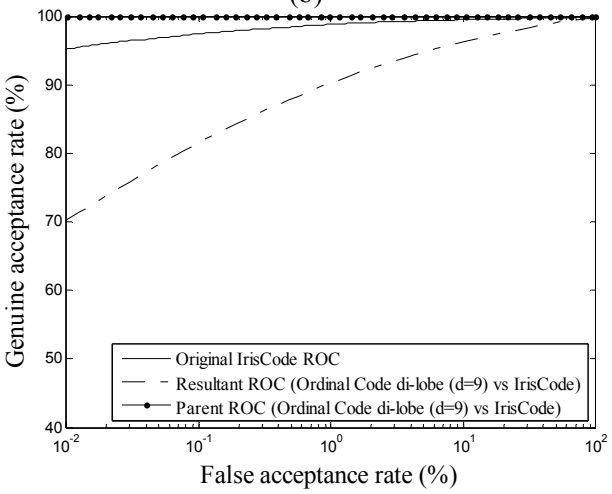
(a)


Genuine acceptance rate (%)
False acceptance rate (%)
Original IrisCode ROC
Resultant ROC (Ordinal Code di-lobe (d=5) vs IrisCode)
Parent ROC (Ordinal Code di-lobe (d=5) vs IrisCode)

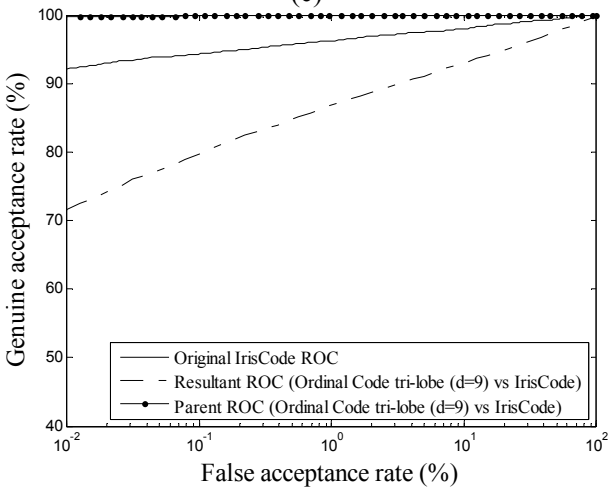
(b)


Genuine acceptance rate (%)
False acceptance rate (%)
Original IrisCode ROC
Resultant ROC (Ordinal Code di-lobe (d=9) vs IrisCode)
Parent ROC (Ordinal Code di-lobe (d=9) vs IrisCode)

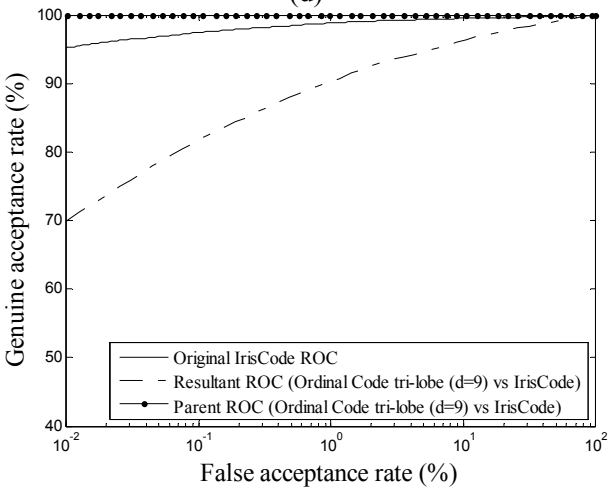
(c)


Genuine acceptance rate (%)
False acceptance rate (%)
Original IrisCode ROC
Resultant ROC (Ordinal Code di-lobe (d=9) vs IrisCode)
Parent ROC (Ordinal Code di-lobe (d=9) vs IrisCode)

(d)


Genuine acceptance rate (%)
False acceptance rate (%)
Original IrisCode ROC
Resultant ROC (Ordinal Code tri-lobe (d=9) vs IrisCode)
Parent ROC (Ordinal Code tri-lobe (d=9) vs IrisCode)

(e)


Genuine acceptance rate (%)
False acceptance rate (%)
Original IrisCode ROC
Resultant ROC (Ordinal Code tri-lobe (d=9) vs IrisCode)
Parent ROC (Ordinal Code tri-lobe (d=9) vs IrisCode)

(f)

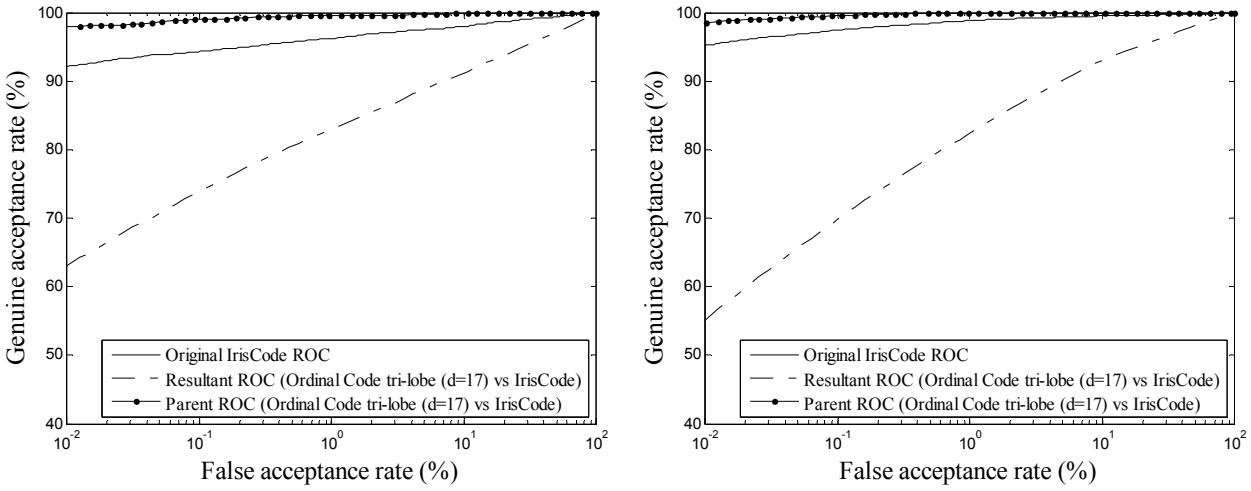
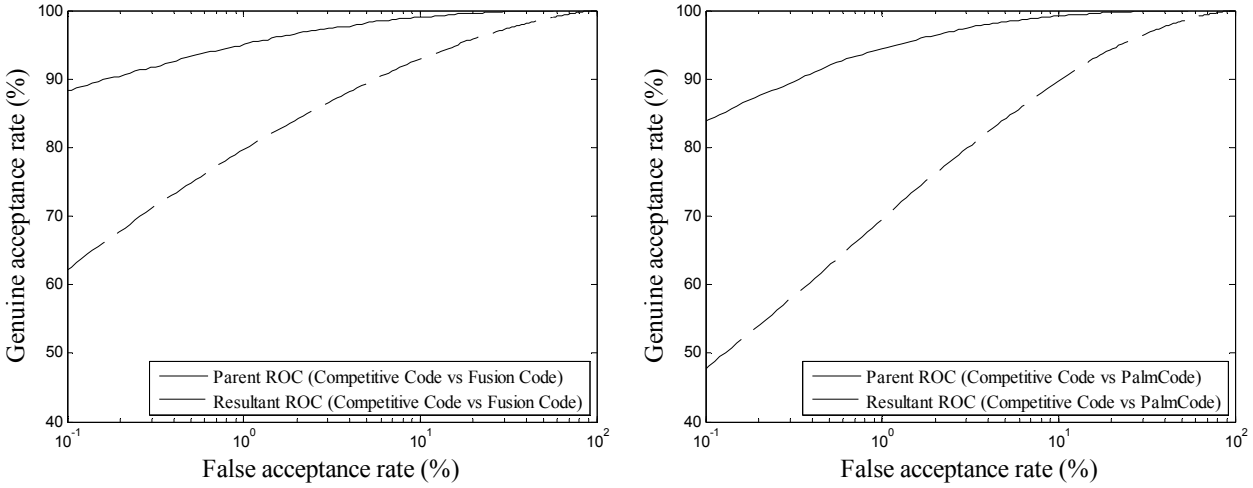


Fig. 12 ROC curves of the IrisCode. The first column presents the results from the UBIRIS.v1 database, and the second column presents the results from the WVU iris database. Rows 1-4 are results from iris images estimated from the convex polyhedral cones of di-lobe (d=5), di-lobe (d=9), tri-lobe (d=9) and tri-lobe (d=17), respectively.

4.2. Evaluation of Palmprint Recognition Methods

In this section, three palmprint recognition methods, Competitive Code [5], Fusion Code [31], and PalmCode [29], were re-implemented to evaluate the algorithm presented in Section 3.2. Competitive Code is a high-order generalization of IrisCode. In this experiment, the central rays of Competitive Code were matched with the original images using Fusion Code and PalmCode. Five thousand palmprint images in the database were tested. Half of the images were collected in the first session, and the rest were collected in the second session. This experiment studied the risk of using the central rays to reveal relationships among templates produced by different palmprint identification methods. If templates produced by different methods can be matched, user privacy can be invaded, because the same person registered in different systems can be known. If a hamming distance of Fusion Code or PalmCode, obtained by matching the central ray of a Competitive Code with an original image, is significantly shorter than other hamming distances from matching the central ray with other images, the Competitive Code and original image can be regarded as from the same palm. The imposter distributions from the cross-matching between the central rays of Competitive Code, and original images, and corresponding genuine distributions were used to plot ROC curves. Fig. 13 shows the Fusion Code and PalmCode ROC

curves and provides their parent ROC curves. These results indicate that if two palmprint templates produced by two algorithms (e.g., Competitive Code and Fusion Code) are from the same image, the probability of detecting that they are generated from the same palm is over 0.83, when the probability of falsely seeing that two independent templates are from the same palm is 0.001. If two templates are generated from different images of the same palm, this probability drops to 0.62 for Fusion Code and 0.48 for PalmCode. If attackers set the threshold at false acceptance rate of 0.01, these two probabilities increase, respectively to 0.79 and 0.7.



(a) (b)

Fig. 13 ROC curves of using convex polyhedral cones to reveal relationships among templates from palmprint identification algorithms. All convex polyhedral cones were generated by Competitive Code. (a) Fusion Code ROC curves and (b) PalmCode ROC curves.

4.3. Evaluation of Biohashing

In this experiment, 2,500 palmprint images were used to train a feature extractor, and another 2,500 images were used to test the algorithm derived in Section 3.3; 500 different palms were tested. Each palm had 5 images in the training set and 5 in the testing set. We only used images collected in the first session in this experiment, because we wished to simulate a system with extremely high accuracy. Linear discriminant analysis was used as a feature extractor [30]. The dimension of the original feature vectors was 99 [30]. We used cosine distance to match the original feature vectors and generated the corresponding ROC curve for comparison. This ROC curve is labeled as LDA+Cosine. The Biohashing

template sizes were 60 and 90 bits. Each testing image was matched with 5 training images from the same palm to produce five hamming distances, and the smallest distance was considered a genuine hamming distance. Similarly, each testing image was matched with all other images. The minimum hamming distance from the same palm in the database was considered an imposter hamming distance. The corresponding ROC curves were labeled as Biohashing (K bits), where K is the number of bits in the templates. We also estimated the central rays of the intersection of the convex polyhedral cones formed by compromised Biohashing templates to match newly reissued templates. More clearly, we matched the $W = [w_1, \cdots, w_{m_{\Omega_{q+1}}}]^T$ generated from an original image with the $\hat{W} = [\hat{w}_1, \cdots, \hat{w}_{m_{\Omega_{q+1}}}]^T$ generated from an estimated central ray, where $\boldsymbol{\Omega}_{q+1}$ represents a newly re-issued random matrix. We only matched W with $\hat{W}$ from the same palm. Only genuine distributions were produced from these matches. Using these genuine distributions with the original Biohashing impostor distributions, the probability of a break-in based on the central rays of the intersections of the convex polyhedral cones formed by compromised templates can be estimated. The ROC curves generated from the original impostor distributions and these genuine distributions are referred to as the ROC curves of the CPC algorithm (C*T*), where *CT* denotes the number of compromised templates used to compute the central rays. We also implemented a constrained optimization method for comparison [26]. To estimate the original biometric vectors protected by Biohashing, this method uses information from unrelated feature vectors (e.g., from different persons). In this experiment, each original vector was estimated by 100 feature vectors from 100 different palms. When the number of comprised templates is one, the CPC algorithm is same as our previous work [25]. However, we neither discussed convex polyhedral cones nor used multiple compromised templates to estimate original feature vectors in that work [25]. Fig. 14 shows the ROC curves, clearly indicating that the CPC algorithm can effectively break into Biohashing. The probability of a successful break-in is over 40% when the central rays are estimated from only 7 compromised templates with 60 bits and system threshold is set at false acceptance rate of 0.001%; the probability of a successful break-in is over 50% when the central rays are estimated from only 4 compromised templates with 90 bits and system threshold

is set at false acceptance rate of 0.001%. As the number of compromised templates increases, the probability of a successful break-in also increases. When the number of compromised templates is enough, the CPC algorithm can perform even better than the original Biohashing.

As mentioned above, Biohashing can defend attacks from the constrained optimization method if the decision threshold is high enough [26]. Readers may wonder why the constrained optimization method cannot perform better than the CPC algorithm (*CT*=1) if both use information from one compromised template, but the constrained optimization method utilizes additional images from other palms to estimate the original features. This phenomenon can be explained by the geometric structure of a convex polyhedral cone. The constrained optimization method has two steps. Given a set of unrelated feature vectors $\kappa_1 \cdots \kappa_K$, this method seeks a solution $\ddot{x}_k$ that has a minimum distance from κ_k and fulfills all constraints from a compromised template. Mathematically, $\ddot{x}_k$ is obtained from optimizing

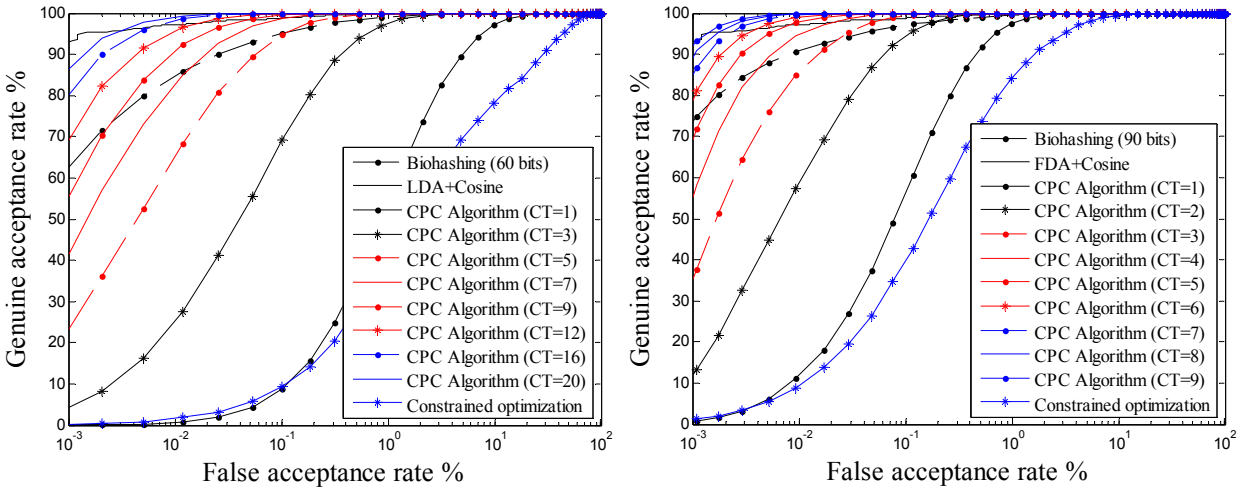
$$\ddot{x}_k = \arg\min_x \left\| x - \kappa_k \right\| \quad \text{subject to } x^T \zeta_j \le 0 \text{ if } w_j = 0 \text{ and } x^T \zeta_j \ge 0 \text{ if } w_j = 1 \ \forall j , \tag{32}$$

where w_j is a bit in a compromised template and ζ_j is a column vector from the compromised random matrix. Note that both Eq. 32 and the CPC algorithm require information from the compromised random matrix **Ω**. After obtaining $\ddot{x}_k$ s, the constrained optimization method computes the weight average, i.e.,

$$x_f = \frac{\sum_{k=1}^{K} \ddot{x}_k / d_k^2}{\sum_{k=1}^{K} 1 / d_k^2}, \tag{33}$$

where d_k is the hamming distance between the Biohashing templates of κ_k and $\ddot{x}_k$. To understand why this constrained optimization method cannot surpass the CPC algorithm (*CT*=1), let us consider the case of only one unrelated feature κ_1 first. Because *K*=1, we can derive that $x_f = \ddot{x}_1$. Note that κ_1 is a feature from another person and κ_1 therefore cannot fulfill all constraints in Eq. 27, i.e., $\exists\ j$ such that $\kappa_1^T \zeta_j < 0$ if $w_j = 1$ or $\kappa_1^T \zeta_j \ge 0$ if $w_j = 0$. In other words, κ_1 is not inside the convex polyhedral cone of the compromised template. The solution obtained from Eq. 32 is on the boundary of the convex polyhedral

cone (Fig. 15). If more than one unrelated feature is used, the final solution x_f is a weighted average of a set of points on the boundary of the convex polyhedral cone. It can be considered as a method of estimating the central ray of the convex polyhedral cone. If the unrelated feature vectors are used appropriately, methods that outperform the CPC algorithm (*CT*=1) are likely to be developed [20].



(a) (b)

Fig. 14 ROC curves of using convex polyhedral cones to attack Biohashing with (a) 60 bits and (b) 90 bits.

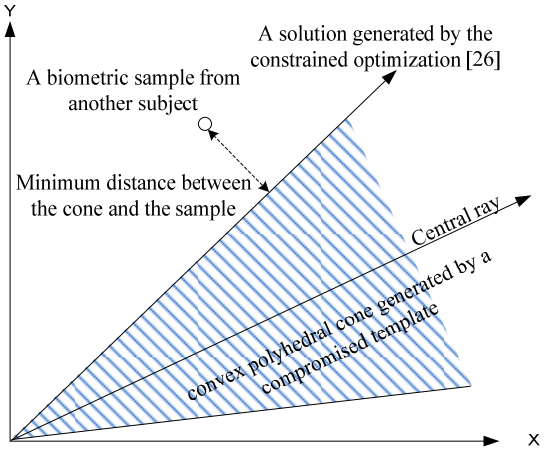


Fig. 15 Illustration of the phenomenon that the CPC algorithm (*CT*=1) and the constrained optimization method perform similarly.

5. Discussion

Though the number of theoretical properties of IrisCode has recently been discovered, our understanding of this influential iris recognition algorithm remains incomplete. In this paper, we indicate that templates

produced by IrisCode and its generalization are convex polyhedral cones. From a geometric perspective, IrisCode cuts down the hyperspace formed by normalized iris images into 2^{2048} convex polyhedral cones. Each convex polyhedral cone represents one iris template, but an iris is represented by a set of convex polyhedral cones that are near in terms of hamming distance.

Using the geometric properties of convex polyhedral cones, a simple algorithm that can often be implemented in one MATLAB command line has been developed to estimate the central rays of IrisCode and GIrisCode. The experimental results show that these central rays can match iris and palmprint templates produced by different methods and break into protected biometric templates. They imply that the central rays are enough to reveal the relationships among templates produced by different methods and break into systems without liveness detectors and quality checkers and that Biohashing has a limited capability to reissue new templates. The central rays of a compromised template can also be submitted into the data link between a biometric sensor and a feature extractor [32]. Except for replay attack at the sensor level, all other attacks do not depend on the visual quality of the preprocessed images (iris image) generated from the central rays. For some methods, central rays are, in fact, preprocessed images with reasonable visual quality. Fig. 16 shows preprocessed images generated from the central rays of Ordinal Codes (di-lobe d=5). The DC and the contrast of the original images are used to display the central rays to avoid visual differences caused by these two factors. Because Ordinal Code uses only the left and the right parts of an iris, Fig. 16 only shows these two parts [9].

Without solid security analyses, any biometric template that is a convex polyhedral cone cannot be assumed to be secure. The simplicity of the geometric algorithm implies that even junior hackers without advanced image processing knowledge can invade a system. Even worse, the central ray is also the expected ray and the optimal ray of an objective function on a group of distributions. Finally, it should be re-emphasized that this paper aims not to reconstruct high quality images from templates but instead to provide a deeper understanding of the geometric structures of IrisCode and its variants and investigate the risk from this geometric information.

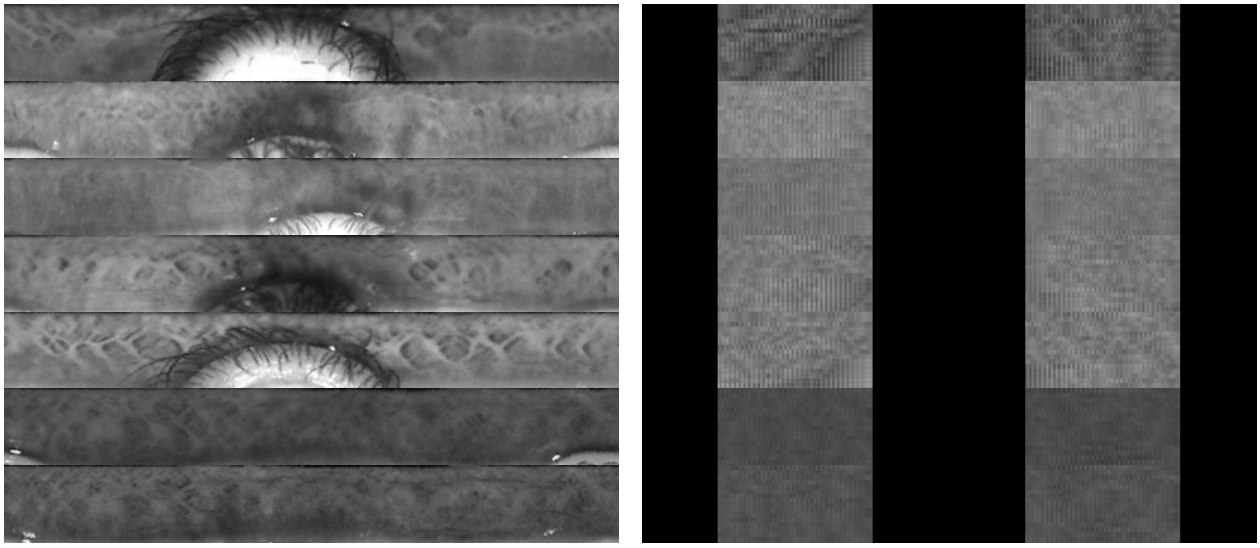


Fig. 16 Visual comparison between the original images and the central rays of Ordinal Code (di-lobe d=5). The first column presents the original images, and the second column presents preprocessed images from the central rays.

Acknowledgements

We would like to thank the University of West Virginia, the University of Beira Interior and the Hong Kong Polytechnic University for sharing their databases, Prof. Daugman for his valuable comments, and Mr. Jose Thomas Thayil for drawing all three-dimensional figures. This work is partially supported by Academic Research Fund Tier 1 (RG6/10) from the Ministry of Education, Singapore.

References

[1] J.G. Daugman, "High confidence visual recognition of persons by a test of statistical independence", *IEEE 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, B*, vol. 37, no. 5, pp. 1167-1175, 2007.
[4] A.W.K Kong, D. Zhang and M. Kamel, "An analysis of IrisCode", *IEEE TIP*, vol. 19, no. 2, pp. 522-532, 2010.
[5] A.W.K. Kong and D. Zhang, "Competitive coding scheme for palmprint verification", *in Proc. International Conference on Pattern Recognition*, vol. 1, pp. 520-523, 2004.
[6] A.K. Jain, S. Prabhakar, L. Hong, and S. Pankanti, "Fiterbank-based fingerprint matching", *IEEE TIP*, vol. 9, no. 5, pp. 846-859, 2000.
[7] 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.
[8] L. Ma, T. Tan, Y. Wang, and D. Zhang, "Efficient iris recognition by characterizing key local variations", *IEEE TIP*, vol. 13, no. 6, pp. 739-750, 2004.

[9] Z. Sun and T. Tan, "Ordinal measures for iris recognition", *IEEE TPAMI*, vol. 31, no. 12, pp. 2211-2226, 2009.
[10] L. Zhang, L. Zhang, D. Zhang and H. Zhu, "Online finger-knuckle-print verification for personal authentication", *Pattern Recognition*, vol. 43, pp. 2560-2571, 2010.
[11] H.A Park and K.R. Park, "Iris recognition based on score level fusion by using SVM", *Pattern Recognition Letters*, vol. 28, pp. 2019-2028, 2007.
[12] H. Proença and L.A. Alexandre, "Toward noncooperative iris recognition: a classification approach using multiple signatures", *IEEE TPAMI*, vol. 29, no. 4, pp. 607- 612, 2007.
[13] 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.
[14] 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.
[15] W. K. Kong and D. Zhang, "Palmprint texture analysis based on low-resolution images for personal authentication", *in Proceeding of ICPR*, pp. 807-810, 2002.
[16] A.T.B. Jin, D.N.C. Ling and A. Goh, "Biohashing: two factor authentication featuring fingerprint data and tokenized random number", *Pattern Recognition*, vol. 37, pp. 2245-2255, 2004.
[17] A. Kong, K.H Cheung, D. Zhang, M. Kamel and J. You, "An analysis of Biohashing and its variants", *Pattern Recognition*, vol. 39, no. 7, pp. 1359-1368, 2006.
[18] K.P. Hollingsworth, K.W. Bowyer and P.J. Flynn, "The best bits in an iris code", *IEEE TPAMI*, vol. 31, no. 6, pp. 964-973, 2009.
[19] A. Kong, "An analysis of Gabor detection", *in Proc. of International Conference on Image Analysis and Recognition*, pp 64-72, 2009.
[20] A.W.K. Kong, "IrisCode decompression based on the dependence between its bit pairs", *IEEE TPAMI*, vol. 34, no. 3, pp. 506-520, 2012.
[21] B. Grünbaum, Convex Polytopes, Second Edition, Springer-Verlag, New York, 2003.
[22] G.M. Ziegler, Lectures on Polytopes, Springer-Verlag, New York, 1995.
[23] C.B. Barber, D.P. Dobkin, and H.T. Huhdanpaa, "The quickhull algorithm for convex hulls", *ACM Transactions on Mathematical Software*, vol. 22, no. 4, pp. 469-483, 1996.
[24] J. Telgen, "Minimal representation of convex polyhedral sets", *Journal of Optimization Theory and Applications*, vol. 38, no. 1, pp. 1-24, 1982.
[25] K.H. Cheung, A.W.K. Kong, J. You and D. Zhang, "An analysis on invertibility of cancelable biometrics based on BioHashing", *in Proceedings of the International Conference on Imaging Science, Systems, and Technology*, pp. 40-45, 2005.
[26] A. Nagar, K. Nandakumar and A. K. Jain, "Biometric template transformation: a security analysis", *in Proc of Media Forensics and Security II*, 2010.
[27] H. Proenca and L.A. Alexandre, "UBIRIS: a noisy iris image database", *in Proc. of the 13th International Conference on Image Analysis and Processing*, vol. 1, pp. 790-977, 2005.
[28] A. Ross and S. Shah, "Segmenting non-ideal irises using geodesic active contours", *in Proc. of Biometrics Symposium*, pp. 1-6, 2006.
[29] D. Zhang, W.K. Kong, J. You and M. Wong, "On-line palmprint identification", *IEEE TPAMI*, vol. 25, no. 9, pp. 1041-1050, 2003.
[30] A.B.J. Teoh, A. Goh and D.C.L. Ngo, "Random multispace quantization as an analysis mechanism for Biohashing of biometric and random identity inputs", *IEEE TPAMI*, vol. 28, no. 12, pp. 1892-1901, 2006.
[31] A.W.K. Kong and D. Zhang, "Feature-level fusion for effective palmprint authentication", *in Proceeding of International Conference on Biometric Authentication*, pp. 520-523, 2004.
[32] N.K. Ratha, J.H. Connell and R.M. Bolle, "Enhancing security and privacy in biometrics-based authentication systems", *IBM Systems Journal*, vol. 40, no. 3, pp. 614-634, 2001.
[33] J. Daugman, Personal Communication.