algorithms

MDPI

Article

From Iris Image to Embedded Code: System of Methods

Ivan Matveev [1] and Ilia Safonov [2,*]

[1] Federal Research Center "Computer Science and Control" of the Russian Academy of Sciences, Vavilov Street, 44/2, 119333 Moscow, Russia
[2] Computer Science and Control Systems Department, National Research Nuclear University MEPhI, Kashirskoye Highway, 31, 115409 Moscow, Russia
* Correspondence: ilia.safonov@gmail.com

Abstract: Passwords are ubiquitous in today's world, as are forgetting and stealing them. Biometric signs are harder to steal and impossible to forget. This paper presents a complete system of methods that takes a secret key and the iris image of the owner as input and generates a public key, suitable for storing insecurely. It is impossible to obtain source data (i.e., secret key or biometric traits) from the public key without the iris image of the owner, the irises of other persons will not help. At the same time, when the iris image of the same person is presented the secret key is restored. The system has been tested on several iris image databases from public sources. It allows storing 65 bits of the secret key, with zero possibility to unlock it with the impostor's iris and 10.4% probability to reject the owner in one attempt.

Keywords: biometric cryptosystem; iris identification; error-correcting codes

check for updates

Citation: Matveev, I.; Safonov, I. From Iris Image to Embedded Code: System of Methods. *Algorithms* **2023**, *16*, 87. https://doi.org/10.3390/a16020087

Academic Editors: Francesco Bergadano and Giorgio Giacinto

Received: 15 December 2022
Revised: 19 January 2023
Accepted: 3 February 2023
Published: 6 February 2023

1. Introduction

Nowadays, cryptographic algorithms are widely used for information protection. A large number of them, as well as their applications, have been invented and introduced [1]. These algorithms and systems are mathematically grounded and reliable. The weak link in their implementation and usage, as usual, is human. Cryptography requires keys, i.e., sequences of digits, which should be reproduced precisely. While a human is able to remember and reproduce a personally invented password (though there are difficulties here already), it is practically impossible to memorize a long sequence of pseudorandom symbols, which is created automatically [2]. Meanwhile, humans possess biometric features that are simple to retrieve, difficult to alienate, and contain a significant amount of information. The disadvantage of biometric traits is their variability: it is impossible to exactly replicate the measurement results, we can only say that two sets of traits taken from one person are in some sense closer than the sets obtained from different people. It is of great interest to combine these two approaches, i.e., to develop methods for obtaining unique cryptographic keys from variable biometry data of a given person.

The eye iris is the most suitable biometric modality among all non-invasive ones due to its highest information capacity. The number of degrees of freedom of the iris template was evaluated as 249 [3]. It promises to be almost as good as a strong symmetric cryptography key length of 256 bit, while the net coming fingerprint is reported to have 80 bits [4]. In order to design a practically usable system it is advisable to base it on the iris. Up to now a major focus in developing automated biometric is building an identification system, i.e., the system, which executes a scenario: sample biometric features once, record, take them sometime later and decide whether these samples belong to the same person.

The workflow of the biometric identification system can be combined of the blocks: capture, segmentation, template generation, and template matching, see Figure 1.

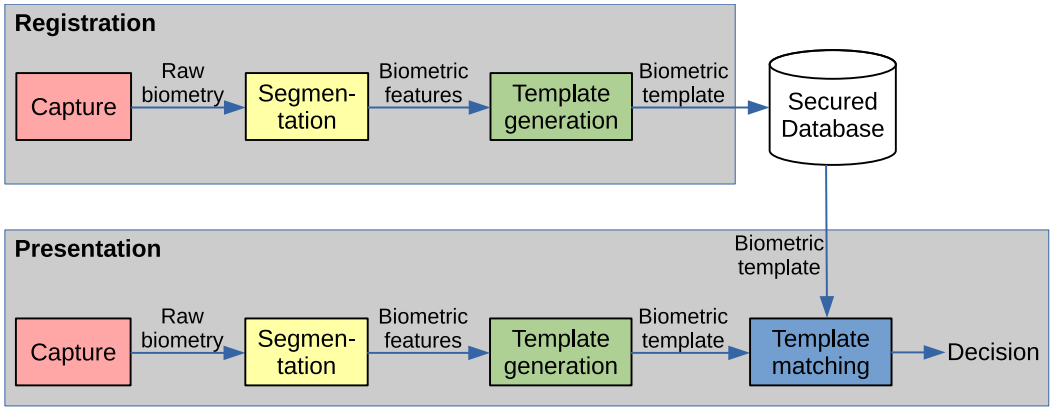

Figure 1. Biometric system workflow.

Note that in this scenario the biometric template should be securely stored and exclude the intruder from obtaining it. Here, a different problem is solved, thus only capture, segmentation and partly template generation blocks are inherited, and matching is replaced by embedding/extracting the cryptographic key into/from the biometric features.

The explanation here goes alongside the data processing: from the source iris image to the embedding of the secret key. The capture process, i.e., obtaining eye images with a camera device, is beyond the scope of this paper. We start from the image segmentation task and present a framework for locating the iris in an eye image. In the next section clue methods of the framework are described. Then feature extraction and matching methods are given. Following is the discussion of the application scenario of embedding the secret key to biometric features. The successful extraction of the embedded key depends on the difference between registered and presented biometric features, the value of this difference is determined based on several databases. In the next section the methods of encoding and decoding the key are presented, and the selection of their optimal parameters is discussed.

The contribution of this work is comprised of the following.

- The system of iris segmentation methods is presented which combines preliminary detection with refinement steps. The first steps use the most common geometric properties of the eye and accept the widest range of image conditions, while the final steps take care of details. The core of the system is a special base radii detection method.
- The cascade of error correction codecs adopted to iris code nature. A novel step of pseudorandom bit shuffling is introduced, accompanied by bit dubbing. This contradicts known methods, which do not use bit dubbing and deliberately avoid bit shuffling.
- The combination of the iris segmentation system and error correction cascade results in a practically applicable method, proven for several databases of variable image quality.

2. Eye Segmentation Framework

Methods, algorithms and applications of iris biometrics have attracted much attention during the last three decades [5–7] and continue developing rapidly in recent years [8]. The main trend of the latest development is switching from heuristic approaches and hand-crafted methods to employing neural networks in various tasks. A wide variety of artificial neural networks has emerged and is applied to iris segmentation, starting from earlier research with fully connected nets [9] to latest applications of attention-driven CNNs [10], U-Nets [11], hybrid deep learning [12]. Another trend comes from the in-born ability of neural networks to *classify* objects (say, pixels, textures, images) rather than *calculate* their positions and other numerical properties. Due to this, most of the works in iris segmentation rely on detecting masks, i.e., pixels belonging to regions of the iris (or pupil, or else) in the image. Positions and sizes of pupil and iris are then derived from these masks. Surely, detecting masks is what one calls segmentation; however, such an approach ignores the clear and simple geometry of the iris and is prone to detecting irises of

unnatural shape as is shown in [12]. Some works [13,14] apply a neural network to detect the position of the iris as a number; however, it seems a strained decision.

Here we adopt a "classical" method. The obvious approach to iris region extraction in eye imaging is a chain of operations that starts with the detection of the pupil (the most distinctive area that is dark and has an expressed circular shape). Then outer iris border is presumably located. Finally, the visible iris part is refined by cutting off the areas distorted by reflections, eyelids and eyelashes. Most researchers and developers follow this method. Detection of each iris feature is usually carried out only one time and it is not recalculated any more even after obtaining other features, which can serve for refinement. For instance, in [15–18] full set of iris features is detected; however, pupil parameters are obtained at the first step and are not revised any more.

Only a few papers develop something different from this sequence "first pupil, then iris, once determined, and never reviewed". In [19,20], the position of the iris center is estimated first which makes pupil detection more robust. In [21], pupil parameters are refined using iris size after the iris is located. In [21,22], detection methods run iteratively several times for refinement. In [20,23], a multi-scale approach is used, and methods run in several scales. However, none of these works use various types of methods for detecting any iris feature.

Here we develop a *system of methods* for segmentation of the iris in an eye image. Evaluating of each of parameters is performed at several steps. The main idea of this system is that at first the most general characteristics of objects are defined, which are then sequentially supplemented by more specific and refined ones. Beginning steps do not need to output precise final parameters, used as final. Instead, they should be robust and general and tolerate a wide range of conditions, i.e., detect the object of any quality. Later steps should have the highest possible precision and may reject poor quality data.

Iris region in frontal images is delimited by two nearly concentric nearly circular contours, called inner and outer borders. Hereinafter the contour separating iris and pupil is referred to as *inner border*, *pupil border* or simply *pupil*, and the contour delimiting iris and sclera is called *outer border* or *iris*. In most cases pupil border is wholly visible in the image, but some part of the iris border is frequently overlapped by eyelids and eyelashes.

Since the pupil and iris are almost concentric, one *eye center* point may serve as an approximate center for both contours. It can be considered the most general geometric property of the iris, and the first step of eye detection should be locating this eye center. Note that only the position of the center is to be found, rather than the size of any contour. Excluding size and allowing approximate detection involves both concentric borders in the process. This is especially significant for eyes with poorly visible inner boundaries, where pupil location alone fails frequently. A modification of Hough method [24] is used.

It is very likely that after center detection pupil size should be estimated. To the best of our knowledge, this is carried out in all works where iris segmentation starts from eye center location, as in [19]. However, this method is not stable and universal for a wide range of imaging conditions. Detecting the radius may easily mistake the outer border for the inner, especially for images with poor inner border contrast [25]. Here we decide to employ both correlated contours around the detected center, and detect sizes of them simultaneously. Hereinafter this detection is referred to as *base radii* detection, meaning that it finds approximate (base) radii of inner and outer circles around a given center. The method relies on circular projections of gradient [26]. Base radii detection produces approximate center coordinates and radii of pupil and iris circles, which satisfy some reasonable limitations. Furthermore, the quality of detection is calculated. The quality should be high enough to pass the image to further processing.

Then both boundaries are re-estimated with better precision (refined). Pupil refinement is carried out by a specially developed version of the shortest path method [27]. Iris is refined by the same method as that of base radius. The difference is that the position of the pupil is now fixed and only the iris center and radius are being searched. Iris segmentation here results in detecting two nearly concentric circles, which are approximating the inner

and outer borders of the iris ring. Occlusion detection [28] is carried out to ensure the quality of iris data, i.e., to reject strongly occluded irises from further processing, but apart from this the occlusion mask is not used.

Summing up, the segmentation stage of the system employs five steps: center detection, base radii detection, pupil refinement, iris refinement and occlusion detection, see Figure 2.

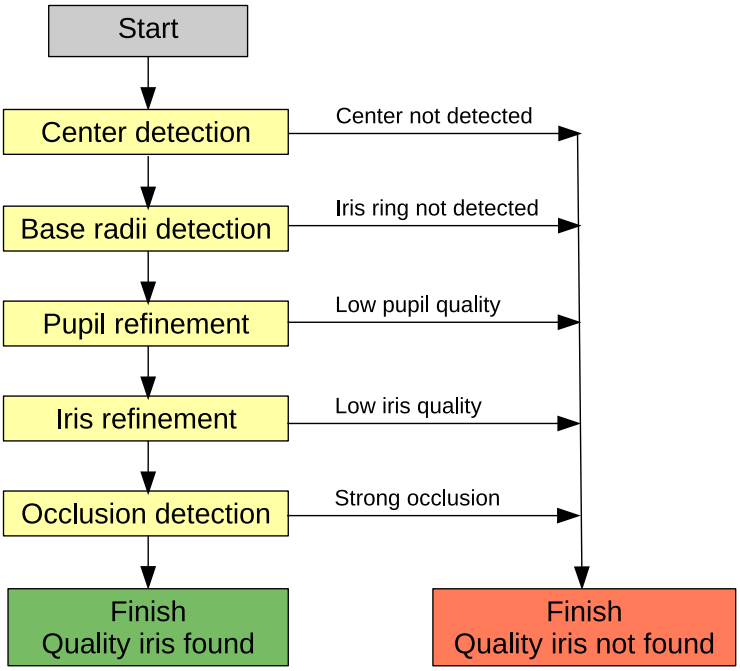

Figure 2. Workflow of iris segmentation methods.

At each stage of segmentation, quality value is estimated and the process is terminated if the quality is below acceptable.

3. Eye Segmentation Methods

Methods of iris segmentation are briefly presented in this section.

3.1. Center Detection

The algorithm finds the coordinates $(x_C, y_C) = \vec{c}$ of eye center in the image $b(\vec{x})$, and does not need to estimate pupil or iris size. There is also no need to find the center position precisely, it is sufficient to locate it somewhere inside the pupil. Thus, pixels of both pupil and iris borders are used in Hough's procedure. Furthermore, the algorithm has low computational complexity since only two parameters are estimated and a two-dimensional Hough accumulator is used.

The algorithm runs the following five steps.

Step 1. Gradient calculation.

Consider rectilinear coordinate system Oxy in the image with the center in the left bottom corner and axes Ox and Oy directed along its borders. Denote brightness $b(\vec{x})$ in image point $\vec{x}$. Brightness gradient $\vec{\nabla} b(\vec{x}) = \vec{g}(\vec{x})$ is estimated by standard Sobel masks [29].

Step 2. Outlining voting pixels.

We need edge pixels to vote. These are selected with the help of a gradient value threshold. Cumulative distribution of brightness gradient values in pixels over the image is

calculated, and set Ω_1 of pixels with brightness gradient in the upper 5% of this distribution are selected:

$$\begin{aligned} H(G) &= |\{\vec{z} : \|\vec{g}(\vec{z})\| \leqslant G\}| \,, \\ \Omega_1 &= \{\vec{x} : H(\|\vec{g}(\vec{x})\|) \geqslant (1-\tau_s)N\} \,, \end{aligned} \tag{1}$$

where $|S|$ is power (count of elements) of set S, N is total number of image pixels, $\tau_s = 0.05$ is the share of points being selected.

Step 3. Voting to accumulator.

Hough methods use *accumulator* function, which is defined over a parameter space. We detect the eye center, which is some point in the image, and its parameters are its coordinates in the image. Thus, the parameter space is 2D vector $\vec{x}$ and the accumulator is $A(\vec{x})$ with the same size as the source image.

Ray from some given point $\vec{x} \in \Omega_1$ in anti-gradient direction $-\vec{\nabla} b(\vec{x})$ is the locus of all possible dark circles with border passing through this point. A set of such rays, drawn in the accumulator, traced from each pixel coordinated selected at step 2 will produce clotting at the center of any roundish dark object. The more circle-like this object is, the more expressed will be its central clotting.

Step 4. Accumulator blurring.

The accumulator $A(\vec{x})$ is subject to a low-pass filter, to suppressed noise such as singular sporadic rays produced by non-circular edges in the image. Denote the result as $A_B(\vec{x})$.

Step 5. Maximum location.

Maximum position

$$\vec{c} = arg \max_{\vec{x}} A_B(\vec{x}) \tag{2}$$

in blurred accumulator corresponds to the center of the best round-shaped object in the image. It is the most probable eye center. However, local maxima exist in any image due to noise. In order to decide whether there is a noticeable circular object, one can compare the value of local maxima against the values produced by noise. Since τ_s pixels of the image are voting and for each point voting procedure draws a segment of approximately $0.5W$ pixels, where W is a linear size of the image, the average brightness level is near $0.5\tau_s W$. Selecting desirable signal to noise ratio P_{SNR}, one can write the condition of accepting the located maximum (2) for eye center:

$$Q_C = \max_{\vec{x}} A_B(\vec{x}) > \frac{1}{2} P_{SNR} \tau_s W \,. \tag{3}$$

If condition (3) does not hold, the decision is made that there is no eye in the image $b(\vec{x})$.

3.2. Base Radii Detection

The algorithm simultaneously locates two iris boundaries as circle approximations: inner (pupil) (x_P, y_P, r_P) and outer (iris) (x_I, y_I, r_I) starting from the center $\vec{c}$ (2). In this section, we set $(x_C, y_C) = \vec{c}$ as coordinate origin. Anti-gradient vector at the boundary of the dark circle and the direction to the circle center coincide or form a small angle. As the pupil and iris are both dark circles on the brighter background, one can state the following condition for pixels $\vec{x}$ of their boundaries:

$$\phi(\vec{x}) = \arccos \frac{\vec{g}(\vec{x}) \cdot \vec{x}}{\|\vec{g}(\vec{x})\| \, \|\vec{x}\|} < \tau_\phi \quad . \tag{4}$$

We use a threshold value $\tau_\phi = 45^0$.

Furthermore, the condition for gradient value (1) is applicable. Pixel $\vec{x}$ satisfying the conditions (1), (4) probably belongs to the inner or outer boundary. Call it *candidate*. Define the set of candidate pixels as Ω_2:

$$\Omega_2 = \left\{\vec{x} : \phi(\vec{x}) < \tau_\phi, H(\|\vec{g}(\vec{x})\|) \geqslant (1-\tau_s)N\right\} . \tag{5}$$

For each radius r a ratio of candidate count at this radius to the count of all pixels at this radius is estimated:

$$\Pi(r) = \frac{|\{\vec{x} : \|\vec{x}\| \in [r-0.5, r+0.5), \vec{x} \in \Omega_2\}|}{|\{\vec{x} : \|\vec{x}\| \in [r-0.5, r+0.5)\}|} . \tag{6}$$

If there is a dark circle of some radius ρ with the center near the coordinate origin its border pixels are likely to belong to the set Ω_2, and are likely to have distance ρ to the coordinate origin. Thus, $\Pi(\rho)$ will be big, i.e., have local maximum. Other contours will not vote to the same radius of circular projection and will not form local maxima therein.

The image plane is divided into four quadrants, left, right, top and bottom by the lines $y = x$ and $y = -x$. In each quadrant, a *sub-projection* is calculated separately according to (6). Positions of local maxima on the right, left, top, and bottom sub-projections are:

$$\mu_\alpha(n) = \arg\operatorname*{loc\,max}_n \Pi_\alpha(r) , \quad \alpha = \{R, L, T, B\} . \tag{7}$$

The quality of maxima is simply the value of histogram at the point

$$q_\alpha(n) = \Pi_\alpha(\mu_\alpha(n)) . \tag{8}$$

If not occluded, each of the two circular contours (inner and outer borders) gives a local maximum in each sub-projection. Other maxima may arise due to occlusions such as eyelashes and eyelids or due to other details in eye images, including patterns of the iris itself. Combining local maxima positions (7) gives set of hypothetical pupils:

$$\begin{aligned} x_P^{i,j} &= \frac{1}{2}(\mu_R(i) - \mu_L(j)) , \quad i = \overline{1, n_R} , \quad j = \overline{1, n_L} , \\ y_P^{k,l} &= \frac{1}{2}(\mu_T(k) - \mu_B(l)) , \quad k = \overline{1, n_T} , \quad l = \overline{1, n_B} , \\ r_P^{i,j,k,l} &= \frac{1}{4}(\mu_R(i) + \mu_L(j) + \mu_T(k) + \mu_B(l)) . \end{aligned} \tag{9}$$

Qualities of combinations are also defined from values (8):

$$q_P^{i,j,k,l} = \frac{1}{4}(q_R(i) + q_L(j) + q_T(k) + q_B(l)) . \tag{10}$$

Irises are estimated by just the same formulas:

$$\begin{aligned} x_I^{i,j} &= \frac{1}{2}(\mu_R(i) - \mu_L(j)) , \quad i = \overline{1, n_R} , \quad j = \overline{1, n_L} , \\ y_I^{k,l} &= \frac{1}{2}(\mu_T(k) - \mu_B(l)) , \quad k = \overline{1, n_T} , \quad l = \overline{1, n_B} , \\ r_I^{i,j,k,l} &= \frac{1}{4}(\mu_R(i) + \mu_L(j) + \mu_T(k) + \mu_B(l)) , \\ q_I^{i,j,k,l} &= \frac{1}{4}(q_R(i) + q_L(j) + q_T(k) + q_B(l)) . \end{aligned} \tag{11}$$

The nature of the pupil and iris imposes certain limitations on their locations and sizes. We use the following four inequalities: pupil size is not less than 15% of iris size and not

more than 75% of iris size; center of the iris is inside pupil; pupil cannot be displaced too much from iris center. This can be written as:

$$r_P > 0.15 r_I \,, \;\; r_P < 0.75 r_I \,, \;\; d < r_P \,, \;\; 2(r_I - r_P - d) > r_I - r_P + d \,, \tag{12}$$

where $\vec{c}_P = (x_P, y_P)$, $\vec{c}_I = (x_I, y_I)$ are centres of pupil and iris, $d = \|\vec{c}_P - \vec{c}_I\|$ is a distance between these centres.

From all possible variants of pupil-iris pair given by (9)–(11) we select those satisfying conditions (12). The quality of combination is a sum of pupil and iris qualities (10) and a weighted quality of fitting to conditions (12):

$$\begin{aligned} Q &= q_P + q_I + \gamma q_{fit} \,, \\ q_{fit} &= \min\left\{ \frac{r_P - 0.15 r_I}{r_P}, \frac{0.75 r_I - r_P}{r_P}, \frac{r_P - d}{r_P}, \frac{r_I - r_P - 3d}{r_I - r_P} \right\} . \end{aligned} \tag{13}$$

The combination with the best quality is selected. If Q is below the given threshold, it is supposed that the eye in the image is squinted and upper and lower eyelids cover a big share of the iris border. In this case, the variant with absent top and bottom iris local maxima is tested. The formulas (9) and (10) are modified accordingly, iris center vertical position is taken equal to that of the pupil: $y_I \equiv y_P$. If Q is below the threshold again, it is decided that there is no feasible iris ring and in the image. Other types of occlusion are not treated, the iris images are considered too bad for processing in this case. Thresholds for Q and value of γ in (13) are estimated experimentally so as to reject the biggest share of erroneously detected irises while preserving good outcomes. So, the method runs in six steps:

Step 1. Gradient calculation.

This step is common with center detection.

Step 2. Candidates selection.

This step is similar to Step 2 of center detection. In addition to gradient value condition (1) angular condition (4) is imposed.

Step 3. Circular projecting.

Calculating circular projections (6) in four quadrants.

Step 4. Enumeration of maxima.

Finding local maxima (7) in projections. Prior to this the projections are smoothed with a Gaussian filter to suppress redundant local maxima originating from noise.

Step 5. Enumerations of hypothetical irises.

Finding coordinates and radii of inner and outer circles from combinations of maxima (9), which hold conditions (12).

Step 6. Selecting the best iris.

Pair of circles is selected according to the qualities (8), (10), (13).

If no feasible iris is detected in step 5, the result is "no eye detected".

A sample of the projection combination is presented in Figure 3. Real positions of pupil and iris borders, taken from expert marking are depicted by arrows. There is no local maxima corresponding to the iris border in the top projection $\Pi_T(r)$ since the iris border is occluded. Such minor obstacles do not prevent choosing correct combination.

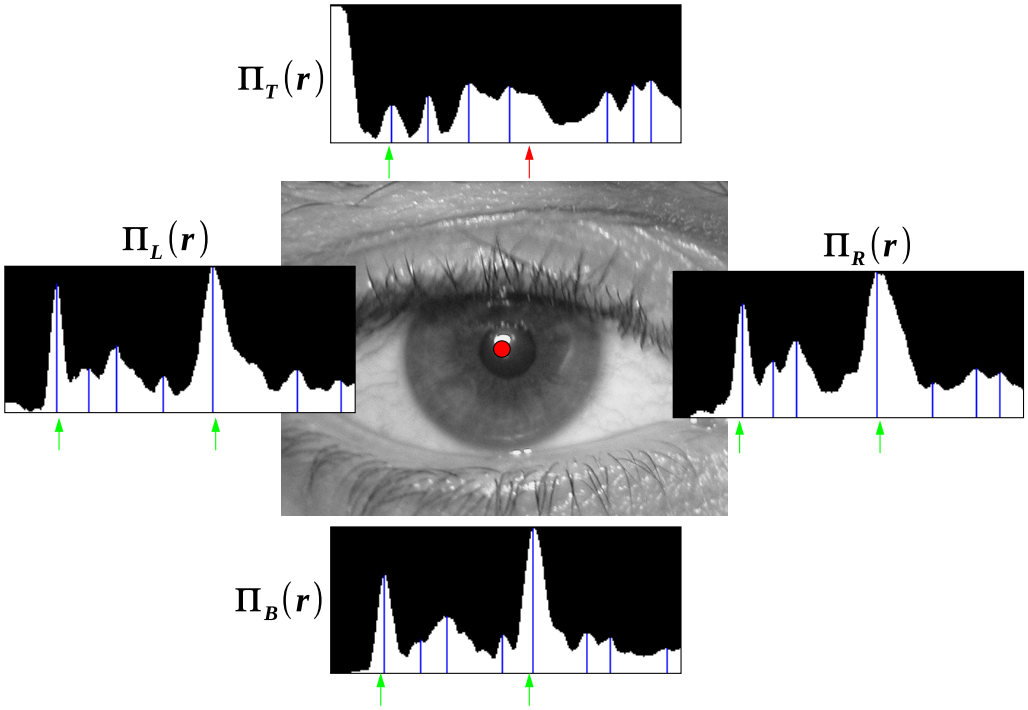

Figure 3. Four circular projections, their maxima positions and correct position of borders.

3.3. Pupil Refinement

Circular shortest path method constructs a closed contour in a circular ring [30]. The ring is centered at a given point and has inner and outer radii. CSP method is a kind of optimal path method, i.e., it optimized the functional, which is the cost of the path. We take the ring concentric to the approximate pupil circle and spread 30% of its radius inside and outside.

In order to ease calculations polar transformation is carried out. The ring shape in the source image is unwrapped to a rectilinear raster. Radial and angular coordinates of the ring are mapped to abscissa and ordinate. Thus, the problem of locating the circular shortest path is reduced to a problem of detecting the optimal path from the left to the right side of the rectangle such that terminal points of the path have the same ordinate. Contour is represented as a function $\rho(\phi)$, $\phi \in [0; 2\pi]$, $\rho(0) = \rho(2\pi)$ with limited derivative $d\rho/d\phi < 1$. In a discrete rectilinear raster of size $W \times H$ the contour is turns to a sequence of points: $\{(n, \rho_n)\}$, $n \in [0; W-1]$. Limitations to the derivative transforms to $|\rho_{n+1} - \rho_n| \leqslant 1$, edge condition is set as $|\rho_{W-1} - \rho_0| \leqslant 1$.

Consider points (n, ρ') and $(n+1, \rho'')$ from adjacent columns of the raster. Denote the cost of passing between them as

$$C((n, \rho'), (n+1, \rho'')) \equiv C_n(\rho', \rho'') = C_n^{(I)}(\rho', \rho'') + C_n^{(O)}(\rho', \rho'') \; . \tag{14}$$

This cost is a sum of inner and outer parts.

Inner cost is a function of contour shape, designed in a way to promote its smoothness:

$$C_n^{(I)}(\rho', \rho'') = \begin{cases} 0 \; , \; \rho' = \rho'' \; , \\ \tau_i \; , \; |\rho' - \rho''| = 1 \; , \\ \infty \; , \; |\rho' - \rho''| > 1 \; . \end{cases} \tag{15}$$

Value of $\tau_i > 0$ is a parameter defining the magnitude of a "force", which pulls the contour towards a straight line. Optimizing the inner part alone would give horizontal lines in polar raster, i.e., ideal circles with the given center in source image.

Outer cost is designed to make the contour pass through border pixels. So it is low in boundary points (the gradient vector is big and perpendicular to the local direction of the contour) and is high otherwise. The outer part is the cost of passing the point (n, ρ'):

$$C^{(O)}(n,\rho) = \begin{cases} 0\ , & (x,y) \in \Omega_3\ , \\ \tau_o;\ , & (x,y) \notin \Omega_3\ , \end{cases} \tag{16}$$

where Ω_3 is the set of points defined by (5), x and y are the coordinates of the source image point, which was mapped to (n, ρ).

Optimal contour $S = \{\rho_n\}_{n=1}^{W}$ is the one minimizing the total cost:

$$S^* = arg \min_S \sum_{n=1}^{W} C_n(\rho_n, \rho_{n+1})\ . \tag{17}$$

This discrete optimization problem can be solved by various methods. Here the method works in quite a narrow ring and the exhaustive search is faster due to small overhead.

Denote sum in (17) as Σ. In the best case $\Sigma = 0$, in the worst case $\Sigma = W(\tau_i + \tau_o)$. Mapping this into the range $[0; 1]$ where value 1 stands for best we obtain the quality

$$Q_{ref} = 1 - \frac{\Sigma}{W(\tau_i + \tau_o)}\ . \tag{18}$$

The contour is considered acceptable if $Q_{ref} > 0.5$, otherwise the decision is made that the pupil border cannot be detected with the required precision and the segmentation is terminated.

The algorithm runs in five steps.

Step 1. Candidates selection.

The same gradient calculation as in the first step of previous methods is used. Then the conditions (1), (4) are imposed as in Step 2 of base radii detection. However, a smaller angular threshold $\tau_\phi = 30^0$ is set since the center position is known with better precision.

Step 2. Polar transform.

The transform creates an image (rectangular raster) $P(\phi, \rho)$, $\phi \in [0, W-1]$, $\rho \in [0; H-1]$ by calculating a brightness value in each of its pixels (ϕ, ρ). This brightness is taken from source image $b(x, y)$ where its coordinates are estimated as

$$\begin{aligned} x(\phi,\rho) &= \left(r_1 + \frac{r_2 - r_1}{H}\rho\right)\cos\left(\frac{2\pi\phi}{W}\right)\ , \\ y(\phi,\rho) &= \left(r_1 + \frac{r_2 - r_1}{H}\rho\right)\sin\left(\frac{2\pi\phi}{W}\right)\ , \end{aligned} \tag{19}$$

where r_1 and r_2 are the inner and outer radii of the ring in the source image, and the coordinate origin of the source image is placed at the center of the ring. The brightness of the point of the polar image is obtained by bilinear interpolation:

$$\begin{aligned} N(\rho,\phi) = \\ (1-\{x\})(1-\{y\})b(\lfloor x\rfloor, \lfloor y\rfloor) + \\ \{x\}(1-\{y\})b(\lfloor x\rfloor + 1, \lfloor y\rfloor) + \\ (1-\{x\})\{y\}b(\lfloor x\rfloor, \lfloor y\rfloor + 1) + \\ \{x\}\{y\}b(\lfloor x\rfloor + 1, \lfloor y\rfloor + 1)\ , \end{aligned} \tag{20}$$

where $\lfloor a\rfloor$ and $\{a\}$ define integer and fractional parts of a.

Step 3. Optimal path tracking.

Finding S^* according to (14)–(17).

Step 4. Transforming to original coordinates.

Restore the coordinates of the optimal path from $O\rho\phi$ polar system back to the source image Oxy system.

Step 5. Estimating equivalent circle.

Pupil border contour is not a circle precisely; however, we can define an *equivalent circle*, with area and center of mass same as those of the figure enclosed into the pupil border contour. The center and radius of the equivalent circle are:

$$\begin{aligned} &x_{eq} = \frac{M_x}{M}\,,\ \ y_{eq} = \frac{M_y}{M}\,,\ \ r_{eq} = \sqrt{\frac{M}{\pi}}\,, \\ &M = |\Omega_4|\,,\ \ M_x = \sum_{(x,y)\in\Omega_4} x\,,\ \ M_y = \sum_{(x,y)\in\Omega_4} y\,, \end{aligned} \tag{21}$$

where Ω_4 is the area inside contour S^* in source image. This equivalent circle is further used as the pupil border, and it happens to be a better model due to its stability [31].

4. Experiments with Iris Segmentation

Iris segmentation here results in detecting two nearly concentric circles, which are approximating inner and outer borders of iris ring.

Assessment of the iris segmentation quality can be carried out in the following ways:

- Matching against manual segmentation by a human.
- Matching against rivals disclosed in the literature.
- Applying obtained segmentation further to iris recognition. Under the assumption that more precise detection reduces the number of classification errors this will indirectly estimate segmentation quality.

In order to compare the results of the proposed system with the known analogs, the following publicly available iris image databases were used: CASA-3-Lamp and CASIA-4-Thousand [32] (totally 54,434 images), BATH [33] (31,988 images), NDIRIS [34] (64,980 images), UBIRIS-1 [35] (1207 images).

4.1. Matching against Manual Segmentation

All images were processed by a human expert, who marked two circles approximating iris borders in each of them or rejected if the iris was not visible or of poor quality. (In fact, there were very few, less than a hundred altogether, images rejected at this stage.) We assume that the expert did it accurately; therefore this segmentation is taking for ground truth. Denote the manually marked circles as $(x, y, r)_P^*$ for pupil and $(x, y, r)_I^*$ for the iris. Values of absolute and relative errors of eye center detection averaged in databases

$$\varepsilon_{C,abs} = \langle \Delta \rangle\,,\ \ \varepsilon_{C,rel} = \left\langle \frac{\Delta}{r_P^*} \right\rangle\,,\ \ \Delta = \left((x_C - x_P^*)^2 + (y_C - y_P^*)^2 \right)^{1/2} \tag{22}$$

are given in Table 1.

It can be seen that for all databases except for the small bases MMU and UBIRIS, which contain low-resolution images, and UBIRIS, which contain images with small pupil size, the mean absolute deviation of the detected eye center from the true center of the pupil does not exceed four pixels and the relative deviation does not exceed one-tenth of the radius.

Table 1. Errors of eye center detection.

Database	$\varepsilon_{C,abs}$, Pixels	$\varepsilon_{C,rel}$, %
BATH	2.85	5.27
CASIA	2.94	7.47
MMU	3.22	15.5
NDIRIS	3.12	6.30
UBIRIS	8.29	22.3

The next method of the system is base radius detection. Table 2 presents the average deviations of the detected centers and radii of the pupil and the iris from those marked by human experts.

$$\begin{aligned} \varepsilon_{P,abs} &= \left\langle \left((x_P - x_P^*)^2 + (y_P - y_P^*)^2 \right)^{1/2} \right\rangle , \ \varepsilon_{rP,abs} = \langle r_P - r_P^* \rangle , \\ \varepsilon_{I,abs} &= \left\langle \left((x_I - x_I^*)^2 + (y_I - y_I^*)^2 \right)^{1/2} \right\rangle , \ \varepsilon_{rI,abs} = \langle r_I - r_I^* \rangle , \end{aligned} \tag{23}$$

Table 2. Errors of base radii detection, pixels.

Database	$\varepsilon_{P,abs}$	$\varepsilon_{rP,abs}$	$\varepsilon_{I,abs}$	$\varepsilon_{rI,abs}$
BATH	2.15	1.66	6.23	2.05
CASIA	2.58	1.60	17.41	4.21
MMU	3.31	2.05	13.01	5.11
NDIRIS	2.33	2.73	5.34	2.34
UBIRIS	6.03	5.78	7.68	11.35

It is seen that the mean error in detecting the pupil center is reduced compared with the first column of Table 1.

Table 3 shows the errors for the final circles of the pupil and the iris obtained by the system, calculated according to (23).

Table 3. Errors of final iris parameters detection, pixels.

Database	$\varepsilon_{P,abs}$	$\varepsilon_{rP,abs}$	$\varepsilon_{I,abs}$	$\varepsilon_{rI,abs}$
BATH	0.52	1.42	2.48	1.71
CASIA	1.05	1.13	2.44	1.62
MMU	0.97	1.77	1.92	4.35
NDIRIS	0.84	1.14	1.97	2.26
UBIRIS	2.27	5.37	3.25	5.82

4.2. Matching against Other Methods

Table 4 compares the computation time and errors in determining the pupil parameters for the presented system and its analogs. The comparison was carried out with the methods described in [3,36–39].

The third method to assess the algorithm of iris segmentation, i.e., applying its results to iris recognition is disclosed further.

Table 4. Matching against other methods.

Database	Error, Pixels	Method					
		Wildes	**Daugman**	**Masek**	**Ma et al.**	**Daugman-2**	**Presented**
BATH	$\varepsilon_{P,abs}$	3.44	3.73	5.32	4.29	3.27	0.52
	$\varepsilon_{rP,abs}$	4.38	4.54	6.72	4.65	3.19	1.42
CASIA	$\varepsilon_{P,abs}$	5.37	2.15	3.67	4.79	1.19	2.44
	$\varepsilon_{rP,abs}$	6.12	4.39	5.15	5.39	3.02	1.62
MMU	$\varepsilon_{P,abs}$	3.15	2.61	4.98	3.92	1.14	0.97
	$\varepsilon_{rP,abs}$	3.96	4.18	5.78	4.67	3.76	1.77
NDIRIS	$\varepsilon_{P,abs}$	6.37	2.13	5.59	5.92	1.79	0.83
	$\varepsilon_{rP,abs}$	7.51	3.53	7.23	7.38	3.11	1.14

5. Feature Extraction and Matching

We use the standard approach [3] here, which first transforms the iris ring to a so-called *normalized* image. This image is a rectangular raster, it is obtained from the iris ring by the polar transformation, analogous to (19), (20), where r_1 and r_2 are set to the radius of pupil and iris, respectively. In fact, more elaborate version of (19) is used:

$$\begin{aligned} x(\phi,\rho) &= \left(1-\frac{\rho}{H}\right)x_1\left(\frac{2\pi\phi}{W}\right)+\frac{\rho}{H}x_2\left(\frac{2\pi\phi}{W}\right), \\ y(\phi,\rho) &= \left(1-\frac{\rho}{H}\right)y_1\left(\frac{2\pi\phi}{W}\right)+\frac{\rho}{H}y_2\left(\frac{2\pi\phi}{W}\right), \\ x_1(\phi) &= x_P + r_P\cos(\phi)\,, y_1(\phi) = y_P + r_P\sin(\phi)\,, \\ x_2(\phi) &= x_I + r_I\cos(\phi)\,, y_2(\phi) = y_I + r_I\sin(\phi)\,, \end{aligned} \tag{24}$$

where x_P, y_P, r_P are the position and radius of pupil and x_I, y_I, r_I are the position and radius of iris. In comparison to (19) this variant accounts for the difference of pupil and iris centres.

The key idea of standard iris feature extraction is to convolve the normalized iris image with a filter, calculating the most informative features of the texture. Earlier Gabor wavelet was used for feature extraction. In one-dimensional space, it is represented as

$$g_{\sigma\lambda}(x) = \exp\left(-\frac{x^2}{2\sigma^2}\right)\exp\left(-i\frac{x}{\lambda}\right), \quad G_{\sigma\lambda}(u) = \exp\left(-\frac{(u-\lambda^{-1})^2\sigma^2}{2}\right), \tag{25}$$

where σ defines the width of the wavelet in the spatial domain, λ is the wavelength of modulation of the Gaussian by a harmonic function. By introducing inverse values $S = 1/\sigma$ and $W = 1/\lambda$, a simplified representation in the frequency domain can be obtained:

$$G_{SW}(u) = \exp\left(-\frac{(u-W)^2}{2S^2}\right). \tag{26}$$

It turned out that the modification of the Gabor wavelet called Log-Gabor function is better for feature extraction. Log-Gabor is given in the frequency domain as:

$$G_{SW}(u) = \exp\left(\frac{-\log^2(u/W)}{2\log^2(S/W)}\right) = \exp\left(-\frac{(\log u - \log W)^2}{2\log^2 L}\right). \tag{27}$$

This is equivalent to (26), in which each variable is replaced by its logarithmic counterpart. $L = S/W = \lambda/\sigma$ represents the ratio of the modulation wavelength to the width of the Gaussian. Research has shown that Log-Gabor wavelets are most likely optimal for the template generation problem. Therefore, we use this type of filter. The parameter λ is essentially the characteristic size of the objects in the image extracted by this filter, and L

is the number of periods of the harmonic function in Equation (25) which have sufficient amplitude and influence to the result. Optimal values of λ and L are selected according to [40].

Iris features $V(\phi,\rho)$ are calculated by convolution of the normalized image (20) with a Gabor or Log-Gabor filter, the transformation is performed in the spectral domain:

$$\begin{aligned} V(\phi,\rho) =& N(\phi,\rho) * g_{\sigma\lambda}(\phi) = \\ =& \mathcal{F}^{-1}\{\mathcal{F}\{N(\phi,\rho)\}\mathcal{F}\{g_{\sigma\lambda}(\phi)\}\} = \\ =& \mathcal{F}^{-1}\{\mathcal{F}\{N(\phi,\rho)\}G_{\lambda L}(u)\} \,. \end{aligned} \tag{28}$$

where σ and λ define the width of the wavelet along the angular axis and the modulation frequency, s is the width along the radial axis, $\mathcal{F}$ is the Fourier transform. The features used to form the patterns are computed as binary values of real and imaginary parts of the array $V(\phi,\rho)$:

$$\begin{aligned} T_{Re}(\phi,\rho) &= \mathcal{H}[\Re(V(\phi,\rho))] \,, \\ T_{Im}(\phi,\rho) &= \mathcal{H}[\Im(V(\phi,\rho))] \,, \end{aligned} \tag{29}$$

where $\mathcal{H}[\cdot]$ is the Heavyside function. So, eye image $b(x,y)$ produces a template $T(\phi,\rho)$, and each element of the template contains two bits.

We use features raster of 13 pixels in radial direction r and 256 pixels in tangential direction ϕ. Since each pixel produces two bits in (29) the total size of the template is 6656 bit [40].

Although here we do not build a classification system, which calculates a distance between templates and compares it against a classification threshold, template matching is implicitly present, as it will be shown below. Thus, we need to describe the matching method.

In a standard iris recognition approach templates T_1 and T_2 are matched by normalized Hamming distance:

$$\rho_0(T_1,T_2) = \frac{1}{|\Omega|}|\{T_1(\phi,\rho) \neq T_2(\phi,\rho), (\phi,\rho) \in \Omega\}| \,, \tag{30}$$

where $\Omega = M_1 \cap M_2$ is the intersection of the visible areas (presenting true data) of the two irises. Because of the uncertainty of the iris rotation angle, a more complex distance formula is used. The rotation of the original image of the eye is equivalent to a cyclic shift of the normalized image along the $O\phi$ axis. Therefore, one of the templates (together with the mask) is subjected to several shift and compare operations:

$$\begin{aligned} \rho(T_1,T_2) &= \min_{\psi} \rho_\psi(T_1,T_2) \,, \\ \rho_\psi(T_1,T_2) &= \frac{1}{\Omega(\psi)}|\{T_1(\phi+\psi,\rho) \neq T_2(\phi,\rho), (\phi,\rho) \in \Omega(\psi)\}| \\ \Omega(\psi) &= M_1(\phi+\psi) \cap M_2(\phi) \,, \end{aligned} \tag{31}$$

where $\psi \in [-S;S]$ is the image rotation angle.

Here things may be simplified. For the embedding method, only irises with low occlusion levels are acceptable. Thus, it is supposed that masks M_1 and M_2 cover all of the iris area, and Ω set spans all templates. Omitting mask, rewriting $|\{T_1 \neq T_2\}|$ as $\sum T_1 \oplus T_2$ and using single order index i instead of coordinates (ϕ,ρ) put (30) as:

$$\rho_0(T_1,T_2) = \frac{1}{N}\sum_{i=1}^{N} T_1(i) \oplus T_2(i) \,, \tag{32}$$

where $T(i)$ is the i-th bit of the template, operation $\oplus$ is the sum modulo 2, N is the size of the template. Furthermore, (31) transforms to

$$\begin{aligned} \rho(T_1, T_2) &= \min_{\psi} \rho_{\psi}(T_1, T_2) \, , \\ \rho_{\psi}(T_1, T_2) &= \frac{1}{N} \sum_{i=1}^{N} T_1(i(\psi)) \oplus T_2(i) \, , \end{aligned} \tag{33}$$

where $i(\psi)$ index is recalculated accordingly.

The recognition system is designed to supply the following conditions with the lowest possible errors:

$$\begin{aligned} T_1, T_2 \text{ taken from same person} &\implies \rho(T_1, T_2) \leqslant \theta \, , \\ T_1, T_2 \text{ taken from different persons} &\implies \rho(T_1, T_2) > \theta \, . \end{aligned} \tag{34}$$

Violation of the first condition in (34) is called *false reject* and its probability is referred to as *false reject rate* (FRR). FRR of the system is estimated in tests as the ratio of the number of false rejects to the number of all matches of biometric traits of the same persons. Analogously, violation of the second condition in (34) is called *false accept* and its probability is named *false accept rate* (FAR). The threshold θ is chosen from a trade-off between FRR and FAR.

6. Selecting the Embedding Method

There are many works, where biometry is used in combination in combination with other security measures such as usual secured passwords, for instance [41,42]. Here, we intend to develop a system that uses only data transmitted insecurely—the only protection is the iris of the owner.

We also limit ourselves to the case of symmetric encryption. During encoding the *message* M and the secret *key* K are combined into the *code* by the *encoder* function Φ: $C = \Phi(M, K)$, and during decoding the message is reconstructed from code and key by *decoder* functions Ψ: $M = \Psi(C, K)$. If key K is not present, it is impossible to obtain M from C, thus the code C can be made public. Symmetric encryption requires that K is repeated exactly. Not a single bit of it can be changed.

The central problem in automatic biometry systems can be put as developing the optimal classifier. The classifier consists of a distance function between two biometric data samples $\rho(D_1, D_2)$ and a threshold θ (34). The function ρ can be treated as a superposition of two sub-functions. The first one is the calculation of the biometric template T from source data $T = T(D)$, Second sub-function is the calculation of the distance itself $\rho(T_1, T_2)$. Features should be selected, which are stably close for the same person and stably far for different persons with respect to function ρ. As a rule, the elements of biometric templates are highly correlated. On the contrary, cryptographic keys are deliberately developed so as to have uncorrelated bits. However, the entropy (information amount) of an iris template is comparable to that of currently used cryptographic keys [43]. This suggests that it is possible to implement a cryptographic key in biometrics without reducing its robustness.

It should be noted that most of the works presenting the application of cryptographic methods to biometrics, develop the scenario of *cancelable biometrics* [44]. Its essence is producing such biometric templates that source biometric data cannot be extracted or guessed anyhow from any number of templates. Cancelable biometrics is nothing but a kind of fuzzy hashing [45]. Formally, an additional step is introduced in the calculation of the distance function ρ. Distance $\rho(S_1, S_2)$ is calculated, $S = S(T)$ is the hash function. Obviously, the recognition problem is still being solved here. Thus, cancelable biometrics is just a remake of identification and cannot be used for our purposes.

There are two approaches to how to process volatile biometrics, leading them to an unchanging cryptographic key. The first approach employs already computed biometric features constituting the template T, which are supplemented with error correction using

different variants of redundant coding. This approach is used here. In the second approach [46] biometric features are not obtained in explicit form. Instead, a neural network is trained, which directly produces a target key from raw biometric data D. The advantage of this approach is said to be less coding redundancy by using continuous data at all stages and quantization only at the end. Disadvantages are the unpredictability of neural network training, lack of guaranteed quality of performance, including uncertainty in retaining quality in a wider set of data than that used in training.

The task of reproducing a cryptographic key is accomplished by *biometric cryptosystems* (BC) [45,47], also called *biometric encryption* [48]. There are two classes of BCs, which implement different approaches: *key generation* and *key binding*.

Methods of key generation, i.e., direct production of the key from raw biometry or template without using supplementary code are studied in [49–51]. Biometric template T is mapped into the space of cryptographic keys (usually bit strings) by a special function: $K(T) : T \to \{0,1\}^n$, where n is the length of the key. One function is used for registration and recognition. The conditions must hold

$$\begin{aligned} T_1, T_2 \text{ taken from one person } &\implies K_1 = K_2 \,, \\ T_1, T_2 \text{ taken from different persons } &\implies K_1 \neq K_2 \,. \end{aligned} \tag{35}$$

These conditions are closely related to (34); however, in (35) the task is to reproduce the sequence of bits. The results of the methods without supplementary data are not very hopeful for practical applications. Error level is generally very high in this approach. In [50] the authors report FRR $= 24\%$ at FAR $= 0.07\%$ even for homogeneous high-quality images [32]. In [51], the idea is based on assumption that two iris codes can be mapped to some "closest" prime number and this number will be the same for the codes from one person. Considering the variability of iris codes even for ideal conditions this is unlikely to happen. The authors do not report the study of recognition errors.

Scenario with *helper code* demonstrates much better performance. During registration the encoder takes the template T_1, computes the key $K_1 = K(T_1)$, encrypts the message M with K_1 and additionally outputs some helper code $h = \Phi(T_1)$. Immediately after this the original T_1, M, and K_1 are destroyed, leaving only encoded message M' and helper code h. The original template T_1 or key T_1 cannot be recovered from M' and h. During presentation another set of biometric traits T_2 is obtained and the key $K_2 = \Psi(T_2, h)$ is calculated. Functions Φ and Ψ are designed so as to satisfy (35). Thus, by providing biometrics and helper code, the registered user can obtain the original key $K_2 = K_1$, and hence the message M. At the same time, the intruder, even knowing h, will not be able to obtain K_1 [52], so the helper code h can be made non-secret.

The biometric data itself may be used as a key: $K \equiv T$. In this case, at the stage of presentation, original biometrics T_1 is restored from presented T_2. This scenario is called *secure sketch* [53]. However, the works based on secure sketches and available in the literature show rather modest results. For example, the system [54] is workable under the assumption that intraclass variability of features is below 10%. In practice, the variability is more than 20%. This conditions the inoperability of the proposed method.

The *key binding* scheme in the above terms looks like a pair of encoder function $C = \Phi(K_1, T_1)$ and decoder function $K_2 = \Psi(C, T_2)$, which holds the (35) condition. The advantage is that K_1 is set externally, rather than created by the developed algorithm. From this point of view, K_1 can be perceived as a message M, which is external to the encryption system. This immediately simplifies the biometric cryptosystem to a symmetric encryption scenario. The difference is that the secret key K must be the same in encoding and decoding in symmetric encryption, whereas the biometric features (also secret) differ: $T_1 \neq T_2$. This scenario is called *fuzzy extractor* [53].

If Ψ is an inverse of Φ and biometric data are composed of real numbers the scenario is referred to as *shielding functions* [55]. So-called *fuzzy vault* [43] is another popular method of key embedding. It is founded on Shamir's secret separation scheme [56]. Here rather

low, practically meaningful error values are obtained: [57] reports FRR = 0.78% and [58] reports FRR = 4.8% at zero FAR. However, both results are shown using a single small image database (less than 1000 samples).

The most promising for use in iris biometry is *fuzzy commitment* scenario [59]. In [60], a simple algorithm is proposed. The basic idea is to use employ the *error correcting coding* (ECC) [61]. ECC is widely used in data transmission over noisy channels. Generally, data transmission involves a pair of functions also called encoder and decoder. The encoder $R = \Phi_p(K)$ maps the transmitted message K into a larger redundant code R. Then R is passed through the transmission channel, which alters each of its symbols independently with the probability q, and the altered code R' is received at the other side. The decoder $K = \Psi_p(R')$ is able to restore K back from R' under the condition that no more than a p share of values were altered. Call p as *tolerated error probability*. Thus, if $q < p$ then the message is restored with a probability close to 1. Otherwise, the probability to restore K is close to 0. One can design Φ and Ψ for a wide range of transition error probabilities $p \in [0; 0.5)$. Redundancy grows as p approaches to 0.5, for $p = 0.5$ it becomes infinite.

Here ECC is used as follows. The encoder and decoder are constructed so as to have a tolerated error probability equal to the classification threshold of the biometric recognition system: $p = \theta$. Upon registration, a password K_1 is constructed and the user's template T_1 is obtained. The code $R_1 = \Phi_p(K_1)$ (it generally looks like pseudorandom numbers) is bitwise summed modulo 2 (exclusive or) to the iris template yielding the public code $C = R_1 \oplus T_1$. After C is calculated, template T_1, message K_1, and redundant code R_1 are destroyed. None of them can be extracted from C alone. Thus, it is possible to expose C publicly and transmit it through unprotected channels. Upon presentation iris of a person is registered once more and a new template T_2 is formed. Of course, it is not equal to the original one. Since $R_2 = C \oplus T_2 = (R_1 \oplus T_1) \oplus T_2$, then $R_1 \oplus R_2 = T_1 \oplus T_2$. If the templates T_1 and T_2 are taken from one person, the distance is very likely to be less than the classification threshold: $\rho(T_1, T_2) \leqslant \theta$, so $\rho(R_1, R_2) \leqslant \theta$. By the nature of (32) it means that less than p share of bits differ in R_1 and R_2 and the original secret key $K_1 = K_2 = \Psi_p(R_2)$ will be recovered. On the other hand, if the templates are classified as belonging to different persons $\rho(T_1, T_2) \geqslant \theta$, the probability of restoring the original K_1 is close to zero. The scenario of operation is shown in Figure 4.

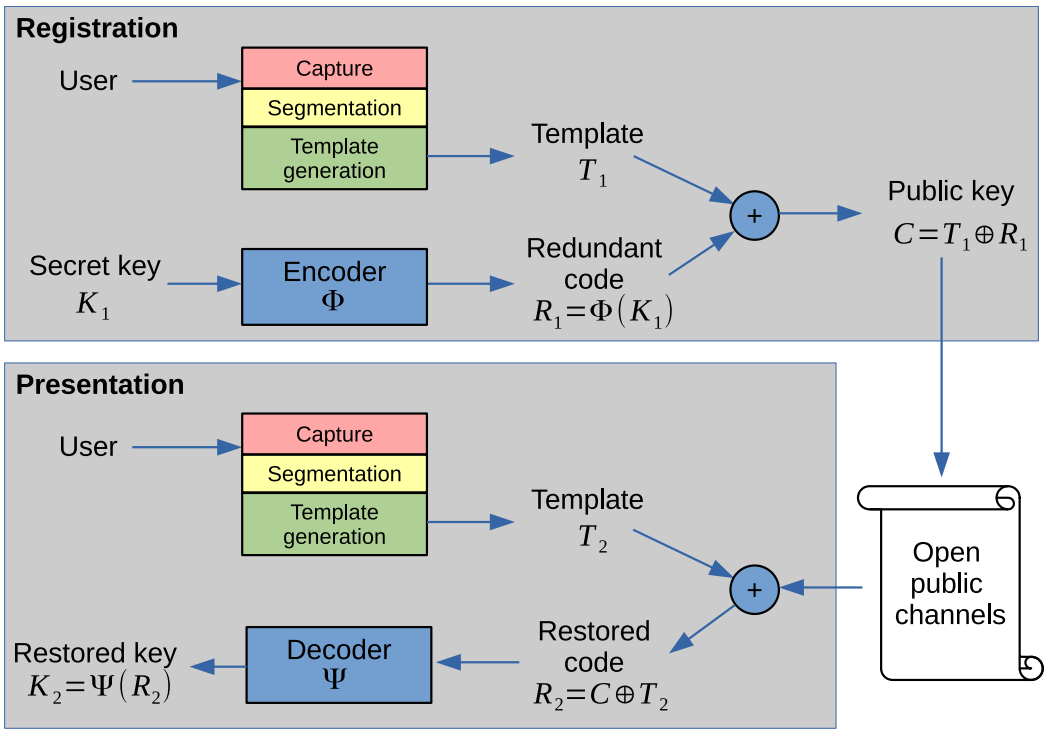

Figure 4. Scenario of the method [60].

Work [60] proposes a cascade of two ECC algorithms: Reed–Solomon [62] and Hadamard [61]. Reed–Solomon coding handles an entire block of data of length L, processing it as a set of L/s s-bit symbols. Any arbitrary symbols (not bits!) can be different as long as their number is not greater than pL. In [60], this coding is aimed to combat group errors appearing from various occlusions (eyelashes, eyelids), which cover significant areas of the iris. Hadamard coding processes small chunks of data (few bits), and corrects no more than 25% of the errors in each chunk. For Hadamard code to be most successful in error correction, the errors (deviations of T' from T) should be evenly scattered across the template with a density of no more than 25%. This coding is designed to deal with single pixel deviations arising from camera noise. The key K is encoded by Reed–Solomon code, the result is processed by Hadamard code.

This cascade performs well if the share of altered bits in one person's templates does not exceed 25%. However, in practical databases and applications this share is bigger which leads to an unacceptably high (more than 50%) false reject probability. To overcome this difficulty, it is proposed [42] to introduce additional template masking: every fourth bit of the iris templates is set to zero. Due to this, the proportion of altering bits in the templates of one person is reduced below 20%. This easy solution ruins the very idea of security: if some bits of the template are fixed, then appropriate bits of redundant code are made known to code crackers and can be used to attack the code. A critique of this method in terms of resistance to cracking is given in [46]. The attack is carried out by gradually restoring the original template.

Here we attempt to refine the fuzzy extractor [60] in a more feasible method and build a practically applicable key embedding method. Based on the iris feature extraction system, experiments against several publicly available iris databases are carried out. Two steps are added to the encoder tail (and hence, decoder head): majority coding of single bits and pseudorandom bit mixing. Three of these four steps have parameters, which affect their properties, including error tolerance and size. Optimal values of these parameters are selected to fit the redundant code size into the iris template size, keep the error tolerance near the target level, and maximize the size of encoded key.

7. Determining the Threshold

So, if the registration template T and the presentation template T' are at Hamming distance (32) below the threshold θ, then the encrypted message M is recovered with high confidence; otherwise, the probability to recover it is extremely low (order of random guess). Thus, the threshold θ separates "genuine" and "intruder" templates T' with respect to T.

It is necessary to determine the value of the threshold p, which will be used for separating "genuines" and "impostors". With this value redundant coder Φ_p and decoder Ψ_p will be devised, capable to restore message for "genuine" template and making it impossible for "impostor" template.

The following publicly available databases were used for the experiments: CASIA-4-Thousand [32], BATH [33], ICE subset of NDIRIS [34], UBIRIS-1 [35].

Table 5 gives a list of databases used with the obtained thresholds.

Table 5. Database characteristics and thresholds.

Database	Number of		θ at FAR$= 10^{-4}$	FAR at $\theta = 0.35$, $\times 10^{-4}$	FRR at $\theta = 0.35$, $\times 10^{-2}$
	Eyes	Images			
BATH	1600	31988	0.402	0.03	4.46
CASIA	2000	20000	0.351	0.97	6.71
ICE	242	2953	0.395	0.011	7.13
UBIRIS	240	1207	0.401	0.001	5.18

For each database the following numbers are given:

- The number of individual eyes. It does not match the number of participating persons as some persons in the database supplied images from both eyes.
- The number of eye images. Each eye produced from one to several hundred images, depending on the database collection scenario.
- Value of the threshold θ, at which the false accept rate (FAR) is 10^{-4}. This value is the probability of a random guess of a four-digit pin code. It is used for reference in developing biometric recognition systems so that they would have the same or less probability of being defeated as pin-code-based systems.
- False accept rate at $\theta = 0.35$.
- False reject rate at $\theta = 0.35$.

Since $FAR(\theta)$ is a monotonous growing function, we select a minimal $\theta(FAR = 10^{-4})$ from the fourth column of the table. It is $\theta = 0.351$ for the CASIA database. Other databases have even smaller FAR with this value of θ.

So, the value of $\theta = 0.35$ is the tolerated error probability p for constructing the ECC. Table 5 shows the values of false accept and false reject rates for this threshold. The maximum false reject rate does not exceed 8%.

8. Error Correction Cascade

We describe the applied methods in the sequence of their execution by the decoder, which is also the method "from simple to complex". In the beginning data unit is a single bit, at the end it is the whole message. The problem is to devise an error correction method, which encodes a message into a block of data with redundancy, and then is able to reconstruct the message if no more than $p \leqslant 0.35$ share of these bits is altered. Popular Walsh-Hadamard and Reed-Muller [63] methods can be used only for $p < 0.25$, thus they are not directly applicable. Furthermore, the errors of neighboring elements of the template are strongly correlated, whereas almost all ECC methods show the best performance against uncorrelated errors.

8.1. Decorrelation by Shuffling

It is more difficult to design methods usable against correlated errors, and their performance is worse compared to the case of uncorrelated errors. Much of the effort in this case is directed precisely at decorrelation. Luckily, the whole block of data is available in our task (rather than sequentially feeding with symbols as in many transmission channel systems) and a simple method of decorrelation can be applied which is the quasi-random shuffling of iris template bits. A bit of the template array T is placed from position i into position $ij \mod N$:

$$\tilde{T}(ij \mod N) = T(i)\,, \quad i = \overline{0, N-1}\,, \tag{36}$$

where j is a number, relatively prime to total number N of bits in array. Relatively prime condition guarantees $ij \mod N$ number being unique for $i = \overline{0, N-1}$ and the shuffling $T \to \tilde{T}$ being reversible. After shuffling the neighboring bits of $\tilde{T}$ are taken from bits, which were far away from each other in T and their errors are uncorrelated. If one rule of shuffling is always applied then bits in all templates change their position in the same method, and calculation (32) affects the same pairs of bits. Thus, Hamming distance is unchanged and all developments from it are preserved.

This method does not change the size of the code.

8.2. Bit-Majority Coding

The error rate $p = 0.35$ is too big for most error correction codes. Practically, the only possibility here is the majority coding of single bits. It is applicable for $p < 0.5$. At the coding stage, the bit is repeated n times. At the decoding stage, the sum of n successive bits is counted. If it is below $n/2$, zero bit value is decoded, otherwise a unit. It is easy to

see that odd values of n are preferable. If p is the error probability of a single bit and bits are altered independently the error probability of decoded bit is

$$p_D(p) = 1 - \sum_{l=0}^{(n-1)/2} \binom{n}{l} p^{n-l}(1-p)^l = 1 - (1-p)^n \sum_{l=0}^{(n-1)/2} \binom{n}{l} \left(\frac{p}{1-p}\right)^l . \quad (37)$$

If the error probability of one bit of the code is $p = 0.35$, then bit majority coding with $n = 7$ will transmit bits with error probability $p_D = 0.2$. This value is below 0.25 and allows the use of Hadamard codes. Majority coding with $n = 15$ will give $p_D = 0.12$ for $p = 0.35$. A larger duplication is possible, but results in a larger code size.

The parameter of this method, affecting its size and error probabilities is the bit repetition rate n.

8.3. Block Coding

Denote the set of all bit strings of length n as $\mathbb{B}^n$. This set can be viewed as a set of vertices of n-dimensional binary cube. Consider the string of length k called *message* here: $M \in \mathbb{B}^k$. There can be 2^k different messages. Consider the set of 2^k strings of length $n > k$ called *codes*. There is a one-to-one correspondence of messages and codes. Since the code length is greater than the message length, the coding is redundant and it is possible to alter some bits of the code but still be able to restore the corresponding message. The idea of block coding is to select such 2^k codes out of their total amount of 2^n, that the probability of restoration error is minimal. The set of selected codes is called *code table* $\mathbb{C}$. In terms of the n-dimensional binary cube, this means selecting 2^k vertices so as to maximize minimal Hamming distance between selected vertices:

$$\mathbb{C}^* = \arg\max_{\mathbb{C}} \min_{\substack{u,vs.\in\mathbb{C} \\ u\neq v}} \rho(u,v) , \quad \rho^* = \min_{\substack{u,vs.\in\mathbb{C}^* \\ u\neq v}} \rho(u,v) , \quad (38)$$

where ρ is the distance (32).

Hadamard coding is based on a Hadamard matrix, which is constructed iteratively:

$$H_0 = (0) , \quad H_{n+1} = \begin{pmatrix} H_n & H_n \\ H_n & \overline{H}_n \end{pmatrix} , \quad (39)$$

where $\overline{H}$ is a bit inversion of all elements of H. Hadamard matrix H_k is a square matrix with 2^k rows. It gives the coding table naturally: each row number is the message and the row contents is the code. It can be proven that for Hadamard codes $\rho^* = 2^{k-1}$. We use so-called augmented Hadamard codes, where another 2^k strings are added to the code table. These strings are bitwise inverted versions of the strings obtained from (39). For this code table $\rho^* = 2^{k-2}$.

There is a simple and well-known estimation (called *Hamming boundary*) of the probability of block coding error, which for augmented Hadamard codes is:

$$p_H \leqslant 1 - P_{corr} = 1 - (1-p)^n \sum_{l=0}^{(n-1)/4} \binom{n}{l} \left(\frac{p}{1-p}\right)^l , \quad (40)$$

where p is the probability of bit inversion. Since this stage inputs the output of bit majority decoding, the value of p here is the value of p_D from (37). Let us redefine $p_D \to p$ in this section for simplicity. Furthermore, one can note that (40) is the same as (37) except for the upper summation limit.

However, this is a rather rough estimate, which grows worse with the increase in n. For small n exact calculations performed by simple exhaustive search, the results are given below. The message is decoded assuming that the original code is distorted minimally, i.e., for the code C we will look for the closest code $C^* \in \mathbb{C}$. Let us call it *attractor*. There can

be several attractors (several codes can have the same minimal distance to C). If there are several attractors, a random one is chosen. Let us denote the set of attractors for C as $A(C)$.

The probability of decoding the correct message is that of choosing the correct attractor

$$P_{corr} = \sum_M P(M) \sum_C p(C^*|C) p(C|C^*) \,, \tag{41}$$

where $P(M)$ is the probability to obtain message M as input. Message M is encoded by code $C^* \in \mathbb{C}$. Then $p(C|C^*)$ is the probability to obtain distorted code C while transmitting C^*, $p(C^*|C)$ is probability to recover C^* (hence, correct M) from distorted code C. Suppose the probability of all messages is the same. Then the sum by M is reduced and

$$P_{corr} = \sum_C p(C^*|C) p(C|C^*) \,. \tag{42}$$

Without loss of generality, due to the symmetry of Hadamard codes [61], we can assume C^* to be a zero code, i.e., a string of zero bits (as is in standard code). Then the probability of obtaining a certain code C from zero code is $p^{b(C)}(1-p)^{n-\beta(C)}$, where $\beta(C) \equiv \rho(0, C)$ is number of unit bits in C, and

$$P_{corr} = \sum_C p(0|C) p^{\beta(C)} (1-p)^{n-\beta(C)} \,, \tag{43}$$

where $p(0|C)$ is the likelihood to obtain zero code from C. Define the set of attractors for string C as code table entries with minimal distance to the code:

$$\begin{aligned} &A(C) = \left\{ C' \in \mathbb{C} : \rho(C', C) = \rho_{\min}(C) \right\} \,, \\ &\rho_{\min}(C) = \min_{C' \in \mathbb{C}} \rho(C, C') \,. \end{aligned} \tag{44}$$

Define the cardinality of this set as $\alpha = |A(C)|$ and state that $\alpha = 0$ if there is another code table entry more close to C then the correct code: $\exists C' \in \mathbb{C}, C' \neq C^* : \rho(C', C) < \rho(C^*, C)$. Then we can write

$$p(0|C) = \begin{cases} 0 \,, & \alpha = 0 \,, \\ 1/\alpha \,, & \alpha \neq 0 \,. \end{cases} \tag{45}$$

For small values of n all points of code space $\mathbb{B}^n$ can be enumerated and their distribution by distance to zero β and attractor number α can be estimated:

$$\begin{aligned} &H(\tilde{\beta}, \tilde{\alpha}) = \left| \left\{ C : 0 \in A(C), |\alpha(C)| = \tilde{\alpha} \,, \;\; \beta(C) = \tilde{\beta} \right\} \right| \,, \\ &\tilde{\beta} \in [0; n] \,, \;\; \tilde{\alpha} \in [1; 2^k] \,. \end{aligned} \tag{46}$$

Substituting to (43) we get:

$$P_{corr} = \sum_{\alpha \neq 0} \sum_{\beta} \frac{H(\beta, \alpha)}{\alpha} p^{\beta} (1-p)^{n-\beta} = \sum_{\beta} p^{\beta} (1-p)^{n-\beta} \sum_{\alpha \neq 0} \frac{H(\beta, \alpha)}{\alpha} \tag{47}$$

and decoding error

$$p_H = 1 - P_{corr} = 1 - (1-p)^n \sum_{\beta} h(\beta) \left(\frac{p}{1-p} \right)^{\beta} \,, \;\; h(\beta) = \sum_{\alpha \neq 0} \frac{H(\beta, \alpha)}{\alpha} \,. \tag{48}$$

The formula is the same as (40) except for the coefficients and summation limits. The values of $h(\beta)$ for the augmented Hadamard code of order 5 ($n = 2^5 - 1 = 31$, $k = 5 + 1 = 6$) are given in Table 6. All meaningful values are given, and values for other α and β are zero. For instance, if the code is distorted in 12 bits or more, it will be never

recovered to the correct value, since there will be another valid code from the code table, closer to distorted value.

Table 6. Values $h(\beta)$ and $p_H(\beta)$ for Hadamard code $k = 6$, $n = 31$.

β	$h(\beta)$	$p_H(\beta)$
0	1	0.
1	31	0.
2	465	0.
3	4495	0.
4	31,465	0.
5	169,911	0.
6	736,281	0.
7	2,629,575	0.
8	7,490,220	0.026
9	13,798,100	0.164
10	8,265,964	0.508
11	427,924	0.810

Up to values $\beta = 7$ only one attractor is chosen, i.e., at this or less divergence the message is definitely recovered. This corresponds to Hamming boundary (40). However, even with larger divergences, up to $\beta = 11$ there is a significant probability of correct recovery. This plays a big role since the majority of distorted codes fall outside of Hamming boundary but still have the significant probability to restore the message correctly. Thus, the probability (40) is overestimated. For example, for the considered code and error $p = 0.250$, the formula (40) gives $p_H = 0.527$, which would seem to prevent using such a code. However, the calculation using the formula (48) gives $p_H = 0.261$, which is fairly suitable for use in the next step of coder.

The Hadamard coding parameter is only the word length k. The size of the codeword n is dependent: $n = 2^{k-1}$ for the augmented variant.

8.4. Reed–Solomon Message Coding

The unit of encoding for Reed–Solomon's algorithm is the entire message, which is divided into codewords of fixed size, s bits each. A stream of L bits is cut into $k = \lceil L/s \rceil$ words. Then additional words can be added up to the total count of n by the coding algorithms. It turns out that if no more than $t = \lfloor (n-k)/2 \rfloor$ codewords are altered then it is possible to recover the message. So, Reed–Solomon code corrects no more than t of errors, where t is half the number of redundant words. Denoting $p = t/n$, we get

$$p \leqslant \frac{n-k}{2n} \,. \tag{49}$$

This number is an estimate of the tolerated error probability of a codeword. The Reed–Solomon method also imposes a limitation to a codeword count in the whole message:

$$n \leqslant 2^s - 1 \,. \tag{50}$$

The error probability in (49) is determined by the previous step: $p = p_H$. Hence, possible Reed–Solomon codes here are determined by codeword length s and message length L.

9. Selection of Code Parameters

Four ECC methods are organized in a chain. Encoder runs Reed–Solomon, Hadamard, bit majority and shuffling to obtain the redundant code. The decoder executes this chain in reverse order. The encoder should obtain the code of size no more than the size of the iris template, i.e., $N = 6656$ bits for the presented system. The code size cannot be larger, duplication and masking are unacceptable, as they make it trivial to break such a code.

Furthermore, of course, it is desired to embed the message of reasonable size. This is a discrete constrained optimization problem.

ECC methods used here have the following parameters affecting their characteristics: (1) decorrelation has no parameters; (2) majority coding is governed by the bit duplication count n; (3) Hadamard coding depends on word size k, Reed–Solomon coding is parameterized by word size s and message length L. Combinations of (n, k, s, L) values yield different encoding with specific code length $C(n, k, s, L)$ and error probabilities. The errors are the aforementioned FRR and FAR. False rejection is a failure to recover the embedded key after presenting the same person's biometrics. False acceptance is recovering the person's key with another person's biometric. The errors depend on ECC parameters: $FRR(n, k, s, L)$ and $FAR(n, k, s, L)$. Furthermore, the formal statement of the problem is

$$\begin{gathered} FRR(n, k, s, L) \to \min , \\ s.t.\ \ FAR(n, k, s, L) \leqslant 10^{-4} , \ \ C(n, k, s, L) \leqslant N. \end{gathered} \tag{51}$$

10. Results and Comparison with Literature

The solution of (51) was found: $L = 65$, $n = 13$, $k = 5$, $s = 5$, $FRR = 10.4\%$. Message size $L = 65$ bits is considered satisfactory for "common" user keys. For bigger message sizes there reasonable solution was not discovered. It should be noted that without bit shuffling (not explicitly involved in (51)) the problem is not solved even for $L = 65$.

Table 7 contains the results reported in the literature in contrast with those presented in this work.

Table 7. The results of iris biometric cryptosystems

Authors	FRR/FAR	DataSet	Keybits
Wu et al. [64]	5.45/0.73	CASIA v1	1024
Rathgeb & Uhl [50]	4.92/0.0	CASIA v3	128
Hao et al. [60]	0.42/0.0	70 persons	140
Bringer et al. [65]	5.62/0.0	ICE 2005	40
Kanade08 et al. [42]	2.48/0.0008	ICE 2005	234
Presented	10.4/0.0	mixed	65

The presented system may seem not very successful against its rivals with respect to error level and key length. However, one should note that each of these systems was tested with a single database. Both CASIA databases have images of one eye taken from adjacent video frames that results in the extremely high similarity of iris codes, inaccessible in practice. The same issue concerns [60], they use a small laboratory database and their results cannot be extended to real applications. ICE 2005 database is much closer to the real world, it contains images of varying quality, time and conditions of registration. However, both works [42,65] based on it use interleaving bits. If the bit sequence is fixed and known, this ruins the cryptographic strength. If it is made secret, then it turns to just another secret key, which should be passed securely: the very thing we try to avoid. Although the presented system has the highest FRR, it is practically applicable and has no obvious holes in security.

11. Conclusions

The set of methods allowing to build biometric cryptosystem based on iris images is presented. It contains three main parts: iris segmentation, biometric template generation, and the method of embedding/extracting the cryptographic key to/from biometric features. The original system of iris segmentation methods is described. Its distinction is estimating iris parameters at several steps (initial rough calculation, then refinement), by algorithms of a different kind. The sequence of detection of iris parameters is different from commonly employed as well. The template creation method is a de facto standard Daugman-style convolution. The method for introducing a cryp-

tographic key into iris biometrics is constructed using a fuzzy extractor paradigm. A key of size up to 65 bits can be embedded, for larger sizes no solution has been obtained. The challenge of high variance of biometric features has been overcome by introducing bit majority coding. A high local correlation of errors was removed by quasi-random shuffling. The system was tested on several databases of iris images. A study of cryptography stability is still required.

Author Contributions: Conceptualization, I.M.; methodology, I.M.; software, I.M.; validation, I.M. and I.S.; investigation, I.M.; data curation, I.M.; writing—original draft preparation, I.M.; writing—review and editing, I.S.; supervision, I.M.; project administration, I.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Bertram, L.A.; van Gunther, D. (Eds.) *Nomenclatura: Encyclopedia of Modern Cryptography and Internet Security—From AutoCrypt and Exponential Encryption to Zero-Knowledge-Proof Keys*; Books on Demand: Paris, France, 2019; ISBN 978-3-7460-6668-4. Available online: https://www.amazon.com/Nomenclatura-Encyclopedia-Cryptography-Internet-Security/dp/3746066689 (accessed on 2 February 2023).
2. Chmora, A.L. Key Masking Using Biometry. *Probl. Inf. Transm.* **2011**, *47*, 201–215. [CrossRef]
3. Daugman, J. How Iris Recognition Works. *IEEE Trans. Circuits Syst. Video Technol.* **2004**, *14*, 21–30. [CrossRef]
4. Shekar, B.H.; Bharathi, R.K.; Kittler, J.; Vizilter, Y.V.; Mestestskiy, L. Grid structured morphological pattern spectrum for off-line signature verification. In Proceedings of the International Conference on Biometrics, Phuket, Thailand, 19–22 May 2015; Volume 8, pp. 430–435.
5. Bowyer, K.; Hollingsworth, K.; Flynn, P. Image understanding for iris biometrics: A survey. *Comput. Vis. Image Underst.* **2008**, *110*, 281–307. [CrossRef]
6. Radman, A.; Jumari, K.; Zainal, N. Iris Segmentation: A Review and Research Issues. In *Software Engineering and Computer Systems. ICSECS 2011. Communications in Computer and Information Science*; Mohamad Zain, J., Wan Mohd, W.M.B., El-Qawasmeh, E., Eds.; Springer: Berlin/Heidelberg, Germany, 2011; Volume 179.
7. Bowyer, K.; Hollingsworth, K.; Flynn, P. A survey of iris biometrics research: (2008–2010). In *Handbook of Iris Recognition*; Burge, M., Bowyer, K., Eds.; Springer: London, UK; Heidelberg, Germany; New York, NY, USA; Dordrecht, The Netherlands, 2012.
8. Malgheet, J.R.; Manshor, N.B.; Affendey, L.S. Iris Recognition Development Techniques: A Comprehensive Review. *Complexity* **2021**, *2012*, 6641247. [CrossRef]
9. Liu, N.; Li, H.; Zhang, M.; Liu, J.; Sun, Z.; Tan, T. Accurate iris segmentation in non-cooperative environments using fully convolutional networks. In Proceedings of the 2016 International Conference on Biometrics (ICB), Halmstad, Sweden, 13–16 June 2016; pp. 1–8.
10. Wang, C.; Muhammad, J.; Wang, Y.; He, Z.; Sun, Z. Towards Complete and Accurate Iris Segmentation Using Deep Multi-Task Attention Network for Non-Cooperative Iris Recognition. *IEEE Trans. Inf. Forensics Secur.* **2020**, *15*, 2944–2959. [CrossRef]
11. Huo, G.; Lin, D.; Yuan, M.; Yang, Z.; Niu, Y. Heterogeneous iris segmentation method based on modified U-Net. *J. Electron. Imaging.* **2021**, *30*, 063015. [CrossRef]
12. Meng, Y.; Bao, T. Towards More Accurate and Complete Heterogeneous Iris Segmentation Using a Hybrid Deep Learning Approach. *J. Imaging*. **2022**, *8*, 246. [CrossRef]
13. Korobkin, M.; Odinokikh, G.; Efimov, I.; Solomatin, I.; Matveev, I. Iris Segmentation in Challenging Conditions. *Pattern Recognit. Image Anal.* **2018**, *28*, 817–822. [CrossRef]
14. Hofbauer, H.; Jalilian, E.; Uhl, A. Exploiting superior cnn-based iris segmentation for better recognition accuracy. *Pattern Recognit. Lett.* **2019**, *120*, 17–23. [CrossRef]
15. Cui, J.; Wang, Y.; Tan, T.; Ma, L.; Sun, Z. A Fast and Robust Iris Localization Method Based on Texture Segmentation. In *Biometric Authentication and Testing, National Laboratory of Pattern Recognition*; Chinese Academy of Sciences: Beijing, China, 2004; pp. 401–408.
16. Liu, X.; Bowyer, K.W.; Flynn, P.J. Experiments with an Improved Iris Segmentation Algorithm. In Proceedings of the 4th IEEE Workshop on Automatic Identification Advanced Technologies, New York, NY, USA, 17–18 October 2005; pp. 118–123.
17. Dey, S.; Samanta, D. A Novel Approach to Iris Localization for Iris Biometric Processing. *Intern. J. Biol. Life Sci.* **2007**, *3*, 180–191.
18. Ling, L.L.; de Brito, D.F. Fast and Efficient Iris Image Segmentation. *J. Med. Biol. Eng.* **2010**, *30*, 381–392. [CrossRef]
19. Yuan, W.; Lin, Z.; Xu, L. A Rapid Iris Location Method Based on the Structure of Human Eyes. In Proceedings of the 27th Annual Conf. Engineering in Medicine and Biology, Shanghai, China, 1–4 September 2005; pp. 3020–3023.

20. Pan, L.; Xie, M.; Ma, Z. Iris Localization Based on Multiresolution Analysis. In Proceedings of the 19th International Conference on Pattern Recognition, Tampa, FL, USA, 8–11 December 2008; pp. 1–4.
21. He, Z.; Tan, T.; Sun, Z.; Qiu, X. Toward Accurate and Fast Iris Segmentation for Iris Biometrics. *IEEE PAMI* **2009**, *31*, 1670–1684.
22. Maenpaa, T. An Iterative Algorithm for Fast Iris Detection. In Proceedings of the International Workshop on Biometric Recognition Systems, Beijing, China, 22–23 October 2005; p. 127.
23. Nabti, M.; Ghouti, L.; Bouridane, A. An Effective and Fast Iris Recognition System Based on a Combined Multiscale Feature Extraction Technique. *Pattern Recognit.* **2008**, *41*, 868–879. [CrossRef]
24. Matveev, I. Iris Center Location Using Hough Transform with Two-dimensional Parameter Space. *J. Comput. Syst. Sci. Int.* **2012**, *51*, 785–791. [CrossRef]
25. Proenca, H.; Alexandre, L.A. Iris Segmentation Methodology for Non-cooperative Recognition. *IEEE Proc. Vision Image Signal Process.* **2006**, *153*, 199–205. [CrossRef]
26. Matveev, I. Detection of Iris in Image by Corresponding Maxima of Gradient Projections. In Proceedings of the Computer Graphics, Visualization, Computer Vision and Image Processing 2010, Freiburg, Germany, 27–29 July 2010; pp. 17–21.
27. Novik, V.; Matveev, I.A.; Litvinchev, I. Enhancing Iris Template Matching with the Optimal Path Method. *Wirel. Netw.* **2020**, *26*, 4861–4868. [CrossRef]
28. Solomatin, I.A.; Matveev, I.A.; Novik, V.P. Locating the Visible Part of the Iris with a Texture Classifier with a Support Set. *Autom. Remote Control* **2018**, *79*, 492–505. [CrossRef]
29. Pratt, W.K. *Digital Image Processing: PIKS Scientific Inside*, 4th ed.; Wiley–Interscience: New York, NY, USA, 2007.
30. Sun, C.; Pallottino, S. Circular Shortest Path in Images. *Pattern Recognit.* **2003**, *36*, 709–719. [CrossRef]
31. Gankin, K.A.; Gneushev, A.N.; Matveev, I.A Iris Image Segmentation Based on Approximate Methods with Subsequent Refinements. *J. Comput. Syst. Sci. Int.* **2014**, *53*, 224–238. [CrossRef]
32. CASIA. Iris Image Database, Institute of Automation, Chinese Academy of Sciences. 2010. Available online: http://biometrics.idealtest.org/findTotalDbByMode.do?mode=Iris (accessed on 2 February 2023).
33. Woodard, D.L.; Ricanek, K. Iris Databases. In *Encyclopedia of Biometrics*; Li, S.Z., Jain, A., Eds.; Springer: Boston, MA, USA, 2009.
34. Phillips, P.J.; Scruggs, W.T.; O'Toole, A.J.; Flynn, P.J.; Bowyer, K.W.; Schott, C.L.; Sharpe, M. Frvt2006 and Ice2006 Large-scale Experimental Results. *IEEE PAMI* **2010**, *5*, 831–846. [CrossRef]
35. Proenca, H.; Alexandre, L. UBIRIS: A Noisy Iris Image Database. In Proceedings of the 13th International Conference Image Analysis and Processing, Cagliari, Italy, 6–8 September 2005; pp. 970–977.
36. Wildes, R.P. Iris Recognition: An Emerging Biometric Technology. *Proc. IEEE* **1997**, *85*, 1348–1363. [CrossRef]
37. Daugman, J. New Methods in Iris Recognition. *IEEE Trans. Syst. Man-Cybernatics-Part B Cybernatics* **2007**, *37*, 1167–1175. [CrossRef]
38. Masek, L. Recognition of Human Iris Patterns for Biometric Identification. 2003. Available online: https://www.peterkovesi.com/studentprojects/libor/ (accessed on 2 February 2023).
39. Ma, L.; Tan, T.; Wang, Y.; Zhang, D. Local Intensity Variation Analysis for Iris Recognition. *Pattern Recognit.* **2004**, *37*, 1287–1298. [CrossRef]
40. Matveev, I.A.; Novik, V.; Litvinchev, I. Influence of Degrading Factors on the Optimal Spatial and Spectral Features of Biometric Templates. *J. Comput. Sci.* **2018**, *25*, 419–424. [CrossRef]
41. Kumar, M.; Prasad, M.; Raju, U.S.N. BMIAE: Blockchain-based Multi-instance Iris Authentication using Additive ElGamal Homomorphic Encryption. *IET Biom.* **2020**, *9*, 165–177. [CrossRef]
42. Kanade, S.; Camara, D.; Krichen, E.; Petrovska-Delacrétaz, D.; Dorizzi, B. Three Factor Scheme for Biometric-based Cryptographic Key Regeneration Using Iris. In Proceedings of the Biometrics Symposium, Tampa, FL, USA, 23–25 September 2008; pp. 59–64.
43. Juels, A.; Sudan, M. A Fuzzy Vault Scheme. *Des. Codes Cryptogr.* **2006**, *38*, 237–257. [CrossRef]
44. Patel, V.M.; Ratha, N.K.; Chellappa, R. Cancelable Biometrics: A review. *IEEE Signal Process. Mag.* **2015**, *32*, 54–65. [CrossRef]
45. Rathgeb, C.; Uhl, A. A Survey on Biometric Cryptosystems and Cancelable Biometrics. *EURASIP J. Inf. Secur.* **2011**, *3*, 1–25. [CrossRef]
46. Akhmetov, B.S.; Ivanov, A.I.; Alimseitova, Z.K. Training of Neural Network Biometry-Code Converters. *Izv. NAS RK Ser. Geol. Tech. Sci.* **2018**, *1*, 61.
47. Itkis, G.; Chandar, V.; Fuller, B.W.; Campbell, J.P.; Cunningham, R.K. Iris Biometric Security Challenges and Possible Solution. *IEEE Signal Process. Mag.* **2015**, *32*, 42–53. [CrossRef]
48. Cavoukian, A.; Stoianov, A. Encryption, Biometric. In *Encyclopedia of Biometrics*; Li, S.Z., Jain, A.K., Eds.; Springer: Boston, MA, USA, 2015.
49. Sutcu, Y.; Sencar, H.T.; Memon, N.A. Secure Biometric Authentication Scheme Based on Robust Hashing. In Proceedings of the 7th Workshop Multimedia and Security, New York, NY, USA, 1–2 August 2005; pp. 111–116.
50. Rathgeb, C.; Uhl, A. Privacy preserving key generation for iris biometrics. In *Communications and Multimedia Security*; De Decker, B., Schaumueller-Bichl, I., Eds.; Springer: Berlin/Heidelberg, Germany, 2010; pp. 191–200.
51. Therar, H.M.; Mohammed, E.A.; Ali, A.J. Biometric Signature based Public Key Security System. In Proceedings of the International Conference Advanced Science and Engineering, Duhok, Iraq, 23–24 December 2020; pp. 1–6.

52. Davida, G.I.; Frankel, Y.; Matt, B.; Peralta, R. On the Relation of Error Correction and Cryptography to an Offline Biometric Based Identification Scheme. In Proceedings of the Workshop on Coding and Cryptography, Paris, France, 11–14 January 1999; pp. 129–138.
53. Dodis, Y.; Reyzin, L.; Smith, A. Fuzzy Extractors: How to Generate Strong Keys from Biometrics and Other Noisy Data. *SIAM J. Comput.* **2008**, *38*, 97–139. [CrossRef]
54. Yang, S.; Verbauwhede, I. Secure Iris Verification. In Proceedings of the 2007 IEEE International Conference on Acoustics, Speech and Signal Processing—ICASSP'07, Honolulu, HI, USA, 15–20 April 2007; Volume 2, pp. 33–136.
55. Linnartz, J.-P.; Tuyls, P. New Shielding Functions to Enhance Privacy and Prevent Misuse of Biometric Templates. In Proceedings of the 4th International Conference Audio- and Video-Based Biometric Person Authentication, Guildford, UK, 9–11 June 2003; pp. 393–402.
56. Shamir, A. How to Share a Secret. *Commun. ACM* **1979**, *22*, 612–613. [CrossRef]
57. Lee, Y.J.; Bae, K.; Lee, S.J.; Park, K.R.; Kim, J. Biometric Key Binding: Fuzzy Vault Based on Iris Images. In Proceedings of the 2nd International Conference Biometrics, Seoul, Republic of Korea, 27–29 August 2007; pp. 800–808.
58. Wu, X.; Qi, N.; Wang, K.; Zhang, D. An Iris Cryptosystem for Information Security. In Proceedings of the International Conference Intelligent Information Hiding and Multimedia Signal Processing, Harbin, China, 15–17 August 2008; pp. 1533–1536.
59. Juels, A.; Wattenberg, M. A Fuzzy Commitment Scheme. In Proceedings of the 6th ACM Conference Computer and Communications Security, Singapore, 1–4 November 1999; pp. 28–36.
60. Hao, F.; Anderson, R.; Daugman, J. Combining Crypto with Biometrics Effectively. *IEEE Trans. Comput.* **2006**, *55*, 1081–1088.
61. Morelos-Zaragoza, R.H. *The Art of Error Correcting Coding;* John Wiley and Sons: Hoboken, NJ, USA, 2006.
62. Reed, I.S.; Solomon, G. Polynomial Codes over Certain Finite Fields. *J. Soc. Ind. Appl. Math.* **1960**, *8*, 300–304. [CrossRef]
63. Reed, I.S. A Class of Multiple-error-correcting Codes and the Decoding Scheme. *Trans. Ire Prof. Group Inf. Theory* **1954**, *4*, 38–49. [CrossRef]
64. Wu, X.; Qi, N.; Wang, K.; Zhang, D. A Novel Cryptosystem based on Iris Key Generation. In Proceedings of the 2008 Fourth International Conference on Natural Computation, Jinan, China, 18–20 October 2008; pp. 53–56.
65. Bringer, J.; Chabanne, H.; Cohen, G.; Kindarji, B.; Zemor, G. Optimal iris fuzzy sketches. In Proceedings of the 2007 First IEEE International Conference on Biometrics: Theory, Applications, and Systems, Crystal City, VA, USA, 27–29 September 2007; pp. 1–6.