An Analysis of IrisCode

[1]**Adams Kong**, *IEEE Member*, [2]**David Zhang**, *IEEE Fellow*,
and [3]**Mohamed Kamel,** *IEEE Fellow*

[1] Forensics and Security Laboratory,
School of Computer Engineering, Nanyang Technological University,
Nanyang Avenue, Singapore, 639798

[2]Biometrics Research Centre
Department of Computing, The Hong Kong Polytechnic University
Kowloon, Hong Kong

[3]Pattern Analysis and Machine Intelligence Research Group
University of Waterloo,
200 University Avenue West, Ontario, Canada

Corresponding author:

Adams Kong

School of Computer Engineering, Nanyang Technological University,

Nanyang Avenue, Singapore, 639798

Phone: (65) 6513-8041

Fax: (65) 6792-6559

E-mail: adamskong@ntu.edu.sg

Abstract — IrisCode is an iris recognition algorithm developed in 1993 and continuously improved by Daugman. It has been extensively applied in commercial iris recognition systems. IrisCode representing an iris based on coarse phase has a number of properties including rapid matching, binomial impostor distribution and a predictable false acceptance rate. Because of its successful applications and these properties, many similar coding methods have been developed for iris and palmprint identification. However, we lack a detailed analysis of IrisCode. The aim of this paper is to provide such an analysis as a way of better understanding IrisCode, extending the coarse phase representation to a precise phase representation, and uncovering the relationship between IrisCode and other coding methods. Our analysis demonstrates that IrisCode is a clustering algorithm with four prototypes; 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; Gabor function can be considered as a phase-steerable filter and the bitwise hamming distance can be regarded as a bitwise phase distance. We also discuss the theoretical foundation of the impostor binomial distribution. We use this analysis to develop a precise phase representation which can enhance accuracy. Finally, we relate IrisCode and other coding methods.

Keywords: Biometrics, iris recognition, Daugman algorithm, palmprint recognition, phase, Gabor filter

1. Introduction

Various biometric systems have been developed for governmental and commercial applications. Most of these systems can verify, 1-to-1 match or identify a person in a small database, 1-to-many match. Real time large-scale identification is still a challenging problem in terms of matching speed and accuracy. Of existing biometric technologies, [1]IrisCode developed in 1993 and continuously improved by Daugman [1-4] is able to identify a person in an extremely large database in real time. Although recently there has been some debate as to the accuracy of IrisCode [5], IrisCode has been extensively deployed in

[1] In 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.

commercial iris recognition systems for various security applications and more than 50 million persons have been enrolled using IrisCode. Following the key idea of IrisCode, researchers have developed different coding methods for use in iris and palmprint recognition [6-18, 27].

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

$$h_{\mathrm{Re}} = 1 \quad if \quad \mathrm{Re}\left(\int_{\rho}\int_{\phi} I(\rho,\phi) e^{-(r_0-\rho)^2/\alpha^2} e^{-(\theta_o-\phi)^2/\beta^2} e^{-i\omega(\theta_0-\phi)} \rho d\rho d\phi\right) \geq 0, \tag{1}$$

$$h_{\mathrm{Re}} = 0 \quad if \quad \mathrm{Re}\left(\int_{\rho}\int_{\phi} I(\rho,\phi) e^{-(r_0-\rho)^2/\alpha^2} e^{-(\theta_o-\phi)^2/\beta^2} e^{-i\omega(\theta_0-\phi)} \rho d\rho d\phi\right) < 0, \tag{2}$$

$$h_{\mathrm{Im}} = 1 \quad if \quad \mathrm{Im}\left(\int_{\rho}\int_{\phi} I(\rho,\phi) e^{-(r_0-\rho)^2/\alpha^2} e^{-(\theta_o-\phi)^2/\beta^2} e^{-i\omega(\theta_o-\phi)} \rho d\rho d\phi\right) \geq 0, \tag{3}$$

$$h_{\mathrm{Im}} = 0 \quad if \quad \mathrm{Im}\left(\int_{\rho}\int_{\phi} I(\rho,\phi) e^{-(r_0-\rho)^2/\alpha^2} e^{-(\theta_o-\phi)^2/\beta^2} e^{-i\omega(\theta_0-\phi)} \rho d\rho d\phi\right) < 0, \tag{4}$$

where r_0, θ_0, ω, α and β are the parameters of the Gabor filters [2]. The bitwise hamming distance is used to measure the difference between two IrisCodes. The first version of the bitwise hamming distance is defined as $HD = \sum_{i=1}^{2048} (A_i \otimes B_i) / 2048$, where A_i and B_i are the bits in two IrisCodes and $\otimes$ represents the bitwise operator, XOR. The current IrisCode uses a mask to exclude the corrupted bits from eyelashes, reflection, eyelids, and low signal-to-noise ratio [2]. The hamming distance between two IrisCodes is redefined as

$$HD = \frac{\sum_{i=1}^{2048} ((A_i \otimes B_i) \cap (A_i^M \cap B_i^M))}{\sum_{i=1}^{2048} (A_i^M \cap B_i^M)}, \tag{5}$$

where A^M and B^M are respectively the masks of IrisCodes A and B and $\cap$ represents bitwise operator AND.

The desirable qualities of IrisCode are well known. It is robust against local brightness and contrast variations because of the zero DC Gabor filters and the coding scheme. In rotating between any adjacent phase quadrants, only a single bit in IrisCode changes and this can enhance the robustness of the genuine distribution. In rotating between one phase quadrant to the opposite phase quadrant, both two bits in IrisCode change, which is to say that the distances between phase quadrants are retained. We refer to this representation as cyclic representation. Another well-known property of IrisCode is that it produces a binomial impostor distribution with high degrees-of-freedom (which does not necessarily imply high accuracy [30]). Making use of this property, the decision threshold is dynamically changed according to a predictable false acceptance rate from the binomial impostor distribution. Some researchers have developed an iris individuality model [24] based on the binomial impostor distribution. The key to high speed matching is the bitwise hamming distance. IrisCode can perform one million comparisons per second using a computer with a 3G Hz processor. This speed can be further improved by applying Beacon Guided Search [20].

It is generally believed that the cores of IrisCode are the operators, "≥" and "<" in the Eqs. 1-4 and the bitwise hamming distance. These two operators allow each feature value to be represented by one bit then two encoded features are compared using the bitwise hamming. Researchers replace the Gabor filters in IrisCode with different filters and transformations including [2]quadratic spline wavelet, Haar wavelet frame, log Gabor filters, independent component analysis, directional filter banks and dissociated tripole filters [10-18] to develop new coding methods for iris recognition. Based on this understanding, some researchers claim that IrisCode is a local ordinal feature [11]. Some researchers further believe that the impostor distribution of their coding method also follows a binomial distribution [19]. However, this understanding of IrisCode is incomplete and limits the design of new coding schemes for feature

[2] The coding scheme in [10] does not explicitly use the two operators, "≥" and "<". In fact, it can be rewritten based on these two operators. Readers can refer appendix A.

representation. One weakness of this understanding is that each filter response or coefficient provides only one bit of information. It lacks representational flexibility. Furthermore, some claims are controversial. Developing an iris individuality model with a solid theoretical foundation also requires a complete understanding [24].

A detailed analysis of IrisCode is important for understanding IrisCode, for designing new coding schemes, and for clarifying the relationship between IrisCode and other coding methods. Nevertheless, such an analysis has not been found in the literature. In this paper, we investigate the relationship between IrisCode and clustering algorithms, the property of the Gabor function, and the relationship between the bitwise hamming distance and bitwise phase distance. We also discuss the theoretical foundation of the binomial impostor distribution. Making use of this analysis, we develop an algorithm for precise phase representation with effective filtering and matching. Finally, we study the relationship between IrisCode and other coding methods.

The rest of this paper is organized as follows. Section 2 describes the properties of IrisCode from the point of view of clustering. Section 3 presents an algorithm for precise phase representation. Section 4 discusses the theoretical basis for the binomial impostor distribution. Section 5 shows the relationship between IrisCode and different coding methods. Section 6 offers some concluding remarks.

2. Understanding IrisCode from a Clustering Point of View.

In this section, we demonstrate that IrisCode is a clustering algorithm and study the properties of the Gabor function. The relationship between bitwise hamming distance and bitwise phase distance is presented in Section 3.

2.1 IrisCode — A Clustering Algorithm

Let $M_R(\rho, \phi)$ and $M_I(\rho, \phi)$ be the real and imaginary parts of a Gabor filter. For convenience, we use M_R to denote $M_R(\rho, \phi)$. We use the same notations for other symbols. The definitions of M_R and M_I are as follows:

$$M_R(\rho,\phi) = e^{-(r_0-\rho)^2/\alpha^2} e^{-(\theta_o-\phi)^2/\beta^2}\left(\cos(-\omega(\theta_0-\phi))\right), \tag{6}$$

$$M_I(\rho,\phi) = e^{-(r_0-\rho)^2/\alpha^2} e^{-(\theta_o-\phi)^2/\beta^2}\left(\sin(-\omega(\theta_0-\phi))\right). \tag{7}$$

We do not remove the DC term of the Gabor filter in Eq. 6 since in this paper we assume that the DC of the iris patch for filtering has been removed.

We define a continuous *periodic* function,

$$Z(\varphi) = (\cos(\varphi)M_R + \sin(\varphi)M_I), \tag{8}$$

with respect to the phase parameter, φ, where $\varphi \in [0, 2\pi)$. $Z(\varphi)$ is called a filter-generating function. Using $Z(\varphi)$, we can obtain four filters by substituting $5\pi/4$, $7\pi/4$, $\pi/4$, and $3\pi/4$ to φ. The four filters are

$$Z_0 = Z(5\pi/4) = (-M_R - M_I)/\sqrt{2}, \tag{9}$$

$$Z_1 = Z(7\pi/4) = (M_R - M_I)/\sqrt{2}, \tag{10}$$

$$Z_2 = Z(\pi/4) = (M_R + M_I)/\sqrt{2}, \tag{11}$$

$$Z_3 = Z(3\pi/4) = (-M_R + M_I)/\sqrt{2}. \tag{12}$$

These four filters are shown in Fig. 1 and will later be regarded as the cluster centers. We define

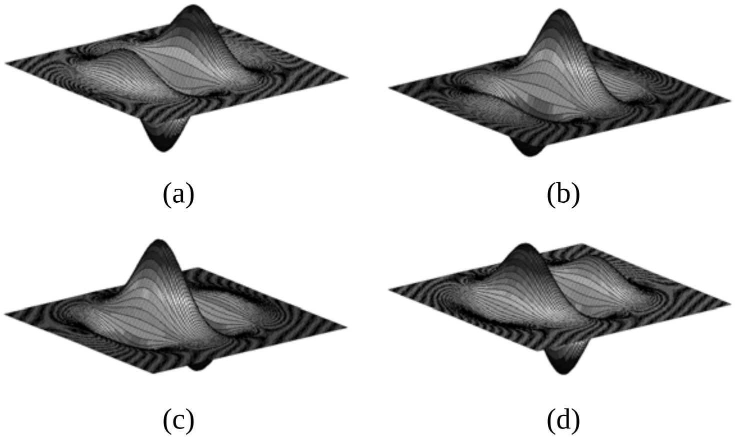
$$j = \arg\max_i \left(\int_\rho \int_\phi \rho I Z_i \, d\rho d\phi \right), \tag{13}$$

as a clustering criterion, where j is called the winning index and I is the iris image in the dimensionless polar coordinate system. It is equivalent to the cosine measure between Z_i and ρI, i.e., $j = \arg\max_i \int_\rho \int_\phi \rho I Z_i d\rho d\phi / \|\rho I\| \|Z_i\|$ since the four filters have the same power, i.e., $\int_\rho \int_\phi Z_i^2 d\rho d\phi = C$, where C is a constant, which can be proved by using the orthogonal property between M_R and M_I i.e., $\int_\rho \int_\phi M_I M_R d\rho d\phi = 0$. The winning index is an integer representation of φ. For fast matching, we have to encode the winning index. Table 1 gives the coding table under the heading “Coded winning indexes”.

The difference between two encoded winning indexes is also measured by their bitwise hamming distance, as in IrisCode. Table 1 compares bits of IrisCode and the binary representation of the winning indexes, demonstrating their equivalence. In the other words, IrisCode is a clustering algorithm and the cosine measure is the clustering criterion.

Table 1. Comparison of IrisCode, winning index and coded winning index

IrisCode		Winning index	Coded winning indexes	
h_{Im}	h_{Re}		Bit 2	Bit 1
0	0	0	0	0
0	1	1	0	1
1	1	2	1	1
1	0	3	1	0



(a) (b)

(c) (d)

Fig. 1 The four filters used in IrisCode, a) Z_0, b) Z_1, c) Z_2 and d) Z_3.

2.2 Properties of the Gabor Function

Let us study the physical meaning of φ in the filter generating function. Reordering the terms in Eq. 8, $Z(\varphi)$ can be rewritten as

$$Z(\varphi) = e^{-(r_0-\rho)^2/\sigma^2} e^{-(\theta_0-\phi)^2/\beta^2} \cos(-\omega(\theta_0-\phi)-\varphi)\,. \tag{14}$$

It is clear that φ is the phase of a Gabor function and the filter generating function can be rewritten as a

Gabor function. According to Eqs. 9-12 and 14, it is very clear that IrisCode is a *periodic feature*.

Next, we study the locus of the filter generating function, also the Gabor function in Eq. 14 with respect to the phase parameter. We discretize M_R, M_I and Z to obtain three vectors, $\vec{M}_R$, $\vec{M}_I$ and $\vec{Z}(\varphi)$, respectively and define two orthonormal vectors, $\vec{v}_R = \vec{M}_R / \|\vec{M}_R\|$ and $\vec{v}_I = \vec{M}_I / \|\vec{M}_I\|$. Using these two vectors, $\vec{Z}(\varphi)$ can be rewritten as

$$\vec{Z}(\varphi) = (\cos(\varphi)\|\vec{M}_R\|\vec{v}_R + \sin(\varphi)\|\vec{M}_I\|\vec{v}_I), \tag{15}$$

a linear combination of $\vec{v}_R$ and $\vec{v}_I$. The coordinate of $\vec{Z}(\varphi)$ in the two dimensional space spanned by $\vec{v}_R$ and $\vec{v}_I$ is $(\|M_R\|\cos(\varphi), \|M_I\|\sin(\varphi))$ satisfying the following equality

$$\frac{(\|M_R\|\cos(\varphi))^2}{\|M_R\|^2} + \frac{(\|M_I\|\sin(\varphi))^2}{\|M_I\|^2} = 1. \tag{16}$$

Obviously, the locus of $\vec{Z}(\varphi)$ with respect to φ is an ellipse on the two dimensional space. In fact, the locus can be further constrained. Under suitable parameterization, it is a circle. The mathematical proof is given in Appendix B. Even if the condition in Appendix B cannot be fulfilled, the locus of $\vec{Z}(\varphi)$ can be approximated by a circle in many applications of Gabor filters because their $\|M_R\|$ is approximately equal to their $\|M_I\|$ [33].

3. The Precise Phase Representation Algorithm

To design a precise phase representation, we begin by taking more sample points from the filter generating function to generate more prototypes for the clustering algorithm. We use uniform sampling to obtain the filters, i.e., [3] $Z(i\pi / n + \eta)$, where i=0, 1,…2n-1 and η is an offset. For the design of bitwise phase distance, the number of prototypes is set to 2n, where n is called the order of the coding scheme.

[3] Please note the i in $Z(i\pi / n + \eta)$ is not $\sqrt{-1}$.

For IrisCode, n is 2 and η is $5\pi/4$. It should be noted that the clustering criterion based on the cosine measure is independent of the contrast of the image. The brightness of the iris is normalized. To embed other inherent properties of IrisCode such as rapid matching and cyclic representation, we have to design a novel coding scheme to encode the winning indexes and a distance measure for rapid matching. In the following subsections, we ignore the offset η without loss of generality.

3.1 Phase Distance

Since $\vec{Z}(\varphi)$ is on a two dimensional ellipse and φ is the phase, the distance between $\vec{Z}(\omega)$ and $\vec{Z}(\gamma)$ can be measured by the phase distance between ω and γ defined as $\min(|\omega-\gamma|, 2\pi-|\omega-\gamma|)$. If we use uniform sampling to obtain $\vec{Z}(\omega)$ and $\vec{Z}(\gamma)$ *i.e.,* $\omega = 2\pi p/2n$ and $\gamma = 2\pi q/2n$ where p and q are two integers between 0 and $2n$-1, the phase distance can be rewritten as $\min\left(\frac{\pi}{n}|p-q|, \frac{\pi}{n}\left(2n-|p-q|\right)\right)$. The phase distance can be further simplified as

$$\min\left(|p-q|, \left(2n-|p-q|\right)\right), \tag{17}$$

if π/n is defined as one unit distance. As a result, the phase distance between $\vec{Z}(\omega)$ and $\vec{Z}(\gamma)$ is defined only based on their winning indexes, p and q, the integer presentation of their phases. The phase distance between any two adjacent winning indexes is 1, as in IrisCode. The cyclic representation has been embedded in the precise phase representation.

3.2 Bitwise Matching and Coding Scheme

The bitwise hamming distance supporting high speed matching is one of the keys to large-scale iris identification in real time. However, the winning index and phase distance in Eq. 17 are integer representations. To embed high speed matching in the precise phase representation, we have to design a new coding scheme to encode the integer winning indexes and to develop a bitwise phase distance. A coding table $A = [a_{i,j}]$ is designed for this purpose, where $1 \le i \le n$; $1 \le j \le 2n$ and $a_{i,j}$ is defined as in Fig. 2.

```
$if \quad j \le n \quad and \quad 1 \le i < j$,
        $a_{i,j} = 1$
$elseif \quad j > n \quad and \quad j - n \le i \le n$,
        $a_{i,j} = 1$
$else$
        $a_{i,j} = 0$
```

Fig. 2. Pseudo code of the coding table.

For illustration, Table 2 shows two coding tables for *n*=3 and 4. These tables have a structure where each winning index is represented by the column of *A* with the result that *n* bits are used to encode one winning index.

It should be noted that when *n*=1 and *n*=2, the bitwise winning indexes form a [4]cyclic code [25] and a gray code. However, when *n* is greater than two, the bitwise winning indexes do not form a cyclic code since cyclically shifting a bitwise winning index can generate a code that does not exist in the coding table. For instance, (1 0 0) is a code in Table 3a but (0 1 0) does not exist in this table. They also do not form a gray code since gray code uses *n* bits to represent 2^n integers. For example, (1 0 1) is a code in gray code but it does not exist in Table 3a. Cyclic code and gray code are not essential properties for precise phase representation since we have embedded cyclic representation in it.

When *n* is greater than two, the coding scheme does not fully utilize the code space. Only 2*n* code words in the space are used. This design is to ensure that adjacent phases are encoded with code words in which only one bit has changed, thus maintaining a distance metric.

[4] The definitions of cyclic code [25] and cyclic representation are different.

To achieve high speed matching, we need a bitwise matching for the encoded winning indexes. We discover that phase distance and bitwise hamming distance have an equivalent relationship, i.e. $\sum_{i=1}^{n} a_{i,j} \otimes a_{i,j+k} = \min(k, 2n-k)$. The mathematical proof is given in Appendix C. We refer to this bitwise hamming distance for precise phase representation as the bitwise phase distance since it measures the phase difference between two prototypes in the two dimensional ellipse.

Table 2. The coding tables for (a) *n*=3 and (b) *n*=4

(a)

Winning index	0	1	2	3	4	5
Bit 0	0	1	1	1	0	0
Bit 1	0	0	1	1	1	0
Bit 2	0	0	0	1	1	1

(b)

Winning index	0	1	2	3	4	5	6	7
Bit 0	0	1	1	1	1	0	0	0
Bit 1	0	0	1	1	1	1	0	0
Bit 2	0	0	0	1	1	1	1	0
Bit 3	0	0	0	0	1	1	1	1

3.3 Phase-Steerable Filtering

We have developed an algorithm for precise phase representation that embeds the properties of IrisCode including fast matching and cyclic representation. In the scheme we have presented so far, the number of filters needed would have to increase with the desired precision of the extracted phase information. Namely, 2*n* filters would be needed if the desired phase precision is *n* bits. However, now we will show that only the two cardinal filters M_R and M_I are actually required, regardless of the desired phase precision, because Gabor filters are phase-steerable.

Let us consider the clustering criterion, $j = \arg\max_{i} \int_{\rho}\int_{\phi} \rho I Z(i\pi/n) d\rho d\phi / \|\rho I\| \|Z(i\pi/n)\|$, where $\|\rho I(\rho,\phi)\|$ is independent of i and $\|Z(i\pi/n)\|$ can be pre-computed. These two terms would not dramatically increase the computational burden even when *n* is large. The major computation cost comes from $\int_\rho\int_\phi \rho I Z(i\pi/n)d\rho d\phi$.

Substituting Eq. 8 into $\int_\rho\int_\phi \rho I Z(i\pi/n)d\rho d\phi$, we have

$$\int_\rho\int_\phi \rho I(\cos(i\pi/n)M_R + \sin(i\pi/n)M_I)d\rho d\phi$$

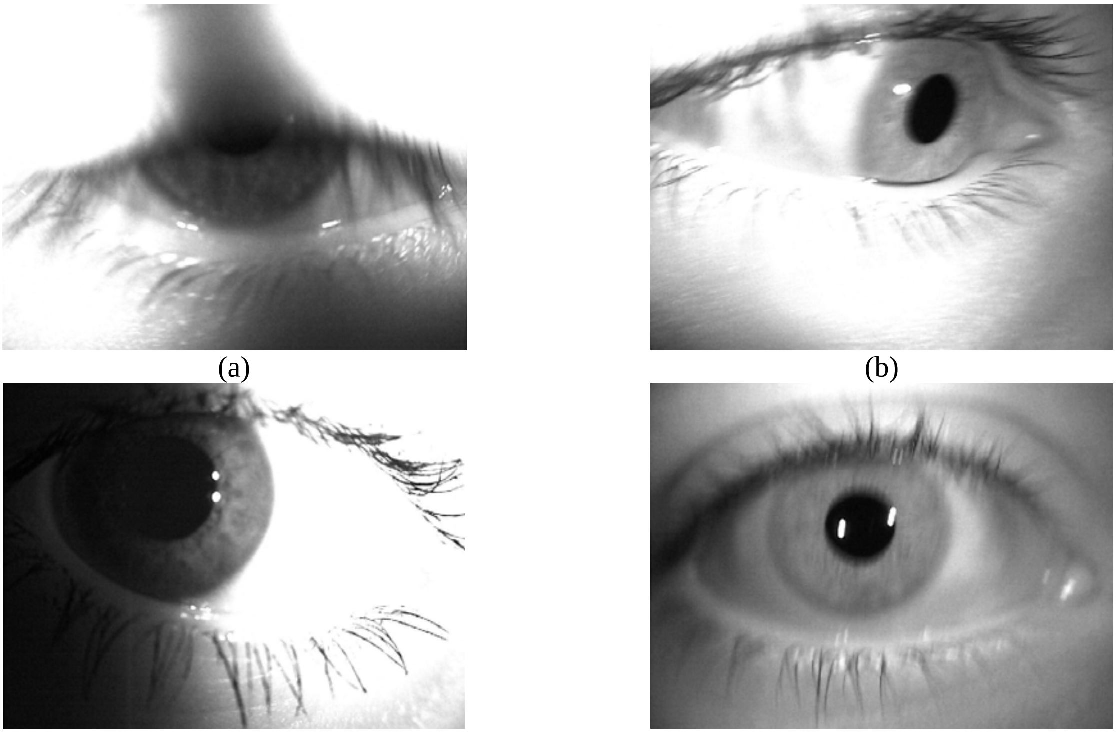
$$= \cos(i\pi/n)\int_\rho\int_\phi \rho I M_R d\rho d\phi + \sin(i\pi/n)\int_\rho\int_\phi \rho I M_I d\rho d\phi \qquad (18)$$

Eq. 18 shows that $Z(\varphi)$ is a phase-steerable filter [31] and we need only two filters, M_R and M_I for any precision of the phase. We have successfully reduced the computation complexity of filtering from *O*(*n*) to *O*(1). If the locus of $\vec{Z}(\varphi)$ is a circle, the solution phase, $\arg\max_\varphi \int_\rho\int_\phi \rho I Z(\varphi)d\rho d\phi / \|\rho I\|\|Z(\varphi)\|$ can be computed by $\varphi = \tan^{-1}\left(\int_\rho\int_\phi \rho I M_I d\rho d\phi \Big/ \int_\rho\int_\phi \rho I M_R d\rho d\phi\right)$. The mathematical proof is given in Appendix D. If the locus of $\vec{Z}(\varphi)$ is not a circle, φ can still be approximated by $\varphi_c = \tan^{-1}\left(\int_\rho\int_\phi \rho I M_I d\rho d\phi \Big/ \int_\rho\int_\phi \rho I M_R d\rho d\phi\right)$ and their error bound is $|\varphi - \varphi_c| \le \tan^{-1}(\frac{1}{\sqrt{K}}) - \tan^{-1}(\sqrt{K})$, where $K = \|M_I\|^2 / \|M_R\|^2$. The mathematical proof is given in [33].

3.4 Re-implementation of IrisCode.

To evaluate the performance of precise phase representation, we re-implemented IrisCode including pupil, limbus, and eyelid detection, eyelash segmentation, normalization, coding and matching for comparison.

Accurately re-implementing IrisCode is highly difficult since it is a complex computer vision system. Moreover, previously published work has not in every case clearly disclosed all details e.g. the computational details involved in eyelid detection. Further, since our re-implementation made use of the West Virginia University (WVU) iris database, which contains many challenging images, it was necessary to make a number of modifications in the preprocessing. Some examples are shown in Fig. 3. In the following we first describe iris segmentation, then normalization and filtering, and then matching. Our results are discussed in Section 3.5.



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

Fig. 3 Iris images in WVU iris database

3.4.1 Iris Segmentation

In our re-implementation, we first estimate the location of the specular reflections and scrub them. To localize a pupil boundary, we then apply the integro-differential operator,

$\max_{(r,x_o,y_o)}\left|G_\sigma(r)*\frac{\partial}{\partial r}\oint_{r,x_o,y_o}\frac{I(x,y)}{2\pi r}ds\right|$ reported in Daugman's publications. This boundary is used to initialize an active contour that can be used to accurately estimate the pupil boundary [3].

The original integro-differential operator for localizing a limbus boundary is separated to two integro-differential operators. One is for the left limbus boundary and the other is for the right limbus boundary. We also modify the integro-differential operator as follows:

$$\max_{(r,x_o,y_o)}\left|G_\sigma(r)*\frac{\partial}{\partial r}\oint_{S(r,x_o,y_o)}W(s)\frac{I(x,y)}{L(S)}ds\right|, \tag{19}$$

where, *S* is the left or right circular path controlled by *r*, x_0 and y_0, *L(S)* is the length of *S* and *W(s)* is a weighting function. The purpose of using two integro-differential operators is to handle the images captured under very uneven lighting environments (e.g. Fig. 3(c)). If the two boundaries have similar radii and centers, the center of mass of the two boundaries is regarded as the center of the limbus and the average distance between the center of mass and the two boundaries is regarded as the radius of the limbus; otherwise, we apply the integro-differential operator again on the weaker boundary and limit the searching space by the parameters of the stronger boundary. Then, an active contour is used to accurately estimate the limbus boundary [3].

To detect potential points of eyelids, a set of classifiers is sent to different locations. The locations of the classifiers depend on the parameters of the limbus and pupil. Then, ordinal statistics is employed to retain partial detected points for curve fitting. Daugman does not, as is widely believed, use the integro-differential operator to detect eyelids, but the details of the algorithm have not been disclosed [26]. A statistical test and some pair knowledge are used to segment the eyelashes [4, 21]. Fig. 4 shows a segmented iris.

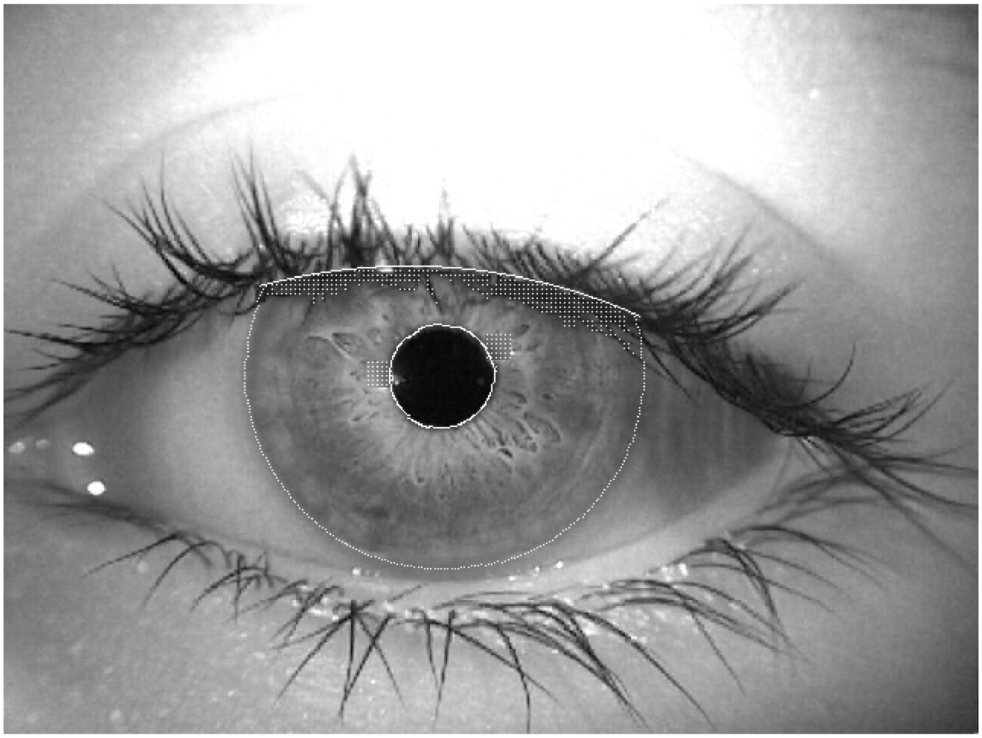


Fig. 4 A segmented iris

3.4.2 Normalization and Filtering

We use the dimensionless polar coordinate system to normalize an iris. Many researchers display the normalized iris as a rectangular sheet. Strictly speaking, the normalized iris is the surface of a cylinder and it has only two cuts from the pupil and limbus boundaries. The size of a normalized iris is 64 by 512 pixels.

We use sixteen Gabor filters with different parameters, determined by 300 different iris images to extract phase information. In parameter training, we have Gabor filters of two sizes, 8 by 105 and 24 by 105 pixels. The smaller one is applied to the sample points close to pupil and limbus boundaries. The *d'* index defined as $d'=\left|\mu_g-\mu_i\right|/\sqrt{(\sigma_g^2+\sigma_i^2)/2}$, where $\mu_{g(i)}$ and $\sigma_{g(i)}^2$ are the mean and variance of a genuine (impostor) distribution, is employed as the objective function for training. This index measures the degree of separation between the two distributions in units of their standard deviations, and thus it is analogous to Z-scores, Fisher scores, and other ways of measuring separation between distributions, which a biometric system aims to optimize.

In the training, we first select one set of parameters for the Gabor filter closest to the pupil boundary. Then we retain the first optimized parameters and select another set of parameters for the same sample points. The other sets of parameters are selected in the same way. We independently optimize the parameters of precise phase representation with different orders.

3.4.3 Matching

A raw hamming distance (HD_{raw}) is computed from Eq. 5. The number of effective bit matchings is different in each comparison because of eyelashes and eyelids. To obtain the same decision confidence, Daugman rescales the hamming distance with the following equation,

$$HD_{norm} = 0.5 - (0.5 - HD_{raw})\sqrt{m/960}\ , \tag{20}$$

where m is the number of actually compared bits and 960 is the average number of actually compared bits [3]. For precise phase presentation, normalized hamming distance is defined as

$$HD_{norm} = 0.5 - (0.5 - HD_{raw})\sqrt{(m/(n960/2))}\ , \tag{21}$$

where n is the order of the coding scheme.

3.5 Experimental Results

[5]The WVU iris database contains 3,099 iris images from 472 irises. The first 300 images were used to train the parameters. The remainders were used in testing. In the following experiments, we will use the Equal Error Rate (EER) and Receiver Operating Characteristic (ROC) curves as performance indexes.

Using the WVU database [32], Ross and Shah use geodesic active contours for segmentation and achieve an EER of 12.03% when matching left irises and an EER of 14.19% when matching right irises. They also examine Masek’s approach [19] and report EERs of 13.51% when matching left irises and

[5] Some mislabeled images are corrected by the authors

13.07% when matching right irises. Furthermore, they implement integro-differential operators for comparison and report EERs of 35.0% when matching left irises and 33.4% when matching right irises. These approaches can achieve EERs in the range of 1.9% and 4.29% in the CASIA-1 iris database. These results show clearly that the WVU iris database contains large numbers of low quality iris images.

3.5.1 Validation of Precise Phase Representation

To validate the precise phase representation, we design the following experiment. When *n*=2, the performance of precise phase representation and IrisCode should be the same. Fig. 5 shows the ROC curves of IrisCode and precise phase representation with order 2. As the theoretical predication, the two curves overlap completely. The EER of matching left irises is 1.5% and the EER of matching right irises is 1.3%. Compared with the previously reported EERs on this database, the quality of our re-implementation should be acceptable.

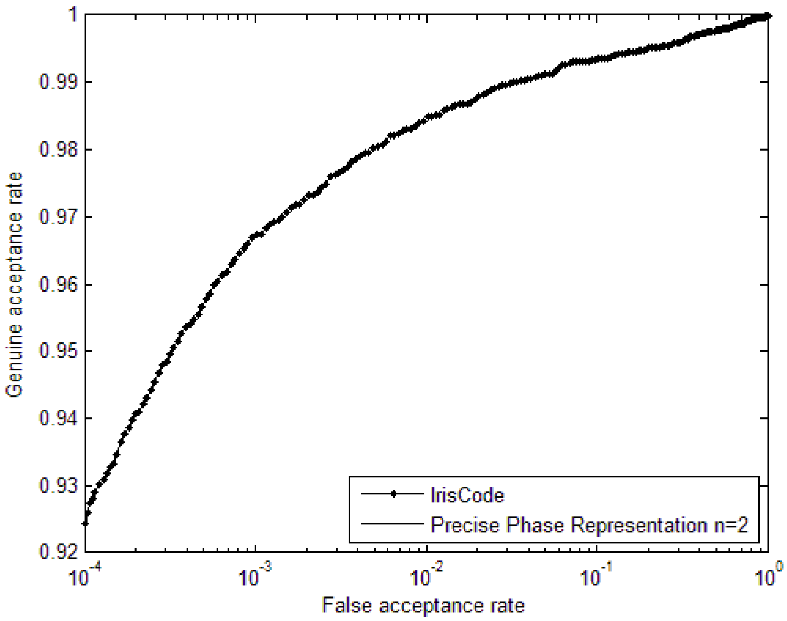


Fig. 5 Validation of precise phase representation

3.5.2 Comparison of IrisCode and Precise Phase Representation

To examine the effectiveness of precise phase representation, we compare IrisCode and precise phase representation for n=3, 4, and 5. We match iris images from the same eyes and from different eyes to respectively obtain 9,224 genuine matchings and 3,906,518 impostor matchings for each representation. The genuine and impostor matchings are used to estimate the genuine and impostor distributions. From the ROC curves in Fig. 6 we can see that the precise phase representation of order 5 is always the most accurate with an EER of 1.1%. Comparing IrisCode and precise phase representation of order 5, precise phase representation of order 5 has 1.7% improvement in genuine acceptance rate when the false acceptance rate is 10^{-4}.

In our own previous work using the CASIA-1 database, we had found that the precise phase representation of order 5 does not perform better than the precise phase representation of order 4 [28]. On this database, however, increasing the precision of phase always improves the accuracy. This is because the parameters of Gabor filters in precise phase representation of different orders are optimized independently for the WVU database. Fig. 7 shows the ROC curves of matching left and right irises.

It should be noted that the matching speed of precise phase representation is slower than IrisCode. The matching speed of precise phase representation of order n can be estimated by $T_n = 2 \times T_2 / n$, where T_n is the matching speed (in the unit of number of matching per second) of the precise phase representation and T_2 is the matching speed of IrisCode. IrisCode uses only two bits to represent one filter response while precise phase representation of order n requires n bits to represent one filter response. For some applications such as identifying a person in a residential building for access control, one million comparisons per second is much more than enough. We can use precise phase representation to achieve high accuracy for these applications. Although the WVU iris database contains many low quality iris images of the sort that may be acquired in less controlled environments (e.g. iris on the move) [29], precise phase representation is effective on these images. We cannot claim that precise phase representation is more accurate than IrisCode as these *identification* algorithms are intended for use on databases of different sizes but we can say that precise phase representation does allow us to balance speed and accuracy.

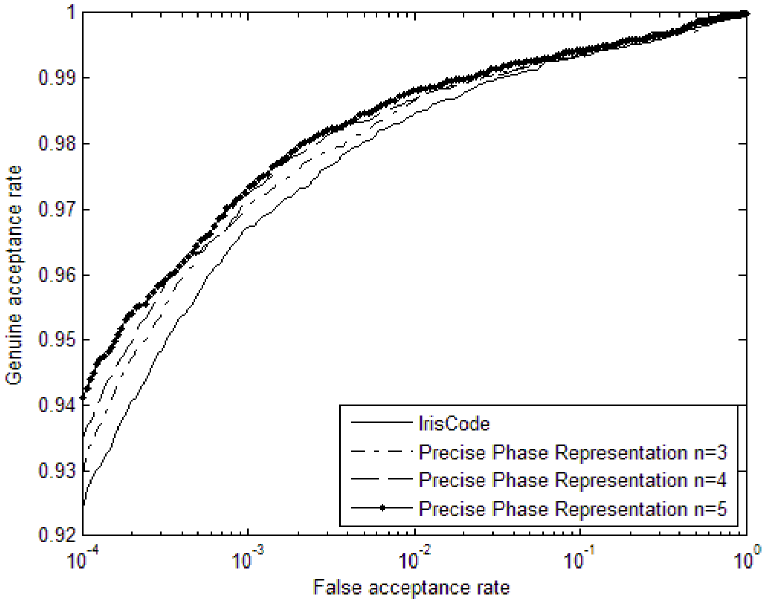
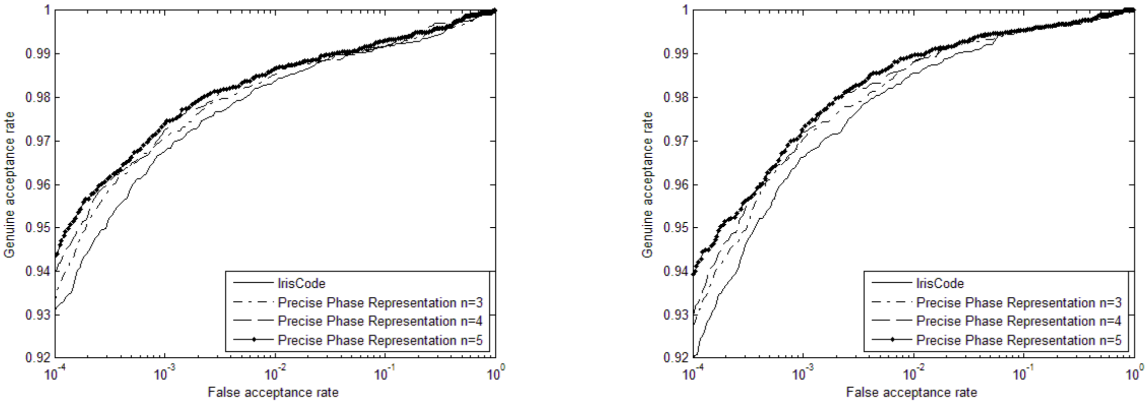


Fig. 6 ROC curves of IrisCode and precise phase representation for *n*=3, 4 and 5.



(a) (b)

Fig. 7 ROC curves of IrisCode and precise phase representation for *n*=3, 4 and 5. (a) Matching left irises and (b) matching right irises.

4. Theoretical Foundation of Binomial Impostor Distribution

In the previous sections, we used a number of properties of IrisCode to develop an algorithm for precise phase representation with effective filtering and rapid matching. However, we have not touched upon the binomial impostor distribution in IrisCode which is used to predict the false acceptance rate under different thresholds and different matching condition such as the variation in the number of actually compared bits. The binomial impostor distribution of IrisCode has been experimentally validated on a large database [3].

In practice, the impostor distribution of IrisCode is binomial. Since all of the feature values of similar coding methods including our precise phase representation designed for iris and palmprint [6-18, 27] are binary and two feature codes are compared using a hamming distance, some might also expect that the impostor distributions of these coding methods would also follow a binomial distribution.

Let us review the three assumptions of a binomial distribution. A random variable X following a binomial distribution should satisfy the following conditions.

1) X is defined as $\sum_{i=1}^{N} T_i$, where T_i is a Bernoulli variable.
2) All T_i has the same probabilities of success, p.
3) All T_i are independent.

We refer to (2) as a stationary condition and (3) as an independent condition. Obviously, the hamming distances of IrisCode and other coding methods satisfy condition (1). Daugman validated condition (2) in 1993 [1]. However, all coding methods violate the independent condition because of correlations among texture features, and because of correlations introduced by the bandlimited filters. If the sum of correlated and stationary Bernoulli variables followed a binomial distribution unconditionally, the impostor distributions of all coding methods would have to be binomial. However, no such mathematical theorem has been discovered. Even if the correlation is of the first order Markovian type, the distribution of the sum of correlated stationary Bernoulli trails can be bimodal and trimodal shapes [22-23]. As a result, we cannot guarantee that the impostor distributions of the other coding methods and precise phase representation also follow binomial distributions.

5. The Relationship between IrisCode and Other Coding Methods.

Many coding methods have been developed for iris and palmprint identification that are quite similar [6-18, 27]. The most common approach is to replace the Gabor filters in IrisCode with other linear transforms or filters. According to this analysis, IrisCode is a clustering with four prototypes. As it happens, most the other coding methods can also be regarded as clustering algorithms but with two prototypes. Let us formally define these approaches. Let F be a linear filter used in their coding methods. Their coding schemes can be summarized in the following equations,

$$h = 1 \quad if \quad \int_{\rho}\int_{\phi} FId\rho d\phi \geq 0\,, \tag{22}$$

$$h = 0 \quad if \quad \int_{\rho}\int_{\phi} FId\rho d\phi < 0\,, \tag{23}$$

where h is a resultant bit. We refer to this coding scheme as a standard coding scheme. To uncover the relationship between IrisCode and the standard coding scheme, we define a filter generating function $(-1)^{\upsilon+1}F$, where $\nu \in \{0, 1\}$. This filter generating function can generate only two filters, F and $-F$. Since these two filters have the same power, i.e., $\|F\| = \|-F\|$, the clustering criterion can be rewritten as

$$j = \arg\max_{i}\left(\int_{\rho}\int_{\phi}(-1)^{i+1} FId\rho d\phi\right). \tag{24}$$

If j=0, then we have $\int_{\rho}\int_{\phi} - FId\rho d\phi > \int_{\rho}\int_{\phi} FId\rho d\phi$ and $0 > \int_{\rho}\int_{\phi} FId\rho d\phi$. If j=1, we have $\int_{\rho}\int_{\phi} FId\rho d\phi > \int_{\rho}\int_{\phi} - FId\rho d\phi$ and $\int_{\rho}\int_{\phi} FId\rho d\phi > 0$. Using the first order coding scheme defined in Fig. 2 to encode the winning index j, we obtain Eqs. 22 and 23.

In addition to the standard coding scheme, other coding methods based on Gabor filters and log Gabor filters employ the order 2 coding scheme in precise phase representation [12, 15]. The authors also develop a Competitive Code for palmprint identification [8]. Its filter generating function is the negative real part of a Gabor function. We assign different values to the orientation parameter so as to generate six filters and use order 3 coding scheme to encode the winning indexes. We employ bitwise phase (angular [8]) distance to measure two different Competitive Codes.

The standard coding method, IrisCode, and Competitive Code respectively employ the order 1, 2 and 3 coding schemes in precise phase representation. Their differences are in their filter generating functions. Most coding methods including standard coding method, IrisCode, Competitive Code and precise phase representation are under the same framework given in Fig. 8. The cores of this framework include filter generating function, clustering, coding scheme and bitwise phase matching. It is worth to mention that both loci of the filter generating functions of the standard coding method and IrisCode are always on two-dimensional planes. However, the locus of the filter generating function of Competitive Code is on a higher dimensional plane.

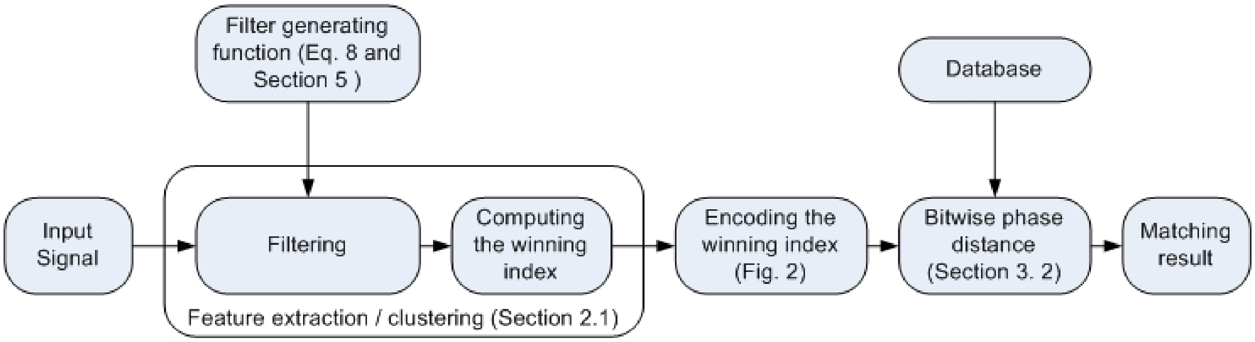

Filter generating function (Eq. 8 and Section 5)
Database
Input Signal
Filtering
Computing the winning index
Encoding the winning index (Fig. 2)
Bitwise phase distance (Section 3. 2)
Matching result
Feature extraction / clustering (Section 2.1)

Fig. 8 A common framework employed by most of the existing coding methods.

6. Conclusion

IrisCode first appeared sixteen years ago yet to our knowledge this is the first paper to provide a detailed analysis of this method. The analysis made here makes a number of contributions. It presents a complete analysis of IrisCode, demonstrating that IrisCode is a clustering algorithm and that the locus of a Gabor function is on a two dimensional ellipse with respect to the phase parameter. It also proves the equivalent relationship between the bitwise hamming distance and bitwise phase distance and shows that Gabor function can be considered as a phase-steerable filter. It then uses these properties and this relationship to develop a precise phase representation algorithm. This algorithm inherits the properties of IrisCode including robustness against brightness and contrast variations and rapid matching based on bitwise operators and cyclic representation. Our experiments have shown that given the same quality of preprocessing precise phase representation is more accurate than IrisCode. Precise phase representation is a flexible representation for balancing the tradeoff between matching speed and identification accuracy. This paper has also discussed theoretical issues regarding the binomial impostor distribution of IrisCode and other coding methods. Finally, using the filter generating function and the coding scheme defined in this study, we have shown the relationships between IrisCode and other iris and palmprint recognition coding methods.

Acknowledgements

The authors would like to thank Prof. Daugman for his valuable comments. We also would like to thank the University of West Virginia for sharing their database.

References:

[1] J.G. Daugman, “High confidence visual recognition of persons by a test of statistical independence”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, no. 11, pp. 1148-1161, 1993.

[2] J. Daugman, “How iris recognition works”, *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 14, no. 1, pp. 21-30, 2004.

[3] J. Daugman, “Probing the uniqueness and randomness of IrisCode: results from 200 billion iris pair comparisons”, *Proceedings of the IEEE*, vol. 94, no. 11, 2006.

[4] J. Daugman, “New methods in iris recognition”, *IEEE Transactions on System, Man and Cybernetics Part B*, vol. 37, No. 5, pp. 1167-1175, 2007.

[5] E.M. Newton and P. Jonathon Phillips, “Meta-analysis of third-party evaluations of iris recognition”, NISTIR 7440, August 2007.

[6] A.W.K. Kong, D. Zhang and M. Kamel, “Palmprint identification using feature-level fusion”, *Pattern Recognition*, pp. 478-487, 2006.

[7] D. Zhang, W.K. Kong, J. You and M. Wong, “On-line palmprint identification”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 9, pp. 1041-1050, 2003.

[8] A.W.K. Kong and D. Zhang, “Competitive coding scheme for palmprint verification”, *in Proceedings of International Conference on Pattern Recognition*, 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 Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 279-284, 2005.

[10] L. Ma, T. Tan, Y. Wang, D. Zhang, “Efficient iris recognition by characterizing key local variations”, *IEEE Transactions on Image Processing*, vol. 13, no. 6, pp. 739-750, 2004.

[11] Z. Sun, T. Tan and Y. Wang, “Iris recognition based on non-local comparisons”, *Lecture Notes in*

Computer Science, Springer, vol. 3338, pp. 491-497, 2004.

[12] E. Krichen, M.A. Mellakh, S. Garcia-Salicetti, B. Dorizzi, "Iris identification using wavelet packets", *in Proceedings of International Conference on Pattern Recognition*, 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", *Lecture Notes in Computer Science,* Springer, vol. 2688, pp. 861-868, 2003.

[14] K. Bea, S. Noh and J. Kim, "Iris feature extraction using independent component analysis", *Lecture Notes in Computer Science*, 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 Proceedings of the Third 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 Proceeding 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 Proceedings* of *International Conference on Image Processing*, 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] F. Hao, J. Daugman and P. Zielinski, "A fast search algorithm for a large fuzzy database", *IEEE Transactions on Information Forensics and Security*, vol. 3, no. 2, pp. 203-212, 2008

[21] W.K. Kong and D. Zhang, "Detecting eyelash and reflection for accurate iris segmentation", *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 17, no. 6, pp. 1025-1034, 2003.

[22] R. Viveros, K. Balasubramanian and N. Balakrishnan, "Binomial and negative binomial analogues under correlated Bernoulli trials", *The American Statistician*, vol. 48, no. 3, pp. 243-247, 1994.

[23] R.W. Katz, "Comment on: Binomial and negative binomial analogues under correlated Bernoulli trials", *American Statistician*, vol. 49, no. 3, 1995.

[24] R.M. Bolle, S. Pankanti, J.H. Connell and N.K. Ratha, "Iris individuality: a partial iris model", *in Proceedings of the 17th International Conference on Pattern Recognition*, vol. 2, pp. 927-930, 2004.

[25] S. Lin, An Introduction of error-correcting codes, Prentice-Hall Inc., New Jersey, 1970.

[26] J. Daugman, Personal Communication

[27] C. Sanchez-Avila and R. Sanchez-Reillo, "Two different approaches for iris recognition using Gabor filters and multiscale zero-crossing representation", *Pattern Recognition*, vol. 38, pp. 231-240, 2005.

[28] A.W.K. Kong, Palmprint Identification Based on Generalization of IrisCode, PhD thesis, University of Waterloo, 2007 (http://uwspace.uwaterloo.ca/handle/10012/2708)

[29] J.R. Matey, O. Naroditsky, K. Hanna, R. Kolczynski, D.J. LoIacono, S. Mangru, M. Tinker, T.M. Zappia and W.Y. Zhao, "Iris on the move: acquisition of images for recognition in less constrained environments", *Proceedings of the IEEE*, vol. 94, no. 11, pp. 1936-1947, 2006.

[30] J.L. Wayman, "Degrees of freedom as related to biometric device performance" http://www.engr.sjsu.edu/biometrics/publications_degrees.html

[31] W.T Freeman and E.H Adelson, "The design and use of steerable filters", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 9, pp. 891-906, 1991.

[32] A. Ross and S. Shah, "Segmenting non-ideal irises using geodesic active contours", *in Proceedings of Biometrics Symposium*, pp. 1-6, 2006

[33] A.W.K. Kong, “An analysis of Gabor detection”, *in Proceeding of International Conference on Image Analysis and Recognition*, pp. 64-72, 2009.

Appendixes

A.

The appendix shows that the coding method in [9] is based on the operators “≥” and “<” and hamming distance.

Ma *et al* first apply a quadratic spline wavelet to decompose the one dimensional iris signals into several scales but only two scales are used in the their feature extraction process. They then record the locations, d, and types of all the local extremum points (minimum and maximum points) as features. To obtain stable feature points, they remove pairs of adjacent points having an amplitude difference smaller than a predetermined threshold. Since there are only two types of extremum points and since two adjacent points of a maximum point must be minimum points, just one pointer, p_i, will suffice to denote all the type information in scale i. The pointer, p_i, is set to 1 if the first extremum point is a minimum point; otherwise p_i is set to –1. Finally, for each decomposed signal S_i, the pointers and locations of feature points are stored as the following form

$$f_i = \{d_1, d_2, ...d_i, ...d_m; d_{m+1}, d_{m+2}, ..., d_{m+n}; p_1, p_2\}\,. \quad (25)$$

To exploit XOR operations for effective matching, as in IrisCode, the original features in each scale are transformed into a binary feature vector of a fixed length, L. Fig. 11 illustrates this process, which is called feature transform. If p_j is –1, the first d_1-1 components in the binary feature vector are set to 0; otherwise they are set to 1. Then, all the components in the binary feature vector corresponding to maximum and minimum points are set to 0 and 1, respectively. All the other components between d_i and d_{i+1} are set to 0 if d_i corresponds to a maximum point;

otherwise, they are set to 1.

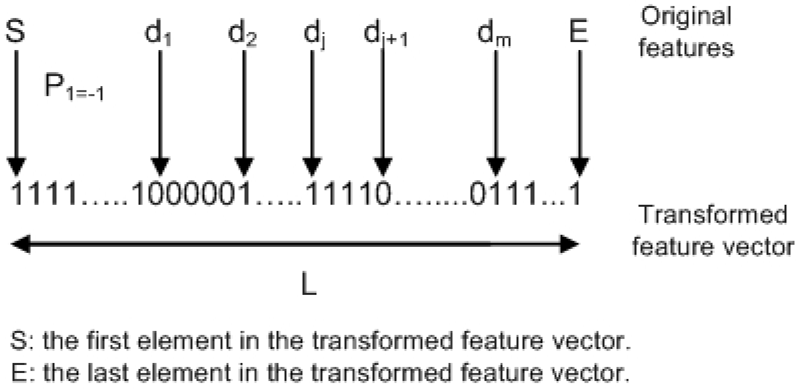

S
d1
d2
dj
dj+1
dm
E
Original features
P1=-1
1111.....1000001.....11110........0111...1
Transformed feature vector
L
S: the first element in the transformed feature vector.
E: the last element in the transformed feature vector.

Fig. 11 Illustration of Ma et al's feature transform

This coding scheme is more complicated than other coding schemes but is very similar to the standard coding scheme defined in Eqs. 22-23. To demonstrate the relationship between the coding scheme of Ma *et al* and other coding schemes, we apply a process to S_i, the wavelet transformed signal, to obtain a signal v_i for coding based on "≥" and "<". Each extremum point in v_i has a corresponding extremum point in S_i with the same type, location and amplitude but the amplitude difference between each pair of adjacent extremum points in v_i has to be larger than the predetermined threshold. A simple way to obtain v_i is to interpolate the extremum points in Eq. 25. Since the maximum and minimum points in v_i correspond to the zero-crossings in dv_i / dx, their coding scheme can be rewritten as follows.

If $dv_i / dx \,|\, x = x_j$ >0, then B_j=1;

otherwise B_j=0.

Therefore, the Ma *et al* coding scheme can be rewritten based on the operators "≥" and "<" and can be considered as standard coding scheme.

B.

Eq. 16 demonstrates that the locus of $\vec{Z}(\varphi)$ is an ellipse with respect to φ. In fact, the locus can be further constrained. Under suitable parameterization, it is a circle. Mathematically, $\lim_{k\to\infty}(\|M_R\|^2 - \|M_I\|^2) = 0$, where $k = \omega\beta$.

Considering $\|M_R\|^2 - \|M_I\|^2$

$$= \iint \left(e^{-\frac{\rho^2}{\alpha^2}} e^{-\frac{\phi^2}{\beta^2}} \cos(\omega\phi) \right)^2 d\rho d\phi - \iint \left(e^{-\frac{\rho^2}{\alpha^2}} e^{-\frac{\phi^2}{\beta^2}} \sin(\omega\phi) \right)^2 d\rho d\phi$$

$$= \iint e^{-\frac{2\rho^2}{\alpha^2}} e^{-\frac{2\phi^2}{\beta^2}} \left(\cos^2(\omega\phi) - \sin^2(\omega\phi)\right) d\rho d\phi$$

$$= \int e^{-\frac{2\rho^2}{\alpha^2}} d\rho \int e^{-\frac{2\phi^2}{\beta^2}} (\cos^2(\omega\phi) - \sin^2(\omega\phi)) d\phi$$

$$= \frac{\alpha\sqrt{2\pi}}{2} \int e^{-\frac{2\phi^2}{\beta^2}} (\cos^2(\omega\phi) - \sin^2(\omega\phi)) d\phi$$

Let $\gamma = \frac{k}{\beta}\phi$. Thus

$$= \frac{\alpha\beta\sqrt{2\pi}}{2k} \int e^{-\frac{2\gamma^2}{k^2}} (\cos^2(\gamma) - \sin^2(\gamma)) d\gamma$$

$$= \frac{\alpha\beta\sqrt{2\pi}}{2k} \int e^{-\frac{2\gamma^2}{k^2}} \cos(2\gamma) d\gamma$$

Let $2\gamma = \tau$

$$= \frac{\alpha\beta\sqrt{2\pi}}{4k} \int e^{-\frac{\tau^2}{2k^2}} \cos(\tau) d\tau$$

$$= \frac{\alpha\beta\sqrt{2\pi}}{4k} \sqrt{2\pi} k e^{-\frac{k^2}{2}}$$

$$= \frac{1}{2}\alpha\beta\pi e^{-\frac{k^2}{2}}$$

Since, α, β, and k are greater than zeros, $\|M_R\|^2 - \|M_I\|^2$ is always greater than zero.

However, $\lim_{k\to\infty}\left(\|M_R\|^2 - \|M_I\|^2\right) = \lim_{k\to\infty} \frac{1}{2}\alpha\beta\pi e^{-\frac{k^2}{2}} = 0$

C.

This appendix shows the equivalent relationship between the bitwise hamming distance and the phase distance in Eq. 17.

Let two winning indexes be *j-1*, and *j-1+k*, where $1 \le j \le j + k < 2n$. Their phase distance is $\min(k, 2n - k)$. Using the coding scheme given in Fig. 2, the winning indexes are represented by j^{th} and $j+k^{\text{th}}$ column vectors of matrix *A*. We would like to prove

$$\sum_{i=1}^{n} a_{i,j} \otimes a_{i,j+k} = \min(k, 2n - k).$$

Since all $a_{i,j}$ are either zero or one, $\sum_{i=1}^{n} a_{i,j} \otimes a_{i,j+h} = \sum_{i=1}^{n} \left| a_{i,j} - a_{i,j+k} \right|$

Case 1:

If $j \le n$ and $j + k \le n$

From the definition of *A*, we know $\sum_{i=1}^{n} \left| a_{i,j} - a_{i,j+k} \right| = k$

Case 2:

If $j > n$ and $j + k > n$

As in Case 1, we know $\sum_{i=1}^{n} \left| a_{i,j} - a_{i,j+k} \right| = k$

Case 3:

If $j \le n$ and $j + k > n$ and $k \le n$

Consider $a_{i,j}=1$ and $a_{i,j+k}=1$ (26)

From the definition of *A*, we have $1 \le i < j$ and $j + k - n \le i \le n$

Then, $j + k - n \le i < j$

The number of *i* satisfying condition (26) is $\max(0,\ j-(j+k-n))$. (27)

Since $k \le n$, $\max(0,\ j-(j+k-n)) = n-k$

Consider $a_{i,j}=0$ and $a_{i,j+k}=0$ (28)

From the definition of *A*, we have $i \ge j$ and $i < j+k-n$

Then $j \le i < j+k-n$

The number of *i* satisfying condition (28) is $\max(0,\ j+k-n-j)$ (29)

Since $k \le n$, $\max(0,\ k-n)$ =0

Thus, $\sum_{i=1}^{n} \left| a_{i,j} - a_{i,j+k} \right| = n-(n-k) = k$

Case 4:

If $j \le n$ and $j+k > n$ and $k > n$

Consider $a_{i,j}=1$ and $a_{i,j+k}=1$. (30)

From (27), number of *i* satisfying (30) is $\max(0,\ j-(j+k-n))$

Since $k > n$, $\max(0,\ n-k) = 0$

Consider $a_{i,j}=0$ and $a_{i,j+k}=0,$ (31)

From (29), number of *i* satisfying (31) is $\max(0,\ j+k-n-j)$

Since *k*>*n*, $\max(0,\ k-n) = k-n$

Thus, $\sum_{i=1}^{n} \left| a_{i,j} - a_{i,j+k} \right| = n-(k-n) = 2n-k$

Thus, $\sum_{i=1}^{n} a_{i,j} \otimes a_{i,j+k} = k$ for Cases, 1-3 and $\sum_{i=1}^{n} a_{i,j} \otimes a_{i,j+k} = 2n-k$ for Case 4. Since $2n-k \ge k$ for Cases 1-3 and $2n-k < k$ for Case 4, $\sum_{i=1}^{n} a_{i,j} \otimes a_{i,j+k} = \min(k,\ 2n-k)$.

D.

This appendix shows that when the locus of $\vec{Z}(\varphi)$ is a circle, $\arg\max_{\varphi}\int_{\rho}\int_{\phi}\rho IZ(\varphi)d\rho d\phi/\|\rho I\|\|Z(\varphi)\| = \varphi_1$ or φ_2 , where $\varphi_1 = \tan^{-1}\left(\int_{\rho}\int_{\phi}\rho IM_I d\rho d\phi \Big/ \int_{\rho}\int_{\phi}\rho IM_R d\rho d\phi\right)$, $\varphi_1 \in [0,\pi)$ and $\varphi_2 = \varphi_1 + \pi$.

Since the locus of $\vec{Z}(\varphi)$ is a circle, $\arg\max_{\varphi}\int_{\rho}\int_{\phi}\rho IZ(\varphi)d\rho d\phi/\|\rho I\|\|Z(\varphi)\|$ can be rewritten as $\arg\max_{\varphi}\int_{\rho}\int_{\phi}\rho IZ(\varphi)d\rho d\phi$. Using Eq. 18, $\int_{\rho}\int_{\phi}\rho IZ(\varphi)d\rho d\phi = \cos(\varphi)C_R + \sin(\varphi)C_I$, where $C_I = \int_{\rho}\int_{\phi}\rho IM_I d\rho d\phi$ and $C_R = \int_{\rho}\int_{\phi}\rho IM_R d\rho d\phi$. Considering $\frac{d\cos(\varphi)C_R + \sin(\varphi)C_I}{d\varphi} = -\sin(\varphi)C_R + \cos(\varphi)C_I$ and setting $-\sin(\varphi)C_R + \cos(\varphi)C_I = 0$, we have $\varphi = \tan^{-1}(C_I/C_R)$. Since $\cos(\varphi)C_R + \sin(\varphi)C_I$ is a continuous periodic function, one of the φ_i corresponds to a maximum and the other corresponds to a minimum

Using the second order derivative, we can demonstrate that the φ_i corresponding to the maximum satisfies the following inequalities. If $\cos(\varphi_i) \neq 0$, $-C_R/\cos(\varphi_i) < 0$. If $\cos(\varphi_i) = 0$, $-\sin(\varphi_i)C_I < 0$.