

Improved Iris Recognition Through Fusion of Hamming Distance and Fragile Bit Distance

Karen P. Hollingsworth, Kevin W. Bowyer, and Patrick J. Flynn

Abstract—The most common iris biometric algorithm represents the texture of an iris using a binary iris code. Not all bits in an iris code are equally consistent. A bit is deemed *fragile* if its value changes across iris codes created from different images of the same iris. Previous research has shown that iris recognition performance can be improved by masking these fragile bits. Rather than ignoring fragile bits completely, we consider what beneficial information can be obtained from the fragile bits. We find that the locations of fragile bits tend to be consistent across different iris codes of the same eye. We present a metric, called the *fragile bit distance*, which quantitatively measures the coincidence of the fragile bit patterns in two iris codes. We find that score fusion of fragile bit distance and Hamming distance works better for recognition than Hamming distance alone. To our knowledge, this is the first and only work to use the coincidence of fragile bit locations to improve the accuracy of matches.

Index Terms—Iris biometrics, fragile bits, score fusion.

I. INTRODUCTION

RELIABLE identification of people is required for many applications such as immigration control, aviation security, or safeguarding of financial transactions. Research [2] and experience to date in actual applications [3] have demonstrated that the texture of a person’s iris is unique and can be used as a means of identification. Improving the accuracy and reliability of iris recognition is the goal of many current research endeavors [4].

The canonical iris recognition system involves a number of steps [4]. First, a camera acquires an image of an eye. Next, the iris is located within the image. The annular region of the iris is “unwrapped”, or transformed from raw image coordinates to normalized polar coordinates. A texture filter is applied to a set of locations on the iris, and the filter responses are quantized to yield a binary iris code. Finally, the iris code is compared with a known iris code in the gallery, and a similarity or distance score is reported. In an identity-verification application, the system uses the reported score to decide whether the two compared iris codes are from the same subject or from different subjects.

A. Fragile Bit Masking

Not all bits in the iris code are consistent across different images of the same iris. The concept that some bits in the iris code are less consistent than others was first published by Bolle et al. [5]. Since then, there has been a number of papers investigating fragile bits [1], [6]–[10].

Karen Hollingsworth, Kevin Bowyer, and Patrick Flynn (kholling, kwb, flynn@cse.nd.edu) are with the University of Notre Dame.

This is an extended and revised version of the paper, Using Fragile Bit Coincidence to Improve Iris Recognition, Proc. IEEE conf. on Biometrics: Theory, Applications, and Systems [1] © 2009 IEEE.

We now give a brief introduction to the concept of fragile bits. Readers interested in a more detailed analysis may refer to our previous work [7]. In creating an iris code, a traditional iris biometrics system applies Gabor filters to a number of locations on an iris image and obtains a complex-valued filter response for each location. Each complex number is quantized to two bits; the first bit is set to one if the real part of the complex number is positive, and the second bit is one if the imaginary part is positive. Consider multiple images of the same iris. A filter applied to one location on the iris produces a complex value. Across all images, the complex values for that location will be similar, but not exactly the same. Similarly, the bit from the binary quantization could be the same across all iris codes, or it may differ in some of the codes. A bit in a subject’s iris code is *consistent* if it assumes the same value for most images of that subject. A bit is *fragile* if it varies in value some substantial percent of the time. For a filter applied to a specific location in a single image, if the real part of the complex number has a large magnitude, then the corresponding bit will likely be consistent. On the other hand, if the real part is close to zero (or close to the vertical axis of the complex plane), the corresponding bit is fragile. Similar logic applies to the imaginary bits.

To illustrate this concept, we took 54 images of the same iris, and for each image, looked at the filter response for one particular spot on the iris. The resulting 54 complex numbers are shown in Figure 1. For this spot on the iris, all of the filter responses had a positive real value, but the imaginary part was positive about half of the time and negative the other half of the time. Therefore, the corresponding real bit in the iris code was consistent, and the corresponding imaginary bit was fragile.

In discussing fragile bits in the iris code, it is important to note that we are not saying that parts of the iris itself are unstable. The iris structure is generally considered highly robust, changing very little over time. Instead, bit fragility occurs when the inner product between a filter and a particular part of the iris produces a result with small magnitude, or with a phase close to the quantization boundary. Therefore, the consistency of each bit in the iris code is dependent upon a combination of (1) the iris texture at a certain position, (2) the filter used to analyze that texture, and (3) the quantization method for the filter response. If the iris texture at a particular point happens to have locally odd symmetry, for example, then its projection onto a cosine-phase wavelet may be close to zero, so the corresponding bit in the iris code may be determined by noise. A small change in the local coordinate system (due to segmentation uncertainty) can then easily flip the even bit, but not the odd bit. Thus, it is the result of how phase is encoded by the wavelets that is sensitive to sensor

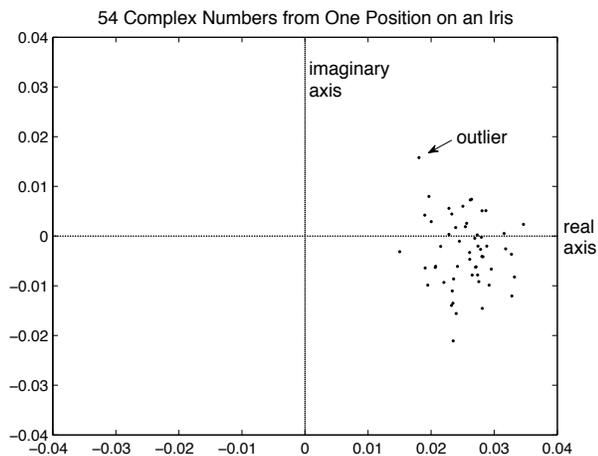


Fig. 1. These 54 complex numbers, each from the same region on 54 different images of the same subject's eye, all correspond to the same location on the iris code. Each complex number is mapped to two bits. This particular part of the iris code had a highly consistent real bit, and a highly fragile imaginary bit. (Figure reprinted from Hollingsworth et al., *IEEE Transactions on Pattern Analysis and Machine Intelligence* [7] ©2008 IEEE.)



Fig. 2. The consistency of each bit in the iris code is dependent upon a combination of the iris texture at a certain position and the filter used to analyze that texture. These two fragile bit patterns both correspond to the same iris, but a different filter has been used, and consequently, different fragile bit patterns emerge. Black regions represent fragile bits in the iris code, and white regions represent consistent bits.

noise or to minor coordinate system variation, not the iris or iris texture itself that is fragile.

In a previous paper [7], we used two different filters to create two different iris codes from the same eye. The iris codes were different for the two different filters, and similarly, the locations of fragile bits were different. This situation is illustrated in Figure 2. In this figure, we applied two different filters to the same iris and examined the fragile bit patterns. All factors were kept constant between the two analyses, except for the filter used. The two fragile bit patterns are clearly different. Similar experiments on other irises show the same phenomenon. Based on our previous experiments [7], we see that any type of filter, when used with a quantization technique like the one described above, would result in some bits being less consistent than others, and that the locations of the fragile bits would occur on the quantization boundaries. A different quantization technique would produce different patterns of fragile and consistent bits.

This concept suggests a simple optimization to iris recog-

nition algorithms. Assume a traditional quantization method, meaning quantization based on the four quadrants of the complex plane. When generating an iris code, we sort the real parts of the complex numbers, and identify a fraction of numbers with the smallest magnitude. Daugman suggested rejecting the lowest quartile of values [11]. The corresponding bits in the iris code are masked, or not considered when computing distance scores. The bits corresponding to the smallest imaginary values are likewise masked. With this modification, the final score in a comparison is based on fewer bits, but each bit used is more consistent. We call this modification *fragile bit masking*.

B. Motivation of Proposed Method

When using fragile bit masking [7], we mask a significant number of bits because the filter response produced an output with small magnitude. Rather than completely ignoring all information from those locations, we would like to find a way to make some beneficial use of those bits. We know that the values (zero/one) of those bits are not stable. However, the physical locations of those bits should be stable and might be used to improve our iris recognition performance.

We call the physical locations of fragile bits a *fragile bit pattern*. Figure 3 shows some iris images and Figure 4 shows the corresponding fragile bit patterns. Figure 4(a) and Figure 4(b) both show subject number 2463, and Figure 4(c) and Figure 4(d) both show subject 4261. The fragile bit patterns in Figure 4(a) and Figure 4(b) are more similar to each other than the fragile bit patterns in Figure 4(a) and Figure 4(c).

To compute the Hamming distance between two iris codes, we first take the logical AND of the masks for the two iris codes. Figure 5 shows the fragility masks obtained by ANDing pairs of fragility masks together. For example, Figure 5(a) is the comparison mask obtained by combining Figure 4(a) and 4(b). Figure 5(a) and 5(b) both show masks obtained when computing the Hamming distance for a match comparison (same subject). Figure 5(c) and 5(d) show masks for nonmatch comparisons. The fragile bit patterns for the match comparisons coincide more closely than the fragile bit patterns for the nonmatch comparisons. By looking at how well two fragile bit patterns align, we can make a prediction on whether those two irises are from the same subject or from different subjects. We can fuse that information with the Hamming distance information and get an improved prediction over using the Hamming distance alone.

C. Organization of paper

The rest of this paper is organized as follows. In section II we talk about related research. Section III describes the data set used for our experiments. Section IV defines a new metric, the *fragile bit distance* (FBD), which quantifies the difference between two fragile bit patterns. In section V, we present graphs of the distributions of FBD and Hamming distance. In section VI we talk about how to fuse Hamming distance with FBD. In section VII we show that the proposed method results in a statistically significant improvement over using Hamming

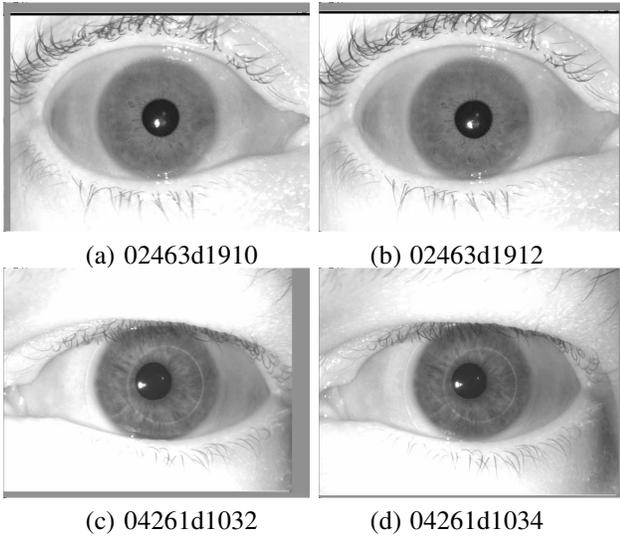


Fig. 3. Example images from our data set. These images were captured using an LG4000 iris camera.

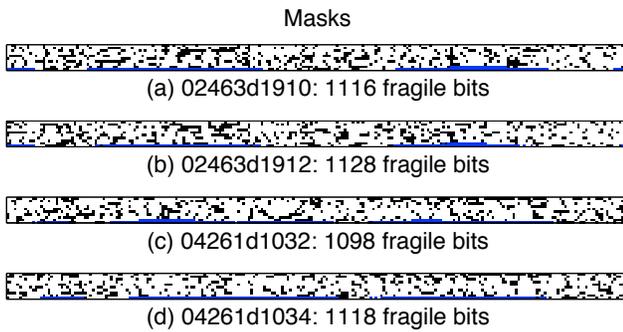


Fig. 4. These are the fragile bit patterns (imaginary part) corresponding to the images in Figure 3. Black pixels are bits masked for fragility. Dark blue pixels on the bottom of each rectangle are regions masked for occlusion. We use 4800-bit iris codes and mask 25% of the bits (or 1200 bits) for fragility. Some of the bits are masked for occlusion, and so slightly less than 1200 bits are masked for fragility.

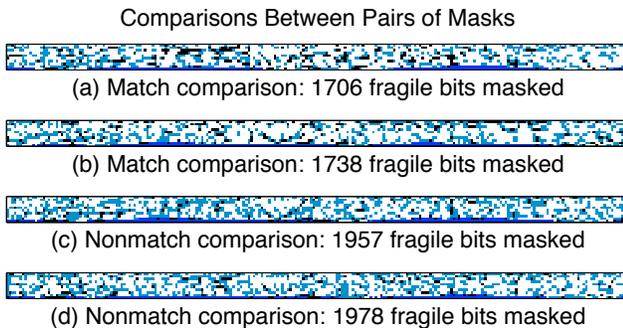


Fig. 5. These are comparisons of fragile bit patterns, each obtained by ANDing two fragile bit masks together. For example, Figure 5(a) is the comparison mask obtained by combining Figure 4(a) and 4(b). Black pixels show where the two fragility masks agreed. Light blue pixels show where they disagreed. White pixels were unmasked for both iris codes. There is more agreement in same-subject comparisons than there is when comparing masks of different subjects.

distance alone. Section VIII presents experiments showing the effect of changing the amount of fragile bit masking used. Finally, section IX shows how our method performs when using smaller template sizes.

II. RELATED WORK

For a broad survey of the iris biometrics field, readers may look at [4]. To see papers that evaluate different distance metrics for use in biometrics, we refer readers to [12] and [13]. Boulton [12] shows how robust distance measures can improve performance, and Ross and Govindarajan [13] employ Thresholded Absolute Distance and Euclidean distance to compare two biometric templates. In this section, we review some specific iris biometrics papers that relate closely to our work. We mention papers that (1) use portions of the iris code for specific tasks, (2) fuse Hamming distance with additional information, (3) extend the iris code to store additional phase information, or (4) investigate fragile bits in the iris code.

A. Using portions of the iris code for specific tasks

When searching a large database of iris codes, Hao et al. [14] compare small segments of iris codes instead of the entire code. They require three small segments to match exactly before taking the time to retrieve from disk and compare the entire iris code or record. This strategy allows them to achieve a 300-times speedup over an exhaustive search of 632,500 iris codes, with only a slight drop in performance. Mukherjee et al. [15] also present a method of improving response time when searching an iris database. They examine the structure of the iris code for clustering and indexing purposes.

Gentile et al. [16] also look at subsets of pixels in the iris code. Their full-length iris code was 5760 bytes. By subsampling rows in their iris code, they obtained a “short-length iris code” of 450 bytes (not including the mask). Short-length iris codes had the advantage of reducing the matching time per pair. The iris codes we use in this paper are 600 bytes long.

These papers are similar to ours in that they present some analysis of the structure of the iris code, but they differ in that they aim to reduce the time required for matching, while our work aims to improve recognition performance.

B. Research on fusing Hamming distance with added information

A small subset of iris biometrics research investigates combining Hamming distance with other information. A work by Sun et al. [17] aims to characterize global iris features using the following feature extraction method. First, they introduce a *local binary pattern* operator (LBP) to characterize the iris texture in each block of the iris image. The image block information is combined to construct a global graph. Finally, the similarity between two iris images is measured using a graph matching scheme. They fuse the LBP score with Hamming distance using the sum rule. They report that using Hamming distance alone yields an equal error rate (EER) of 0.70%, but the score fusion of Hamming distance with their LBP method yields an EER of 0.37%.

As an alternative to the sum rule, Sun et al. [17] state that the LBP score could be combined with Hamming distance using cascaded classifiers. Since their LBP method is slower than computing the Hamming distance, they suggest calculating the Hamming distance first. If the Hamming distance is below some low threshold, the comparison is classified as a match. If the Hamming distance is above some high threshold, the comparison is classified as a nonmatch. If the Hamming distance is between those two thresholds, use the LBP score to make the decision.

Vatsa et al. [18] characterize iris texture using Euler numbers. They use a Vector Difference Matching algorithm to compare Euler codes from two irises. Vatsa et al. combine Hamming distance and Euler score using a cascaded classifier.

Zhang et al. [19] use log-Gabor filters to extract 32 global features characterizing iris texture. To compare the global features from two iris images, they use a weighted Euclidean distance (WED) between feature vectors. Zhang et al. use cascaded classifiers to combine the global WED with a Hamming distance score. However, unlike Sun et al. [17] and Vatsa et al. [18], they propose using their global classifier first, and then using Hamming distance. In their experiments, using Hamming distance alone gave a false accept rate (FAR) of 8.1% when the false reject rate (FRR) was 6.1%. The fusion of WED and Hamming distance gave FAR = 0.3%, FRR = 1.9%.

Park and Lee [20] generate one feature vector using the binarized directional subband outputs at various scales. To compare two binary feature vectors, they use Hamming distance. A second feature vector is computed as the blockwise normalized directional energy values. Energy feature vectors are compared using Euclidean distance. To combine scores from these two feature vectors, Park and Lee use a weighted average of the two scores. Using the binary feature vectors alone gives an EER of 5.45%; the energy vectors yield an EER of 3.80%; when the two scores are combined, the EER drops to 2.60%.¹

All of the above mentioned papers combine Hamming distance scores with some other scores at the matching score level to improve iris recognition. Sun et al. [17] combine scores by summing. Three of the papers [17], [18], [20] use cascaded classifiers. Park and Lee [20] use a weighted average. Our work is similar to these papers in that we also consider combining two match scores to improve performance. We differ from these other works in that we are the first to use a score based on the location of fragile bits in two iris codes.

C. Extending the iris code to store additional information

Rather than designing a new type of feature extraction method, Kong et al. [21] extended the traditional method of creating a binary iris code. The traditional method, proposed by Daugman [2] coarsely quantizes each filter response to one of four values – 00, 01, 10, or 11 – based on the phase of the

complex number. Kong et al. [21] quantized the plane more finely. They tried using six regions in the complex plane and encoding the responses as one of six values – 000, 100, 110, 111, 011, or 001. They also considered using eight or ten regions. They found that a more precise phase representation improved performance, at the cost of requiring a larger iris code and consequently, a slightly slower matching speed.

Our work is similar to Kong's in that we essentially divide the complex plane into five regions – the four traditional regions, and a fifth region containing the axes of the complex plane. While each region in Kong's representation is treated equally, our method assigns a heavier value to the four traditional regions, and a lighter value to the region around the axes.

D. Research on Fragile Bits

Research on fragile bits is a more recent trend in iris biometrics literature. One of our previous papers, [7], presented evidence that not all bits in the iris code are of equal consistency. We investigated possible causes of fragile bits including small inconsistencies in segmentation and granularity of image alignment during matching. We also considered the effect of different filters on bit fragility; we used 1D log-Gabor filters and multiple sizes of a 2D Gabor filter, and found that the fragile bit phenomenon was apparent with each filter tested. The largest filter tended to yield fewer fragile bits than the smaller filters. With both types of filters, the fragile bits came from filter responses near quantization boundaries. We performed an experiment comparing (1) no masking of fragile bits (baseline) with (2) masking bits corresponding to complex filter responses near the axes of the complex plane. We masked fragile bits corresponding to the 25% of filter responses closest to the axes. Using a data set of 1226 images from 24 subjects, we found that fragile bit masking improved the separation between the match and nonmatch score distributions.

Other researchers have also begun to investigate the effects of masking fragile bits. Barzegar et al. [8] investigated fragile bit masking using different thresholds. They compared (1) no fragile bit masking to (2) fragile bit masking with thresholds of 20%, 30% and 35%. They found that using a threshold of 35% produced the lowest error rates on the CASIA-IrisV3 data set. Our own initial investigations have shown that the optimal fragility threshold may depend partly on the quality of the iris images being used; therefore, we feel that further investigation into the proper fragility threshold would be worthwhile.

Dozier et al. [9] also tried masking fragile bits and found an improvement in performance. However, they used a different method than Hollingsworth et al. [7] and Barzegar et al. [8]. Hollingsworth et al. [7] and Barzegar et al. [8] approximated fragile bit masking by masking filter responses near the axes of the complex plane. In contrast, Dozier et al. used a training set of ten images per subject to find consistency values for each bit in the iris code. Then for that subject, they only kept bits that were 90% or 100% consistent in their training set, and masked all other bits. In addition, they also considered only those bits that had at least 70% coverage in their training set; that is, if a bit was occluded by eyelids or eyelashes in four

¹These EERs are not as good as some EERs reported in other iris papers (e.g. [17]). It is not clear from the paper why these EERs are worse, but it is possible that their segmentation routine was not as robust as others, or their data set was more challenging.

or more of the training images, they masked that bit. Dozier et al. tested their method on six subjects from the ICE data set.

In a similar paper, Dozier et al. [10] again showed the benefit of masking fragile bits. In this work, they used a genetic algorithm to create an iris code and corresponding mask for each subject. Once again, they used ten training images per subject in generating their fragile bit masks.

Each of the above mentioned papers showed the benefit of masking fragile bits, but in every case, they simply discarded all information from the fragile bits. None of them considered employing the locations of fragile bits as an extra feature to fuse with Hamming distance.

The only paper that showed a benefit from using the locations of fragile bits is the earlier version of this paper [1]. In that earlier work, we introduced the idea of comparing fragile-bit-locations between two irises, and tested our idea on a data set of 9784 images. In this paper, we have more than doubled the size of our data set. We have further analyzed the distribution of fragile bit distance, added statistical tests evaluating our proposed method, and investigated the effect of varying the amount of fragile bit masking used.

III. DATA AND SOFTWARE

We acquired a data set of 19,891 iris images taken with an LG4000 iris camera [22] at the University of Notre Dame. Some example images are shown in Figure 3 and the camera is shown in Figures 6 and 7. The images are 640 pixels by 480 pixels. All images in this set were acquired between January 2008 and May 2009. A total of 686 different people attended acquisitions sessions, so there are 1372 different eyes in the data set. Each subject attended between one and eighteen acquisition sessions. At each session, we usually acquired three left eye images and three right eye images. The minimum number of images for any one subject is four (two of each iris), and the maximum is 108 (54 of each iris).

For our experiments, we used the current version of our in-house iris biometric software. This software is based on the IrisBEE software [23]. It uses one-dimensional log-Gabor filters for extracting the iris texture from the segmented and unwrapped iris image. The frequency response of the log-Gabor filter is

$$G(f) = \exp\left(\frac{-(\log(f/f_0))^2}{2(\log(\sigma/f_0))^2}\right) \quad (1)$$

where f_0 is the center frequency and σ gives the bandwidth of the filter [24]. Our software uses a center wavelength of 12 pixels, and a filter bandwidth of 0.5 [25]. One modification that we made to the IrisBEE software is that our software now uses active contours for segmentation. Additionally, we added fragile bit masking to the software; we use a default fragile bit masking threshold of 25% [7]. In section VIII of this paper, we investigate the effects of changing this threshold. A third modification involves the size of the iris code. We took the default 240 by 20 normalized iris image, and averaged neighboring rows to create a smaller image to use when generating the iris code. We averaged pixel values from rows one and two, three and four, and so forth, so that the final



Fig. 6. Images in our data set were captured using this LG4000 iris camera [22].



Fig. 7. The LG4000 iris camera captures images of both eyes at the same time.

normalized iris image was reduced to a 240 by 10 image. This row-averaging resulted in a smaller iris code, with no loss in performance [25]. From each pixel in the normalized image, we get two bits in the iris code, so the final iris code size is 240 by 10 by 2, or 4800 bits. In addition, we store one 4800-bit matrix with occlusion information and one 4800-bit matrix with fragility information.

IV. FRAGILE BIT DISTANCE

Figure 5 provides some indication of what we should expect when comparing two fragile bit patterns. In a genuine comparison, the locations of the fragile bits coincide. In an impostor comparison the locations of the fragile bits do not. When we compare two iris codes, we mask any bit that is fragile in either of the two fragile bit patterns. Therefore, we expect more bits to be masked for fragility in impostor comparisons than in genuine comparisons.

We can theoretically predict how many bits will be unmasked in an impostor comparison. In this analysis, we make

the assumption that the fragility of bits is independent of position and that each position is independent of all other positions. Consider the iris code for a single, unoccluded image. We mask 25% of bits for fragility, and leave 75% of bits unmasked. Now consider a comparison of two unoccluded images from different subjects. We expect $(75\%)(75\%) = 56.25\%$ of the bits to be unmasked, and 43.75% of the bits to be masked.

In contrast, a genuine comparison will have fewer masked bits. In two identical images, the fragile bits will coincide exactly and the comparison will have 75% unmasked bits and 25% masked bits. However, two different images of the same iris are not identical because of differences in lighting, dilation, distance to the camera, focus, or occlusion. Therefore, on average, more than 25% of the bits will be masked in a genuine comparison.

We define a metric called the *fragile bit distance* (FBD) to measure how well two fragile bit patterns align. In order to compute fragile bit distance, we need to store occluded bits and fragile bits separately. Therefore, each iris template will consist of three matrices: an iris code i , an occlusion mask m , and a fragility mask f . Unmasked bits are represented with ones and masked bits are represented with zeros. Specifically, unoccluded bits and consistent bits are marked as ones, while occluded and fragile bits are zeros. We do not want FBD to be affected by occlusion, so we consider only unoccluded bits when computing the FBD.

Take two iris templates, template A and template B. The FBD is computed as follows:

$$FBD = \frac{\|m_A \cap m_B \cap \overline{f_A \cap f_B}\|}{\|m_A \cap m_B\|} \quad (2)$$

where \cap represents the AND operator, and the line over $\overline{f_A \cap f_B}$ represents the NOT operator. The norm ($\|\cdot\|$) of a matrix tallies the number of ones in the matrix.

In above equation, $\overline{f_A \cap f_B}$ is a matrix storing all bits masked for fragility. $m_A \cap m_B$ is a matrix marking all bits unoccluded by eyelashes and eyelids. The FBD expresses the fraction of unoccluded bits masked for fragility in the comparison. This metric is large for impostor comparisons, and small for genuine comparisons.

Our theory predicts that we will have an average FBD of 0.4375 for impostor comparisons, and an average FBD of somewhere between 0.25 and 0.4375 for genuine comparisons. We tested these predictions on our data set of 19,891 images. The average FBD for genuine and impostor comparisons are reported in Table I, with standard deviations reported in parentheses.

Table I: Average FBD for Genuine and Impostor Comparisons (std dev)

	Avg. Genuine FBD	Avg. Impostor FBD
Theoretical value	between 0.25 & 0.4375	0.4375
LG4000 images	0.4047 (0.0149)	0.4397 (0.0097)

The average impostor FBD was within one standard deviation of the theoretical prediction. Also, the average genuine FBD was less than the average impostor FBD.

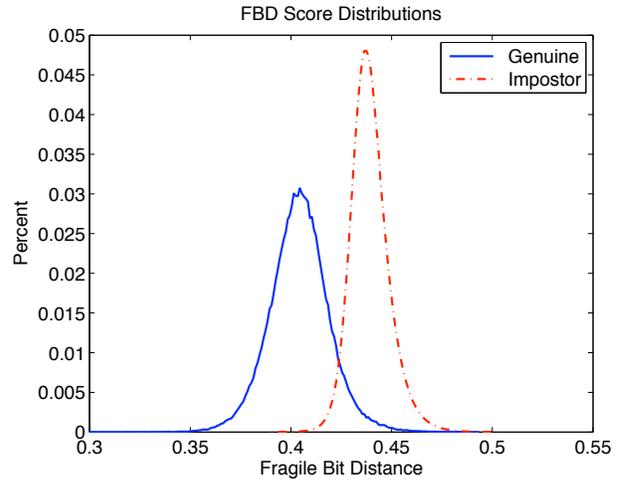


Fig. 8. Score distributions for fragile bit distance.

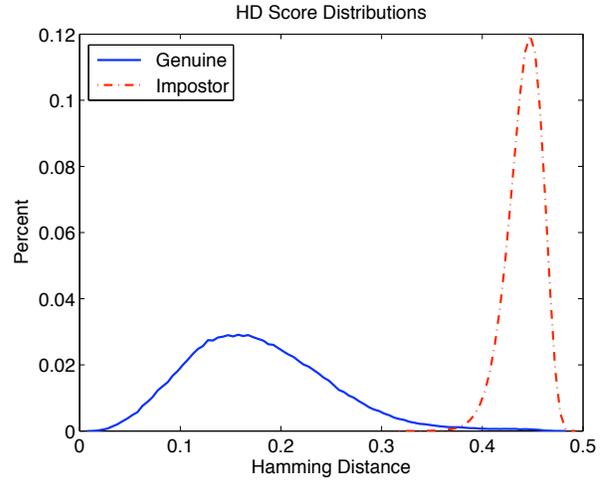


Fig. 9. Score distributions for Hamming distance.

V. SCORE DISTRIBUTIONS FOR HAMMING DISTANCE AND FRAGILE BIT DISTANCE

We graphed the genuine and impostor score distributions for fragile bit distance (FBD) from all possible comparisons in our 19,891-image data set. Figure 8 shows the result. In comparison, Figure 9 shows the genuine and impostor score distributions for Hamming distance. There is more separation between the genuine and the impostor score distributions for Hamming distance than there is for FBD. The means, standard deviations, and d' values for both HD and FBD are given in Table II.

Table II: Distributions of FBD and HD

Metric	Match μ (σ^2)	Nonmatch μ (σ^2)	d'
HD	0.1798 (0.0705)	0.4440 (0.0179)	5.1406
FBD	0.4047 (0.0149)	0.4397 (0.0097)	2.7857

Figure 10 shows the joint distribution of FBD and Hamming distance. Figure 13 shows the same joint distribution, zoomed-

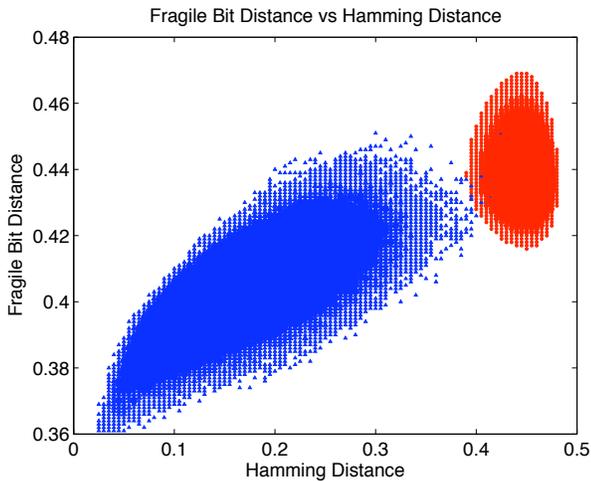


Fig. 10. Joint score distributions for Hamming distance and FBD. Genuine scores are shown in blue. Impostor scores are shown in red.

in on the area of the graph where the genuine and impostor score distributions meet. Each blue point in these figures represents at least 0.003% of the 247,872 match comparisons in our experiment, and each red point represents at least 0.003% of the 197,229,390 nonmatch comparisons. Selecting a single threshold of Hamming distance (e.g. $HD = 0.35$) would separate genuine and impostor comparisons better than any threshold we might choose for FBD. Using FBD, we achieve an equal error rate of 6.34×10^{-2} on this data set. Using Hamming distance, the equal error rate is 8.70×10^{-3} .

VI. FUSING FRAGILE BIT DISTANCE WITH HAMMING DISTANCE

Even though the FBD is not as powerful a metric as the Hamming distance, we can combine the features to create a better classifier than Hamming distance alone. To combine Hamming distance and FBD, we first tried a weighted average technique, using the same approach as [20]. We combined the two scores using the equation,

$$Score_W = \alpha \times HD + (1 - \alpha) \times FBD. \quad (3)$$

We varied the parameter α in steps of 0.1 from 0 to 1, and calculated the equal error rate for each run. Figure 11 shows how the equal error rate changes as α varies. The lowest equal error rate was 8.02×10^{-3} , which was obtained using an α value of 0.6.

The benefit of using a weighted average can be seen visually in Figure 13. This figure shows the joint distribution of Hamming distance and FBD scores. The vertical line marked “ $HD = \text{constant}$ ” shows how using Hamming distance would separate the genuine and impostor scores. The diagonal line marked “ $0.6HD + 0.4FBD = \text{constant}$ ” shows that a better separation between genuine and impostor scores is achieved using the weighted average.

Multiplication can be used as an alternative method of score fusion:

$$Score_M = HD \times FBD. \quad (4)$$

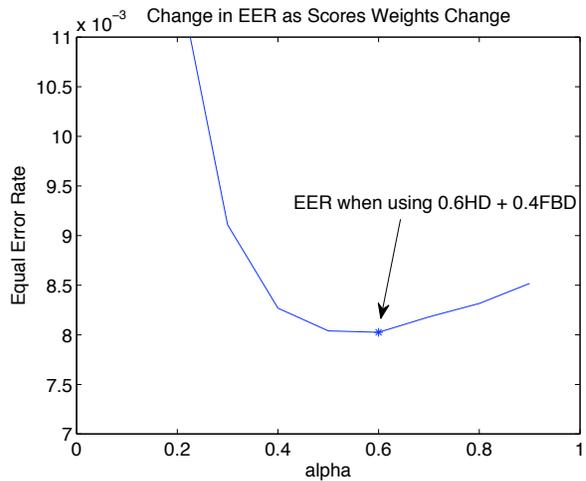


Fig. 11. We fused FBD and HD using the expression, $\alpha \times HD + (1 - \alpha) \times FBD$. We found that an α value of 0.6 yielded the lowest equal error rate.

When using multiplication, the equal error rate was 7.99×10^{-3} . Fusing by multiplication and fusing by weighted average yielded similar results.

An ROC curve showing the results of these tests is shown in Figure 12. The weighted average method (red line) and the multiplication method (blue line) perform similarly, so the corresponding ROC curves are overlapping. Both fusion methods are clearly performing better than the baseline (black line). This figure shows 95% confidence intervals for multiple points on the graph. To create the confidence intervals, we used a bootstrapping method. For each bootstrap, we sampled the 686 subjects with replacement to form a set of test subjects to use for a probe, and compared irises of these subjects to all gallery images. From the resulting scores we formed an ROC curve. Given an ROC curve for each of 50 bootstraps, we computed the mean true accept rate and corresponding standard deviation for multiple false accept rates. Then we used these values to construct the 95% confidence limits for each point. The confidence intervals for the baseline method do not overlap with the confidence intervals for the fusion methods at low false accept rates (0.001 to 0.003). At higher false accept rates, there is some overlap in the confidence intervals. At a false accept rate of 0.02, the mean performance for the fusion methods equals the upper limit of the baseline confidence interval.

Table III shows summary statistics of these experiments including the equal error rate (EER) and the false reject rate at an operating point of $FAR=0.001$ (FRR at $FAR=0.001$). These error rates were computed over the entire data set (without bootstrapping).

Method	EER	FRR at $FAR=0.001$
HD (baseline)	8.70×10^{-3}	1.40×10^{-2}
0.6HD + 0.4FBD	8.02×10^{-3}	1.25×10^{-2}
HD \times FBD	7.99×10^{-3}	1.23×10^{-2}

Based on the values in Table III, we see that both methods

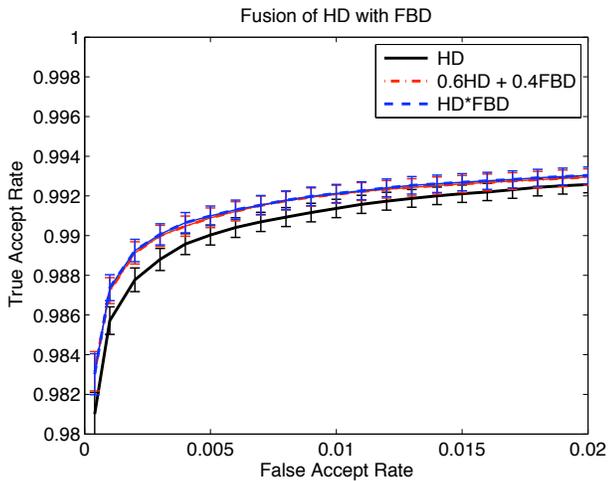


Fig. 12. Fusing Hamming distance with FBD performs better than using Hamming distance alone. Fusing by multiplying and fusing by weighted averaging yield similar results.

of fusing Hamming distance with FBD performed better than using Hamming distance alone. By incorporating FBD, we improved the accuracy of our iris matcher.

One caveat with using FBD is that in order to compute FBD, we have to store the fragility mask separately from the occlusion mask. Therefore, our iris template is 50% larger than it would be if we did not use FBD.

VII. TESTS OF STATISTICAL SIGNIFICANCE

The proposed fusion between Hamming distance and FBD works better than the baseline of Hamming distance alone. We performed a statistical test to determine whether this difference was statistically significant. The null hypothesis for this test is that there is no difference between the baseline Hamming distance method and the proposed fusion of Hamming distance and FBD. The alternative is that there is a significant difference. To test for statistical significance, we randomly divided the subjects into ten different test sets. For each test set, we measured the performance of using Hamming distance alone, and of using fusion of Hamming distance and FBD. Then we used a paired t-test to see whether the proposed method obtained a statistically significant improvement. The results are given in Table IV for weighted average fusion. Table V shows the results for fusion using multiplication. The t-test showed statistically significant improvement of the proposed method over the baseline for both EER and false reject rate at a false accept rate of 0.1% (FRR at FAR=0.001). Rerunning the same experiment using different random test sets gave similar results.

Table IV: Is $0.6HD + 0.4FBD$ better than HD alone?

Method	Avg. EER	Avg. FRR at FAR=0.001
HD (baseline)	8.68×10^{-3}	1.52×10^{-2}
0.6HD + 0.4FBD (proposed)	8.08×10^{-3}	1.33×10^{-2}
p-value	3.83×10^{-3}	1.44×10^{-3}

Table V: Is $HD \times FBD$ better than HD alone?

Method	Avg. EER	Avg. FRR at FAR=0.001
HD (baseline)	8.68×10^{-3}	1.52×10^{-2}
$HD \times FBD$ (proposed)	8.00×10^{-3}	1.33×10^{-2}
p-value	6.43×10^{-3}	1.90×10^{-3}

Recall that when we performed fusion using a weighted average, we used the equation,

$$Score_W = \alpha \times HD + (1 - \alpha) \times FBD \quad (5)$$

and we found that using a weight value of $\alpha = 0.6$ worked best. We tested whether performance using this value of alpha was statistically different from performance using other values of alpha. For this test, we again divided the subjects randomly into ten different test sets. We varied the parameter α in steps of 0.1 from 0 to 1. For a given value of α , we computed the equal error rate for each of the test sets, then found the average equal error rate for this value of α across all test sets. The results are shown in Table VI. Next, we performed a paired t-test to determine whether the given value of α produced significantly different results than using $\alpha = 0.6$. The p-values for these tests are also shown in Table VI. We found that at a significance level of $p=0.05$, values of α between 0.4 and 0.7 were not significantly different from $\alpha = 0.6$. However, other values of α were significantly different.

Table VI: Is $\alpha HD + (1 - \alpha)FBD$ statistically significantly different from $0.6HD + 0.4FBD$?

α	Avg. EER	p-value	Yes/No?
0	6.28×10^{-2}	2.81×10^{-8}	Yes
0.1	2.09×10^{-2}	1.75×10^{-6}	Yes
0.2	1.16×10^{-2}	1.79×10^{-4}	Yes
0.3	8.98×10^{-3}	1.24×10^{-2}	Yes
0.4	8.21×10^{-3}	4.29×10^{-1}	No
0.5	8.08×10^{-3}	9.62×10^{-1}	No
0.6	8.08×10^{-3}	-	-
0.7	8.19×10^{-3}	1.18×10^{-1}	No
0.8	8.37×10^{-3}	2.56×10^{-2}	Yes
0.9	8.56×10^{-3}	5.25×10^{-3}	Yes
1.0	8.68×10^{-3}	3.68×10^{-3}	Yes

VIII. EFFECT OF MODIFYING THE FRAGILE BIT MASKING THRESHOLD

Recall that fragile bit masking ignores the bits corresponding to complex filter responses close to the axes of the complex plane (see section I-A). In the experiments presented up to this point, we masked 25% of bits in each iris code for fragility. We used 25% because it is the value used in previous work [7] and recommended by Daugman [11]. In this paper, we wanted to study how changing the threshold would affect our results.

We ran experiments varying the threshold used for fragile bit masking. First we ran one test with 0% fragile bit masking. We ran an all-vs-all test (comparing all images to all other images in the data set) and computed the performance using Hamming distance alone. The equal error rate for that test was 8.26×10^{-3} .

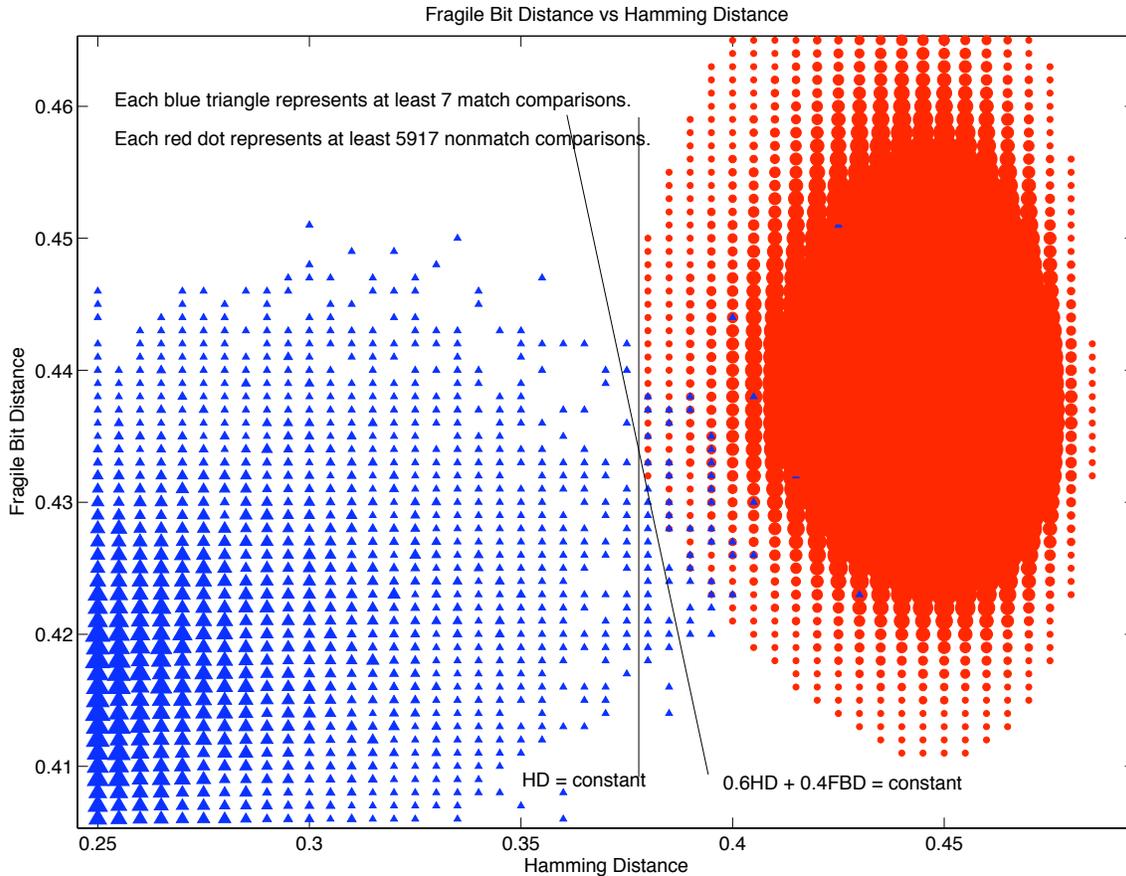


Fig. 13. A zoomed-in view of the joint score distributions for Hamming distance and FBD. Genuine scores are shown in blue. Impostor scores are shown in red. Each point represents at least 0.003% of the comparisons.

Next, we varied the threshold from 5% to 30% in increments of 5%. At each threshold, we ran three all-vs-all tests. The first test was using Hamming distance alone. The second test was using the a weighted average of Hamming distance and fragile bit distance: $0.6HD + 0.4FBD$. The third test used the multiplication of Hamming distance and fragile bit distance: $HD \times FBD$.

At low levels of fragile bit masking, we would not expect fragile bit distance to be as powerful a metric because there is less fragile bit information available. However, the weighted average fusion method is robust enough that we could expect $0.6HD + 0.4FBD$ to still achieve good performance. Multiplication is considered a less robust fusion method, in the sense that outliers of one of the factors in the multiplication can dramatically alter the product. The experimental results of using 5% fragile bit masking for all three tests are shown in Figure 14. As expected, the weighted average method still performed well. In fact, the Hamming distance test and the weighted average test gave very similar results, with the weighted average being marginally better than Hamming distance. The multiplication method did not perform well at this level of masking.

At 15% fragile bit masking, the multiplication fusion

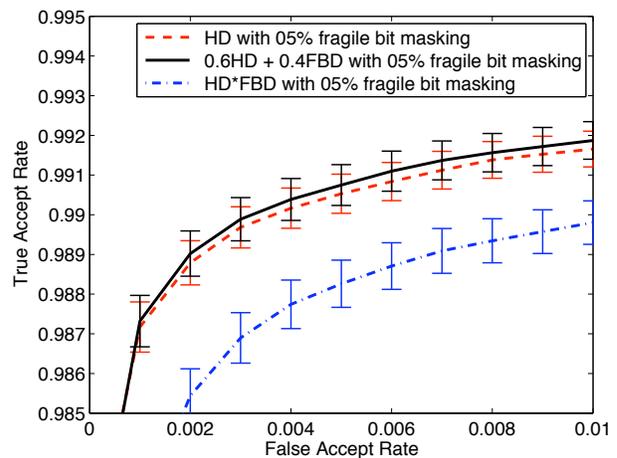


Fig. 14. We considered the effect of masking only 5% of the bits in the iris code for fragility. Using this value, we compared the performance of (1) Hamming distance (HD) with performance of (2) fusing HD and FBD with a weighted average ($0.6HD + 0.4FBD$) and (3) fusing HD and FBD with multiplication ($HD \times FBD$). At this low levels of fragile bit masking, the multiplication method performs poorly. In contrast, the difference between HD and the weighted average method is small, and the ROC curves for these two methods overlap.

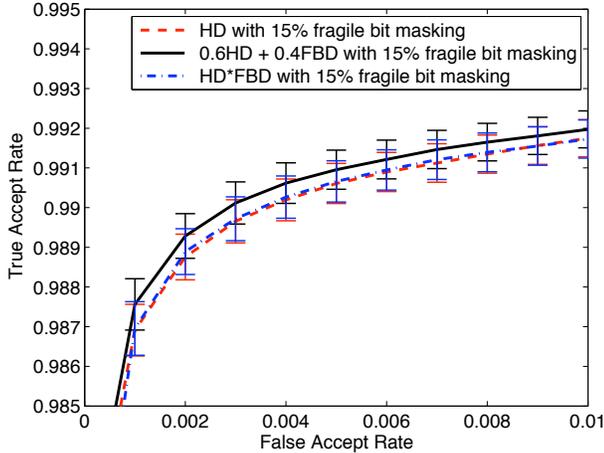


Fig. 15. Using 15% fragile bit masking, the difference between HD and a weighted average method ($0.6HD + 0.4FBD$) is larger than at lower levels of fragile bit masking. The multiplication method ($HD \times FBD$) is also much improved at this level of fragile bit masking.

method did much better; performance matched the performance of Hamming distance alone. The weighted average method still performed the best (Figure 15). Between 20% and 30% fragile bit masking, performance of the multiplication method matched performance of the weighted average method, and both noticeably outperformed Hamming distance (See Figures 16, and 17). The general trend showed that at higher levels of fragile bit masking, the difference between using Hamming distance and using one of the two fusion methods was larger (see Figures 15, 16, and 17). The ROC curves in Figures 14, 15, 16, and 17 all show 95% confidence intervals for a number of points on the curve.

The best performance using Hamming distance alone was achieved using 5% fragile bit masking; at this threshold, the equal error rate was 8.15×10^{-3} . The best performance using the weighted average of Hamming distance and FBD was achieved using a 25% fragile bit threshold; the equal error rate on this test was 8.02×10^{-3} . The best performance for the multiplication of Hamming distance and fragile bit distance was 7.99×10^{-3} , and this was achieved using a 25% fragile bit masking threshold.

We observe that the fusion of Hamming distance and fragile bit distance has greater benefit when a higher level of fragile bit masking is used. We only tested fragile bit masking thresholds up to 30% on our data set because for our data and software, our experiments indicate that increasing the fragile bit masking further would not improve performance. On the other hand, other researchers have found that fragile bit masking of 35% worked best on the CASIA version 3 data set [8]. We postulate that any system that uses a fragile bit masking level of 15% or higher could benefit from using fragile bit distance in addition to Hamming distance.

IX. EFFECT OF USING A SMALLER IRIS CODE

In our iris recognition software, we averaged rows in the 240 by 20 normalized image to create a smaller normalized

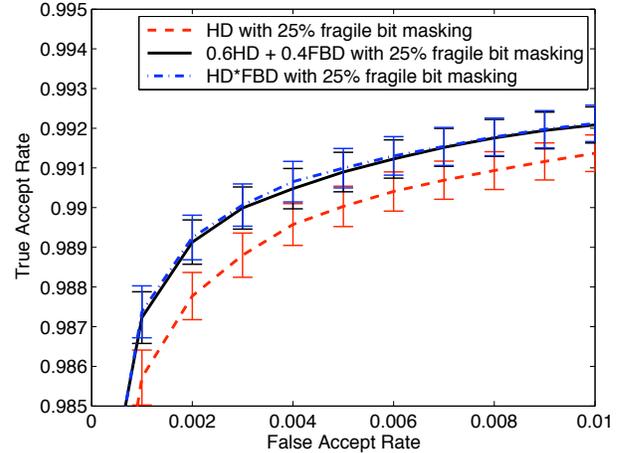


Fig. 16. Using 25% fragile bit masking, the weighted average method ($0.6HD + 0.4FBD$) and the multiplication method ($HD \times FBD$) are both clearly better than HD alone.

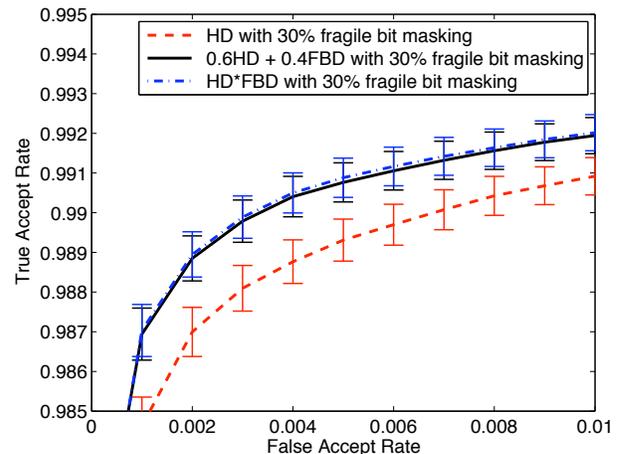


Fig. 17. Using 30% fragile bit masking, the weighted average method ($0.6HD + 0.4FBD$) and the multiplication method ($HD \times FBD$) are both clearly better than HD alone.

image of size 240 by 10, thus reducing the size of our iris code to 240 by 10 by 2, or 4800 bits. We can further reduce the size of the iris code by subsampling columns. Our approach to subsampling columns is different that our treatment of rows in the template creation. To reduce the number of rows, we averaged rows in the normalized image. In contrast, to reduce the number of columns, we first created a larger iris code, then subsampled the columns.

Using our most recent version of our iris recognition software which uses column subsampling to create a 2400 bit iris code, we reran our experiments to see if FBD could still be helpful when using a smaller template size. With the newer, optimized software, the improvement was not as large, but our proposed technique still achieved superior performance when compared to the baseline of using Hamming distance alone (Table VII).

Table VII: Fusing FBD with Hamming distance when using a smaller iris code

Method	EER	FRR at FAR=0.001
HD (baseline)	2.98×10^{-3}	3.69×10^{-3}
0.6HD + 0.4FBD	2.90×10^{-3}	3.57×10^{-3}
HD \times FBD	2.97×10^{-3}	3.56×10^{-3}

X. CONCLUSION

In this paper, we defined a new metric, the *fragile bit distance* (FBD) which measures how two fragile bit masks differ. Low FBDs are associated with genuine comparisons between two iris codes. High FBD are associated with impostor comparisons.

Fusion of FBD and Hamming distance is a better classifier than using Hamming distance alone. Fusion can be done either by using a weighted average of FBD and Hamming distance, or by multiplying. The multiplication of FBD and Hamming distance reduces the EER of our iris recognition system by eight percent – from 8.70×10^{-3} to 7.99×10^{-3} – a statistically significant improvement.

Fusing FBD and Hamming distance has a greater benefit when higher levels of fragile bit masking are used. At low levels of fragile bit masking, fusion had similar results to using Hamming distance alone on our data. When using fragile bit masking thresholds of 15% or greater, fusion had superior performance.

ACKNOWLEDGEMENT

This research is supported by the National Science Foundation under grant CNS01-30839, by the Central Intelligence Agency, by the Intelligence Advanced Research Projects Activity, by the Biometrics Task Force, and by the Technical Support Working Group under US Army contract W91CRB-08-C-0093. The opinions, findings, and conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of our sponsors.

We would like to thank our anonymous reviewers for suggesting a more precise description of the concepts described in Section I-A.

REFERENCES

- [1] Karen P. Hollingsworth, Kevin W. Bowyer, and Patrick J. Flynn. Using fragile bit coincidence to improve iris recognition. In *Proc. IEEE Int. Conf. on Biometrics: Theory, Applications, and Systems*, pages 1–6, Sept 2009.
- [2] John Daugman. How iris recognition works. *IEEE Transactions on Circuits and Systems for Video Technology*, 14(1):21–30, 2004.
- [3] John Daugman. Probing the uniqueness and randomness of iriscodes: Results from 200 billion iris pair comparisons. *Proceedings of the IEEE*, 94(11):1927–1935, 2006.
- [4] Kevin W. Bowyer, Karen P. Hollingsworth, and Patrick J. Flynn. Image understanding for iris biometrics: A survey. *Computer Vision and Image Understanding*, 110(2):281–307, 2008.
- [5] Ruud M. Bolle, Sharath Pankanti, Jonathon H. Connell, and Nalini Ratha. Iris individuality: A partial iris model. In *Proc. Int. Conf. on Pattern Recognition*, pages II: 927–930, 2004.
- [6] Karen P. Hollingsworth, Kevin W. Bowyer, and Patrick J. Flynn. All iris code bits are not created equal. In *Proc. IEEE Int. Conf. on Biometrics: Theory, Applications, and Systems*, pages 1–6, Sept 2007.

- [7] Karen P. Hollingsworth, Kevin W. Bowyer, and Patrick J. Flynn. The best bits in an iris code. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(6):964–973, Jun 2009.
- [8] Nakissa Barzegar and M. Shahram Moin. A new user dependent iris recognition system based on an area preserving pointwise level set segmentation approach. *EURASIP Journal on Advances in Signal Processing*, pages 1–13, 2009.
- [9] Gerry Dozier, David Bell, Leon Barnes, and Kelvin Bryant. Refining iris templates via weighted bit consistency. In *Proc. Midwest Artificial Intelligence and Cognitive Science (MAICS) Conference*, pages 1–5, Apr 2009. Fort Wayne, Indiana.
- [10] Gerry Dozier, Kurt Frederiksen, Robert Meeks, Marios Savvides, Kelvin Bryant, Darlene Hopes, and Taihei Munemoto. Minimizing the number of bits needed for iris recognition via bit inconsistency and grit. In *Proc. IEEE Workshop on Computational Intelligence in Biometrics: Theory, Algorithms, and Applications*, pages 30–37, Apr 2009.
- [11] John Daugman. personal communication.
- [12] T. Boulton. Robust distance measures for face-recognition supporting revocable biometric tokens. In *IEEE Conference on Automatic Face and Gesture Recognition*, Apr 2006.
- [13] Arun Ross and Rohin Govindarajan. Feature level fusion using hand and face biometrics. In *Proc. of SPIC Conference on Biometric Technology for Human Identification II, Vol.5779*, pages 196–204, Mar 2005.
- [14] Feng Hao, John Daugman, and Piotr Zielinski. A fast search algorithm for a large fuzzy database. *IEEE Transactions on Information Forensics and Security*, 3(2):203–212, June 2008.
- [15] R. Mukherjee and A. Ross. Indexing iris images. In *Proc. of International Conference on Pattern Recognition (ICPR)*, Dec 2008.
- [16] J. Gentile, N. Ratha, and J. Connell. SLIC: short-length iris codes. In *Biometrics: Theory, Applications, and Systems, 2009. IEEE 3rd International Conference on*, pages 1–5, Sept 2009.
- [17] Zhenan Sun, Tieniu Tan, and Xianchao Qiu. Graph matching iris image blocks with local binary pattern. In *Proc. Int. Conf. on Biometrics (Springer LNCS 3832)*, pages 366–372, Jan 2006.
- [18] Mayank Vatsa, Richa Singh, and Afzel Noore. Reducing the false rejection rate of iris recognition using textural and topological features. *Int. Journal of Signal Processing*, 2(2):66–72, 2005.
- [19] Peng-Fei Zhang, De-Sheng Li, and Qi Wang. A novel iris recognition method based on feature fusion. In *Proc. Int. Conf. on Machine Learning and Cybernetics*, pages 3661–3665, 2004.
- [20] Chul-Hyun Park and Joon-Jae Lee. Extracting and combining multi-modal directional iris features. In *Int. Conf. on Biometrics (Springer LNCS 3832)*, pages 389–396, Jan 2006.
- [21] Adams W. K. Kong, David Zhang, and Mohamed S. Kamel. An analysis of iriscodes. *IEEE Transactions on Image Processing*, 19(2), Feb 2010.
- [22] LG IrisAccess 4000. <http://www.lgiris.com/ps/products/irisaccess4000.htm>, accessed Apr 2009.
- [23] P. Jonathon Phillips, Kevin W. Bowyer, Patrick J. Flynn, Xiaomei Liu, and W. Todd Scruggs. The iris challenge evaluation 2005. In *Proc. IEEE Int. Conf. on Biometrics: Theory, Applications, and Systems*, Sept 2008.
- [24] Libor Masek. Recognition of human iris patterns for biometric identification. Master’s thesis, University of Western Australia, 2003. <http://www.csse.uwa.edu.au/~pk/studentprojects/libor/>.
- [25] Tanya Peters. Effects of segmentation routine and acquisition environment on iris recognition. Master’s thesis, University of Notre Dame, 2009.