Optimal Generation of Iris Codes for Iris Recognition

Yang Hu, Konstantinos Sirlantzis, and Gareth Howells

***Abstract*—The calculation of binary iris codes from feature values (*e.g.* the result of Gabor transform) is a key step in iris recognition systems. Traditional binarization method based on the sign of feature values has achieved very promising performance. However, currently, little research focuses on a deeper insight into this binarization method to produce iris codes. In this paper, we illustrate the iris code calculation from the perspective of optimization. We demonstrate that the traditional iris code is the solution of an optimization problem which minimizes the distance between the feature values and iris codes. Furthermore, we show that more effective iris codes can be obtained by adding terms to the objective function of this optimization problem. We investigate two additional objective terms. The first objective term exploits the spatial relationships of the bits in different positions of an iris code. The second objective term mitigates the influence of less reliable bits in iris codes. The two objective terms can be applied to the optimization problem individually, or in a combined scheme. We conduct experiments on four benchmark datasets with varying image quality. The experimental results demonstrate that the iris code produced by solving the optimization problem with the two additional objective terms achieves a generally improved performance in comparison to the traditional iris code calculated by binarizing feature values based on their signs.**

***Index Terms*—iris recognition, iris code, spatial relationship, feature reliability.**

I. Introduction

THE iris has become one of the most reliable biometric traits for human authentication due to some inherent advantages, for example, it is a highly protected internal organ which is visible externally; iris patterns are highly distinctive with a high degree of freedom; iris patterns are relatively stable over time *etc*. State-of-the-art iris recognition algorithms have reported promising performance [1]–[6]. Most of these algorithms use binary features (*i.e.* iris codes). The binary nature of iris codes brings significant advantage in memory and computational cost, enabling the large scale deployment of iris recognition systems. Current nationwide deployments of iris recognition systems in UAE [7] and India [8] are considered successful, with millions of subjects enrolled. Alternative feature extraction and selection approach instead of binary iris codes have also shown high effectiveness [9]–[15].

A traditional iris recognition system mainly consists of three components: iris segmentation, feature extraction and iris matching. Generally, feature extraction can be further divided into two steps: feature value extraction and iris code production. Feature value extraction applies methods such as Gabor transform [1], [2] and Ordinal measure [5] to the iris image, generating a set of feature values. Iris code production obtains binary iris codes from the feature values as the final feature representation of iris. Currently, for traditional iris recognition systems, significant research effort has been devoted to iris segmentation [16]–[19], feature value extraction [1], [2], [5], [20]–[22] and iris matching [23]–[26]. However, for iris code production, most of state-of-the-art methods use simple binarization based on the sign of feature values [1], [2], [5], [21], [27]. Experimentally, this binarization method achieves a generally promising performance, but, currently, there lacks a deeper insight into this method. An important question one may ask is this: “Is it optimal to produce the iris code by binarization based on the sign of feature values?”. A similar question is: “How could we find more effective iris codes given feature values?”. Currently, although a large number of feature selection algorithms are investigated to obtain more effective iris codes [28]–[33], these methods focus on selecting more valuable bits from iris codes, rather than directly calculating optimal iris codes from feature values. Therefore, the above questions remain open.

In this paper, we investigate the issue of iris code production from the perspective of optimization. We illustrate that the traditional iris code based on the sign of feature values is the solution of an optimization problem. This optimization problem seeks iris codes by minimizing the distance between the feature values and iris codes. Such an illustration enables us to apply additional objective terms to this optimization problem, in order to obtain more effective iris codes. We investigate two additional objective terms. The first objective term exploits the spatial relationship of the bits in different positions of an iris code. The idea is that iris pixels are not uncorrelated; instead, they are spatially related [1], [34]. Therefore, as a feature representation of iris textures, the bits at different positions of an iris code should also have spatial relationships. The first objective term exploits such spatial relationships using a Markov Random Field (MRF) model. The second objective term mitigates the influence of less reliable bits in iris codes. As shown in [23], a bit is less reliable if the corresponding feature value is close to the axis in complex Gabor feature space, because such bits are less stable: they are more likely to flip between 0 and 1. Our second objective term aims to mitigate the influence of these less reliable bits. It improves their reliability by enhancing their stability: this term promotes less reliable bits to be assigned uniform bit values to suppress the flipping of these bits. We show that the two additional objective terms can be applied to the original optimization problem not only individually, but

Y. Hu, K. Sirlantzis and G. Howells are with School of Engineering and Digital Arts, University of Kent, Canterbury, UK. e-mail: {yh94, K.Sirlantzis, W.G.J.Howells}@kent.ac.uk

also in a combined scheme. The iris codes obtained by solving the original optimization problem with the two additional objective terms lead to a generally improved performance.

Contributions. Our contribution is as follows. First, We investigate the iris code production from the perspective of optimization. We demonstrate that traditional iris codes based on the sign of feature values are the solution of an optimization problem. Second, based on this optimization problem, we propose two additional objective terms to obtain more effective iris codes. The first objective term exploits the spatial relationship of the bits in different positions of an iris code, while the second objective term mitigates the influence of less reliable bits in iris codes. Experimentally, we find that the two objective terms are able to improve the incorrect iris matching result when using the traditional iris code caused by factors such as the imaging variation in different captures. Third, we propose a scheme to combine the two additional objective terms. We show that the iris code obtained by the combined scheme achieves a generally improved performance, compared to traditional iris codes and the iris codes obtained using each individual objective term.

The remainder of this paper is organized as follows. In section II, we present the proposed method to produce iris codes based on optimization. In section III, we report the result of experimental analysis of the proposed method. Finally in section IV, we concluded the paper.

II. Producing iris codes by optimization

A. Illustration of the traditional iris code production from a perspective of optimization

Let $\boldsymbol{f} = [f_1, f_2, ..., f_n]^{\mathrm{T}} \in \mathbb{R}^n$ be a vector consisting of all features values extracted from an unwrapped iris image. For example, for the most widely used Gabor feature [1], [2], $\boldsymbol{f}$ is constructed by concatenating the real and imaginary parts in all positions into a vector after applying Gabor transform to the unwrapped iris region; each element in $\boldsymbol{f}$ is either the real or imaginary part of the Gabor transform result. Let $\boldsymbol{b} = [b_1, b_2, ..., b_n]^{\mathrm{T}}$ be a binary vector of the iris code corresponding to $\boldsymbol{f}$ with n bits. For the i^{th} bit b_i, traditional binarization method to calculate it is as follows:

$$b_i = \begin{cases} 1 & \text{if } f_i \geq 0 \\ 0 & \text{if } f_i < 0 \end{cases} \tag{1}$$

In other words, traditional iris codes are produced based on the sign of feature values.

In this subsection, we demonstrate that the traditional binarization method can be illustrated as the solution of the following optimization problem:

$$\underset{\mathbf{b} \in \{-1,1\}^n}{\arg\min} \; \|\boldsymbol{b} - \boldsymbol{f}\|_2^2 \tag{2}$$

We note that $\boldsymbol{b} \in \{-1, 1\}^n$ in Eqn. 2 is different from the traditional iris code where $\boldsymbol{b} \in \{0, 1\}^n$. However, we will show that, the solution of Eqn. 2 is equivalent to the tradition iris code in binary feature domain, and it makes no difference in the hamming distance given two iris codes.

The objective function in Eqn. 2 can be expanded as follows:

$$\|\boldsymbol{b} - \boldsymbol{f}\|_2^2 = \|\boldsymbol{b}\|_2^2 + \|\boldsymbol{f}\|_2^2 - 2\boldsymbol{f}^{\mathrm{T}}\boldsymbol{b} \tag{3}$$

Note that $\|\boldsymbol{f}\|_2^2$ is a constant, and $\|\boldsymbol{b}\|_2^2 = n$ due to $\boldsymbol{b} \in \{-1, 1\}^n$. Therefore, Eqn. 2 is equivalent to:

$$\underset{\boldsymbol{b} \in \{-1,1\}^n}{\arg\min} \; -\boldsymbol{f}^{\mathrm{T}}\boldsymbol{b} \tag{4}$$

Since $\boldsymbol{f}^{\mathrm{T}}\boldsymbol{b} = \sum_i f_i b_i$, the optimization problem with respect to b_i reduces to:

$$\underset{b_i \in \{-1,1\}}{\arg\min} \; -f_i b_i \tag{5}$$

To minimize Eqn. 5, the sign of b_i should be consistent with the sign of f_i, thus we can obtain the solution as follows:

$$b_i = \begin{cases} 1 & \text{if } f_i \geq 0 \\ -1 & \text{if } f_i < 0 \end{cases} \tag{6}$$

It can be seen that the solution is equivalent to the traditional iris code (Eqn. 1) if we replace all the -1 by 0 in the obtained iris code $\boldsymbol{b}$, and it does not change the hamming distance given two iris codes. On the other hand, from the perspective of optimization, an explanation is that the solution of Eqn. 5 is equivalent to the solution of the following problem in binary feature space:

$$\underset{b_i \in \{0,1\}}{\arg\min} -f_i b_i \tag{7}$$

Based on the above illustration, we can explain the traditional method of iris code production as follows. Given an unwrapped iris image, the extracted feature values construct a n-dimensional feature vector. In each dimension of the feature vector, two anchor points are set at -1 and 1. The traditional method of iris code production binarizes a feature value by assigning it to the nearest anchor point at the corresponding dimension (*i.e.* Eqn. 2).

B. Iris code production using additional objective terms

Based on the derivation from Eqn. 2 to Eqn. 7 in section II-A, Eqn. 2 formulating the traditional iris code production can be rewritten into the following equivalent problem:

$$\underset{\substack{b_i \in \{0,1\} \\ i=1,2,...,n}}{\arg\min} \; -\sum_i f_i b_i \tag{8}$$

Based on Eqn. 8, we can use more complex models to obtain iris codes by adding terms to the objective function of this optimization problem. We expect that, by adding proper objective terms, we are able to obtain more effective iris codes. In this subsection, we investigate two additional objective terms for iris code production. Note that, in the rest of this paper, we apply both additional objective terms to Eqn. 8, so they are designed based on $b_i \in \{0, 1\}, i = 1, 2, ..., n$. In implementations, $b_i \in \{0, 1\}$ and $b_i \in \{-1, 1\}$ are equivalent for the additional objective term 1, but $b_i \in \{0, 1\}$ is necessary for the additional objective term 2.

Additional objective term 1. This objective term exploits the spatial relationship of the bits in different positions of an iris code. Traditional iris code production method binarizes the feature values in different positions individually (as shown in Eqn. 1 and Eqn. 6). In other words, it considers each feature value separately. However, as studied in [1], [34], the iris texture has inherent correlations along the radial direction. For example, as pointed out in [1], [34], a furrow or ciliary pattern tends to propagate in the radial direction. Therefore, as a feature representation of iris patterns, the bits in an iris code should be dependent along the vertical direction. In this paper, due to the binary nature of iris codes, we assume that such a vertical dependency will lead to a vertical bit-adjacency in an iris code. Accordingly, our first objective term exploits the vertical bit-adjacency in an iris code. It is defined as follows:

$$\sum_{i}\sum_{j\in N^i}|b_i-b_j| \tag{9}$$

where N^i denotes two immediate vertical neighbors of bit i: one at the top of i and one at the bottom of i. Minimizing Eqn. 9 with respect to $b_i \in \{0,1\}, i=1,2,...,n$ prompts each bit in an iris codes to have the same value as its vertical neighbors. Incorporating Eqn. 9 into Eqn. 8 leads to the following optimization problem:

$$\mathop{\arg\min}_{\substack{b_i\in\{0,1\}\\ i=1,2,...,n}} \sum_{i}(-f_i)b_i+\alpha\sum_{i}\sum_{j\in N^i}|b_i-b_j| \tag{10}$$

where α is a parameter controlling the trade-off between the influence of the sign of feature values and the vertical adjacency of iris codes. Eqn. 10 is a first order Markov Random Field with binary labels. It can be solved via graph cuts [35], [36].

Additional objective term 2. This objective term aims to mitigate the influence of less reliable bits in iris codes. Some bits in iris codes are less reliable. Specifically, after Gabor transform, the value of some obtained features are close to the axes in complex Gabor feature space (considering a feature value as a point in complex Gabor feature space, with the real and imaginary parts as the coordinate of this point). Or equivalently, the value of some features are close to the origin in the feature space (*i.e.* close to 0), if we consider the real and imaginary parts separately. As investigated in [23], the bits corresponding to such feature values are less reliable. The reason is as follows. Since such feature values are close to the origin, the noise on these feature values are more likely to change their sign. As a result, in the traditional iris code based on the sign of feature values, the bits corresponding to these feature values are more likely to flip between 0 and 1 under the influence of noise. The flipping of these bits may cause false non-matching in intra-class comparisons, given multiple captures of the same iris. Thus, such bits are less reliable. In other words, a bit is less reliable if the corresponding feature value is close to the origin in the feature space, since such bits are less stable in iris codes.

From the above description of less reliable bits, we can find that the main reason of their low reliability is their low stability: they are more likely to flip between 0 and 1. Therefore, additional objective term 2 aims to improve the reliability of these bits by suppressing their flipping to enhance their stability. This objective term assigns uniform bit values to the bits with the corresponding feature values close to the origin (*i.e.* less reliable bits). By doing this, the value of such bits are no longer influenced by the sign of their corresponding feature values which is less stable (as described above, the sign of a feature value is likely to be changed by noise if this feature value is close to the origin). Thus, we expect that the obtained iris code is more stable and hence more effective.

In this paper, we design the additional objective term 2 to assign less reliable bits uniform bit values of 0. The reason of using 0 rather than 1 is that we find it is more concise to be formulated as an objective term (Eqn. 11). Correspondingly, in the iris code obtained by additional objective term 2, the bit corresponding to a positive feature value close to the origin will have a different bit value from the traditional iris code, while the value of the other bits is the same between the iris code obtained by additional objective term 2 and the traditional iris code.

Additional objective term 2 is formulated as follows:

$$\sum_{i}\sum_{j\neq i}b_ib_j \tag{11}$$

It is obvious that (1) minimizing Eqn. 11 with respect to $b_i \in \{0,1\}, i=1,2,...,n$ encourages a bit to be assigned a value of 0. Furthermore, we can see that Eqn. 11 also models a pairwise correlation of the bits in an iris code. We will show that (2) such pairwise correlation derives a threshold to determine if a bit is less relibale. The overall effect of (1) and (2) is that this objective term assigns bit values of 0 to the found less reliable bits, while it does not influence the value of the other bits.

Incorporating Eqn. 11 into Eqn. 8 leads to the following optimization problem:

$$\mathop{\arg\min}_{\substack{b_i\in\{0,1\}\\ i=1,2,...,n}} -\sum_{i}f_ib_i+\beta\sum_{i}\sum_{j\neq i}b_ib_j \tag{12}$$

where β is a parameter balancing between the first term and the second term.

Solving Eqn. 12 with respect to b_i leads to the following problem:

$$\mathop{\arg\min}_{b_i\in\{0,1\}} -\left(f_i-\beta\sum_{j\neq i}b_j\right)b_i \tag{13}$$

To minimize the objective function, the solution for b_i is as follows:

$$b_i=\begin{cases}1 & \text{if } f_i-\beta\sum\limits_{j\neq i}b_j\geq 0\\ 0 & \text{if } f_i-\beta\sum\limits_{j\neq i}b_j<0\end{cases} \tag{14}$$

It can be seen that Eqn. 14 essentially shifts the feature value towards the negative direction, and it calculates b_i based on the sign of the shifted feature value. By doing this, positive feature values near to the origin (lower than $\beta\sum\limits_{j\neq i}b_j$) will change their signs, and hence the corresponding bits are assigned bit values of 0 instead of 1 (*i.e.* these bits are considered to be less reliable). In Eqn. 14, the shift applied to the feature value, $\beta\sum\limits_{j\neq i}b_j$, essentially serves as the threshold to determine

```
Algorithm 1: Solving Eqn. 12
Input  : vector of feature values $\boldsymbol{f}$; parameter $\beta$
Initialization: obtaining $\boldsymbol{b}^0$ by binarizing $\boldsymbol{f}$ based on the sign of its elements ( Eqn. 2); $t = 0$;
while not converge do
    $\widetilde{\boldsymbol{b}}^t = \|\boldsymbol{b}^t\|_2^2 - \boldsymbol{b}^t$;
    $\widetilde{\boldsymbol{f}}^t = \boldsymbol{f} - \beta\widetilde{\boldsymbol{b}}^t$;
    compute $\boldsymbol{b}^{t+1}$ by binarizing $\widetilde{\boldsymbol{f}}^t$ based on the sign of its element;
    $t \leftarrow t + 1$;
end
Output: $\boldsymbol{b}$
```

if a bit is less reliable (*i.e.* near to the origin). b_i will be considered less reliable and assigned a bit value of 0 instead of 1 only if the corresponding feature value (f_i) is positive but its distance to the origin is lower than $\beta \sum_{j \neq i} b_j$. It can be seen that this threshold is dynamic and it is derived by solving the optimization problem with additional objective term 2 (Eqn. 12). Eqn. 14 only focuses on the less reliable bits corresponding to positive feature values. The reason is that, as illustrated above, the additional objective term 2 is designed to assign uniform bit values of 0 to all less reliable bit; therefore, it does not consider the less reliable bits corresponding to negative feature values, since their bit values are already 0.

According to Eqn. 14, the solution for b_i depends on the values of the other bits. Similarly, the solution of the other bits also depends on the value of b_i. Therefore, the final iris code should be sought by an iterative algorithm with each of its bits updated in every iteration, until the objective function in Eqn. 12 converges to a minimum. To improve the efficiency of the iterative algorithm, we write the solution of Eqn.12 into a vector form as follows. Let $\boldsymbol{b}^t = [b_1^t, b_2^t, ..., b_n^t]^{\mathrm{T}} \in \mathbb{R}^n$ be the iris code at the t^{th} iteration. We define $\widetilde{\boldsymbol{b}}^t = \|\boldsymbol{b}^t\|_2^2 - \boldsymbol{b}^t$ (note that the i^{th} element of $\widetilde{\boldsymbol{b}}^t$ is equal to $\sum_{j \neq i} b_j^t$). Then, $\boldsymbol{b}^{t+1}$ can be calculated based on the sign of each element in $\boldsymbol{f} - \beta\widetilde{\boldsymbol{b}}^t$, similarly to Eqn. 14. At the beginning of the iterative algorithm, the iris code is initialized based on the sign of $\boldsymbol{f}$ (*i.e.* the same as traditional iris code). The whole algorithm is summarized in Algorithm. 1.

We show an example in Fig. 1 to illustrate the process of iris code production using Algorithm. 1. In this figure, we use a 5×5 patch as an example. This patch is extracted from the real process of iris code production for an unwrapped iris image using additional objective 2 (Algorithm. 1). We show the relevant results on this patch in each iteration during the process of iris code production.

In Fig. 1, each column corresponds to one iteration of Algorithm. 1, and the first column corresponds to the initialization stage. Within each column of Fig. 1, we order the sub-figures so that the result in a sub-figure depends on the result in its above sub-figure; the result in the top sub-figure of a column depends on the result in the bottom sub-figure of the previous column. In all, in Fig. 1, the flow of the algorithm is from top to bottom, from left to right.

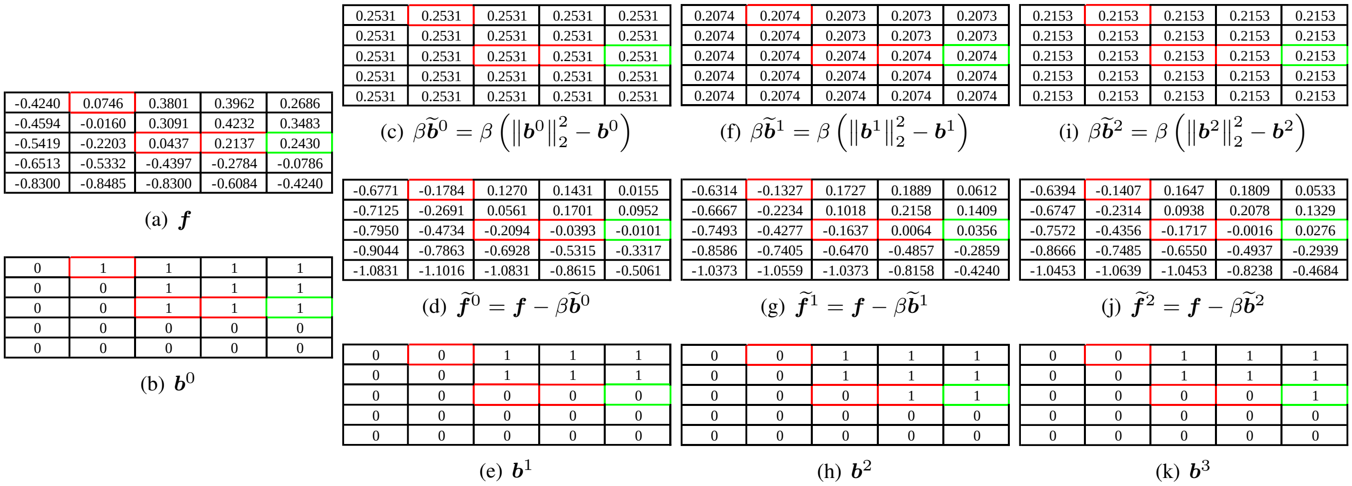
```
Algorithm 2: Combining the two objective terms
Input  : vector of feature values $\boldsymbol{f}$; parameter $\alpha$, $\beta$
Initialization: obtaining $\boldsymbol{b}^0$ by binarizing $\boldsymbol{f}$ based on the sign of its elements ( Eqn. 2); $t = 0$;
while true do
    $\widetilde{\boldsymbol{b}}^t = \|\boldsymbol{b}^t\|_2^2 - \boldsymbol{b}^t$;
    $\widetilde{\boldsymbol{f}}^t = \boldsymbol{f} - \beta\widetilde{\boldsymbol{b}}^t$;
    obtain $\boldsymbol{b}^{t+1}$ by binarizing $\widetilde{\boldsymbol{f}}^t$ based on the sign of its element;
    if convergence then
        compute $\boldsymbol{b}^* \in \{0, 1\}^n$ by solving Eqn. 10 with $\widetilde{\boldsymbol{f}}^t$ as the vector of feature values;
        break;
    end
    $t \leftarrow t + 1$;
end
Output: $\boldsymbol{b}^*$
```

It can be seen that, in the first iteration (the second column from the left in Fig. 1), the produced iris code ($\boldsymbol{b}^1$ in Fig. 1(e)) includes more zeros than the initial iris code ($\boldsymbol{b}^0$ in Fig. 1(b)). This is because a shift ($\beta\widetilde{\boldsymbol{b}}^0$ in Fig. 1(c)) is applied to the initial feature values ($\boldsymbol{f}$ in Fig. 1(a)), and the shifted feature ($\widetilde{\boldsymbol{f}}^0$ in Fig. 1(d)) includes more negative values.

Then, in the second iteration (the third column from the left in Fig. 1), the shift for feature values ($\beta\widetilde{\boldsymbol{b}}^1$ in Fig. 1(f)) is generally lower than the shift in the previous iteration ($\beta\widetilde{\boldsymbol{b}}^0$ in Fig. 1(c)). The reason is that the shift is calculated based on the iris code in the previous iteration; since $\boldsymbol{b}^1$ includes more zeros than $\boldsymbol{b}^0$, $\beta\widetilde{\boldsymbol{b}}^1$ is lower than $\beta\widetilde{\boldsymbol{b}}^0$. The lower shift leads that the shifted features in this iteration ($\widetilde{\boldsymbol{f}}^1$ in Fig. 1(g)) include less negative values than that in the previous iteration ($\widetilde{\boldsymbol{f}}^0$ in Fig. 1(d)). Therefore, the obtained iris code in this iteration ($\boldsymbol{b}^2$ in Fig. 1(h)) includes less zeros than that in the previous iteration ($\boldsymbol{b}^1$ in Fig. 1(e)).

Finally, in the third iteration (last column from the left), the shift for feature values ($\beta\widetilde{\boldsymbol{b}}^2$ in Fig. 1(i)) becomes larger than the shift in the previous iteration ($\beta\widetilde{\boldsymbol{b}}^1$ in Fig. 1(f)), because $\boldsymbol{b}^2$ (Fig. 1(h)) contains less zeros than $\boldsymbol{b}^1$ (Fig. 1(e)). It leads that the produced iris code in this iteration ($\boldsymbol{b}^3$ in Fig. 1(k)) has more zeros than that in the previous iteration ($\boldsymbol{b}^2$ in Fig. 1(h)). In our experiment, the iris code at this 5×5 patch does not change after the third iteration, while the iris code of the whole image does not change after the seventh iteration.

From the above example, we can see that the Algorithm. 1 iterates between smaller and larger shifts of the feature values (shown in Fig. 1(c), Fig. 1(f) and Fig. 1(i)) to produce iris codes with less or more zeros (shown in Fig. 1(e), Fig. 1(h) and Fig. 1(k)). In this case convergence means that, in an iteration, a shift for each dimension is found, such that, in the next iteration, another shift larger or lower than this shift for each dimension does not change the obtained iris code. This shift essentially defines the separation points of the feature values corresponding to less reliable and reliable bits: a bit is considered less reliable if the corresponding feature value is positive but its distance to the origin is lower than this



-0.4240	0.0746	0.3801	0.3962	0.2686
-0.4594	-0.0160	0.3091	0.4232	0.3483
-0.5419	-0.2203	0.0437	0.2137	0.2430
-0.6513	-0.5332	-0.4397	-0.2784	-0.0786
-0.8300	-0.8485	-0.8300	-0.6084	-0.4240

(a) $\boldsymbol{f}$

0	1	1	1	1
0	0	1	1	1
0	0	1	1	1
0	0	0	0	0
0	0	0	0	0

(b) $\boldsymbol{b}^0$

0.2531	0.2531	0.2531	0.2531	0.2531
0.2531	0.2531	0.2531	0.2531	0.2531
0.2531	0.2531	0.2531	0.2531	0.2531
0.2531	0.2531	0.2531	0.2531	0.2531
0.2531	0.2531	0.2531	0.2531	0.2531

(c) $\beta\widetilde{\boldsymbol{b}}^0 = \beta\left(\left\|\boldsymbol{b}^0\right\|_2^2 - \boldsymbol{b}^0\right)$

-0.6771	-0.1784	0.1270	0.1431	0.0155
-0.7125	-0.2691	0.0561	0.1701	0.0952
-0.7950	-0.4734	-0.2094	-0.0393	-0.0101
-0.9044	-0.7863	-0.6928	-0.5315	-0.3317
-1.0831	-1.1016	-1.0831	-0.8615	-0.5061

(d) $\widetilde{\boldsymbol{f}}^0 = \boldsymbol{f} - \beta\widetilde{\boldsymbol{b}}^0$

0	0	1	1	1
0	0	1	1	1
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0

(e) $\boldsymbol{b}^1$

0.2074	0.2074	0.2073	0.2073	0.2073
0.2074	0.2074	0.2073	0.2073	0.2073
0.2074	0.2074	0.2074	0.2074	0.2074
0.2074	0.2074	0.2074	0.2074	0.2074
0.2074	0.2074	0.2074	0.2074	0.2074

(f) $\beta\widetilde{\boldsymbol{b}}^1 = \beta\left(\left\|\boldsymbol{b}^1\right\|_2^2 - \boldsymbol{b}^1\right)$

-0.6314	-0.1327	0.1727	0.1889	0.0612
-0.6667	-0.2234	0.1018	0.2158	0.1409
-0.7493	-0.4277	-0.1637	0.0064	0.0356
-0.8586	-0.7405	-0.6470	-0.4857	-0.2859
-1.0373	-1.0559	-1.0373	-0.8158	-0.4240

(g) $\widetilde{\boldsymbol{f}}^1 = \boldsymbol{f} - \beta\widetilde{\boldsymbol{b}}^1$

0	0	1	1	1
0	0	1	1	1
0	0	0	1	1
0	0	0	0	0
0	0	0	0	0

(h) $\boldsymbol{b}^2$

0.2153	0.2153	0.2153	0.2153	0.2153
0.2153	0.2153	0.2153	0.2153	0.2153
0.2153	0.2153	0.2153	0.2153	0.2153
0.2153	0.2153	0.2153	0.2153	0.2153
0.2153	0.2153	0.2153	0.2153	0.2153

(i) $\beta\widetilde{\boldsymbol{b}}^2 = \beta\left(\left\|\boldsymbol{b}^2\right\|_2^2 - \boldsymbol{b}^2\right)$

-0.6394	-0.1407	0.1647	0.1809	0.0533
-0.6747	-0.2314	0.0938	0.2078	0.1329
-0.7572	-0.4356	-0.1717	-0.0016	0.0276
-0.8666	-0.7485	-0.6550	-0.4937	-0.2939
-1.0453	-1.0639	-1.0453	-0.8238	-0.4684

(j) $\widetilde{\boldsymbol{f}}^2 = \boldsymbol{f} - \beta\widetilde{\boldsymbol{b}}^2$

0	0	1	1	1
0	0	1	1	1
0	0	0	0	1
0	0	0	0	0
0	0	0	0	0

(k) $\boldsymbol{b}^3$

Fig. 1: An example of iris code production using additional objective 2 (Algorithm. 1). This example is based on a 5×5 patch extracted from the real process of iris code production for an unwrapped iris image using additional objective 2 (Algorithm. 1). Each column represents an iteration, and the first column corresponds to the initialization stage. Within each column, the result in a sub-figure depends on the result in its above sub-figure; the result in the top sub-figure of a column depends on the result in the bottom sub-figure of the previous column. The overall flow of the algorithm in this figure is from top to bottom, from left to right. We use red color to mark the less reliable bits sought by the algorithm, and we use green color to mark a possible new less reliable bit introduced by the algorithm.

shift, and all such less reliable bits will be assigned uniform bit values of 0. Note that the algorithm only focuses on the less reliable bits with positive feature values and the reason is explained before.

Specifically, for the 5×5 patch in Fig. 1, the final separation points are defined by $\beta\widetilde{\boldsymbol{b}}^2$ in Fig. 1(i). We use red color to mark the less reliable bits corresponding to the final separation points in Fig. 1. Investigating the original feature values in $\boldsymbol{f}$ in Fig. 1(a), we can see that the feature values marked red (corresponding to less reliable bits) are closer to the origin (0) than the other positive feature values, and there is a generally large difference between the feature values marked red and the other positive feature values.

Also, we find that the Algorithm. 1 to solve Eqn. 12 may introduce new less reliable bits, since some other positive feature values may be shifted to be near to the origin. One possible new less reliable bit is mark green in Fig. 1. As shown in Fig. 1(j), the shifted feature value of this bit is close to the origin in the last iteration. However, in our experiment, we find that, although additional objective term 2 may introduce new less reliable bits, it generally leads to an improved performance, either as an individual additional term (Eqn. 12) or in the proposed combined scheme of the two terms (please see the experimental results in section III-B). We think a possible reason is that, the positive feature values above the separation points found by solving Equ. 12 (*i.e.* using additional objective term 2) are inherently more reliable.

Finally, the convergence of Algorithm. 1 depends on the parameter β. The algorithm will not converge if β is too large. We use the patch in Fig. 1 as an example. When β is a very large positive number, it is possible that the shift in the first iteration ($\beta\widetilde{\boldsymbol{b}}^0$ in Fig. 1(c)) is larger than all the feature values in $\boldsymbol{f}$ (Fig. 1(a)); it leads that the shifted feature values in Fig. 1(d) are all negative; this finally results a $\boldsymbol{b}^1$ with all zeros; then, in the next iteration, since $\boldsymbol{b}^1$ are all zeros, the shift in Fig. 1(f) will be all zeros as well; therefore, the shifted feature values in Fig. 1(g) will be the same as $\boldsymbol{f}$ in Fig. 1(a), and the produced iris code in Fig. 1(h)) will be the same as the initial iris code; the solution will alter between a vector of zeros and the traditional iris code in the following iterations. In our experiment, we find that, in most of the cases, the algorithm converges with a proper choice of β, while it does not converge with an improper β. Please see the experiment section on the setting of β in our experiment.

Combine the two objective terms. We propose a simple yet effective scheme to combine the two additional objective terms. Given a vector of feature values $\boldsymbol{f}$, we firstly use additional objective term 2 to mitigate the influence of less reliable bits, *i.e.* solving Eqn. 12 by Algorithm 1. Then, in the iteration where Algorithm 1 converges, instead of producing the iris code based on the sign of shifted feature values $\widetilde{\boldsymbol{f}}^t = \boldsymbol{f} - \beta\widetilde{\boldsymbol{b}}^t$ (see Algorithm 1), we calculate the final iris code by solving Eqn. 10 using $\widetilde{\boldsymbol{f}}^t$ as the input (*i.e.* using the elements of $\widetilde{\boldsymbol{f}}^t$ as the f_i in Eqn. 10). In other words, in the iteration where Algorithm 1 converges, we seek the final iris code by applying additional objective term 1 to the shifted feature values, in order to exploit the spatial relationship. This scheme essentially combines the effect of the two additional objective terms. First, it shifts the feature values to prompt the less reliable bits to be assigned uniform bit values using additional objective term 2. Then, given the shift feature values, it seeks the final iris code by exploiting the spatial relationship of the bits using additional objective term 1. The whole algorithm is summarized in Algorithm 2.

In Fig. 2, we show examples of (1) traditional iris code, (2) the iris code produced by additional objective term 1, (3) the iris code produced by additional objective term 2 and (4) the iris code produced using the combined scheme of the two additional objective terms. All the iris codes are obtained from the same unwrapped iris image. In the figures of the iris codes (2), (3) and (4), we mark the bits that are different from the traditional iris codes (*i.e.* the changed bits due to the additional objective terms). We use red colour to mark the bits being 1 in the traditional iris code but being 0 in the current iris code, and we use blue colour to mark the bits being 0 in the traditional iris code but being 1 in the current iris code.

It can be seen that, in the iris code produced by additional objective term 1 (Fig. 2(b)), the regions of 1 and 0 are more adjacent, compared to the traditional iris code. Some small regions of 1 or 0 surrounded by large regions with a different bit value in the traditional iris code are assigned the same bit value as their surrounding regions in the iris code produced by additional objective term 1, due to the exploited spatial dependency. In the iris code produced by additional objective term 2 (Fig. 2(c)), the value of some bits in the boundary between the regions of 1 and 0 changes to 0, in comparison to the traditional iris code. It is because the boundary between the regions of 1 and 0 is usually where the feature values change their signs; the feature values located at these boundaries are more likely to be close to the origin, so the corresponding bits are more likely to be less reliable. Finally, in the iris code produced by the combined scheme (Fig. 2(d)), we find the effect of both objective terms. Some bits located at the boundary between the regions of 1 and 0 change to 0 in comparison to the traditional iris code, while the spatial dependency of the bit values is exploited as well.

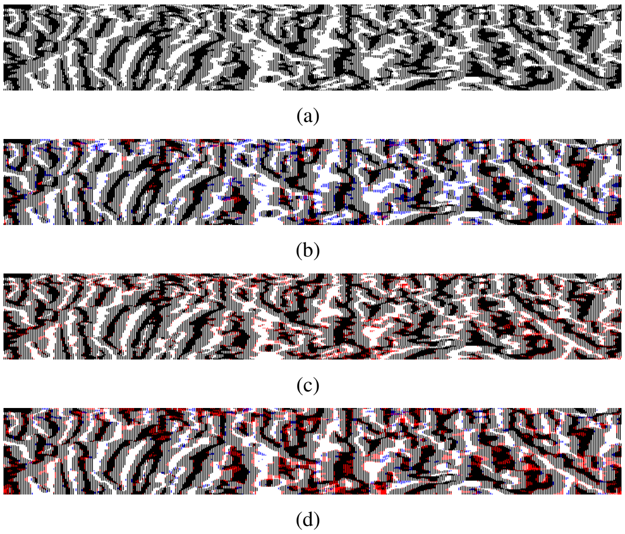
We note that an alternative method to combine the two additional objective terms is to apply them jointly as follows:

$$\underset{\substack{b_i\in\{0,1\}\\ i=1,2,\dots,n}}{\arg\min}\sum_i(-f_i)b_i+\alpha\sum_i\sum_{j\in N^i}|b_i-b_j|+\beta\sum_i\sum_{j\neq i}b_ib_j \tag{15}$$

We do not adopt this combination method due to three reasons. First, minimizing Eqn. 15 requires an iterative algorithm solving a first order MRF with binary labels in each iteration; it is computationally more expensive. Second, we experimentally find that, with Eqn. 15, it is difficult to seek a proper combination of the parameters α and β to achieve a good trade-off between the three terms. Third, we experimentally find that the proposed method of combination is sufficient to achieve an improved performance with less effort in parameter tuning and computation.

III. Experiment

In this section, we conduct experimental analysis to the proposed optimization method to produce iris codes. Firstly, we introduce our experimental setting, including datasets, parameter setting and performance evaluation (section III-A). Then, to study the effect of the proposed method for iris code production, we compare the performance between the traditional iris code and the iris codes produced by the proposed methods (section III-B). After that, we perform separate analysis on the proposed additional objective term 1 (section III-C) and additional objective term 2 (section III-D), aiming to gain a deeper insight into the two objective terms. Finally, we report the computational load of the proposed methods (section III-E).


(a)

(b)

(c)

(d)

Fig. 2: Examples of the iris codes produced by different methods. (a): Traditional iris code; (b): the iris code produced using additional objective term 1; (c): the iris code produced using additional objective term 2; (d): the iris code produced using the combined scheme of the two objective terms. In (b), (c) and (d), the colours mark the changed bits in comparison to the traditional iris code. The red colour marks the bits being 1 in the traditional iris code but being 0 in the current iris code; the blue colour marks the bits being 0 in the traditional iris code but being 1 in the current iris code.

A. Experimental setting

Datasets. We conduct the experiments on four benchmark datasets: CASIA.v4 thousand [37], ND-iris-0405 [38], CAISA.v4 distance [37] and UBIRIS.v2 [39]. The four datasets cover iris data with varying qualities. We show example images of the four datasets in Fig. 3, and we report the information of the data used in our experiment in Tab. I.

CASIA.v4 thousand (also referred as CASIAT in this paper) is a large scale dataset consisting of 2000 eyes from 1000 subjects, captured in near-infrared (NIR) wavelength and a close distance. The dataset includes $20,000$ images captured by a commercial IKEMB-100 camera. The images in this dataset have a high overall quality, despite the influence of glasses and specular reflections in some images. In our experiment, we use CASIA.v4 thousand dataset to represent the iris captures with a satisfactory quality for iris recognition. Correspondingly, we eliminate the images where the iris segmentation fails (we will introduce details on iris segmentation later), since such images usually have a higher noise level which causes the iris segmentation failure.

ND-iris-0405 (also referred as ND0405 in this paper) is a large-scale dataset captured in NIR wavelength and at a close distance. The current version of this dataset includes $64,982$ iris images from 712 eyes, captured by a LG2200

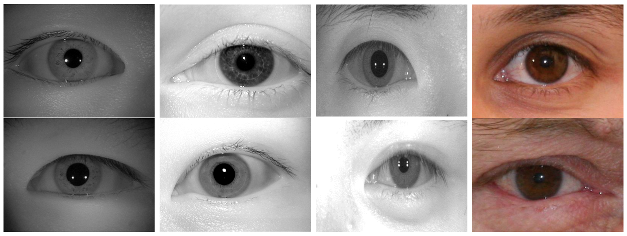


Fig. 3: Examples of the images in all the datasets used in our experiment. (a): CASIA.v4 thousand; (b): ND-iris-0405; (c): CASIA.v4 distance; (d): UBIRIS.v2.

TABLE I: Information on the data used in our experiment.

Dataset	Eyes	Images	Wavelength	Image quality
CASIAT	1790	12108	NIR	Highest
ND0405	712	5613	NIR	Higher
CASIAD	284	5037	NIR	Lower
UBIRIS2	171	1000	Visible	Lowest

TABLE II: Parameter setting on each dataset; recall that n is the number of bits in an iris code

Dataset	α	β	wavelength	sigmaOnf
CASIAT	1.2	$1/2n$	22	0.48
ND0405	1.2	$1/2n$	35	0.49
CASIAD	1.2	$1/2n$	19	0.48
UBIRIS2	0.3	$1/2n$	40	0.38

iris imaging system. Many real-world conditions occur in this dataset, leading to degradations in the captures, including blurring, eyelids occlusion, specular reflection, rotation, off-angle, *etc*. Also, some subjects wear soft contact lenses which cause distortion on iris textures. In our experiment, we use ND-iris-0405 dataset to represent the iris data of a relatively higher quality, with good resolution and clear iris texture, but suffering from some noise and degradations due to real-world conditions. We use a subset of ND-iris-0405 consisting of the first 8 images of each eye, and we eliminate some images with too small eye region from the subset.

CASIA.v4 distance (also referred as CASIAD in this paper) is a dataset captured at a distance under NIR wavelength. It includes both eyes from 142 subjects. The stand-off distance is around 3 meters. The images include most of facial features and patterns. The eye region of most images suffers from blinking, eyelids occlusion, specular reflection and motion blur. In our experiment, we use CASIA.v4 distance dataset to represent the iris data with a relatively lower quality. The captures suffer from the information loss and noise due to the distant capture. We conduct the experiment on a subset of CASIA.v4 distance dataset constructed as follows. We employ all the images from both left and right eyes of all the subjects in our experiment. We detect left and right eye regions from the images using classical Viola-Jones object detector [40]. We correct false detections manually, and we eliminate some images with too small iris regions.

UBIRIS.v2 (also referred as UBIRIS2 in this paper) is a colour iris dataset captured with dynamic lighting conditions. The whole dataset consists of $11,102$ images from 261 subjects. The stand-off distance is between 3 and 8 meters. The images are influenced by specular reflection across the eye region, eyelids occlusion, off-angle and blurring. In our experiment, we use UBIRIS.v2 dataset to represent the iris captures with heavy noise, due to a combination of distant capture and colour wavelength. We conduct the experiment on a subset of UBIRIS.v2 with 1000 images from 171 eyes. This subset was released for NICE II contest [41]. For UBIRIS.v2, we operate on the illumination (Y) channel of YCbCr colour space.

The iris segmentation is performed as follows. For CASIA.v4 thousand dataset which represents the captures with a generally satisfactory quality, we apply Hough transform to the result of Canny edge detection to seek two circular boundaries for limbus and pupil. For this dataset, we manually eliminate the images with the segmentation failure in limbic or pupillary boundary. The reason is that the images with failure in the segmentation of the two boundaries usually have a higher noise level, and we eliminate them to keep the data representing high quality iris captures. The eyelids, shadow and reflections are detected using the algorithm in [42] for the remaining images in CASIA.v4 thousand dataset. For the other three noisy iris datasets, we employ the algorithm in [42] for iris segmentation, including the segmentation of limbus and pupil, and the detection of eyelids, reflection and shadow. The segmentation failures are corrected manually, so that most noisy captures are preserved in these three datasets, enabling them to represent iris captures with varying noise level. The size of unwrapped iris image is set to 100×360. We use 1-D log-Gabor filter [43] to produce the feature values given an unwrapped iris image.

We set the gallery and probe images on each dataset as follows. For CASIA.v4 thousand dataset, we use the first image of each eye as gallery image, and we use the remaining images as probe images. For ND-iris-0405 dataset, we use the first 2 images of each eye as gallery images, and we use the rest of the images as probes. For CASIA.v4 distance and UBIRIS.v2 datasets, we use the first 5 images of each eye as gallery images, and we use the rest of the images as probe images. The number of gallery images is chosen based on the image quality of each dataset. We use more gallery images for the datasets with lower quality.

Parameter tuning and performance evaluation. We use exclusively separated data for parameter setting and performance evaluation. For CASIA.v4 thousand dataset, we use the first 50 eyes to tune the parameter, and we use the remaining 1740 eyes for performance evaluation. For ND-iris-0405 dataset, we use the first 20 eyes to tune the parameters, and we use the remaining 692 eyes to evaluate the performance. For CASIA.v4 distance dataset, we use the first 20 eyes to tune the parameters, and we use the remaining 264 eyes to evaluate the performance. For UBIRIS.v2 dataset, we use the first 19 eyes to tune the parameters, and we use the remaining 152 eyes to evaluate the performance.

There are 4 parameters to be determined in the proposed method of iris code production: α is the parameter in our method based on additional objective term 1 (Eqn. 10), β is the parameter in our method based on additional objective term 2 (Eqn. 12), wavelength and sigmaOnf are 1-D log-Gabor parameters. Based on the above described data for parameter setting, we set these parameters as reported in Tab. II. We find that for additional objective term 2, $\beta = 1/2n$ achieves a generally good convergence and performance on all the datasets. As for additional objective term 1, the α on colour dataset (UBIRIS.v2) is lower than that on NIR datasets. We think the reason is that colour iris images contain a relatively higher amount of noise, and using a small α on colour data is able to prevent the bit-adjacency term incorrectly spreading the influence of noise bits in an iris code.

We evaluate the performance in two tasks: identification and verification. The identification performance is evaluated by cumulative match characteristic (CMC), while the verification performance is evaluated by receiver operating characteristic (ROC).

B. Comparison with traditional iris codes

In this subsection, we make comparisons between the traditional iris code and the iris code produced by the proposed methods. Specifically, we compare the performance of the following four iris codes: traditional iris codes which binarize feature values based on their signs as baseline (referred as Baseline); the proposed iris code using additional objective term 1 (referred as Proposed_OT1); the proposed iris code using additional objective term 2 (referred as Proposed_OT2); the proposed iris code using the combined scheme of the two additional objective terms (referred as Proposed_Comb). We show the CMC and ROC curves of all the comparison methods on all the datasets in Fig. 4.

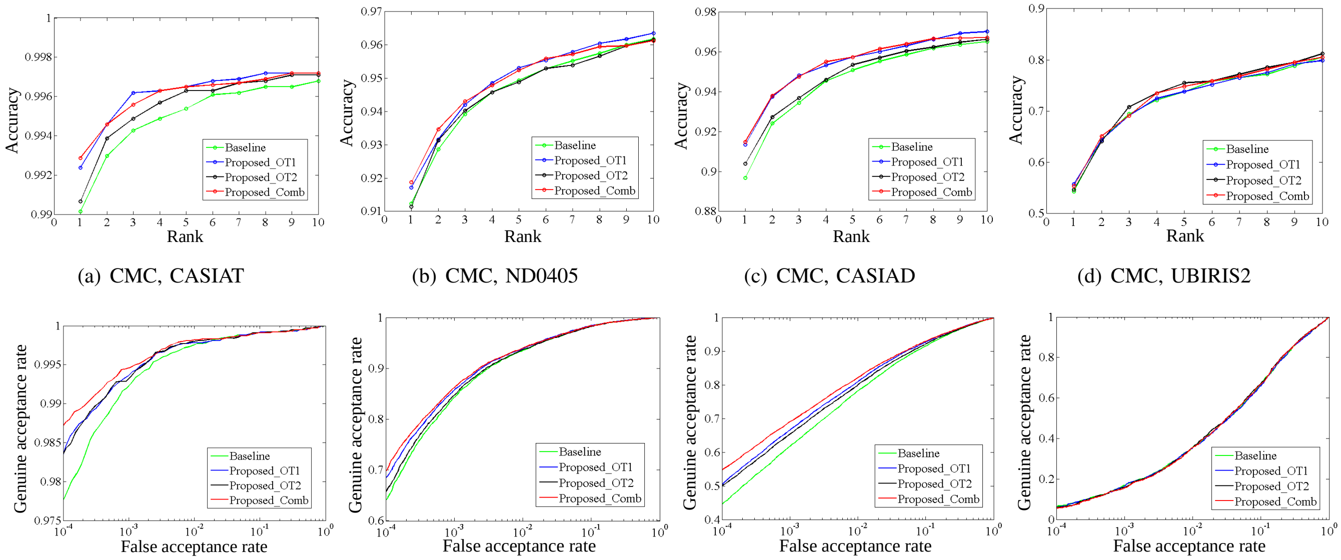
It can be seen that the performance varies on NIR datasets (CASIAT, ND0405, CASIAD) and colour dataset (UBIRIS2). Considering the result on NIR datasets, we find that Proposed_OT1 and Proposed_OT2 generally achieve better CMC and ROC performance, in comparison to Baseline. This result demonstrates that we can obtain more effective iris codes by modeling the spatial relationship of the bits in iris codes (Proposed_OT1) or mitigating the influence of less reliable bits in iris codes (Proposed_OT2), using additional objective terms. Furthermore, we find that Proposed_Comb generally achieves the best rank 1 recognition accuracy in the CMC curves and the best ROC curve, in comparison to the other methods. This result shows that, by combining the two additional objective terms using our scheme, we are able to obtain more effective iris codes, from the same feature vector, than using each individual additional objective term, or using no additional objective terms (*i.e.* Baseline). Especially, Proposed_Comb achieves a generally improved performance compared to Proposed_OT1. The difference between the Proposed_Comb and Proposed_OT1 is that, Proposed_Comb computes an iris code by applying additional objective term 1 to the processed feature values using additional objective term 2 (*i.e.* the feature values are shifted by solving Eqn. 12), while Proposed_OT2 computes an iris code by directly applying additional objective term 1 to the original feature values. In other words, with the same objective term of spatial relationships (additional objective term 1), the feature values obtained based on additional objective term 2 perform better than the original feature values. This observation validates the effectiveness of the proposed additional objective term 2 in handling less reliable bits from another aspect.

The only exception on NIR datasets is that, on ND0405, Proposed_OT2 has a lower rank 1 recognition accuracy than Baseline (see Fig. 4(b)). However, on ND0405, Proposed_Comb still achieves a better rank 1 recognition accuracy than Proposed_OT1. In other words, although additional objective term 2 has a lower rank 1 recognition accuracy than Baseline on ND0405, applying additional objective term 1 to the feature values produced by additional objective term 2 still leads to a better performance than applying additional objective term 1 directly to the original feature values. This result supports the effectiveness of additional objective term 2. Also, Proposed_OT2 still has an improved ROC curve compared to Baseline on ND0405 (see Fig. 4(f)).

On the other hand, for the colour dataset (UBIRIS.v2), we find that applying the additional objective terms leads to little improvement compared to Baseline. We think a reason is that the colour captures in this dataset include too heavy noise. The heavy noise induces a large amount of highly unreliable feature values. Consequently, the spatial relationship exploited based on these feature values is less reliable (Proposed_OT1), and the performance is also less likely to improve by only handling the feature values close to the origin (Proposed_OT2). Therefore, adding these objective terms leads to similar performance to Baseline.

Since the above results show that the noise in iris captures may influence the performance of the proposed iris code, we conduct an experiment to further explore the influence of noise on the proposed method. This experiment is based on the iris matching result on four low quality subsets of CASIAD dataset. Specifically, given the probe set of CASIAD as described in section III-A, we select four low quality subsets from this probe set. These four low quality subsets correspond to the images with low quality in four quality measures: focus, motion, off-angle and occlusion, respectively (we calculate these quality measures using the method in [44], [45]). Each low quality subset includes the images with the lowest 10% quality in the probe set, corresponding to one of the four quality measures above. We perform iris matching between the full gallery set of CASIAD and each of the four low quality subsets, and we investigate the performance of different iris codes on different low quality subsets. Note that we only perform quality selection for the probe set and we preserve the full gallery set. It simulates the scenario that, in real applications, the gallery set usually remains unchanged after registration phase, and the variation of noise types and noise levels mainly exist in the probes.

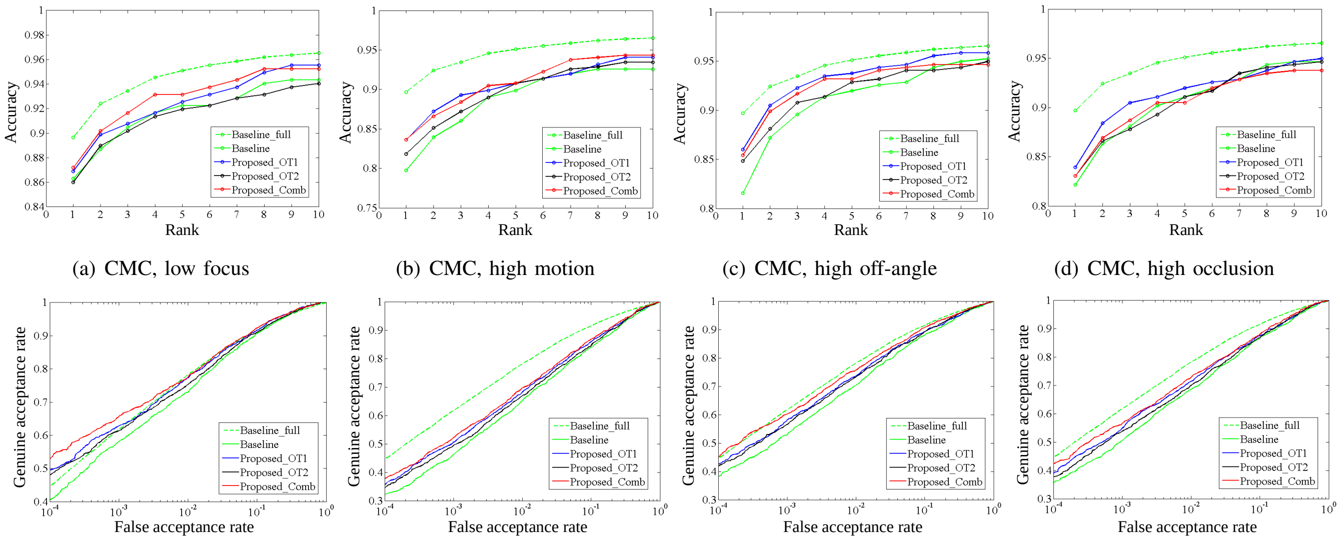
In Fig. 5, we show the CMC and ROC curves of different iris codes on each low quality subset. Following our previous notations, we use Baseline to refer to the performance of traditional iris code, and we use Proposed_OT1, Proposed_OT2, Proposed_Comb to refer to the performance the proposed iris code using additional objective term 1, additional objective term 2 and the combined scheme, respectively. Moreover, in each subfigure, we include the performance of the traditional



(a) CMC, CASIAT (b) CMC, ND0405 (c) CMC, CASIAD (d) CMC, UBIRIS2

(e) ROC, CASIAT (f) ROC, ND0405 (g) ROC, CASIAD (h) ROC, UBIRIS2

Fig. 4: Comparison of different iris codes.



(a) CMC, low focus (b) CMC, high motion (c) CMC, high off-angle (d) CMC, high occlusion

(e) ROC, low focus (f) ROC, high motion (g) ROC, high off-angle (h) ROC, high occlusion

Fig. 5: CMC and ROC curves on CASIAD subsets with low quality.

iris code on the full gallery and probe sets of CASIAD (referred as Baseline_full in Fig. 5). We use Baseline_full as a reference performance of the iris codes on general data in CASIAD without specific selection based on image quality.

We have several observations from the result in Fig. 5. First, Baseline performs worse than Baseline_full on all the four low quality subsets. This is less surprising since the probe set used by Baseline has a significantly lower quality than that used by Baseline_full: each low quality subset includes the data with the lowest 10% quality corresponding to a quality measure in the full probe set.

Second, the noise in each low quality subset is not high enough to influence the verification performance (ROC) of the proposed iris codes. It can be seen that the ROC curves in Fig. 5 is consistent with that in Fig. 4(g): Proposed_OT1 and Proposed_OT2 perform better than Baseline, and Proposed_Comb performs the best among all the iris codes.

Third, the noise in some low quality subsets is high enough to influence the identification performance (CMC) of the Proposed_Comb. It can be seen that, in Fig. 5(b), Fig. 5(c) and Fig. 5(d), although Proposed_OT1 and Proposed_OT2 still achieve generally better performance than Baseline, Proposed_Comb show little improvement over Proposed_OT1 and Proposed_OT2. We think a possible reason is that the noise level in each subset is not high enough to influence the iris code produced by each individual additional objective term (Proposed_OT1 and Proposed_OT2), but the combined scheme accumulates the noise in each individual result, hence

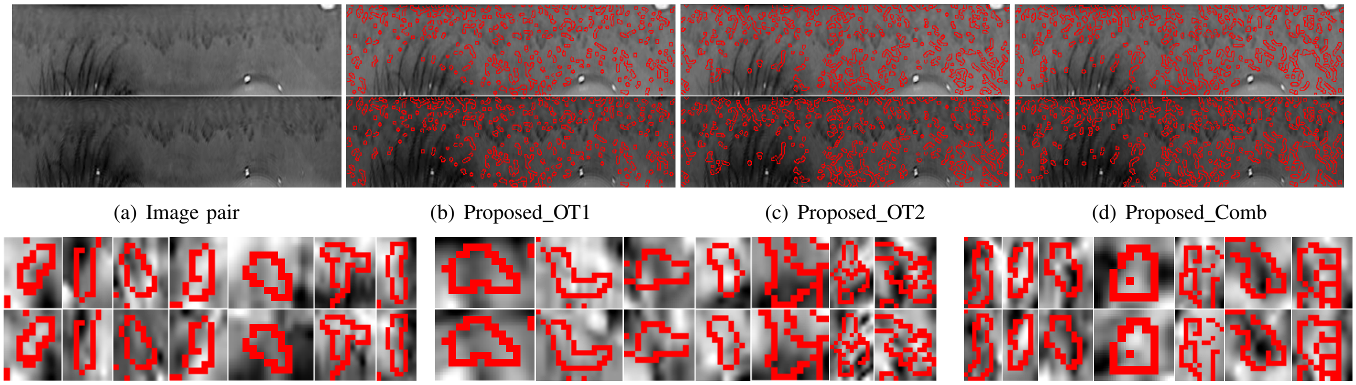

(a) Image pair
(b) Proposed_OT1
(c) Proposed_OT2
(d) Proposed_Comb
(e) Scaled examples, Proposed_OT1
(f) Scaled examples, Proposed_OT2
(g) Scaled examples, Proposed_Comb

Fig. 6: Effect of the proposed iris codes. (a): A pair of unwrapped iris images from the same eye in CASIA.v4 thousand dataset; (b)-(d): the regions that are falsely not matched in the traditional iris code but correctly matched in the iris code produced by Proposed_OT1, Proposed_OT2, Proposed_Comb, respectively; we use red color to mark the edge of such regions in each image; (e)-(g): upscaled examples of the regions in (b)-(d), respectively; each column in (e)-(g) is a pair of image patch at the same position in the two unwrapped images; note that the contrast of each patch in (e)-(g) is enhanced for better visual inspection.

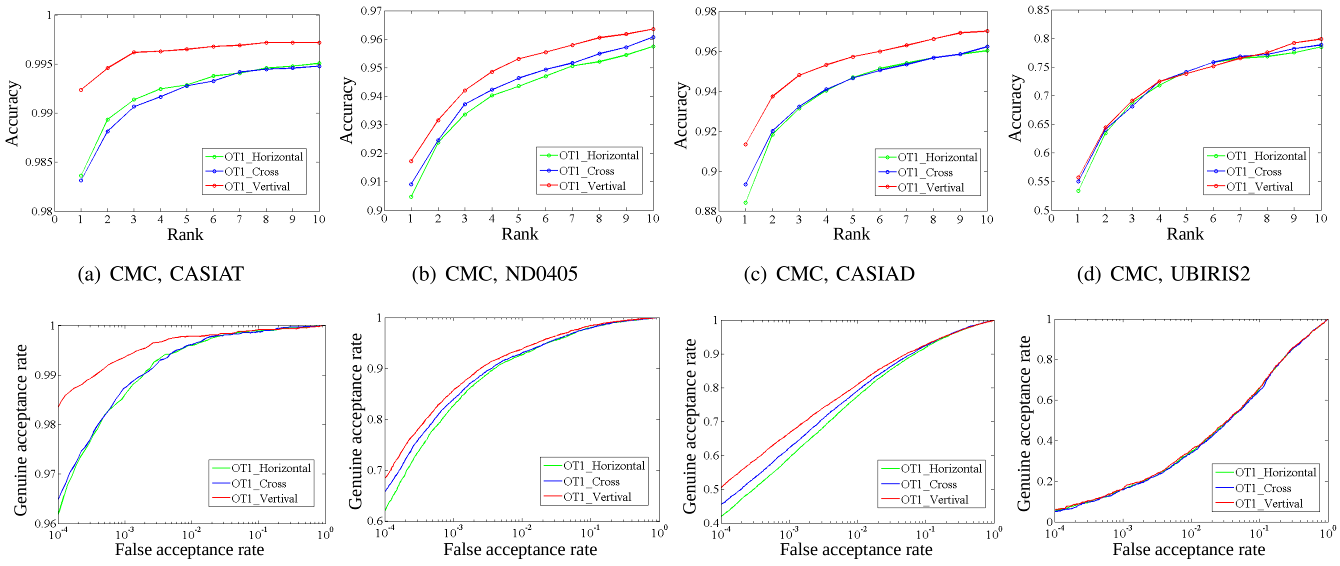
Proposed_Comb is influenced by the accumulated noise and its performance is impaired.

Investigating the performance in the figures with an order of Fig. 4(c)/Fig. 4(g), Fig. 5, Fig. 4(d)/Fig. 4(h), we can obtain an insight on how the noise influences the performance of the proposed iris code (note that we think the noise level is Fig. 4(c)/Fig. 4(g) < Fig. 5 < Fig. 4(d)/Fig. 4(h); it is consistent with the performance shown in these figures). Beginning with Fig. 4(c)/Fig. 4(g), the increasing of noise level will firstly harm the intra-class consistency of the proposed iris code, leading to an impaired CMC performance of Proposed_Comb as shown in Fig. 5. At the point of Fig. 5, we think that the noise influences little on the inter-class separation of the proposed iris code, since the ROC performance of each method is consistent between Fig. 5 and Fig. 4(g). It may be because of the inherent randomness of iris textures. Then, starting from Fig. 5, a further increasing on the noise level begins to influence the inter-class separation of the proposed iris code, degrading the ROC performance as shown in Fig. 4(h).

As the final part of this subsection, in Fig. 6, we show some examples to illustrate how the proposed method improves the iris matching result, compared to the traditional iris code. In Fig. 6(a), we show a pair of unwrapped iris images from the same eye in CASIAT dataset. The two images have lighting variation (the bottom one is darker), and it leads to some variation in local textures. In Fig. 6(b), Fig. 6(c) and Fig. 6(d), using the real part of iris codes as examples, we mark the regions that are falsely not matched in the real part of traditional iris code but correctly matched in the real part of Proposed_OT1, Proposed_OT2 and Proposed_Comb, respectively. We use red color to mark the edge of such regions in each figure. The edge is detected by applying Canny edge detector to a binary image representing the pixels that are falsely not matched in the traditional iris code but correctly matched in the proposed iris codes. Furthermore, for better visual inspection, we scale some of the marked regions in Fig. 6(b), Fig. 6(c) and Fig. 6(d), and we show the scaled examples in Fig. 6(e), Fig. 6(f) and Fig. 6(g), respectively. Note that we enhance the contrast of the scaled region to obtain a better visual effect.

From the example in Fig. 6(e), we find that Proposed_OT1 improves the matching of local textures with imaging variations in the two captures. It can be seen that, in Fig. 6(e), the texture within the marked region has imaging variation between the patches in each pair. For example, in the third column from right in Fig. 6(e), the marked region in the bottom patch is darker and has lower contrast, compared to the marked region in the top patch. We think such imaging variation on local textures causes false non-matching in the traditional iris code. However, the marked regions in each patch pair are correctly matched using Proposed_OT1. We think this is because Proposed_OT1 models the spatial relationship in iris codes: it promotes the bits along the same texture (*i.e.* with a spatial relationship) to be adjacent, and this mitigates the change of bit value due to the imaging variation on local textures.

Investigating the example in Fig. 6(f), we think that Proposed_OT2 enhances the matching of regions corresponding to less reliable bits under imaging variations. Note that the marked regions in Fig. 6(f) correspond to less reliable bits in iris codes. This is because the matching result of these regions is different between the traditional iris code and Proposed_OT2 (*i.e.* the corresponding bit in the traditional iris code is changed in Proposed_OT2), and Proposed_OT2 only changes the less reliable bits in the traditional iris code. It can be seen that the marked regions in each patch pair in Fig. 6(f) are similar but with some variations. We think such imaging variation in the marked regions causes the flipping of less reliable bits (recall that a less reliable bit is more likely to flip between 0 and 1 under the influence of noise). The flipping of less reliable bits leads to inconsistent bits in the traditional iris code between the marked regions in each



(a) CMC, CASIAT

(b) CMC, ND0405

(c) CMC, CASIAD

(d) CMC, UBIRIS2

(e) ROC, CASIAT

(f) ROC, ND0405

(g) ROC, CASIAD

(h) ROC, UBIRIS2

Fig. 7: The experimental results to analyse the spatial relationship of the bits in iris codes.

patch pair, hence resulting false non-matching. In contrast, Proposed_OT2 assigns uniform bit values of 0 to less reliable bits to suppress their flipping, so the marked regions in each patch pair are correctly matched using Proposed_OT2.

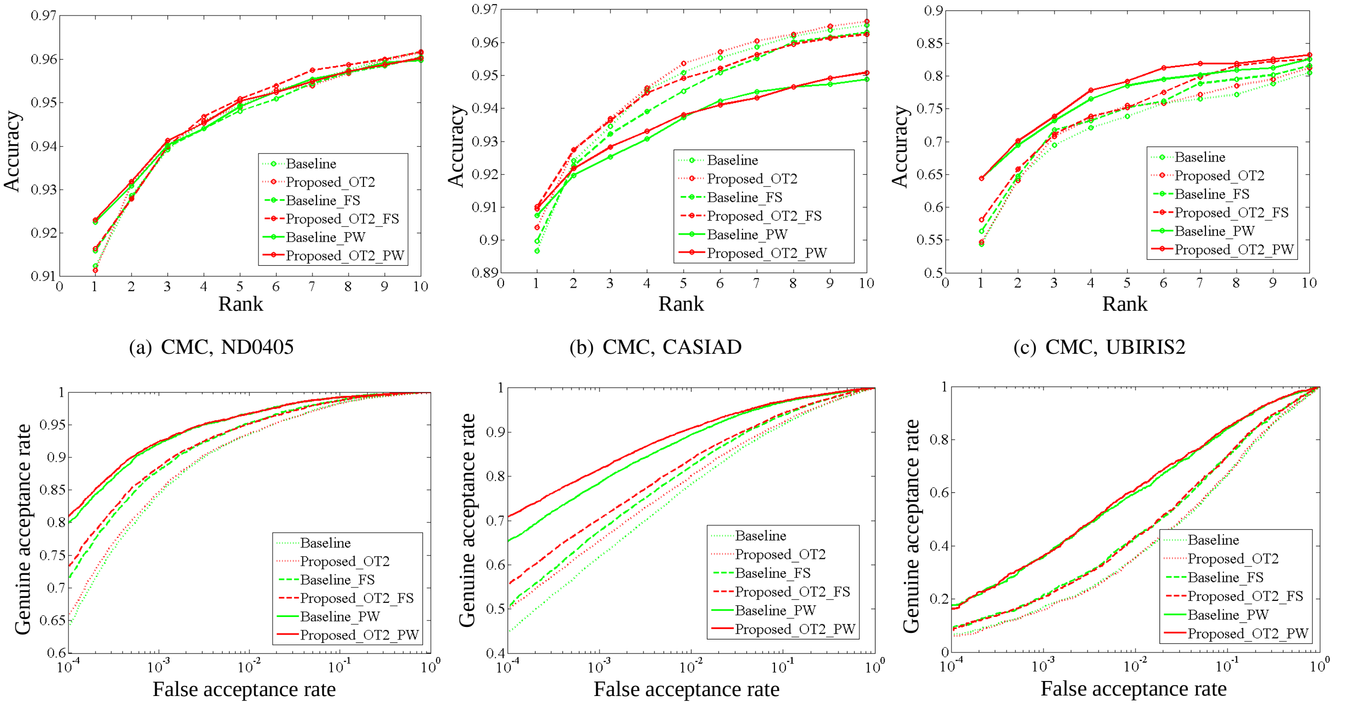
Studying the example in Fig. 6(g), we think that Proposed_Comb may achieve both the above mentioned effects. It improves the matching result of local textures with imaging variations, and it also enhances the matching result of regions corresponding to less reliable bits in iris codes. In our opinion, the left 3 patch pairs in Fig. 6(g) may correspond to the former case, since the corrected regions in these patch pairs are mainly thin and nearly vertical regions that are more likely to correspond to the local texture. Accordingly, the remaining 4 patch pairs are possible to correspond to the latter case.

In all, our experimental results in this subsection show that: (1) when the iris data is less noisy, the iris code produced by the proposed method is able to achieve generally improved identification and verification performance, compared to the traditional iris code; (2) when the iris data is less noisy, the combined scheme of the two additional objective terms leads to a generally improved performance, compared to each individual objective term; (3) in our experiment, the highest noise level where (1) and (2) apply is distant NIR captures; (4) for the iris data with heavier noise (colour captures in our experiment), the iris code produced by the proposed method performs similarly to the traditional iris code; (5) the increasing of noise level firstly influences the intra-class consistency of the proposed iris codes, and the further increasing of noise level begins to impair the inter-class separation.

C. Analysis of the spatial relationship of the bits in iris codes

The radial correlations of iris textures have been demonstrated in researches such as [1], [34]. In this subsection, we study this spatial relationship in iris texture from the point of view of iris code optimization, using the proposed additional objective term 1. We perform the analysis by varying the neighbor type in Eqn. 10, and investigating the performance of the obtained iris codes using additional objective term 1 with different neighbor types. Intuitively, the best performance will be achieved by a neighbor type that is closest to the inherent spatial relationship of iris textures. The reason is that such neighbour type correctly models the inherent spatial dependency in iris textures. In contrast, a neighbor type that is different from the inherent spatial relationship will introduce an incorrect dependency between the bits that should be independent; it will harm the randomness of iris codes, leading to a degraded performance. In other words, only by using a neighbor type which is closest to the inherent spatial dependency of iris textures we can obtain the iris code with the best performance, since it expresses the inherent spatial relationship in iris textures without influencing the inherent randomness part of iris codes. We test three basic neighbor types, vertical neighbor (including 1 top pixel and 1 bottom pixel of the centre pixel, referred as OT1_Vertical), horizontal neighbor (including 1 left pixel and 1 right pixel of the centre pixel, referred as OT1_Horizontal) and cross neighbor (combination of vertical and horizontal neighbors, referred as OT1_Cross).

We show the CMC and ROC curves of the iris codes produced by additional objective term 1 with the three neighbor types on all the datasets in Fig. 7. We find that, on NIR datasets, the iris code obtained with the vertical neighbor consistently outperforms the iris code obtained using the other two neighbor types. As for the colour dataset, although the three neighbor types have similar ROC curves, the iris code with the vertical neighbor achieves a slightly better rank 1 recognition accuracy in the CMC curve. In all, we can conclude that, with the additional objective term 1, the iris code produced using the vertical neighbor generally achieves the best performance. This observation experimentally illustrates that iris texture



(a) CMC, ND0405

(b) CMC, CASIAD

(c) CMC, UBIRIS2

(d) ROC, ND0405

(e) ROC, CASIAD

(f) ROC, UBIRIS2

Fig. 8: The experimental results to analyse the proposed additional objective term 2.

has a spatial dependency along the radial direction, from the perspective of iris code optimization. It is consistent with the demonstrations in previous researches such as [1], [34].

Finally, we find that the iris code with cross neighbor performs generally better than the iris code with horizontal neighbor on all the datasets. We think it is because the cross neighbor includes the information on vertical relationship, while the horizontal neighbor does not.

D. Analysis on the effect of additional objective term 2

The proposed additional objective term 2 aims to handle less reliable bits from the perspective of iris code optimization. However, in the existing researches, the similar aim can be also achieved by iris weight map methods, such as [23]–[26]. The iris weight map methods assign more weight to more reliable bits in iris matching stage to emphasize more reliable bits and suppress less reliable bits. Since both the proposed additional objective term 2 and iris weight map methods are designed for the same aim, it is necessary to study their comparative performance.

We choose two representative iris weight map methods to perform this analysis. The first method is the personalized weight map in [24]. It represents a general iris weight map method with real value weights. The second method is the fisher feature selection in [46]. It represents a special case of iris weight map methods: the iris weight map with binary weights. The result of feature selection can be viewed as an iris weight map with binary weights: a bit is assigned a weight of 1 if this bit is selected, otherwise the weight of this bit is 0.

In this analysis, we investigate the performance of the following six methods: the traditional iris code (Baseline), the iris code produced by additional objective term 2 (Proposed_OT2), Baseline selected by the fisher feature selection for iris matching (Baseline_FS), Proposed_OT2 selected by the fisher feature selection for iris matching (Proposed_OT2_FS), Baseline with the personalized weight map used for iris matching (Baseline_PW), Proposed_OT2 with the personalized weight map used for iris matching (Proposed_OT2_PW). For each dataset, we calculate the iris weight map for each eye using the gallery images described in section III-A. With the iris weight maps, we perform iris matching based on a weighted Hamming distance. Please see [24] for more details.

We show the CMC and ROC curves of all the methods on ND-iris-0405, CASIA.v4 distance and UBIRIS.v2 datasets in Fig. 8. We do not show the results on CASIA.v4 thousand dataset, because the performance of Baseline_PW and Proposed_OT2_PW are exactly the same as that of Baseline and Proposed_OT2 respectively on this dataset, due to our single-image gallery setup. Also, the fisher feature selection is less applicable with the single-image gallery setup.

We have three main observations from the results in Fig. 8. First, Baseline_FS and Baseline_PW perform generally better than Proposed_OT2 on all the datasets. This result demonstrates that the proposed additional objective term 2 can not replace the iris weight map methods. A possible reason is that the iris weight map methods utilizes more information than the proposed additional objective term 2. Specifically, additional objective term 2 seeks the less reliable bits in each iris code separately, while the iris weight map methods compute the bit reliability using multiple iris codes in gallery. Also, for the non-binary iris weight map, the weighted Hamming distance used for iris matching is more effective than the general Hamming distance used by Proposed_OT2.

Second, Proposed_OT2_FS performs generally better than Baseline_FS, similarly for Proposed_OT2_PW and Baseline_PW. That is, applying the iris weight map methods to Proposed_OT2 performs better than directly applying the iris weight map methods to the traditional iris code. This observation suggests that the iris weight map method is applicable to the iris code produced by additional objective term 2, and the performance is better than directly applying the iris weight map method to the traditional iris code. This observation validates the effectiveness of additional objective term 2.

Third, on UBIRIS.v2 dataset, the performance is similar between: (1) Baseline_FS and Proposed_OT2_FS, (2) Baseline_PW and Proposed_OT2_PW. The reason is similar to the one analysed in section III-B: the high noise level in colour iris captures influences the performance of the proposed optimization method for iris code production. As a result, on this dataset, it makes little difference applying iris weight map methods to the iris code produced by each method.

Overall, the experimental analysis in this subsection demonstrates that, in terms of handling less reliable bits, the proposed method can be combined fruitfully with the iris weight map methods. The performance is generally better than directly applying the iris weight map methods to the traditional iris code. However, the proposed method with additional objective term 2 (Proposed_OT2) can not replace alone the iris weight map methods. This observation demonstrates the effectiveness of the proposed additional objective term 2 from the perspective of iris matching.

E. Analysis on the computational cost

The computational cost is a critical factor in real applications. It is expected that the proposed methods have higher computational cost than the traditional binarization method. This is because adding objective terms leads to a more complex algorithm than simple binarization, and it induces more computational load. In this subsection, we study the additional computational cost of the proposed methods. We report the computational cost of four methods: the traditional binarization method (Baseline), the proposed additional objective term 1 (Proposed_OT1), the proposed additional objective term 2 (Proposed_OT2) and the proposed combined scheme (Proposed_Comb). We estimate the computational cost by measuring the elapsed time to produce an iris code given a vector of feature values. The measure is based on a Matlab implementation of all the methods, running on a desktop with Intel i5-3470 quad-core 3.20GHz CPU, 16GB RAM, Windows 7 64bit system and Matlab 2013a 64bit. We report in Tab. III the mean elapsed time of all the methods to produce an iris code on all four datasets.

TABLE III: The computational cost of the comparison methods to produce an iris code (ms)

Method	CASIAT	ND0405	CASIAD	UBIRIS2
Baseline	0.60	0.56	0.56	0.53
Proposed_OT1	47.8	46.0	47.9	39.0
Proposed_OT2	9.50	7.70	8.90	6.30
Proposed_Comb	56.5	51.1	53.1	43.6

It can be seen that, as expected, the computational cost of the proposed methods are generally higher than the traditional binarization method. However, the highest elapsed time to compute an iris code using the proposed methods is 56.5ms in our experimental setting. We think this is acceptable in real applications, because the iris code production is performed only once for each iris capture, and a delay of around 55ms is practically very small.

Comparing between the computational cost of Proposed_OT1 and Proposed_OT2, we find that Proposed_OT1 leads to a generally higher computational cost. It means that solving the MRF in Eqn. 10 requires more computational effort than solving Eqn. 12. Also, Proposed_Comb has the highest computational cost, since it essentially solves both Eqn. 12 and Eqn. 10.

IV. Conclusion

In this paper, we investigate the problem of iris code production from the point of review of optimization. We demonstrate that the traditional iris code can be expressed as the solution of an optimization problem. Furthermore, we propose to apply additional terms to the objective function of this optimization problem to produce more effective iris codes. We investigate two additional objective terms, one exploits the spatial relationship of the bits in an iris code, and the other mitigates the influence of less reliable bits in iris codes. We also propose a scheme to combine the two additional objective terms. The experimental results on benchmark datasets demonstrate that the proposed method leads to a generally improved performance in comparison to the traditional iris code, and the computational cost is acceptable in real applications. Our experimental analysis also provides deeper insights into the proposed additional objective terms as well as the characteristics of iris codes. Future work may focus on: (1) designing more effective objective terms to produce iris codes; (2) examining the proposed method on more datasets covering more data variations, to investigate the stability and robustness of the proposed method to more varying data.

References

[1] J. Daugman, "High confidence visual recognition of persons by a test of statistical independence," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 15, pp. 1148–1161, 1993.

[2] ——, "How iris recognition works," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 14, pp. 21–30, 2004.

[3] R. Wildes, "Iris recognition: an emerging biometric technology," *Proc. IEEE*, vol. 85, pp. 1348–1363, 1997.

[4] K. Bowyer, K. Hollingsworth, and P. Flynn, "Image understanding for iris biometrics: A survey," *Image Vision Comput.*, vol. 110, pp. 281–307, 2008.

[5] Z. Sun and T. Tan, "Ordinal measures for iris recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, pp. 2211–2226, 2009.

[6] A. Kumar and A. Passi, "Comparison and combination of iris matchers for reliable personal authentication," *Pattern Recognit.*, vol. 43, pp. 1016–1026, 2010.

[7] J. Daugman and I. Malhas, "Iris recognition border-crossing system in the uae," *International Airport Review*, vol. 8, pp. 49–53, 2004.

[8] J. Daugman, “600 million citizens of India are now enrolled with biometric id,” *SPIE Newsroom*, 2014.

[9] C. Belcher and Y. Du, “Region-based SIFT approach to iris recognition,” *Opt. Laser Eng.*, vol. 47, pp. 139–147, 2009.

[10] Y. Du, C. Belcher, and Z. Zhou, “Scale invariant gabor descriptor-based noncooperative iris recognition,” *EURASIP J. Adv. Signal Process.*, vol. 2010, pp. 1–13, 2010.

[11] K. Yang and Y. Du, “A multi-stage approach for non-cooperative iris recognition,” *IEEE Int. Conf. Systems, Man and Cybernetics*, pp. 3386–3391, 2011.

[12] H. Mehrotra, B. Majhi, and P. Sa, “Unconstrained iris recognition using F-SHIFT,” *Int. Conf. Information, Communications and Signal Processing*, pp. 1–5, 2011.

[13] Y. Chen, Y. Liu, X. Zhu, F. He, H. Wang, and N. Deng, “Efficient iris recognition based on optimal subfeature selection and weighted subregion fusion,” *Scientific World J.*, vol. 2014, pp. 1–19, 2014.

[14] T. Yang, J. Stahl, S. Schuckers, and F. Hua, “Subregion mosaicking applied to nonideal iris recognition,” *Int. Symp. Computational Intelligence in Biometrics and Identity Management*, pp. 139–145, 2014.

[15] Y. Alvarez-Betancourt and M. Garcia-Silvente, “A keypoints-based feature extraction method for iris recognition under variable image quality conditions,” *Knowl.-Based Syst.*, vol. 92, pp. 169–182, 2016.

[16] Z. He, T. Tan, Z. Sun, and X. Qiu, “Toward accurate and fast iris segmentation for iris biometrics,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, pp. 1670–1684, 2009.

[17] T. Tan, Z. He, and Z. Sun, “Efficient and robust segmentation of noisy iris images for non-cooperative iris recognition,” *Image Vision Comput.*, vol. 28, pp. 223–230, 2010.

[18] C. Tan and A. Kumar, “Unified framework for automated iris segmentation using distantly acquired face images,” *IEEE Trans. Image Process.*, vol. 21, pp. 4068–4079, 2012.

[19] Y. Hu, K. Sirlantzis, and G. Howells, “Improving colour iris segmentation using a model selection technique,” *Pattern Recogn. Lett.*, vol. 57, pp. 24–32, 2015.

[20] L. Ma, T. Tan, Y. Wang, and D. Zhang, “Efficient iris recognition by characterizing key local variations,” *IEEE Trans. Image Process.*, vol. 13, pp. 739–750, 2004.

[21] C. Tan and A. Kumar, “Efficient and accurate at-a-distance iris recognition using geometric key-based iris encoding,” *IEEE Trans. Inf. Forensics Security*, vol. 9, pp. 1518–1526, 2014.

[22] ——, “Accurate iris recognition at a distance using stabilized iris encoding and zernike moments phase features,” *IEEE Trans. Image Process.*, vol. 23, pp. 3962–3974, 2014.

[23] K. Hollingsworth, K. Bowyer, and P.Flynn, “The best bits in an iris code,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, pp. 964–973, 2008.

[24] W. Dong, Z. Sun, and T. Tan, “Iris matching based on personalized weight map,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, pp. 1744–1757, 2010.

[25] C. Tan and A. Kumar, “Adaptive and localized iris weight map for accurate iris recognition under less constrained environment,” *IEEE Int. Conf. Biometrics: Theory, Applications and Systems*, pp. 1–7, 2013.

[26] Y. Hu, K. Sirlantzis, and G. Howells, “Exploiting stable and discriminability iris weight map for iris recognition under less constrained environment,” *Accepted by IEEE Int. Conf. Biometrics: Theory, Applications and Systems*, 2015.

[27] P. Li and G. Wu, “Iris recognition using ordinal encoding of log-euclidean covariance matrices,” *IEEE Int. Conf. Pattern Recongnition*, vol. 31, pp. 964–973, 2008.

[28] K. Roy and P. Bhattacharya, “Optimal features subset selection and classification for iris recognition,” *J. Image and Video Processing*, vol. 2008, 2008.

[29] M. Pereira and A. Veiga, “Application of genetic algorithms to improve the reliabilit of an iris recognition system,” *IEEE Workshop Machine Learning for Signal Processing*, 2005.

[30] K. Chen, C. Chou, S. Shih, W. Chen, and D. Chen, “Feature selection for iris recognition with adaboost,” *Int. Conf. Intelligent Information Hiding and Multimedia Signal Processing*, 2007.

[31] Z. He, Z. Sun, T. Tan, X. Qiu, C. Zhong, and W. Dong, “Boosting ordinal features for accurate and fast iris recognition,” *Int. Conf. Computer Vision and Pattern Recognition*, 2008.

[32] L. Wang, Z. Sun, and T. Tan, “Robust regularized feature selection from iris recognition via linear programming,” *Int. Conf. Pattern Recognition*, 2012.

[33] Z. Sun, L. Wang, and T. Tan, “Ordinal feature selection for iris and palmprint recognition,” *IEEE Trans. Image Process.*, vol. 23, pp. 3922–3934, 2014.

[34] J. Liu, Z. Sun, and T. Tan, “Code-level information fusion of low-resolution iris image sequences for personal identification at a distance,” *IEEE Int. Conf. Biometrics: Theory, Applications and Systems*, pp. 1–6, 2013.

[35] Y. Boykov, O. Veksler, and R. Zabih, “Markov random fields with efficient approximations,” *IEEE Conference on Computer Vision and Pattern Recognition*, 1998.

[36] ——, “Fast approximation energy minimization via graph cuts,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 11, pp. 1222–1239, 2001.

[37] Biometrics ideal test. [Online]. Available: http://biometrics.idealtest.org/dbDetailForUser.do?id=4

[38] K. Bowyer and P. Flynn, “The ND-IRIS-0405 iris image dataset,” *Notre Dame CVRL Technical Report*, 2009.

[39] H. Proenca, S. Filipe, R. Santos, J. Oliveira, and L. Alexandre, “The ubiris.v2: a database of visible wavelength images captured on-the-move and at-a-distance,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, pp. 1529–1535, 2010.

[40] P. Viola and M. Jones, “Rapid object detection using a boosted cascade of simple features,” *IEEE Conf. Computer Vision and Pattern Recognition*, 2001.

[41] H. Proenca and L. Alexandre, “Toward convert iris biometric recognition: experimental results from the nice contests,” *IEEE Trans. Inf. Forensics Security*, vol. 7, pp. 798–808, 2012.

[42] Y. Hu, K. Sirlantzis, and G. Howells, “A robust algorithm for color iris segmentation based on 1-norm regression,” *IEEE Int. Joint Conf. Biometrics*, 2014.

[43] L. Masek and P. Kovesi, “Matlab source code for a biometric identification system based on iris patterns,” *The School of Computer Science and Software Engineering, The University of Western Australia*, 2003.

[44] K. Nguyen, C. Fookes, S. Sridharan, and S. Denman, “Quality-driven super-resolution for less constrained iris recognition at a distance and on the move,” *IEEE Trans. Inf. Forensics Security*, vol. 6, pp. 1248–1258, 2011.

[45] H. Proenca, “Quaity assessment of degraded iris images acquired in the visible wavelength,” *IEEE Trans. Inf. Forensics Security*, vol. 6, pp. 82–95, 2011.

[46] ——, “Iris recognition: What’s beyond bit fragility,” *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 2, pp. 321–332, 2014.