# Robust Long Range Iris Recognition from Video Using Super Resolution

Yung-hui Li

CMU-10-006

Language Technology Institute
School of Computer Science
Carnegie Mellon University
5000 Forbes Ave, Pittsburgh, PA 15213
www.lti.cs.cmu.edu

**Thesis Committee:**
Prof. Marios Savvides, Carnegie Mellon University, Chair
Prof. Bhiksha Raj, Carnegie Mellon University
Prof. Tsuhan Chen, Cornell University
Dr. Nalini Ratha, IBM Research

*Submitted in partial fulfillment of the requirements*
*for the degree of Doctor of Philosophy*
*In Language and Information Technologies*

**Keywords:** Super-Resolution, Iris Recognition, Iris Segmentation, Iris Occlusion Estimation, Multi-frame image fusion

# Abstract

Iris recognition has been developing for over 20 years, but, only in recent years has it been more accessible and widely accepted as one of the most accurate and un-obtrusive biometric modalities. Over the past few years, many companies have developed iris acquisition systems that are more user-friendly. Iris-On-the-Move (IOM) is one such system, offering significant stand-off acquisition distance (3m), which is extremely convenient for users and very suitable for deployment at airports to check passenger identifications and to control access. However, iris images acquired by the IOM and other long range systems are, in most cases, considerably blurred, of low contrast, and lacking detail in the iris texture compared to images from very close proximity sensors (5cm stand-off). This thesis focuses on how to deal with the three most challenging problems in long-range iris recognition: (1) iris segmentation from long range systems, (2) automatic iris mask generation of occluded regions, (3) iris matching performance enhancement using multiple irises from a video sequence. In particular, we emphasize solutions in the context of the IOM system and those that take advantage of an iris image video stream.

If an image of the eye is clear and has strong contrast, it is very easy to find the boundaries of the pupil and iris. However, most images acquired by the long-range iris acquisition system are blurred and noisy, which is the first problem that we propose a solution to: iris segmentation. Even worse, there are always strong specular reflections either in the pupil or on the iris region, which increases the difficulty in achieving good iris segmentation results. For this problem, we propose a novel iris segmentation algorithm which is robust in dealing with specular reflections and image blur and is also computationally efficient.

For the second problem, automatic detection of iris occluded regions from eyelashes and specular reflections, we focus on estimating a mask for the iris texture in the polar coordinate system. Unlike most current methods, we propose a probabilistic, learning-based approach where the system can learn about the pixel distribution from a training data set and create masks for a test data set in an efficient way. We further search the parameter space of Gabor filters in order to optimize the features set that the prposed algorithm learns, for the purpose of minimizing global error rate for large-scale iris recognition.

Iris matching performance enhancement for images captured by long-range iris acquisition devices, the third problem we address, deals with iris images that are blurred, defocused, and noisy due to low quantum efficiency of long-range iris sensors imaging in near-Infrared wavelengths. By exploiting the multi-frame video capture of the long-range iris acquisition system, it is possible to enhance the recognition performance by improving the low-quality iris images acquired from the system using super-resolution methods designed specifically for irises. We show a comprehensive set of empirical results demonstrating the effectiveness of our proposed approach designed for the IOM system that also apply to any video sequence of iris images captured by long-range iris acquisition devices.

# Acknowledgments

I would like to acknowledge and thank the following people. First, I would like to thank my advisor, Marios Savvides, for his guidance, advice, and financial support during the past four years of my PhD study. Dr. Savvides is very considerate of his students, working extremely hard to make sure all of his students are financially supported. His work ethic encourages everyone in the lab to work as hard as they can to achieve the highest quality in our research results. Not only is Dr. Savvides a colleague and mentor, but he is also a close friend of my whole family.

I would also like to thank my colleagues, Khalid Harun, Shreyas Venugopalan, and Taihei Munemoto for their great ideas and useful discussions about iris recognition research and detailed algorithmic design. In addition, I would like to acknowledge Jingu Heo and Sungwon Park, who shared their knowledge of novel algorithms in the field of pattern recognition and machine learning. Our communications generated many new ideas during these past few years.

I would especially like to honor my wife because she is the most important person in my life. She always understands my frustrations, sorrows and worries, encouraging me with her love and kindness. When I have been too busy to come home late at night, she is the person who takes care of every chore in the house all by herself without complaint. She has been very supportive of me, so I could finish my PhD successfully. Over the years, her unconditional love has taught me a lot and has helped me continue to grow as a person. Her sacrifice shows me what true love is all about. It is her love and support that has given me the hope and strength to face all the difficulties and challenges during the past years.

I would also like to give thanks to many of my friends in the Pittsburgh Chinese Church in Oakland (PCCO). They encourage me, give me suggestions, and pray for me whenever suffering and trials come. It is so wonderful to have good friends like them.

Last but not the least, I give thanks and all honor to my God, who is my creator and savior. He is the ultimate source of my joy and strength. It is not by my own power or strength, but through Him, that I can finish the long process of pursuing a PhD. May all the glory be unto Him. He is the almighty God who is worthy to be praised.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

Over the past few decades, biometric recognition has drawn significant attention due to its many applications in the field of law enforcement, surveillance, border control, and national security. The core goal of any biometric recognition system is to recognize the identity of the target person based on his/her physiological or behavioral characteristics. Examples of such characteristics include fingerprint, face, voice, signature, hand geometry, iris and palm print [1].

## 1.1 The Robustness of the Iris Recognition

Among all the usable characteristics for biometric recognition, the pattern of iris texture is one of the few characteristics believed to be the most distinguishable among different people [2]. The iris is the annular area between the pupil and the sclera of the eye, as shown in Figure 1.1. It consists of pigmented fibril-vascular tissues which connect to the sphincter and dilator muscles that control the contraction and dilation of the pupil. It is the randomness of the structure of those tiny tissues that gives the iris pattern its uniqueness for each person. Furthermore, the iris possesses certain properties that make it a very attractive biometric feature; first and foremost, it is thought to remain relatively unchanged throughout a person's lifetime. In fact, the iris pattern is formed in the third month of gestation, and the structure becomes stable around the eighth month [3]. This permanence over time makes iris recognition a reliable biometric compared to other physiological characteristics, such as a face or voice, which change drastically as a person grows old. The iris is also rarely affected by external elements because it is well protected behind the cornea. Even eye surgery, which is typically performed on the cornea or on the retina through the pupil, seldom hurts the iris. In comparison, fingerprints, although unique and relatively permanent, suffer significantly from external elements, such as erosion, cuts and scratches, and medical conditions like Psoriasis. These external elements cause thinning of the finger pads and, ultimately, lead to failures in acquisition by common fingerprint sensors. Although comparably harder to acquire, iris imaging is relatively un-intrusive and, as with face imaging, requires minimal effort from a subject. The subject only needs to stand still or stare at a camera. Because this requires no touching or physical contact, it is easier to accept. Lastly, it is believed to be one of the hardest biometrics to spoof and circumvent.

A typical iris recognition system consists of following stages: (1) iris image acquisition, (2)

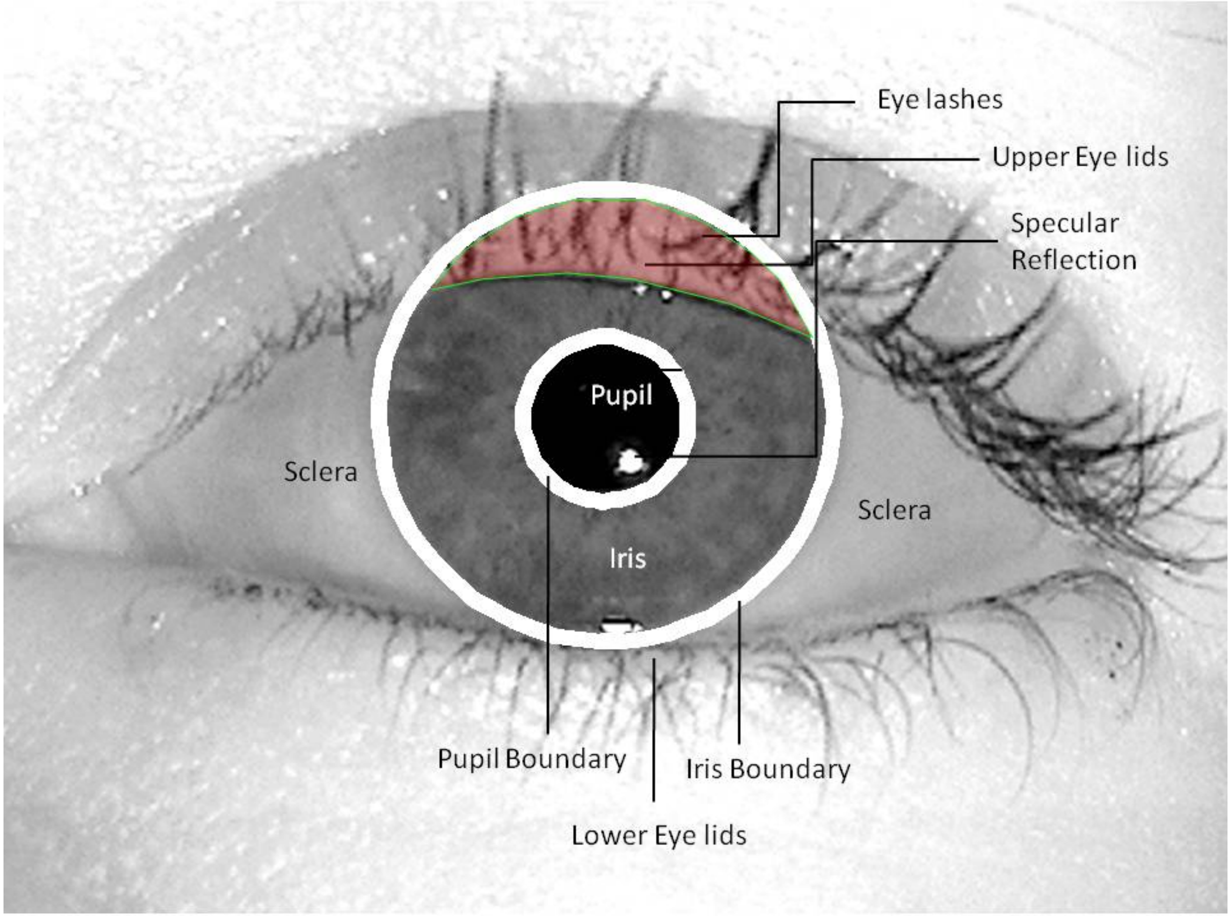


Figure 1.1: Illustration of some important regions of an eye, such as pupil, iris, sclera, eye lashes, eye lids, and specular reflection.

iris image preprocessing, (3) iris texture feature extraction and (4) feature matching. Figure 1.2 shows the flow chart of a typical iris recognition system.

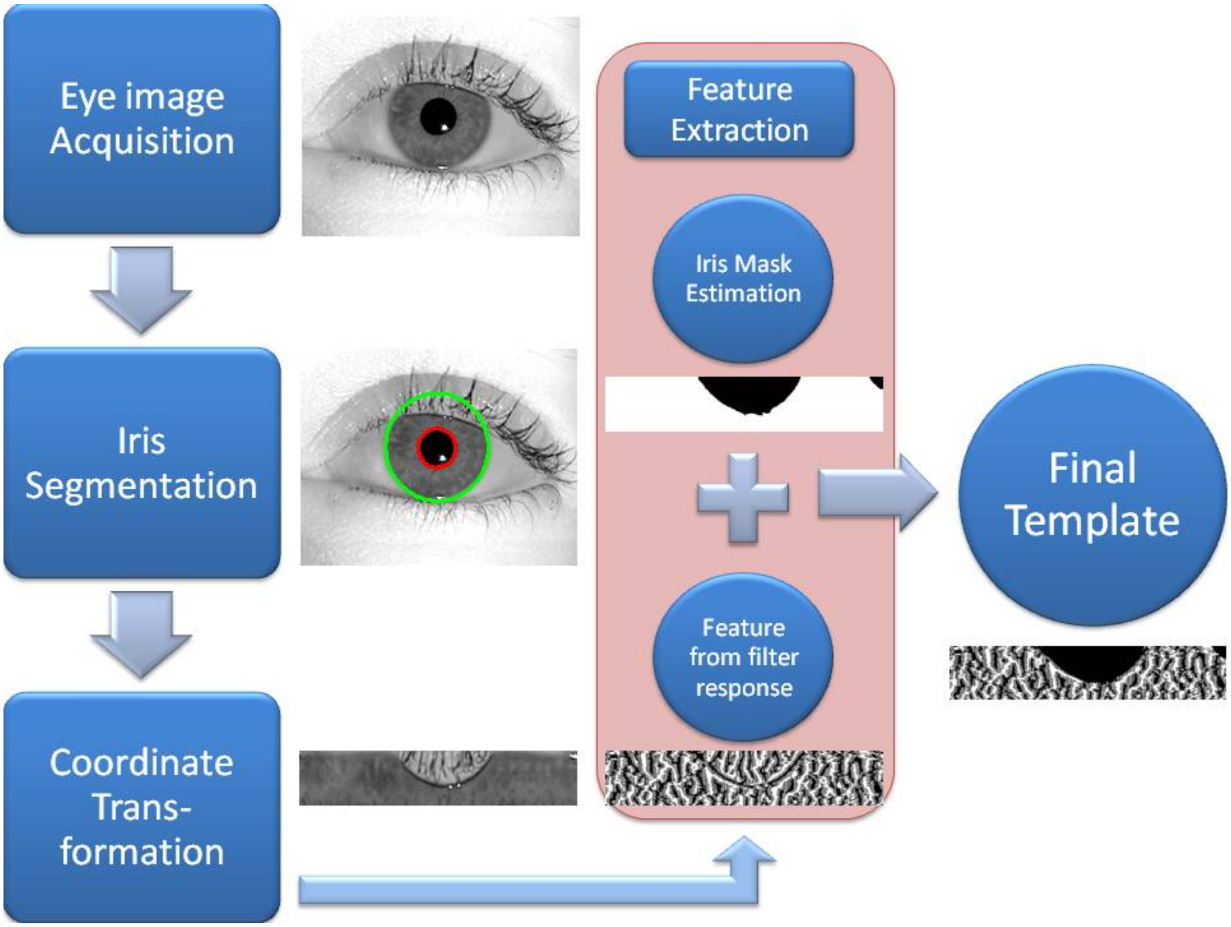


Figure 1.2: The flow chart of the process of a typical iris recognition system. Example images are placed besides to each stage of the process.

## 1.2 Necessity for Non-Intrusive and Long Range Iris Recognition

The advantages of iris recognition are its extraordinarily high recognition rate and its stability through out a person's lifetime. However, for the current and most popular iris recognition systems on the market, the major disadvantage is the stand-off distance: it is too small, typically 5-10cm. This constraint hinders the development of practical un-obtrusive iris recognition systems. If we want to make iris recognition useful in everyday life, this constraint needs to be further relaxed to achieve longer stand-off distance to make the system un-obtrusive. In section 1.3.3, we will introduce a new iris acquisition device which is relatively new to the commercial market and opens the possibility of capturing iris images from a longer distance.

## 1.3 Variability of the Iris Image Acquisition Device and the Resulting Iris Image Quality

Robust iris recognition is inherently related to the quality of the iris image and the acquisition optic. With a good optical image capturing device and sensor and the appropriate lighting to illuminate the iris, the captured image should reveal iris texture with a great level of detail, as shown in Figure 1.3.

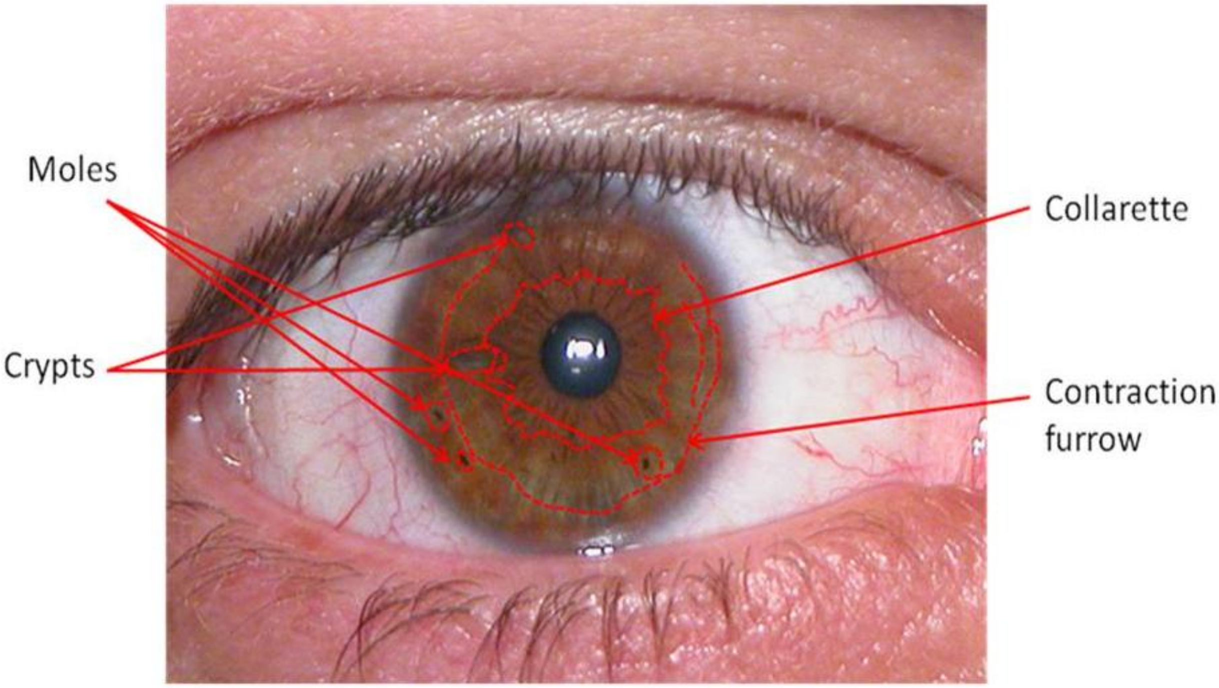
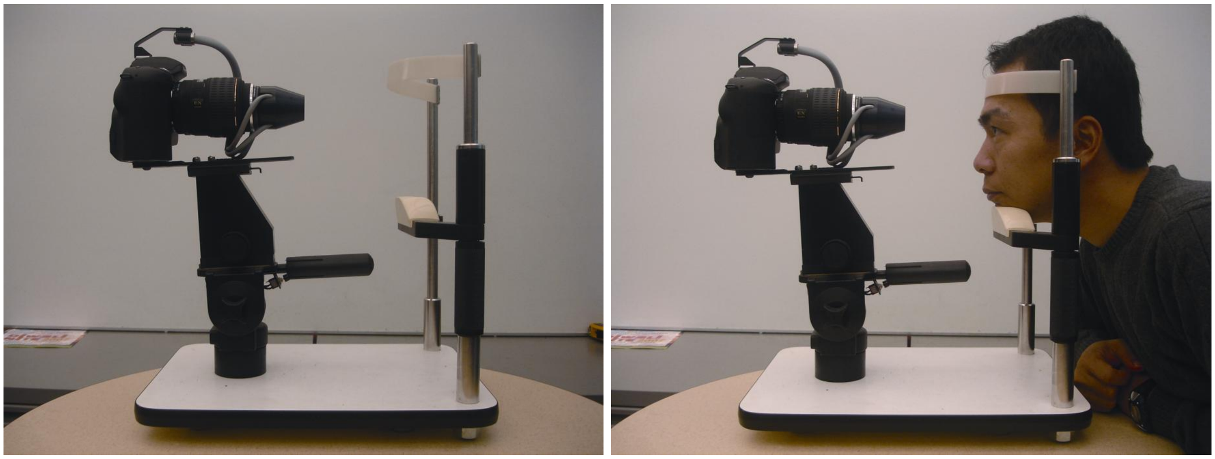


Figure 1.3: Illustration of crypts, furrows, collarette and moles in an iris image.

In Figure 1.3, one can see that the iris texture actually consists of many delicate details, including collarette, contraction furrows, crypts and moles. The location of these tiny muscular structures appear to be random across different eyes of different people. It is this randomness in their existence and location that gives an extremely low probability of false acceptance in a recognition task. In other words, the chances that two different irises would be classified as same is unlikely. Therefore, the quality of the image is extremely important in iris recognition. If we cannot get a high quality iris image for enrollment, it would be extremely difficult to get high recognition performance at later testing stages.

In following sub-sections, three different models of iris acquisition will be introduced. Each of them has its advantages and drawbacks in terms of image quality, as well as user-friendliness and level of non-intrusiveness, which are the two most important aspects of an iris acquisition system. Designers and engineers must take these into acccount in order to find the optimal trade-off between the high quality imagery and ease of use best suited for a particular deployment and operational scenario.



(a)

(b)

Figure 1.4: (a) Example picture of traditional iris acquisition device. (b) Example picture to illustrate how to use it.

### 1.3.1 First Generation Iris Image Acquisition Devices

The original first generation iris acquisition devices usually consisted of a high resolution imaging device (typically a digital camera), one or more light sources (illuminators), and an adjustable mechanical rack to position the subject's head at a very close proximity to the sensor. Figure 1.4 shows how a traditional first generation iris acquisition device looks and how it is operated and positioned with respect to a subject. Figure 1.5 shows an example of the iris image captured by such a device. From Figure 1.5, we can see that the quality of the captured image is so high that every detail of the iris texture is revealed clearly.

The drawbacks of such devices are, obviously, the difficulty of use and the slow turn-over rate of the system (very slow throughput). These systems are impractical in most real world applications, such as access control and border security where the throughput of the system is crucial. Such devices are typically confined to very specific medical applications that require extremely high resolution imagery of the eye.

### 1.3.2 Short Range Iris Image Capturing Devices

In the last decade, a lot of research has been done to enhance the usability of iris imaging devices by relaxing the constraints of the system as much as possible, while maintaining satisfactory matching performance. As a result, many new iris capturing devices have the ability to detect the location of a human face and eyes in an acquired image. Therefore, it does not require users to place their heads in a fixed location, allowing the ability to achieve a certain amount of stand-off distance between the user and the camera.

The LG IrisAccess 4000 is one such commercial device that is popular in the iris commercial market [4]. In this device, the iris camera, face camera and illuminator are all packaged in one relatively small form factor box, which makes it easy to move, install, and deploy in different locations. There are also up/down control buttons located on the front of the panel to allow

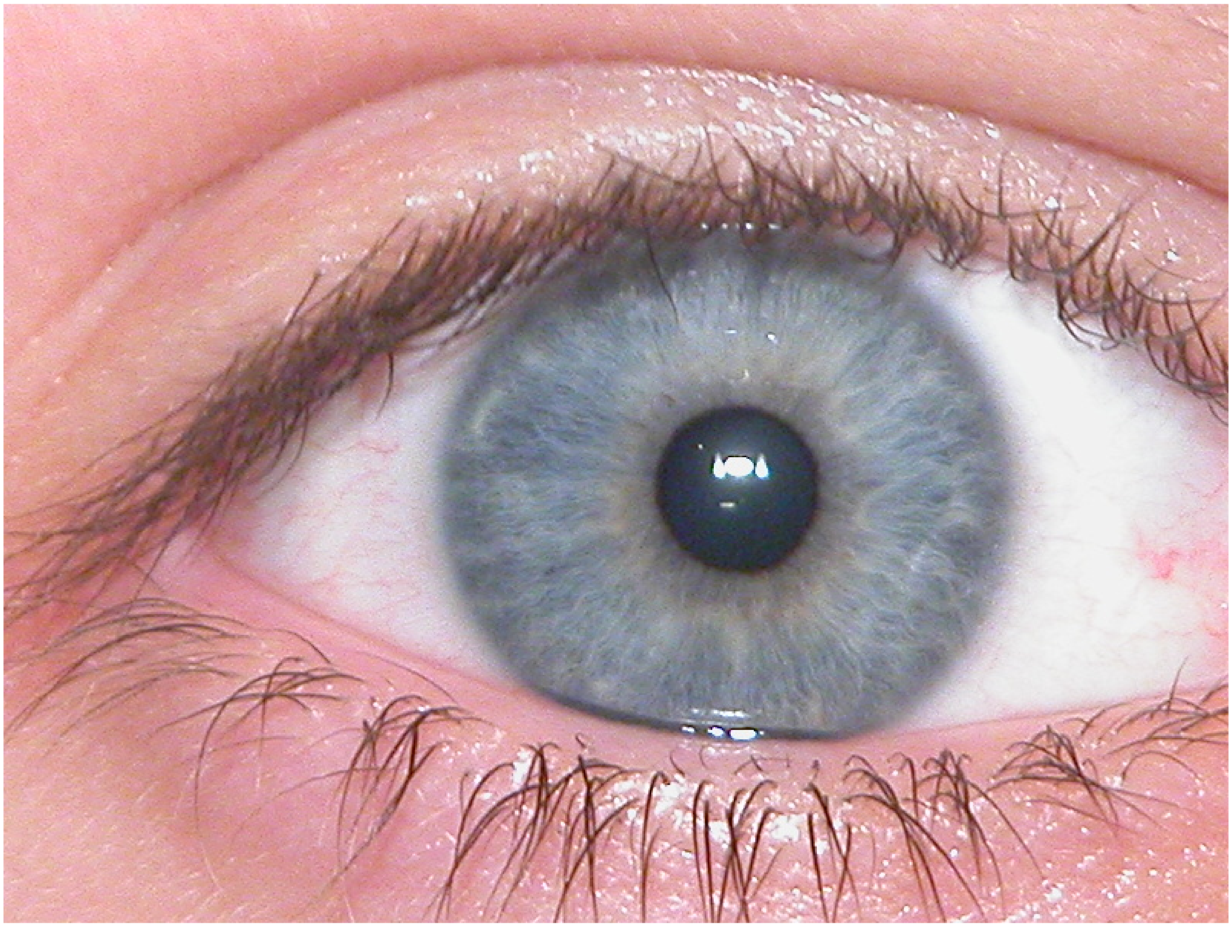
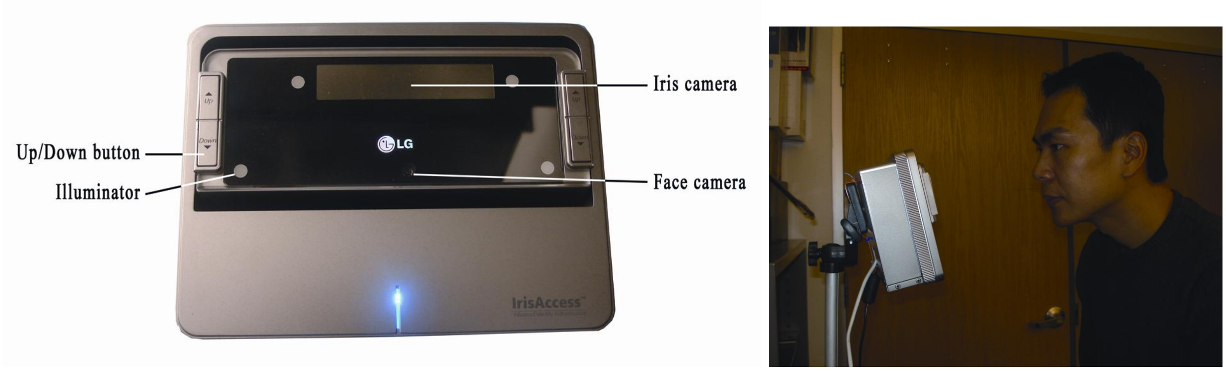


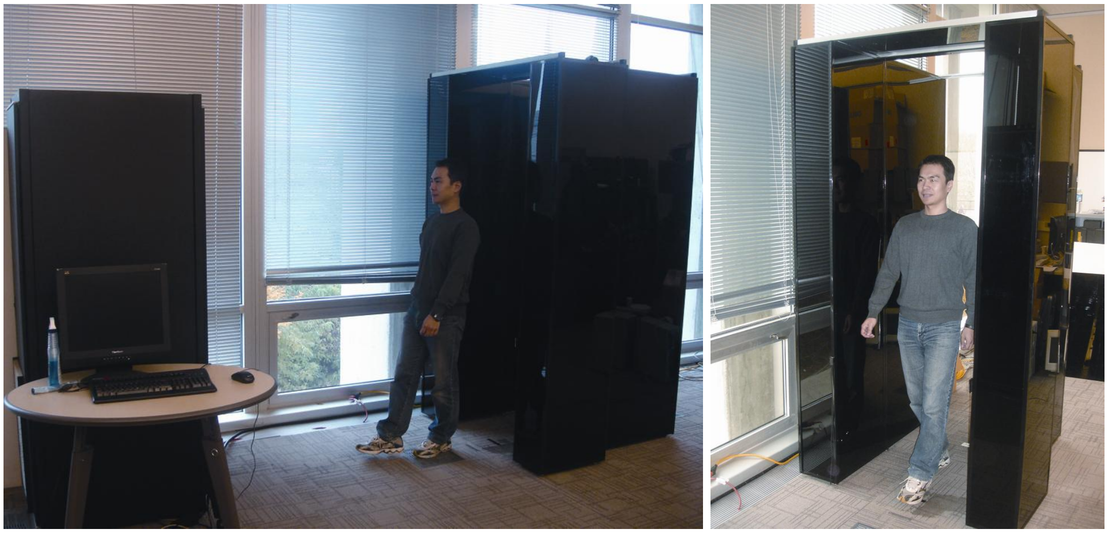
Figure 1.5: An example image captured by traditional iris acquisition device. The image quality is so high that all details of iris texture are revealed clearly.



(a)

(b)

Figure 1.6: (a) LG IrisAccess 4000 iris acquisition device and description of the functional unit. (b) An example image to illustrate how to use it.



(a) (b)

Figure 1.7: (a) IOM system. (b) Example picture of a subject walking through the portal.

users to adjust the angle of the camera, which compensates for people with different heights. Additionally, these new LG models are dual iris capture devices, i.e. they capture both irises from both eyes simultaneously. Conversely, some devices on the current market are single iris capture devices, which take more time to acquire both eyes and can lead to erroneous labeling of the acquired left/right eyes due to human error. Figure 1.6(a) shows a picture of frontal view of LG IrisAccess 4000 camera as well as its functionality buttons. Figure 1.6(b) shows a person using it to capture iris images.

### 1.3.3 Long Range Iris Image Capturing Devices –Iris-On-the-Move

High correct identification rates and low false matching rates make the iris biometric modality ideal for high volume applications such as border control and a myriad of access control applications where there are a large number of subjects that need to be tested in a relatively short amount of time. Therefore, the speed of acquisition and throughput of such a system becomes critical. Recently, high-speed iris capturing devices have become available on the market. These devices are capable of capturing iris images of both eyes in less than 3 seconds per person. One example of a high-speed system is the Iris-On-the-Move (IOM) system, manufactured by Sarnoff Corporation, shown in Figure 1.7. This system consists of three components: the portal through which subjects walk; the cabinet where the imaging devices reside, and a computer to control the device. The capturing process is made extremely easy. All the subjects have to do is to walk through the portal looking straight at the cabinet. The cameras inside the cabinet will automatically capture the iris images. Figure 1.7(b) shows the iris capturing process with a subject walking through the portal.

IOM has the capability of taking a video of subjects as they are walking through the portal. Therefore, the number of the eye images it captures is much more than a traditional iris capturing

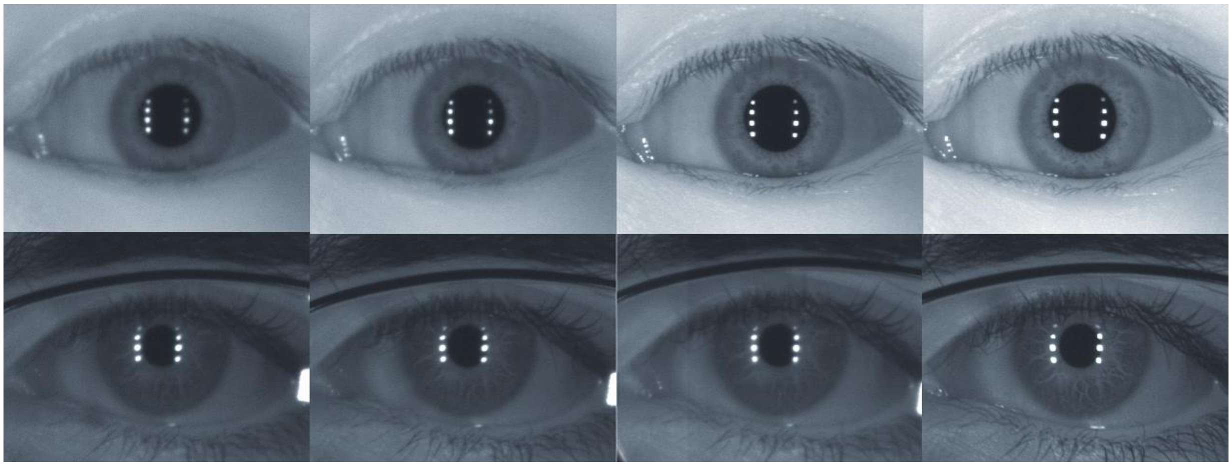


Figure 1.8: Example video frames taken by IOM system.

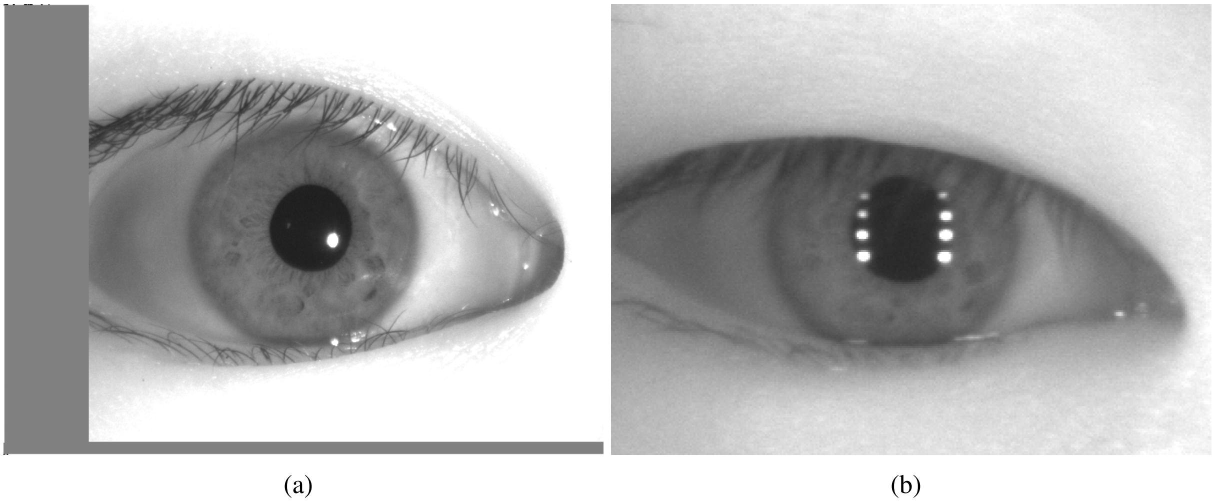
device. Let us assume that it takes one second for a person to walk through the portal, which is a pretty reasonable estimate. The standard video format records video at 30 frames/sec. Assuming that the eye region is inside the camera capture volume for only 1/2 of the time during which the subject is walking through the portal, the average number of iris images that can be captured for a single run should be $30 \times \frac{1}{2} = 15$. Assume out of these 15 images, only 1/2 of them are well focused, then the good quality iris images which can be captured in a single run is about 7 or 8 images. Figure 1.8 shows examples of video frames of a subject's eye captured by the IOM system.

There are both advantages and disadvantages in this system setup. An advantage of the setup is that, given the multiple frames that have been captured, an enrolled iris image can be matched with all of them. The final recognition result can be measured by selecting the best score among them. However, the disadvantage is the image quality of IOM system is usually very low, compared to the images taken by LG camera or a traditional iris acquisition device, as shown in Figure 1.9. Therefore, further improvement is necessary in order for IOM system can be practically usable in a real life scenario.

## 1.4 Basic Optics for IOM system

In this section, we will review some of the basic optical principles and design constraints that go into a long-range iris acquisition device, such as the IOM system. As these optical principles are introduced sequentially in each of the following sub-sections, the design philosophy, constraints, and limitations of long range iris recognition systems will become clear.

We will use Figure 1.10 to illustrate the basic structure of a typical optical imaging system. As shown in Figure 1.10, the purpose of a typical optical imaging system is to capture the light emitted from the object and collect it on a receptive plane, so the image on the plane resembles the details of the target object as much as possible. In most cases, this goal is achieved by using a lens or array of lenses. $D$ denotes the diameter of the entrance pupil, which is the maximal area of the lens that allows for the light to pass through. $\theta$ is the half-angle of the maximal cone of


(a)

(b)

Figure 1.9: The same eye captured by different iris cameras. (a) the image captured by LG IrisAccess 4000 (b) the image captured by Iris-On-the-Move. The detailed iris features are clearly visible in (a) but not very clear in (b).

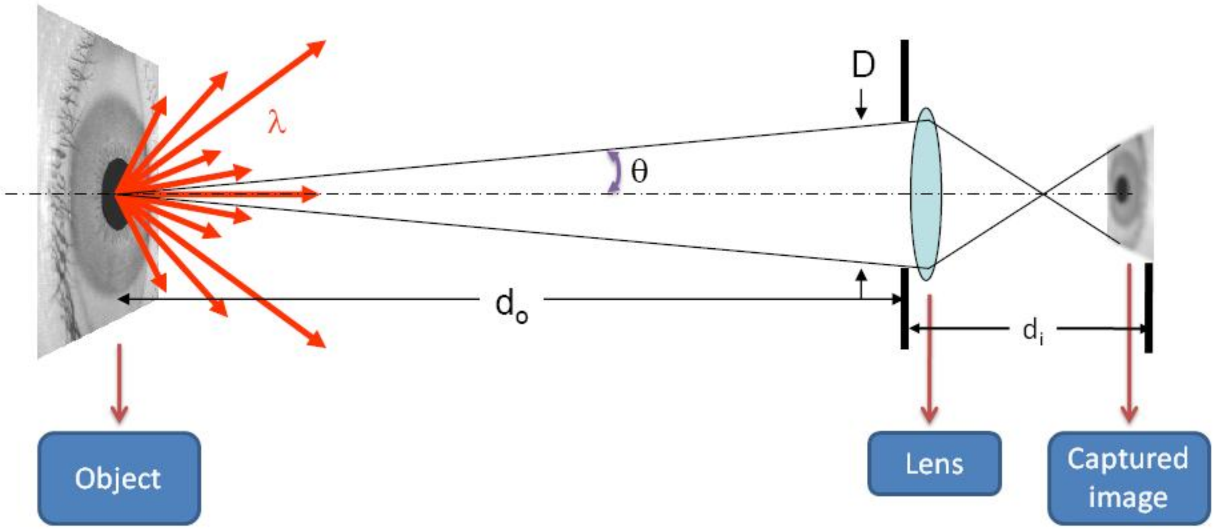


Figure 1.10: Illustration of a typical structure of an optical imaging system.

light that can enter the lens. $d_o$ denotes the distance of the object, which is the distance between the object and the lens. $d_i$ denotes the distance of the image, which is the distance between the image plane and the lens.

### 1.4.1 Numerical Aperture

In optics, the numerical aperture (NA) of an optical system is a dimensionless number that characterizes the range of angles over which the system can accept or emit light. In most areas of optics, the numerical aperture of an optical system is defined as:

$$NA = n \cdot sin\theta \tag{1.1}$$

where $n$ is the index of refraction of the medium in which the light is passing through. For the case of iris image acquisition, the light is passing through air, therefore, $n = 1$. For the purpose of iris acquisition, usually the distance of the object is much larger than the diameter of the entrance pupil, therefore, Eq. (1.1) can be approximated as:

$$NA = n \cdot sin\theta = sin\theta \approx \frac{D}{2d_o} \tag{1.2}$$

### 1.4.2 Minimum Resolvable Separation for Object

Once we know the numerical aperture (NA) and the wavelength ($\lambda$) of the system, we are able to compute the Minimum Resolvable Separation (MRS) at the object side. It is described by the following formula [5]:

$$MRS = \frac{1.22\lambda}{2NA} \approx \frac{1.22\lambda d_o}{D} \tag{1.3}$$

What Eq. (1.3) tells us is the minimal distance of two points on the object side when they can be clearly seen as two separated points on the image plane. This limitation is caused by the optical phenomenon called “diffraction”. Diffraction is directly related to the wavelength of the light. The shorter the wavelength, the smaller MRS we can achieve. However, MRS is also limited by the diameter of the lens. For example, under the normal visible light, the wavelength is about 550nm. Suppose the distance between the object and the optical imaging system is 100m. Table 1.1 shows the MRS value for three optical imaging systems: human eye, eagle eye and telescope.

### 1.4.3 Magnification Rate

For an optical system, if the distance of the object and the distance of the image have been fixed as constraints of the system, the focal length $f$ of the system can be computed by the following equation [5]:

$$\frac{1}{f} = \frac{1}{d_o} + \frac{1}{d_i} \tag{1.4}$$

Table 1.1: The Minimum Resolvable Separation (MRS) of three optical imaging systems, under the condition that the images are taken under normal visible light with $\lambda = 550nm$, and the distance of the object is 100m.

| Optical System | D (cm) | MRS (mm) |
|---|---|---|
| Human Eye | 0.3 | 22.4 |
| Eagle Eye | 1 | 6.71 |
| 8” Telescope | 20 | 0.33 |

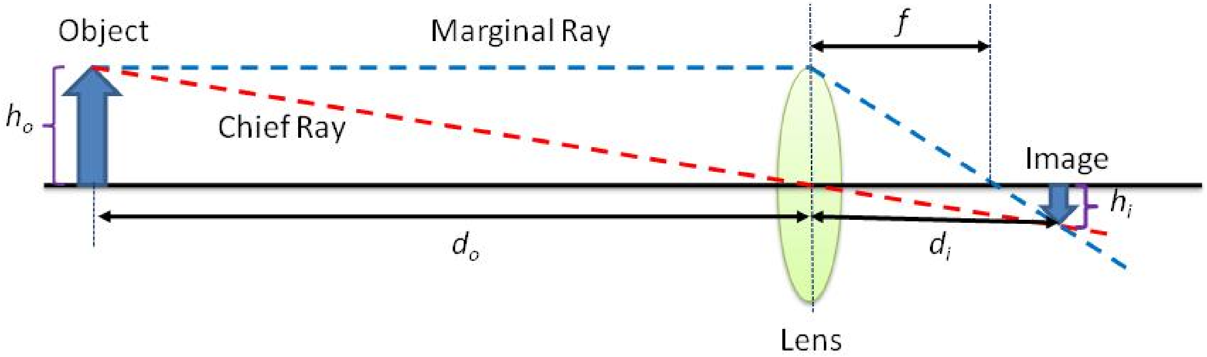


Figure 1.11: The ray path of an optical system. In this figure, one can clearly see the relation between focal length $f$, the distance of the object $d_o$, the distance of the image $d_i$, the height of the object $h_o$, and the height of the image $h_i$.

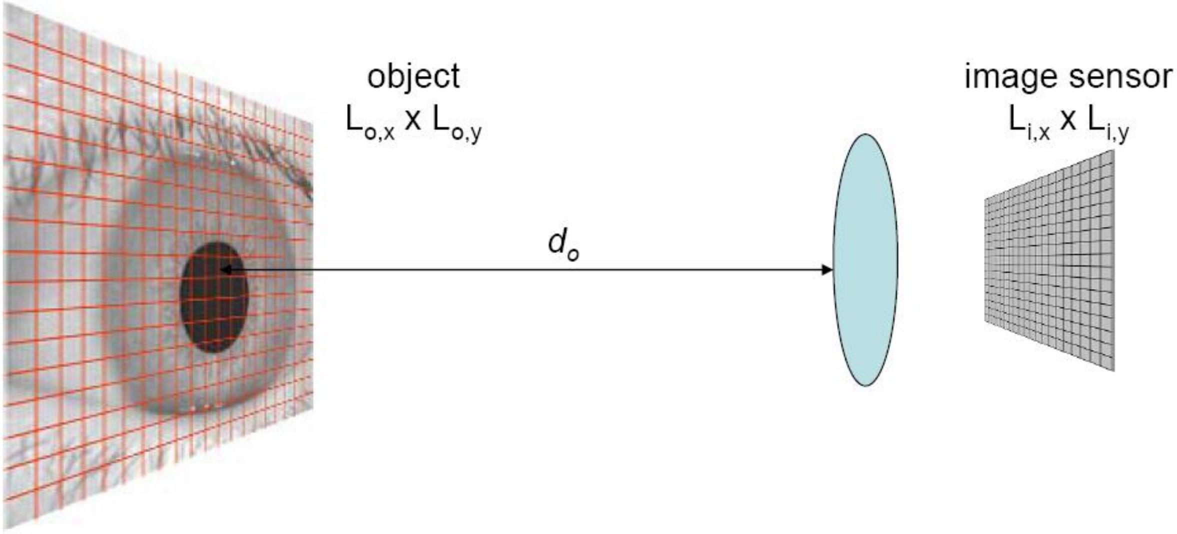


Figure 1.12: The back projection from image sensor array to the target object. The red grid on the target object shows the size of Object Referred Pixels (ORP).

Figure 1.11 shows the relation between focal length $f$, the distance of the object $d_o$ and the distance of the image $d_i$, as well as the ray path of the optical system.

The desired magnification rate of the system can be computed using equation [5]:

$$M \equiv \frac{h_i}{h_o} = \frac{d_i}{d_o} = \frac{f}{d_o - f} \tag{1.5}$$

For example, for an optical system which has a focal length of 50mm, if the object is 10 meters away, then magnification rate is $M = 0.05/(10 - 0.05) = 0.005$. If the lens is replaced with one with a focal length of 500mm, then the magnification rate becomes $M = 0.5/(10 - 0.5) = 0.0528$, which is ten times more than the first lens.

### 1.4.4 Object Referred Pixels

In digital photography, the basic unit that forms an image is a pixel unit. An image that has a higher number of pixels has a “higher resolution” than an image composed of a lower number of pixels. Object Referred Pixels (ORP) defines how much length of the target object is represented by one pixel of the final image. The smaller the ORP, the higher the resolution of the image. Therefore, if we would like to design a good optical system which captures the details of the object, like texture in iris region, we are striving to achieve very low ORP values.

Suppose we use an optical sensor to capture images in the digital form, the semiconductor sensor is arranged as an array in two-dimensional space, as shown on the right side of Figure 1.12. Assume the image sensor array is the size $L_{i,x} \times L_{i,y}$. Now imagine we back-project the grid of the image sensor array to the real target object, and it results in the figure shown on the left side of Figure 1.12. The red grid depicts how much space is represented by one pixel from the image side. The length of a side of the grid is the ORP.

The magnification rate $M$ can be computed from ORP. Suppose the lengths of the physical sensor is $L_{pix,i}$ and the object referred pixels is $L_{pix,o}$, magnification rate $M$ can be computed as following:

$$M \equiv \frac{L_{pix,i}}{L_{pix,o}} = \frac{f}{d_o - f} \approx \frac{f}{d_o} \tag{1.6}$$

Therefore, if we already know the focal length $f$, the distance of the object $d_o$, and the spacing between digital image sensors, we can compute the ORP $L_{pix,o}$ by the following equation:

$$L_{ORP} = L_{pix,i} \cdot \frac{d_o - f}{f} \approx \frac{L_{pix,i} \cdot d_o}{f} \tag{1.7}$$

For example, for a digital optical sensor array that has $8\mu m$ spacing across each sensor array element, if we use a lens that has focal length of $100mm$, and the distance of the object is $2m$, the object referred pixels can be computed as:

$$L_{ORP} \approx \frac{L_{pix,i} \cdot d_o}{f} = \frac{8\mu m \cdot 2m}{100mm} = 160\mu m \tag{1.8}$$

For a normal eye that has a typical iris region of diameter $11mm$, such a system can capture an image of the iris and get a photo that has $11mm/160\mu m \approx 69$ pixels across the entire iris region.

### 1.4.5 Field of View

Field of View (FoV) is the area that is visible through an optical imaging device. Suppose our optical sensor array is composed of $N_{pix} \times M_{pix}$ sensors and spacing between the sensors is $L_{pix}$, then the dimension of the FoV is $L_{ORP} \times \{N_{pix}, M_{pix}\}$.

For example, suppose the sensor array is composed of a $2048 \times 2048$ pixel (4MP) sensor, with $8\mu m$ spacing between pixels; the distance of the object is $d_o = 2m$, and the focal length is $f = 100mm$:

$$FoV = L_{ORP} \cdot N_{pix} \approx \left[ L_{pix} \cdot \frac{d_o}{f} \right] \cdot N_{pix} = \frac{8\mu m \cdot 2m}{100mm} \cdot 2048 = 0.328m \tag{1.9}$$

For the purpose of iris acquisition, if we want the desired image to contain 100 pixels across an $11mm$ iris area, and the sensor array contains $N \times M$ pixel sensors, then $FoV = L_{ORP} \cdot N_{pix} = \frac{11mm}{100} \cdot \{N, M\}$.

### 1.4.6 F-number

For optical imaging devices that have a variable aperture, the F-number of a system is the quantitative measure of the entrance pupil in terms of the focal length of the lens. In mathematical form this is expressed as:

$$F \equiv \frac{f}{D} \tag{1.10}$$

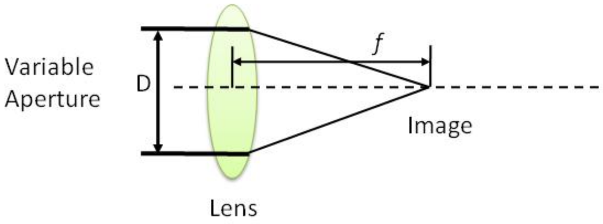


Figure 1.13: Illustration of F-number.

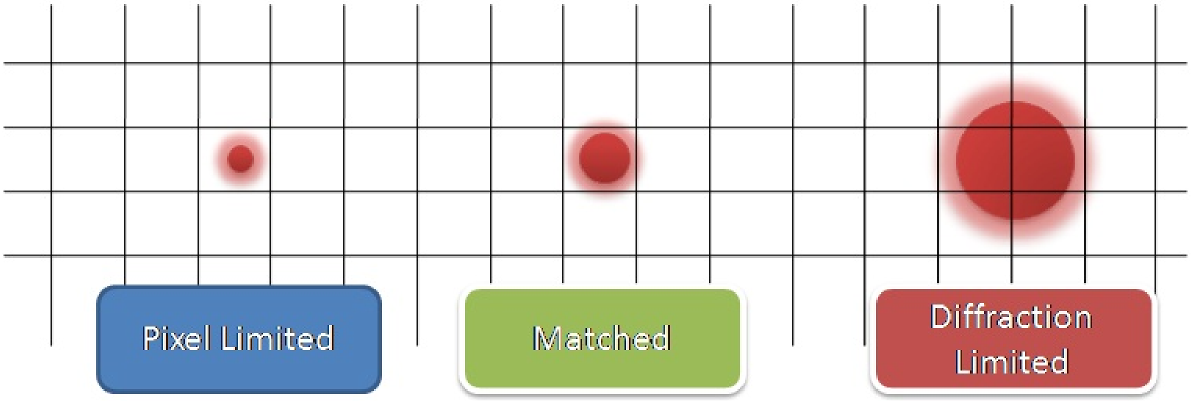


Figure 1.14: Illustration of the difference between pixel-limited and diffraction limited situation.

where $f$ is the focal length and $D$ is the diameter of the entrance pupil, as illustrated in Figure 1.13. By convention, the F-number is expressed as "$f/\#$", with $\#$ replaced with the true quantitative value. For example, if the focal length is 16 times the pupil diameter, the F-number is $f/16$. The greater the F-number, the less light per unit area that transmits to the image plane of the system. The amount of photons transmitted into the optical sensor decreases with the square of the F-number. Therefore, the greater the F-number, the longer it takes the optical sensor to get enough light. In the terminology of photography, it takes a longer exposure, or integration, time. The smaller the F-number, the shorter the exposure time required. For the iris, the longer the exposure, the more IR photons collected, increasing the signal-to-noise ratio in the image acquired; however, the image is prone to motion blur if the subject is not still and is moving, as in the case for the IOM system.

### 1.4.7 Resolution and Diffraction

Recall that in subsection 1.4.2, we introduced the concept of Minimum Resolvable Separation (MRS) on object side, which is limited by the optical phenomenon "diffraction". By using the concept of magnification rate and Eq. (1.6), we can recover the MRS on the image side:

$$MRS_{img} \approx \frac{1.22\lambda d_o}{D} \cdot M \approx \frac{\lambda d_o}{D} \cdot \frac{f}{d_o} = \lambda \frac{f}{D} = \lambda F \tag{1.11}$$

In subsection 1.4.4, we introduced the concept of Object Referred Pixel (ORP), which also

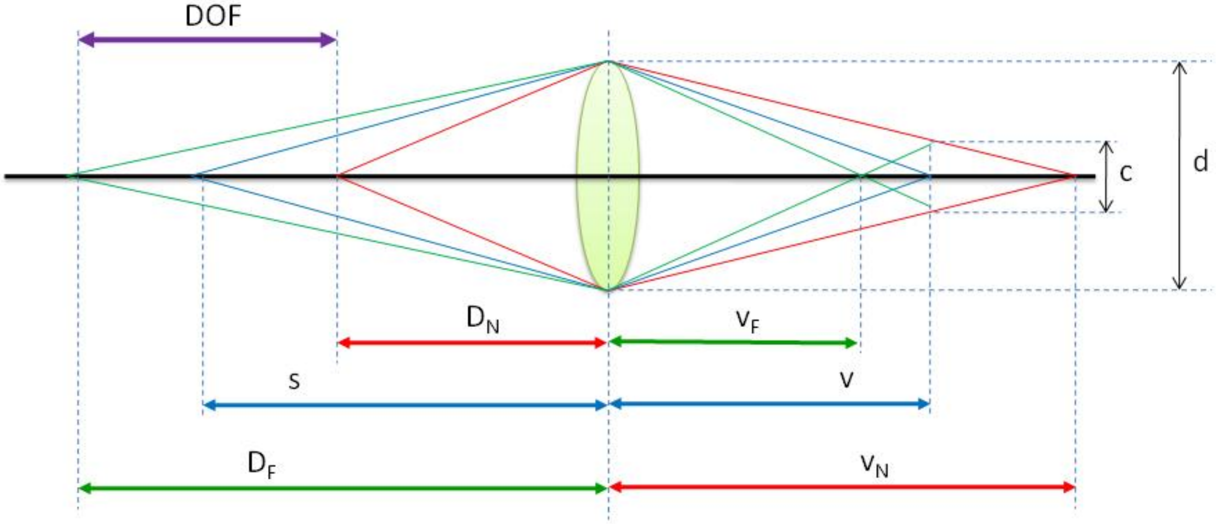


Figure 1.15: Illustration of “Depth of Field”.

tells us the limitations of the resolution of the image. The limitation derived from ORP comes from the physical limitation of semiconductor manufacturing process that produces the digital optical sensor, and the limitation derived from MRS comes from the ubiquitous phenomenon “diffraction”. These two limitations together bound the ability of the optical instrument in terms of its image resolvability. Figure 1.14 shows the difference between pixel-limited resolution and diffraction limited resolution.

By using these two limitations together, we are able to recover the optimal design parameter for our optical system. Let

$$MRS_{img} = L_{pix} \tag{1.12}$$

We can get

$$\lambda F = L_{pix} \tag{1.13}$$

Or

$$F = \frac{L_{pix}}{\lambda} \tag{1.14}$$

For example, for the system that has $L_{pix} = 8\mu m$, imaging with infra-red illumination with wavelength $\lambda = 850nm$, the optimal F-number of the system should be $F = 8\mu m/850nm \approx 10$.

### 1.4.8 Depth of Field

Depth of field (DOF) is another very important concept in photography that is a very important factor in the acquisition of irises. In optics and photography, DOF is the depth of the scene that appears to be acceptably sharp in the final image. Theoretically speaking, a lens can only precisely focus at one particular point. Objects that are in front and behind of that point will appear to be blurred to some degree. However, in digital photography, if the “circle of confusion” of the blurring effect is less than the size of one pixel, it cannot be noticed in the final image. Therefore, the depth of field can be computed using the constraint that the circle of confusion is less than or equal to one pixel.

Figure 1.15 shows a lens system in which we will illustrate and develop the formula for DOF. Suppose the subject at distance $s$, is in focus at image distance $v$. If we freely move the subject either toward the lens or away from the lens, the final image starts to blur. Assume that the diameter of the blur spot, the “circle of confusion” (denoted as $c$), reaches the maximum size that we can tolerate at the distance $D_F$ and $D_N$, then the DOF is defined as $D_F - D_N$. Assume the image appears at $v_F$ and $v_N$ when the subject is at distance $D_F$ and $D_N$, respectively. From triangular similarity, we can get

$$\frac{v_N - v}{v_N} = \frac{c}{d} \tag{1.15}$$

$$\frac{v - v_F}{v_F} = \frac{c}{d} \tag{1.16}$$

By using Eq. (1.10), we can represent $v_F$ and $v_N$ by F-number:

$$v_N = \frac{fv}{f - Fc} \tag{1.17}$$

$$v_F = \frac{fv}{f + Fc} \tag{1.18}$$

We should not forget about the basic equation of thin lens, which states

$$\frac{1}{s} + \frac{1}{v} = f \tag{1.19}$$

Applying Eq. (1.19) with $v_N$, $D_N$, $v_F$ and $D_F$, we get

$$\frac{1}{D_F} + \frac{1}{v_F} = \frac{1}{f}$$

$$\frac{1}{D_N} + \frac{1}{v_N} = \frac{1}{f} \tag{1.20}$$

Solving Eq. (1.17), (1.18) and (1.20), we can get

$$D_N = \frac{sf^2}{f^2 + Fc(s - f)} \tag{1.21}$$

$$D_F = \frac{sf^2}{f^2 - Fc(s - f)} \tag{1.22}$$

$$DOF = D_F - D_N = \frac{sf^2}{f^2 - Fc(s - f)} - \frac{sf^2}{f^2 + Fc(s - f)} \tag{1.23}$$

### 1.4.9 Design of an Iris Acquisition System

Given some of the constraints and according to the optical principles stated in the previous subsections, we are able to design the optimal parameters for a new iris acquisition system.

For example, in long range iris acquisition systems, the most important system constraints are the following: (1) Object Distance $d_o$: How far away can the subject stand relative to the system, so his/her iris images can still be captured? (2) Field of View FoV: How can we enlarge the capture zone so that the system can still capture iris images even when the subject moves slightly away from the appropriate location? (3) Wavelength $\lambda$: For iris acquisition, we usually use infra-red light to image the iris; therefore, $\lambda$ is longer than the normal visible light. (4) Object Referred Pixel $L_{ORP}$: For iris acquisition, we would like to maintain a certain level of $L_{ORP}$, so the iris detail can be fully captured. The higher the $L_{ORP}$, the greater the detail we can see in the resulting images. We also want a suitable Depth-of-Field (DoF) so the iris is in focus, and the user does not have to move in mm precision for the system to acquire an in-focus iris image.

In this example we use the four parameters $\{d_o, FoV, \lambda, L_{ORP}\}$ to demonstrate how to derive other system parameters. First, the size of optical sensors $L_{imager}$ is fixed; therefore, we can compute the magnification rate $M$ by

$$M = \frac{L_{imager}}{FoV} \tag{1.24}$$

Second, from Eq. (1.6), we can get

$$f = \frac{Md_o}{M+1} \tag{1.25}$$

Therefore, we can compute the focal length $f$ from $\{M, d_o\}$. Next, we can compute the number of pixels the system can render on image $N_{pix}$ by

$$N_{pix} = \frac{FoV}{L_{ORP}} \tag{1.26}$$

If we know $N_{pix}$ and the size of the optical sensor $L_{imager}$ , we can compute $L_{pix}$ by

$$L_{pix} = \frac{L_{imager}}{N_{pix}} \tag{1.27}$$

Next, from the parameter $\{L_{pix}, \lambda\}$, we are able to compute the F-number by using the diffraction limit

$$F = \frac{L_{pix}}{\lambda} = \frac{L_{ORP} \cdot M}{\lambda} \tag{1.28}$$

For the depth of field, the parameter "circle of confusion" $c$ is defined as the size of the single pixel, which is $L_{pix}$, and the subject distance is about $3m$, by applying Eq. (1.23) we will be able to compute the DOF.

According to the parameter listed in [6], the DOF of the IOM system should be

$$\begin{aligned}
DOF &= \frac{sf^2}{f^2 - Fc(s-f)} - \frac{sf^2}{f^2 + Fc(s-f)} \\
&= \frac{3 \cdot (0.21)^2}{0.21^2 - 16 \cdot (7.4 \times 10^{-6})(3-0.21)} - \frac{3 \cdot (0.21)^2}{0.21^2 + 16 \cdot (7.4 \times 10^{-6})(3-0.21)} \\
&= 0.0449
\end{aligned} \tag{1.29}$$

## 1.5 Problem Definition and Research Goal

In this thesis, the research goal is to further improve the iris image quality and overall recognition performance of current IOM-type systems which are able to capture multiple images as a person walks through the optimal focal point of the acquisition. Additionally we specifically address the challenge of using IOM-type images for database enrollment. Currently, in order to achieve high recognition performance, the enrolled iris images have to be sharp and clear. Usually people enroll new subjects in the database using iris images captured by short range iris acquisition devices, such as the PIER device or LG4000 system because of the high quality. Then the high-quality images are used to be matched against those acquired through an IOM-type system. However, it would be much more convenient if we could directly enroll data by using images captured by Iris-On-the-Move system. We also inherently make use of the multiple acquired images which have so far been ignored. Typically, only the best-matched iris images have been used in the video sequence of acquire images. Thus, in this thesis we address several challenges :

1. How do we locate the eye (and iris) in the video stream and perform iris segmentation? What are the challenges that we face particularly for the long-range iris acquisition system?
2. How do we segment IOM-type iris images, particularly due to the IOM illumination setup which causes specular reflections on the pupil boundary and on the iris region?
3. On an iris image, which part is the true iris texture region and which part is not? How can we detect those regions efficiently and accurately?
4. Can we further refine the iris mask based on an initial inaccurate estimation?
5. How do we enhance the quality of iris images and make use of the video stream of acquired irises?
6. How do we improve matching scores? The default operating setting of the long-range iris acquisition system may not be the optimal configuration. If we can propose a novel enrollment/identification scheme for the long-range iris acquisition system, is it possible to further enhance the original system performance?

These are the problems that we address in this dissertation.

# Chapter 2

# Robust Iris Segmentation Algorithm for Images Acquired by the Long-Range Iris Acquisition System

The high performance of iris recognition relies on the precise localization of the iris boundary. If the iris boundary is not estimated correctly, it will inadvertently damage recognition performance, even with a great iris feature extraction algorithm and recognition matching algorithm. In this chapter, we describe a novel iris segmentation algorithm designed specifically for IOM and other similar long-range acquisition systems, which is efficient, and, more importantly, very robust to specular reflection and variation in pupil sizes. The irregular spatial distribution of specular reflections and the consistent image blur are two characteristics of iris images captured by the Iris-On-the-Move system, which cause traditional iris segmentation algorithms to fail. Experimental results show the effectiveness and the efficiency of the proposed algorithm, compared to the traditional iris segmentation algorithms reported in literature.

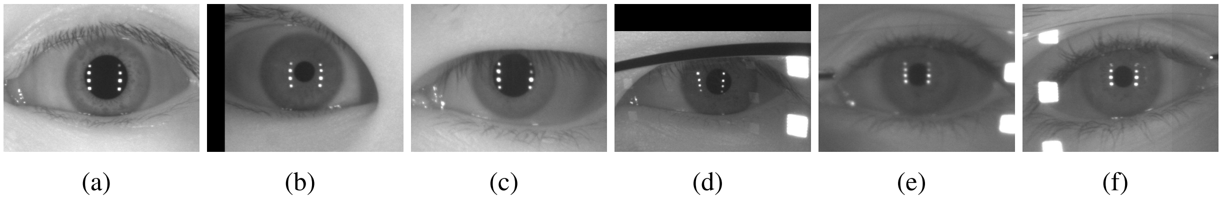


Figure 2.1: Example images, captured by IOM. The specular reflection causes problems because their position relative to the pupillary boundary is indefinite. They can be (a) inside the pupil; (b) outside the pupil (c) on the pupillary boundary (d) span the pupil and iris region. (e) and (f) shows another of great challenges of IOM images, which is image blurring plus strong specular reflection.

## 2.1 Introduction

Iris segmentation is one of the most vital components of any iris recognition system. Incorrect or errorneous segmentation results propagate the error to the next stage of feature extraction, inevitably decreasing the recognition performance. Due to the nature of iris normalization, which is converting the annular iris region to polar coodinate system, any mis-segmentation of the iris area leads to a non-linear deformation of the iris which causes the feature extraction stage to extract features that are mis-aligned in a non-linear fashion. There are many challenges in iris segmentation: the first problem is dealing with the dominant specular reflections on the pupil boundary and on the iris region caused by the IOM near-IR illumination. These specular reflections form artificial edges that are stronger than the ordinary iris boundaries, causing boundary detection algorithm to fail or be misleaded. This is a challenging problem because the specular reflections do not appear at fixed locations on the eye images and their positions depend on the location of the subject as they enter the IOM portal. Specular reflections can be inside the pupil, on the boundary between the pupil and iris, or outside the pupil area and inside the iris region, as shown in Figure 2.1. The random locations of specular reflection greatly impact the performance of iris segmentation algorithm. Another challenge we face with such a system is the inherent image blur that is present in most images. If an eye image is taken in a less constrained environment, where the subjects can freely move their eyes or head, there is a high probability that the captured eye images are blurred due to head or body motion. Image blur can also happen if the iris images are captured outside the optimal focus point (i.e. when the person is moving outside the depth-of-field) of the system. From a segmentation algorithm point of view, most algorithms rely heavily on the edge map of the raw eye images. If the input eye image is blurred, this edge map will not contain a strong edge at the pupillary and limbic boundaries, causing the traditional segmentation algorithms to fail. When a blurred image contains strong specular reflection, as shown in Figure 2.1 (f), the edge detector will pick up the edge contributed by specular reflection, rather by pupillary or limbic boundary. Therefore, the traditional segmentation results will be erroneous in such cases. Another factor that also needs to be considered in the design process is the efficiency of the iris segmentation algorithm. Many iris segmentation algorithms are built upon the algorithms proposed by [7] and [8], which are based on the circular Hough Transform and the integral-differential operator, respectively. The former approach requires significant computation to determine the pupillary and limbic boundary candidates. Therefore, more efficient methods which do not depend on the Hough Transform, and are more efficient than the integral-differential operations, are highly desirable, as long as they can provide the same level of accuracy in locating the pupillary and limbic boundaries. In this thesis, we proposed a novel iris segmentation algorithm which is very robust to specular reflection, blurring of the image (whether due to motion blur or out-of-focus blur), and are very efficient in terms of computational complexity. This is also important from a throughput point of view, the faster the processing, the higher the throughput of people passing through such a system (envision an airport security scenario where delays can cause significant distress).

This chapter is organized as follows: in section 2.2, a literature review is given to show what approaches have been proposed in the field of iris segmentation. In section 2.3, we describe the details of the proposed algorithm and the motivation behind it. In section 2.4, experimental results are presented and compared to other segmentation algorithms. Finally, we provide a summary discussion and conclusion in section 2.5.

## 2.2 Previous Work

The history of the research of iris segmentation is as old as the work related to the iris recognition itself. A number of papers in this area present various approaches to find the pupillary and limbic boundaries. In 1993, Daugman proposed to use an integral-differential operator to search over the parameter space (x,y,r) on the image plane [2]. Wildes proposed to use circular Hough Transform to locate the pupillary and limbic boundaries in 1997 [7]. Most of the research that followed has tried to improve upon these two algorithms in some way. Huang et al. suggested rescaling the image for fast computation, then applying an edge detector and integral-differential operator to locate the limbic and pupillary boundaries, respectively [9]. Liu et al. used the Canny edge detector to detect the rough pupillary boundary, followed by computation of the crossover points from a set of chords on the circle boundary [10]. In 2004, Sung et al. used the Canny edge detector and bisection method to find the pupillary boundary, and they also discovered the collarette boundary by using histogram equalization and high-pass filtering [11]. There are also correlation-based methods reported in the literature. Due to the high cost of computing the circular Hough Transform, Thornton et al. proposed fast approximation to the Hough Transform by 2D cross-correlation with a set of contour filters [12]. The response of contour filters can be stacked into a 3D image response cube, and the maximal value in this three dimensional image cube will indicate optimal circular parameters (x,y,r) of the pupillary and limbic boundary. However, none of these methods consider the case when strong specular reflections are present in the eye image, especially when their locations are locally mixed with either pupillary or limbic boundaries. What makes the problem even worse is that sometimes due to the arrangement of the illuminators, the shape of the specular reflection is similar to that of an arc, or another circle in the eye image. Figure 2.1 shows many examples of this. In such cases, the edge map based algorithms will eventually fail because the edge produced by the specular reflection is much stronger than the pupillary or limbic boundary. Therefore, no matter what kind of circular edge detector is used in the following stage, the segmentation results will not be correct.

Strong specular reflections are frequently present in eye images when we use the IOM system as our iris acquisition device [6]. IOM has the advantage that it can capture iris image from a distance, when subjects are walking through the portal. This design is extremely user-friendly and un-obtrusive for most people. It is very suitable to deploy in cases where high speed and high volume throughput are required. However, due to the design of the IOM system, the eye images captured always have eight white dots, four at each side. The positions of these specular reflections are random and cannot be predicted beforehand. The relative position between specular reflection and the iris boundaries are also unpredictable. If we use traditional iris segmentation algorithms, they fail miserably, as shown in our experimental results in section 2.4. Therefore, a novel iris segmentation method is needed to solve this problem.

## 2.3 Proposed Robust Iris Segmentation Algorithm

In this section, we introduce our proposed robust iris segmentation algorithm, which is especially applicable for IOM-type images. Our basic goal is to correctly locate the pupillary boundary first, then based on the pupillary boundary, estimate the limbic boundary. Our reason for this is that the pixel intensity in the pupil area is typically the lowest, compared to other regions of the eye image. Therefore, a simple technique like hard thresholding on a pixel value can easily and efficiently locate the pupil, which makes our initial stage of the algorithm efficient and simple. We describe each step of our algorithm in the following subsections, while providing images to illustrate the idea.

### 2.3.1 Specular Reflection Removal

In order to achieve an accurate iris segmentation result, we have to remove the specular reflections at an early stage of the segmentation process. Only when we reduce or fully eliminate the effect of these reflections, can the segmentation algorithm perform adequately at the later processing stages. Our proposed scheme is very effective and can reduce or remove strong specular reflections. The basic steps are illustrated as follows: first, we locate the positions of specular reflections by simple thresholding. Due to the nature of the optical setup we can ensure that this approach will always hold for an IOM-type system. This is very effective because specular reflections are always regions with the highest pixel intensity in an IOM-type image. The pixel intensity in the region of skin or sclera is usually lower than the specular reflection. Second, we inpaint the region of specular reflection by an optimal value, so that the regions after painting resemble the neighboruing iris regions. Thus the eye image starts to appear less contaminated. Next we apply median filters to smooth the whole image. Median filtering is an effective technique to remove the noise in an image [13, 14], and if we carefully observe the resulting iris image from step two, we can clearly see there exists a boundary at every specular reflection. By applying median filters we can successfully remove these boundary artifacts and smooth the whole iris image. Figure 2.2 shows an example input image going through the steps of specular reflection removal. As one can see, the resulting image is a smooth version of the input image, without any specular reflections observable.

### 2.3.2 Locating the Pupil

After we remove the specular reflections, it is very easy to locate the pupil. Because the pupil is the darkest region in the whole image, a simple scheme, such as pixel intensity thresholding, is enough to find the pupil. However, some other components around the eye may also have low intensity, for example, eyebrows or an eyeglasses frame. We distinguish the region of pupil from such artifacts by analyzing the geometric property of the connected components. There are many geometric properties that we can choose to optimize the result. For example, the width and height of the pupil area should be confined within a particular range. Many other constraints are useful as well, as long as they provide discriminative property for the pupil from other artifacts. After the pupil area is located, we can measure the center of the mass to locate the center of the

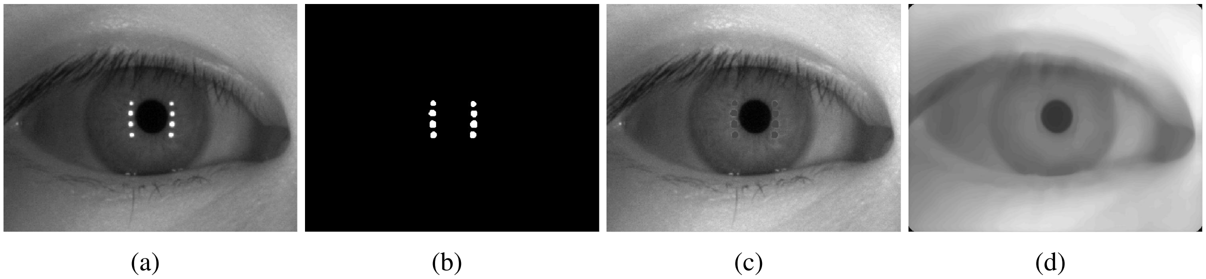


Figure 2.2: Examples of the scheme for specular reflection removal. (a) the input image (b) location of the specular reflection, discovered by hard thresholding (c) image after painting the region of specular reflection with a value which is most likely to appear on iris texture (d) image after applying median filtering on (c).

pupil. Then the radius can be estimated by using the longest distance between the center of mass and any point inside the region. The whole process is illustrated in Figure 2.3.

### 2.3.3 Locating the Iris Boundary

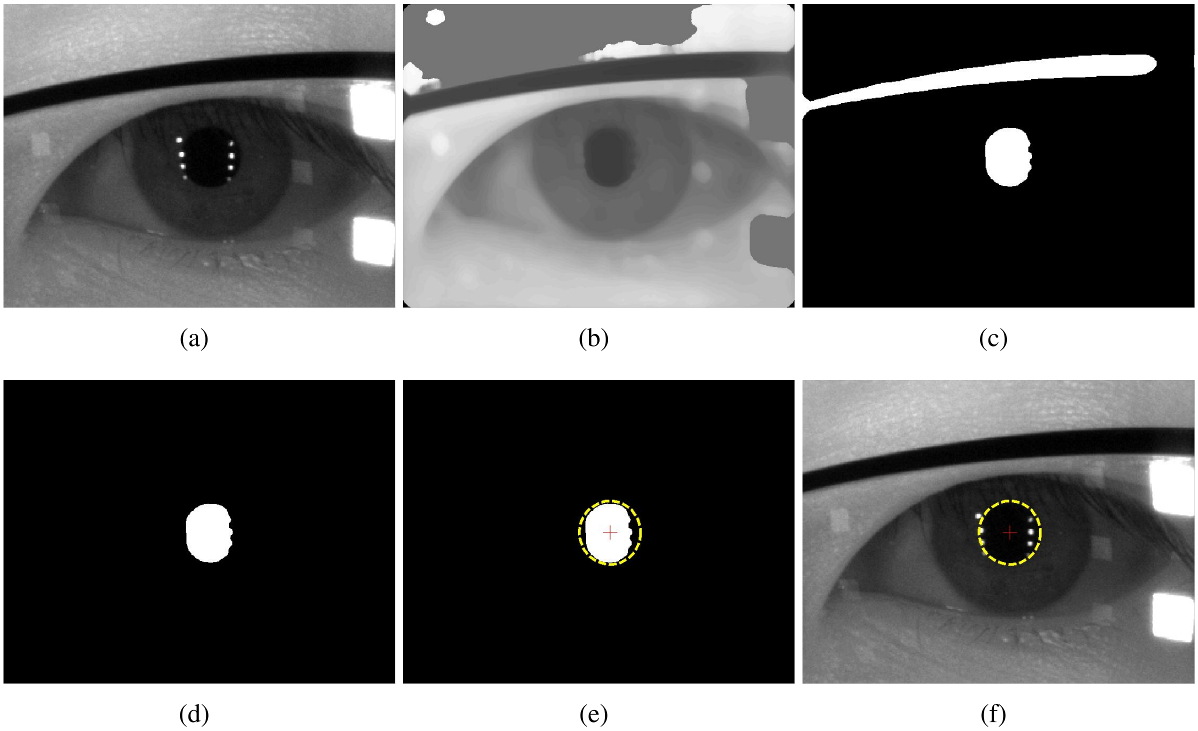
After locating the pupil, the next step is to find the circle for the iris boundary. We use the center of the pupil as a starting point to estimate the limbic boundary. Suppose we draw many straight lines departing from the center of the pupil growing out radially to the sclera region. While doing this, we keep track of the variations in pixel intensity. Since the sclera appears much brighter than the iris, the location where the intensity of pixels grows most rapidly should be the location where those lines hit the boundary. This process is illustrated in Figure 2.4.

### 2.3.4 Recovering the Iris Center and the Radius

The next step is to use those points to locate the center of the iris and to compute the radius of the outer boundary. Using fundamental geometry, we take the few known points from the boundary circle and compute the intersection of the perpendicular bi-sectors of the lines that pass through the pairs of those points. This is illustrated in Figure 2.5(a). After the center is found, the radius can be computed as the average distance from the center to all of those points. A realistic example is given in Figure 2.5(b).

## 2.4 Experiments and Results

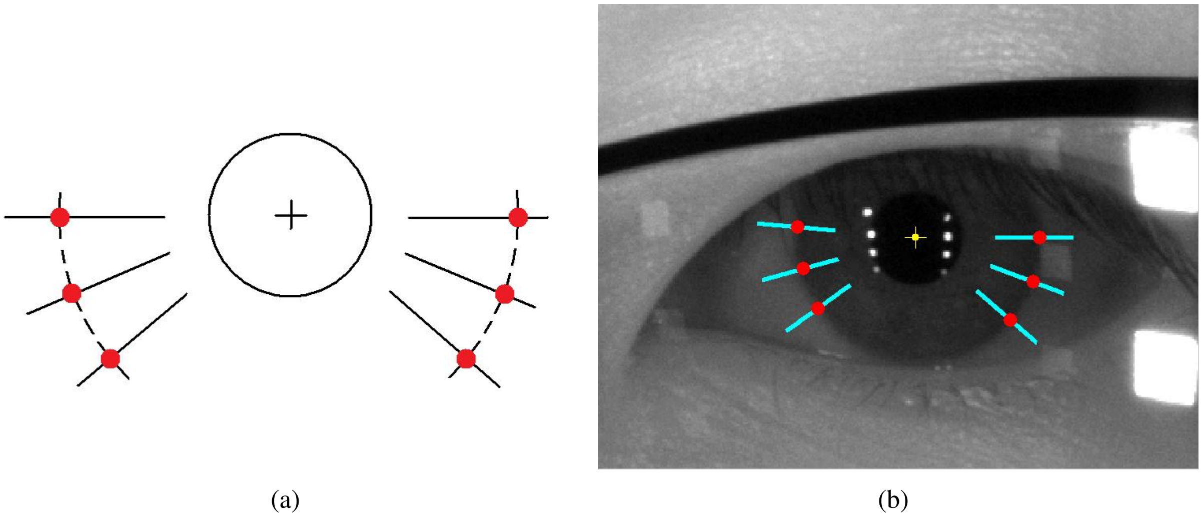
In order to fairly measure the performance of our proposed iris segmentation algorithm, we ran experiments on a large scale database, and also compared the performance with other existing methods. We used two databases to perform iris segmentation, one of which is publicly available, and the other one is collected in our research laboratory. We describe the databases, as well as the experimental results in the following sub-sections.



(a) (b) (c)

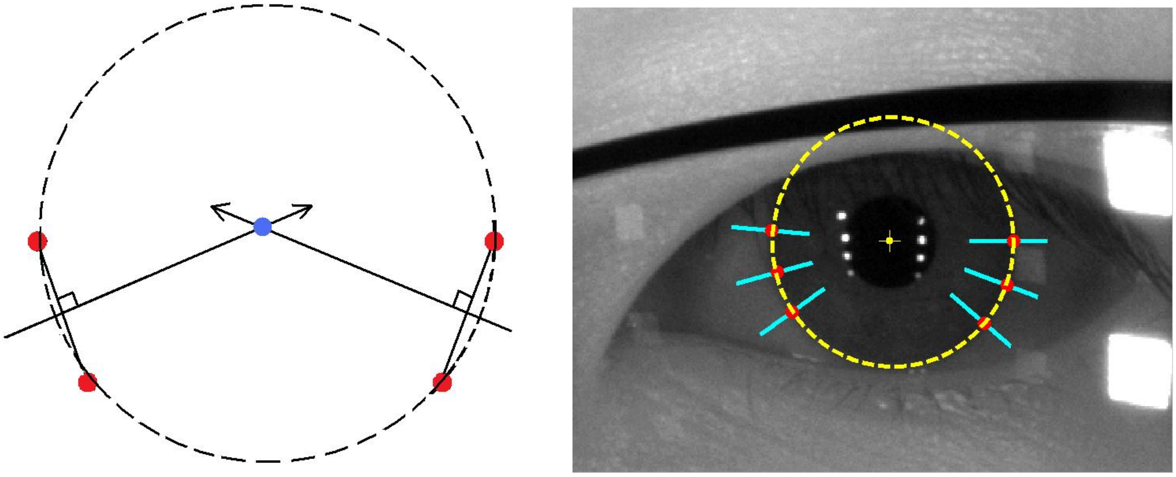
(d) (e) (f)

Figure 2.3: Process of locating pupil. (a) input image (b) after specular reflection removal (c) after pixel intensity thresholding (d) after applying geometrical constraints (e) finding the center of the pupillary circle (f) redraw the segmentation result for the pupil on the input image.



(a) (b)

Figure 2.4: Illustration of how to locate iris boundary. (a) Draw a few straight lines from the center of pupil to sclera region and detect the location where pixel intensity grows most rapidly; (b) a realistic example image to illustrate the idea in (a).



(a)

(b)

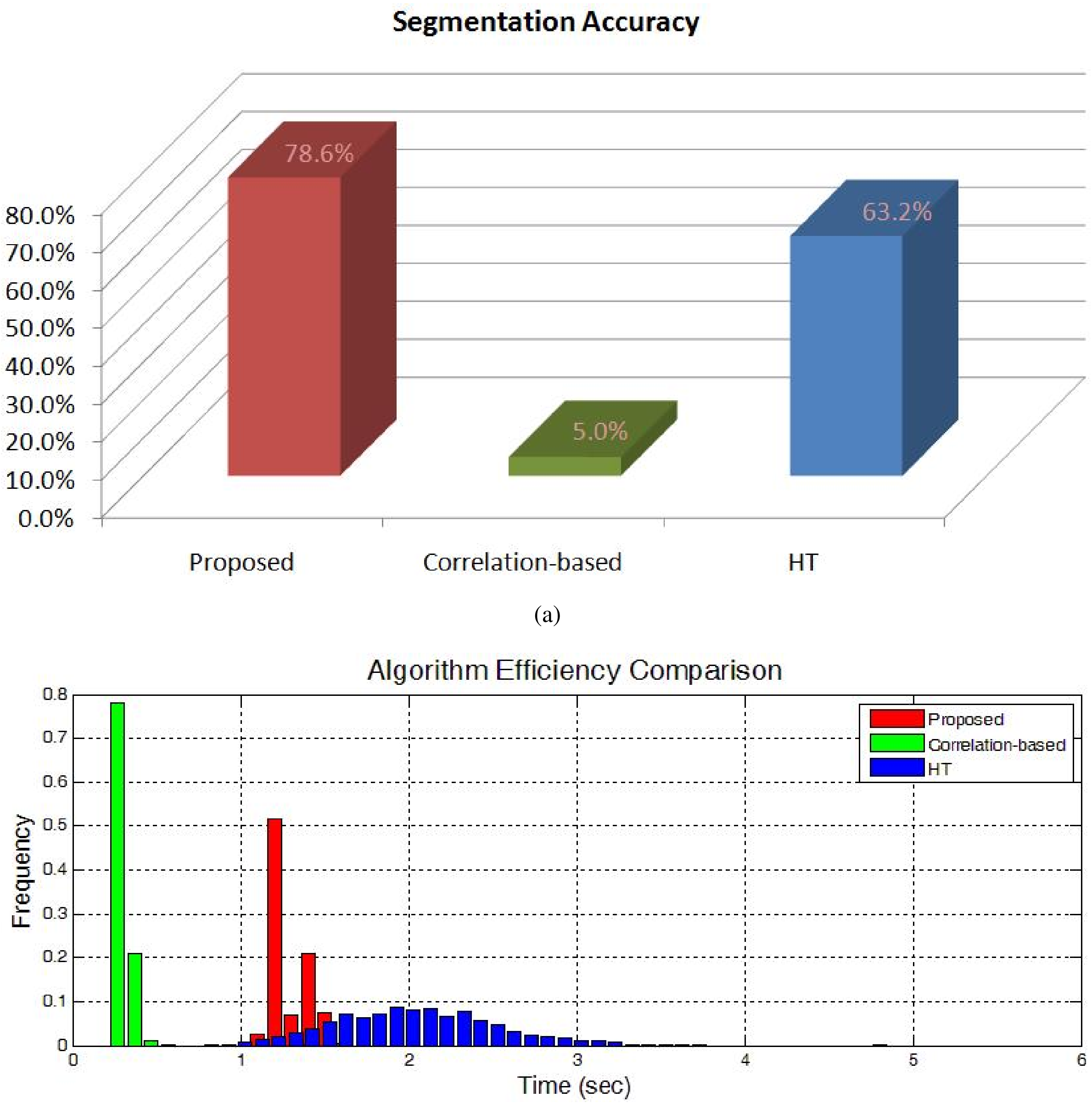
Figure 2.5: Illustration of how to locate iris center and radius. (a) when drawing the perpendicular bi-sectors of the lines that pass through pairs of the points which we found in section 2.3.3, the intersection of all lines should be the center of the iris. After iris center is found, iris radius is found by computing the average distance between center and all the points on the boundary. (b) a realistic example image to illustrate the idea in (a).

### 2.4.1 Experiment on MBGC database

The Multiple Biometric Grand Challenge (MBGC) is a large-scale, multiple biometrics database released by NIST in December 2007 [15]. The goal of the MBGC is to facilitate and direct the biometric research effort into large-scale face and iris recognition. Because the IOM (and similar systems) are emerging as a promising iris acquisition device to be widely used in the public domain in the near future, there is a substantial portion of the face and iris database that is collected by the IOM system. The IOM database is what we use for benchmarking in our experiments.

The MBGC database stores all its IOM data as raw video sequences of the subjects as they walk through the IOM portal, thus we need to pre-process the data before we can use it properly. First, we have to perform eye detection for every frame in every video sequence in order to estimate the region of the eye location. We use a correlation filter-based algorithm for our eye detection method. Second, for all detected eye images, we need to correctly filter out all of the false detected eye regions. We used a simple nearest-neighbor classifier based on projected PCA features to perform this two-class classification task. Since our goal is to perform iris segmentation, at this stage, we simply put all authentic the eye images together without labeling them with the class information. However, we are still able to determine how many classes there are in total by parsing the MBGC XML files. The specifications of the MBGC portal database is listed in 2.1.

For the purpose of comparison, we ran our proposed iris segmentation algorithm, as well as the other two existing algorithms (referred to the baseline algorithms). The first baseline algorithm is Libor Masek's implementation of the circular Hough Transform algorithm, as described



(a)

(b)

Figure 2.6: Experimental results for MBGC database. (a) iris segmentation accuracy for the proposed algorithm, as well as the two baseline algorithms; (b) histogram distribution of the time needed to segment one iris for each algorithm.

Table 2.1: The basic facts of MBGC portal database.

| Properties | Video Sequence | Subjects | Usable Eye Images | Left Eyes | Right Eyes |
|---|---|---|---|---|---|
| Numbers | 139 | 70 | 1384 | 832 | 552 |

in [7] and [16]. This method is the most popular iris segmentation algorithm in the research community. The second baseline algorithm is the correlation-based segmentation algorithm, as described in [12]. This method is known for its speed on the ICE database [17]. Thus, we tested it to see how it performed on the MBGC database.

Each algorithm is used to perform iris segmentation on all of the 1384 eye images in MBGC portal database. Each recovered pupillary boundary is recorded as a triplet $(x, y, r)$, which denotes the x and y coordinate of the center of the circle, and the radius of the circle. The limbic boundaries are also recorded in the same triplet format. We assume and approximate the pupillary and limbic boundaries as two non-concentric circles. We also recorded the time taken to segment each iris image for each algorithm, so we can fairly compare the efficiency and computation cost of the algorithms under the same testing conditions. The results of the segmentation accuracy, as well as the histogram distribution of the segmentation time, is shown in Figure 2.6.

### 2.4.2 Experiment on CMU-IOM database

We collected another IOM iris database in our lab during 2009, consisting of 2682 images, from 60 subjects, 111 iris classes. It contains different ehtnic groups, including Asian, Caucasian and Indian. The ages of the subjects span from 19 to 61.

The experimental procedure is the same as the one described in section 2.4.1. Since we would like to use the images in our iris super-resolution experiments later (as described in section 5.5), we manually created iris segmentation information for each of these 2682 iris images. Therefore, we have recorded the ground truth of the iris segmentation results. This information allows us to measure the accuracy of each algorithm. In this experiment, we do not just label the segmentation results as merely “correct” or “incorrect”. Instead, we use the concept of “threshold” from the correct boundary to describe the accuracy of the iris segmentation. Since the pupillary and limbic boundaries are described as circles, we can set the criteria of correct segmentation as the displacement between the ground truth boundary and the hypothesis boundary. Assume the ground truth circle is denoted as $(x_g, y_g, r_g)$, and the hypothesis circle is denoted as $(x_h, y_h, r_h)$, we can compute a measurement of segmentation distance $D$, defined as:

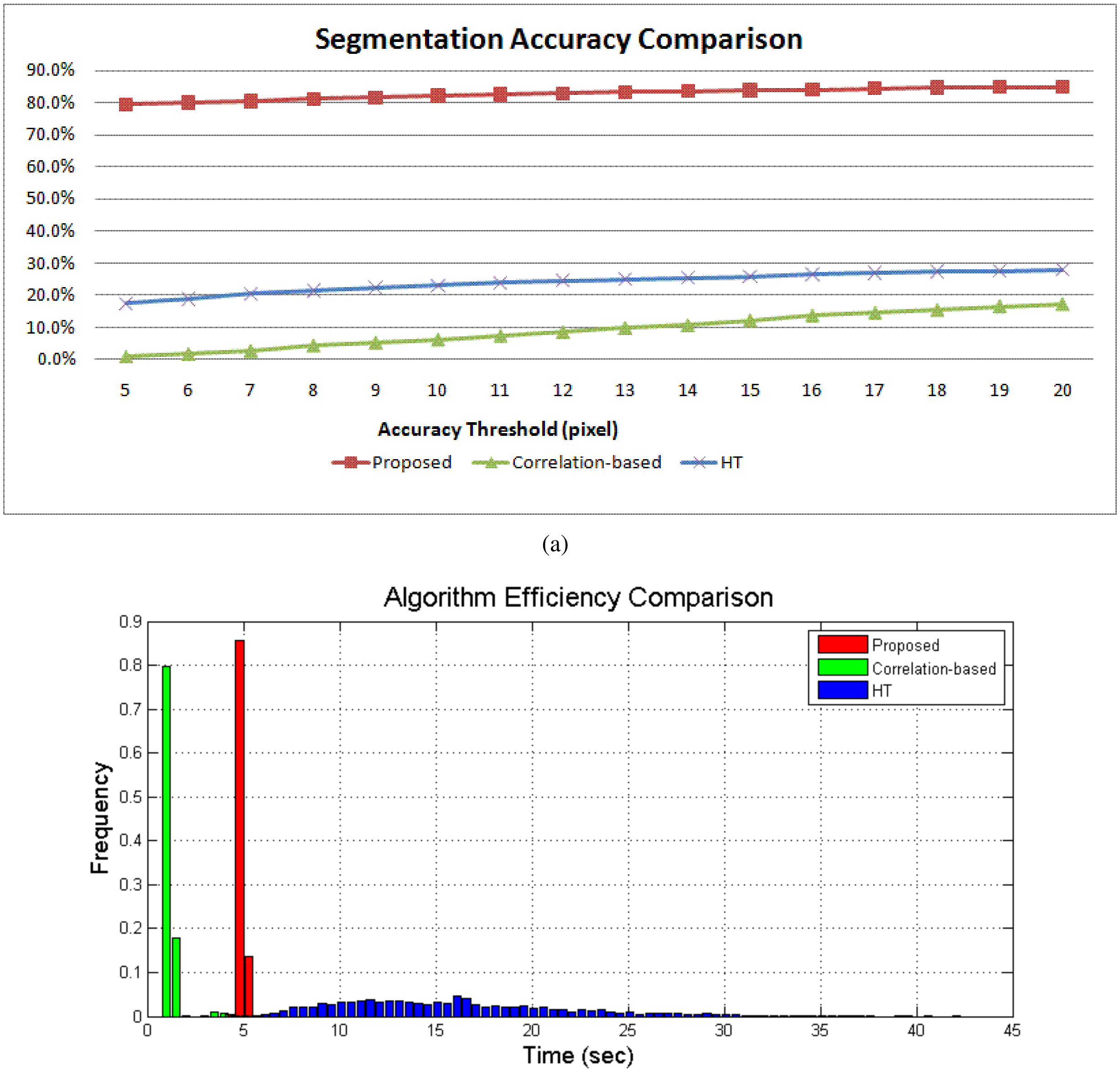
$$D = \sqrt{(x_g - x_h)^2 + (y_g - y_h)^2} + |r_g - r_h| \tag{2.1}$$

Thus, we pick a threshold value for $D$, and use it as a correct/incorrect criterion for classification. If we use different thresholds, it would result in labeling different sets of images as correct or incorrect segmentation. We can draw a chart to show how the segmentation accuracy varies with the threshold value. The experimental results are shown in Figure 2.7.

## 2.5 Conclusions

### 2.5.1 Segmentation Accuracy

From Figure 2.6(a), we can see that, in terms of segmentation accuracy, our proposed method is better than the Hough Transform method, which is in turn better than the correlation-based method. This trend is shown more clearly in Figure 2.7(a). In Figure 2.7(a), since we are able to



(a)

(b)

Figure 2.7: Experimental results for CMU-IOM database. (a) iris segmentation accuracy for all three algorithms. The accuracy is plotted with respect to the different threshold for the segmentation distance value. (b) histogram distribution of the time needed to segment one iris for each algorithm.

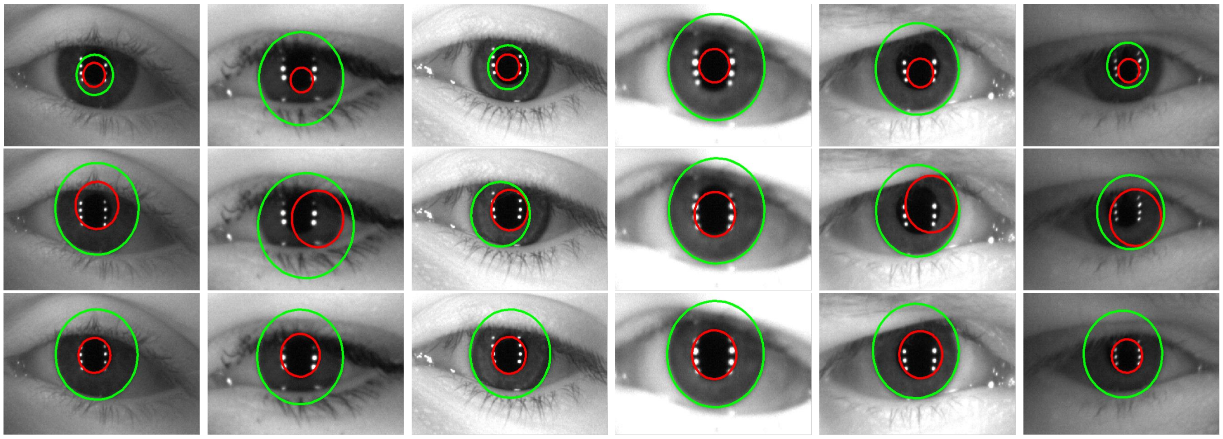


Figure 2.8: Comparison of segmentation results among different algorithms. First row: segmentation results of correlation-based method. Second row: segmentation results of Hough Transform method. Third row: segmentation results of the proposed method.

measure segmentation accuracy based on different thresholds, we can evaluate the algorithm performance under different levels of accuracy requirements. When we relax the accuracy threshold, the segmentation success rate increases, which is intuitive. However, it eventually goes into saturation as the threshold become greater than 18 pixels. No matter what accuracy threshold we are using, the performance gap between the proposed method and the other two baseline methods is significant, which amplifies that the proposed method is obviously more robust than the other existing methods.

Figure 2.8 shows a few segmentation results using all of the three methods for the purpose of comparison. We can see that the hypothesis boundaries of the correlation-based method are very easily affected by specular reflections, which occur in almost every eye image taken by the IOM system. While the Hough transform method is more robust than correlation-based method, the inner boundaries are easily confused with the outer boundaries. Our proposed method is much better than the two baselines and can correctly estimate the two boundaries.

In the MBGC database, the quality of the eye images in most cases is not very good. There are different factors that degrade the image quality. These include:

- Low intensity contrast: the contrast of the whole image is very low, which makes the inner and outer boundaries very hard to locate.
- Defocus and motion blur: if the eye images are taken when the subject is outside the field of focus, the defocus blur happens. If the shutter speed is slower than the moving speed of the subject, motion blur occurs. Both blurring factors degrade the image and make the boundaries softer, which increase the difficulty of recovering and detecting the correct boundaries.
- Noise: there are plenty of eye images in MBGC which are degraded by noise. Image noise occurs when the camera sensor captures the images under low illumination. Since we are imaging in near-IR, the camera sensors only see in the near-IR band. However, due to the low quantum efficiency of the camera sensor (~10%-15%), there is considerable noise

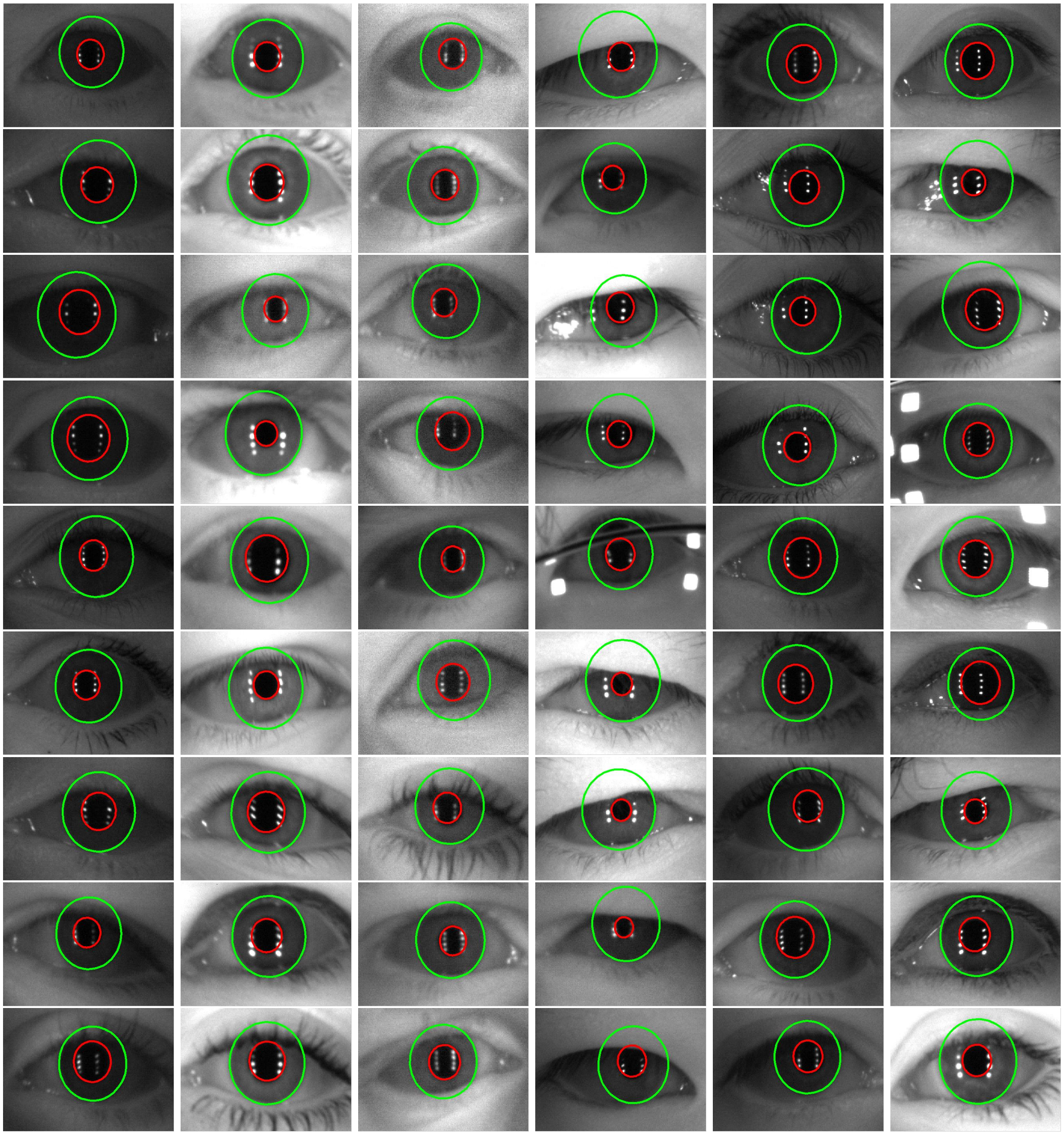


Figure 2.9: Examples of segmentation results of the proposed algorithm, which demonstrates the capability and robustness of the proposed algorithm. Column one: low contrast images. Column two: images suffered from defocus and motion blur. Column three: images degraded by noise. Column four: eye images under serious occlusion. Column five: images suffered from unbalanced illumination. Column six: images that contains unexpected specular reflections.

visible due to low SNR of image acquired. The longer the stand-off the darker the image is acquired due to the drop-off of photon intensity which is inversely proportional to the square of the distance of the light-source to the subject and to the camera. In the case for the IOM, the illumination is not co-located with the camera source and is placed on the side of the portal. Therefore the distance the light has to travel is measured from the person to the camera, which is much longer than the case when it was co-located with the camera source. The latter case would have removed the spurious specular reflections from the iris and placed them inside the pupil, but at the cost of reduced illumination, more noise, and less clarity of the iris as a result of dominating sensor read-out noise.

- Serious occlusion: the iris region of an eye image is easily occluded by other artifacts, such as eyelids or eyeglasses frames.
- Unbalanced lighting: in MBGC database, for unknown reasons, many eye images appear brighter on one side, but darker on the other side.
- Iris specular reflections: as stated above, the recovered boundaries are very easily affected by specular reflections on the ris regions because they cause strong intensity contrast. Because we cannot control the location of the specularity in the final eye images, it is observed that when it falls outside the pupil region, it will lead to confusion of the iris segmentation and detection algorithms.

Our proposed algorithm is proven to be robust and able to handle all of the above problems that we face in the IOM iris segmentation. Figure 2.9 shows a few segmentation results based on the proposed method. Our method successfully handles all of the images which are very challenging for the baseline algorithms, clearly demonstrating the capability of the proposed algorithm.

### 2.5.2 Extensibility of the Proposed Algorithm

Recently, more researchers in the field of iris recognition start to use elliptical segmentation, which means using ellipses to describe pupillary and limbic boundaries, instead of using circles. For our proposed algorithm, although it is described as using circular boundaries, in fact, it can be easily modified to become elliptical segmentation. The only modification is that, during stage four, we can fit all of the points on the outer boundary with an ellipse, instead of a circle. This shows another advantage of our proposed algorithm, which is easily to adapt into different segmentation paradigms.

### 2.5.3 Timing Analysis

Figure 2.6(b) and 2.7(b) shows the histogram distribution of the time needed to segment one iris by each of the three algorithms, for the MBGC and CMU-IOM databases, respectively. The results are consistent for both databases. The correlation-based method is faster than the proposed method, which in turn is faster than Hough Transform method. One might notice that the absolute scale of time seems very different between these two plots. This is because the size of the eye images we used in these two experiments were different. The size of eye images in MBGC is 320x210, but the size we used in CMU-IOM is 640x480. Therefore, it takes much longer for all three algorithms to perform iris segmentation in the CMU-IOM.

Although the correlation-based method appears to be more efficient, in terms of accuracy of segmentation, the correlation-based method is the worst. If our ultimate goal is to perform accurate iris recognition, the importance of accurate iris segmentation cannot be over-emphasized. Without accurate location of the iris region, all the later stages of the iris recognition system (including mask estimation, feature extraction and iris matching) would be adversely affected. Therefore, if we consider both accuracy and efficiency, our proposed algorithm is the best combination from the three candidate algorithms. Additionally, this experiment was performed using Matlab. If the source codes are written in C++, we expect timings to be significantly faster.

# Chapter 3

# Gaussian Mixture Modeling for Automatic Iris Mask Generation

Iris masks are essential in iris recognition to indicate which part of iris texture map is useful and which part is occluded or contaminated with noisy image artifacts such as eyelashes, eyelids, eyeglasses frames, and specular reflections. The accuracy of this iris mask is extremely important as it defines what parts of the iris to use for matching. The performance of the iris recognition system will decrease dramatically when the iris mask is inaccurate, even when the best recognition algorithm is used. Traditionally, people used the rule-based algorithms to estimate iris mask from the iris texture map. However, the accuracy of the iris mask generated this way is questionable. In this chapter, we propose a probabilistic and learning-based method to automatically estimate the iris mask from the iris texture map. The features used in this method are very simple, yet the resulting estimated iris mask is significantly more accurate than the results that the rule-based methods produce. We also demonstrate the effectiveness of the algorithm by performing iris recognition based on masks generated from different algorithms. Experimental results show that the masks generated by the proposed algorithm increase the iris recognition rate on NIST's Iris Challenge Evaluation (ICE) dataset, verifying that iris masking is very important in the recognition stage.

## 3.1 Introduction

Most iris recognition systems consist of four stages: image acquisition, iris segmentation, iris normalization and recognition. In most cases, in the iris normalization stage, iris images are transformed from the Cartesian coordinate system to the polar coordinate system as suggested by Daugman [2]. There are two main advantages of this coordinate transformation. First, it normalizes the texture variation caused by changes in environmental illumination which leads to pupil dilation and contraction. Without performing this transformation, it is extremely difficult to compare two irises of different pupil sizes due to mis-alignment of iris features. The second advantage of this coordinate transformation is that it translates the rotational shift, i.e. head tilt, in the Cartesian coordinate system into a pure translational shift in horizontal direction in the polar domain. It also simplifies the problem in matching stage because it is much more difficult

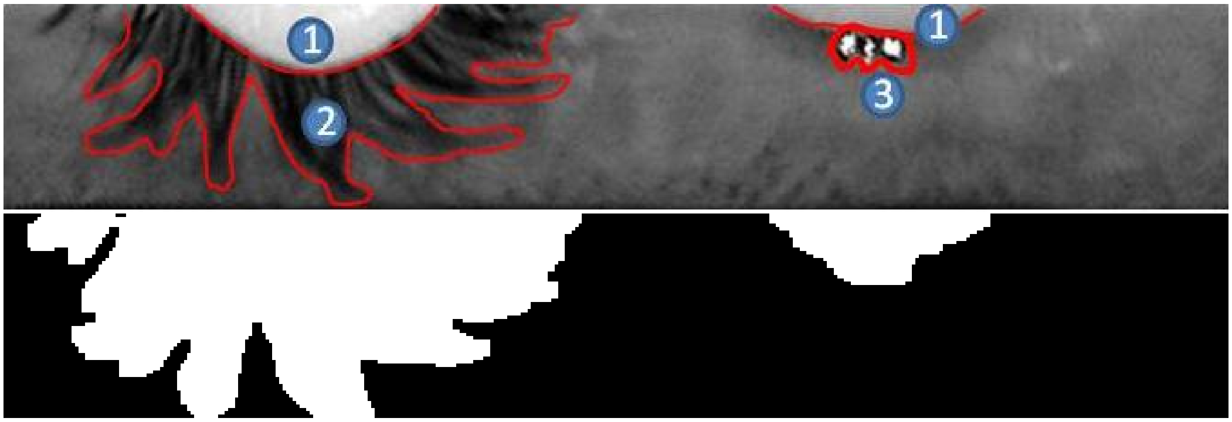


Figure 3.1: Normalized iris texture map (upper picture) and its accurate mask (lower picture), with white color indicates occluded area. In this iris map, there are noises caused by (1) eyelids, (2) eyelashes, and (3) specular reflections, as indicated in the picture. All of these artifacts have to be indicated in the mask in order to achieve high recognition performance.

and error-prone to perform pattern matching with rotational variation than translational variation. This effectively is handled by bit-shifting the whole iris code left/right by several bit columns to account for +- 15 degrees of head rotation. Due to these two significant advantages, most iris recognition systems adopt this type of iris texture normalization in their implementation.

In most cases, after transforming iris texture from the Cartesian coordinate to the polar coordinate, one has to create a mask for the iris map in the polar coordinate. The goal of this mask is to indicate which part in the iris map is truly iris texture, and which part is noise. The sources of the occlusions/artifacts of the iris map may be due to eyelids, eyelashes, shadows, or specular reflections. Figure 3.1 shows example images of an iris texture map and its accurate mask. Note that in the iris texture, the noisy regions consist of texture created by eyelids, eyelashes, and specular reflections. All of these noise artifiacts have to be indicated in the mask in order to be ignored in the spatial feature match process, in order to achieve the high performance of iris recognition.

The accuracy of the iris mask has a great impact to the recognition accuracy of the iris recognition system. Traditionally, the main focus of iris recognition research addresses the power of the matching algorithm and the feature extraction. Most researchers emphasize the novelty and effectiveness of their iris feature extraction and matching algorithm. While these are important issues that play a major role in iris recognition, we have found that the accuracy of the iris mask contributes much more than what researchers thought in the past. Accurate iris masks, combined with good features and effective recognition schemes, make the iris recognition system more successful. However, if the iris mask is inaccurate, the best feature extraction and recognition algorithms can not compensate for this. Thus the overall recognition rate will decrease dramatically. This has been observed empirically in our experiments, and we present it in the later sections.

Because researchers have traditionally focused their attentions on the matching algorithm, the algorithm for automatic generation of the iris mask has not been given significant attention. In this chapter, we propose a probabilistic, bayesian learning-based approach to estimate the iris mask from the original, normalized iris texture map. Another advantage of our proposed algorithm is its efficiency. Since we are modeling the iris intensity probability distribution with a mixture of Gaussians, and the formula of a Gaussian distribution is easy to evaluate, our proposed algorithm can automatically generate an iris mask very efficiently in a short period of time. These advantages make our proposed algorithm not only theoretically attractive but also practical and efficient during the classification stage.

The rest of the chapter is organized as follows: we review previous work about automatic iris generation in Section 3.2 and explain our proposed algorithm in Section 3.3. In Section 3.4, we show our experimental settings and results. In Section 3.5 and 3.6, we discuss discoveries in our experiments and conclude with the contribution of our proposed algorithm, followed by information about our future work.

## 3.2 Previous Work

Daugman's work is one of the earliest in iris recognition [2]. In 1993, he described the normalization scheme for iris texture, calling it the "Doubly Dimensionless Projected Polar Coordinate System". However, when considering which part of the iris texture in the polar coordinate system is the authentic iris, he proposed a very simple method which assumes the top part of the image and the 45° notch at the 6 o'clock position is occluded, for every iris image.

Ma et al. proposed a full framework for an iris recognition system [18]. Although it addressed many issues in the iris recognition process, it did not mention any specific solution for occlusion detection. They merely stated that they discarded the lower part of the normalized iris texture and focused only on the more discriminative regions. This scheme is equivalent to assuming that the lower part of iris texture is occluded. A similar assumption was proposed in the work of Tisse et al [19].

Daugman, in a later work, proposed a more sophisticated algorithm for finding occluded regions in the iris [20]. He proposed to locate the boundary of the eyelids at the segmentation stage. By replacing the circular integration operator (for finding iris and pupil boundary) with the spline parameter, one can approximately locate the eyelid boundaries. Furthermore, in his latest work [21], he proposed a new method, which uses active contours to find the boundary of the eyelids.

Kong and Zhang proposed a model for detecting eyelashes and specular reflections [22]. They proposed using Gabor filters to detect separable eyelashes and used the variance of intensity in the local window to locate clusters of multiple eyelashes. After that, the connectivity criterion is enforced to enhance the robustness of the algorithm. For strong specular reflections, they used a hard threshold on pixel intensity to identify them. For weak specular reflections, they used mean and standard deviation of the local window as an adaptive threshold for pixel intensity to classify.

Zou et al. proposed a procedure for iris occlusion detection [23]. It consisted of four stages: first, they detected the horizontal edges of the eyelids. Second, they performed the morphological operation on those edges to enhance them. Third, they used the segmentation result to localize valid edge candidates. Finally, they used connectivity information to refine the mask. Although this method looks simple and easy to implement, it may not work well for eyelashes because they are usually vertically oriented. Also, the edge-detection based algorithm may incorrectly classify some part of the iris texture into a noisy region when the contrast within the iris texture is large.

In [24], Krichen et al. proposed a probabilistic approach for iris quality measure. They compared the performance of the Gaussian Mixture Model with Fourier-based methods, wavelet-based methods and active-contour based methods. The iris masks estimated by their method seem to be local patch-based, not pixel-based. In [12], Thornton proposed using a discriminative learning method based on FLDA to estimate iris masks in the polar domain.

## 3.3 Proposed Method

All of the previous works performed in mask generation have some limitations. The methods described by Daugman [2], Ma et al. [18], or Tisse et al. [19] are either inaccurate or cannot handle iris images taken in real-time. The method described by Kong and Zhang [22] is basically a rule-based approach, and while this approach may work well for specific settings or environments, it may not work when the environment, e.g. illumination, changes. The parameter-tuning of the rule-based approach is also important to the performance of the iris mask generation; thus making the rule-based approach very sensitive to the environmental setting for any specific database.

To overcome the shortcomings of previous methods, we propose a novel and probablistic method for automatic iris mask generation. The proposed method has to be flexible for all settings and capable to work in environments and across sensors and databases. Additionally, we quantify the performance of our generated iris masks by measuring how closely they fit to the ground-truth of the iris masks.

Let us review this problem from a pattern-recognition perspective. The problem of generating a mask for an iris map can be seen as a two-class classification problem. For each pixel on the iris map, we need to extract robust features that contain discriminative information about whether each pixel belongs to the iris texture or to the occluded region (eyelash, eye, specular reflection etc.). We can then train a classifier by using some training data, and later use the trained classifier to perform classification on unseen data and different datasets.

We propose the use of a Gaussian mixture modeling scheme (GMM) to model the posterior probability distribution of both iris texture and occlusion regions. GMMs have been widely used in all kinds of pattern recognition problems, including speech processing [25], human skin detection [26], real-time tracking [27], hazardous chemical agents detection [28] and bearing damage detection for induction motors [29]. The advantage of GMM is its modeling ability. As long as the number of Gaussians is large enough, GMM can virtually model any distribution. Another advantage of GMM is that its mathematical equation is easy to evaluate; thus the classification speed is very high during the testing stage. This is an important consideration when dealing with problems like automatic mask generation because we have to perform pixel-wise classifications for every iris image. Suppose the size of the normalized iris texture map is 30x180 pixels. The total number of pixels in this map is 30x180=5,400. Our classifier has to perform classification 5,400 times to create a single mask for an iris image.

### 3.3.1 Gaussian Mixture Model

Let us review the basic mathematical foundations for the Gaussian Mixture Model (GMM). The Gaussian distribution of a $D$ dimensional random variable $X$ which has a value $x$ is represented by (3.1)

$$X \sim \mathcal{N}\left(x;\mu,\Sigma\right) = \frac{1}{\left(2\pi\right)^{\frac{D}{2}}\left|\Sigma\right|^{\frac{1}{2}}} e^{\left[-\frac{1}{2}(x-\mu)^{T}\Sigma^{-1}(x-\mu)\right]} \tag{3.1}$$

where $\mu$ is the mean vector and $\Sigma$ is the covariance matrix of the Gaussian distributed random variable $X$.

The probability density function of GMM can be defined as a weighted sum of multiple Gaussian distributions, as shown in (3.2)

$$p(x;\ \theta) = \sum_{c=1}^{C} \alpha_c \mathcal{N}(x;\ \mu_c;\Sigma_c) \tag{3.2}$$

where $\alpha_c$ is the prior probability that the random variable $x$ is generated by the c$^{th}$ Gaussian mixture, satisfying normalization constraints

$$0 \leq \alpha_c \leq 1 \quad and \quad \sum_{c=1}^{C} \alpha_c = 1 \tag{3.3}$$

According to (3.1)-(3.3), the probability density function for a Gaussian mixture model can be completely defined by a parameter list as shown in (3.4)

$$\theta = \{\alpha_1,\ \mu_1,\ \Sigma_1, ...,\ \alpha_C,\ \mu_C,\ \Sigma_C,\} \tag{3.4}$$

We can calculate the number of free parameters for a GMM, given the random variable $X$ of dimension $D$. For each $\mu_i$, there are $D$ free parameters; for each $\Sigma_i$, since it should be symmetric in nature, the number of free parameters is $\left(D^2 - D\right)/2 + D = \frac{1}{2}(D^2 + D)$. For each $\alpha_i$, we have one free parameter. Combining these calculations and (3.3), the total number of free parameters is $C(\frac{1}{2}D^2 + \frac{3}{2}D) + C - 1$.

### 3.3.2 MLE and MAP Estimation for Model Parameters

The process of training GMM is to estimate the parameter list $\theta$, given the observation X. Suppose we have a set of independent, identically distributed samples X=$\{x_1,\ x_2,\ ...,\ x_N\}$ drawn from the same distribution described by the GMM probability density function $p(x;\ \theta)$. The likelihood function can be defined as (3.5)

$$\mathcal{L}(\mathrm{X};\ \theta) = \prod_{n=1}^{N} p(x;\ \theta) \tag{3.5}$$

It depicts the probability that the series of observation X is generated by the distribution governed by $\theta$. The goal of parameter estimation is to find the optimal parameter $\hat{\theta}$ that maximizes the probability:

$$\hat{\theta} = arg \max_{\theta} \mathcal{L}(\mathrm{X};\ \theta) \tag{3.6}$$

$\mathcal{L}(\mathrm{X};\ \theta)$ in (3.5) and (3.6) contains many multiplication operations. We can speed-up the operation by taking logarithm on both sides. The log-likelihood function can be described as (3.7):

$$L(\mathrm{X};\ \theta) = \ln \mathcal{L}(\mathrm{X};\ \theta) = \sum_{n=1}^{N} \ln p(x_n;\ \theta) \tag{3.7}$$

Since the logarithm function is monotonically increasing, we can substitute $\mathcal{L}(\mathrm{X};\ \theta)$with $\mathrm{L}(\mathrm{X};\ \theta)$ in (3.6) and get the same answer.

Parameter estimation by (3.6) is called Maximum-Likelihood Estimation (MLE). Sometimes Maximum a Posteriori (MAP) estimation is used instead of MLE:

$$\hat{\theta}_{MAP} = arg\ \max \{\ln \mathcal{L}(\mathrm{X};\ \theta) + \ln \mathcal{L}(\theta)\} \tag{3.8}$$

### 3.3.3 EM Algorithm

In literature, the Expectation-Maximization (EM) algorithm is used to find the parameter $\hat{\theta}_{MLE}$ or $\hat{\theta}_{MAP}$. EM is an iterative procedure to estimate parameters when part of the data is missing. In training the GMM, we need to estimate multiple parameters at the same time, including the mean and covariance matrix for each Gaussian mixture and determine which training sample belongs to each of the Gaussian mixtures. EM can optimize all of these unknown parameters and can converge to a local maximum of $\hat{\theta}$. There are two main steps in EM algorithm:

The E-step in EM estimating the expectation of the likelihood of the observed data, assuming that we know the optimal model parameters. In other words, it is to evaluate the expectation function $Q(\theta;\ \theta^i)$ in (3.9)

$$Q(\theta;\ \theta^i) = E_{\mathrm{Y}} \left[\ln \mathcal{L}(\mathrm{X},\ \mathrm{Y};\ \theta) | \mathrm{X};\ \theta^i\right] \tag{3.9}$$

where Y is the unknown feature, which is the label that indicates to which Gaussian mixture each sample belongs.

The M-step in EM is to search the parameter space to find the optimal parameter $\hat{\theta}$ that can maximize the likelihood function defined in (3.9):

$$\theta^{i+1} = arg \max_{\theta} Q(\theta;\ \theta^i) \tag{3.10}$$

The two steps will be executed repeatedly until $\theta$ converges to a local maxima. Usually we can define a convergence criterion as a threshold on the difference of the likelihood between two iterations:

$$Q(\theta^{i+1};\ \theta^i) - Q(\theta^i;\ \theta^{i-1}) \leq T \tag{3.11}$$

Alternatively, a convergence criterion can be defined as a threshold on the distance in parameter space:

$$||\theta^{i+1} - \theta^{i}|| \leq \epsilon \tag{3.12}$$

### 3.3.4 Figueiredo–Jain's Extension for GMM Training

EM-based parameter estimation for GMM training has a few drawbacks. First, the number of Gaussian mixtures has to be manually chosen. Therefore, a wrong estimate for the number of Gaussian mixtures may hurt the accuracy of the overall trained GMM. Second, the initialization of the mean of each Gaussian mixture is crucial. Similar to the K-means algorithm, if the initial position of the means of each Gaussian is not initialized appropriately, the EM algorithm may not be able to converge to the true location of the mean of each Gaussian. Due to these two major drawbacks in the EM training algorithm, we decided to use an alternative training approach. Figueiredo and Jain proposed an unsupervised learning method for training GMMs [30]. This method can estimate the number of Gaussian mixtures without human intervention, and can avoid the boundary of the parameter space during the converging stage. The basic idea of Figueiredo–Jain's extension for GMM training (FJ algorithm) is that it dynamically adjusts the number of Gaussians by eliminating Gaussians which are not supported by the observation. Also, during the mixture elimination process, it chooses to eliminate the Gaussian mixtures that are close to becoming singular. This process can avoid converging to small Gaussians on the boundary of parameter space.

The FJ algorithm uses the idea of Minimum Descriptive Length (MDL) and applies it to mixture model training. It is equivalent to using the objective function in (3.13)

$$\begin{aligned} \Lambda(\theta,\, \mathrm{X}) \;\; &= \;\; \frac{V}{2} \sum_{\alpha_c > 0} \ln\left(\frac{N\alpha_c}{12}\right) + \frac{C_{nz}}{2} \ln \frac{N}{12} \\ &\quad + \frac{C_{nz}\,(V+1)}{2} - \ln \mathcal{L}(\mathrm{X},\, \theta) \end{aligned} \tag{3.13}$$

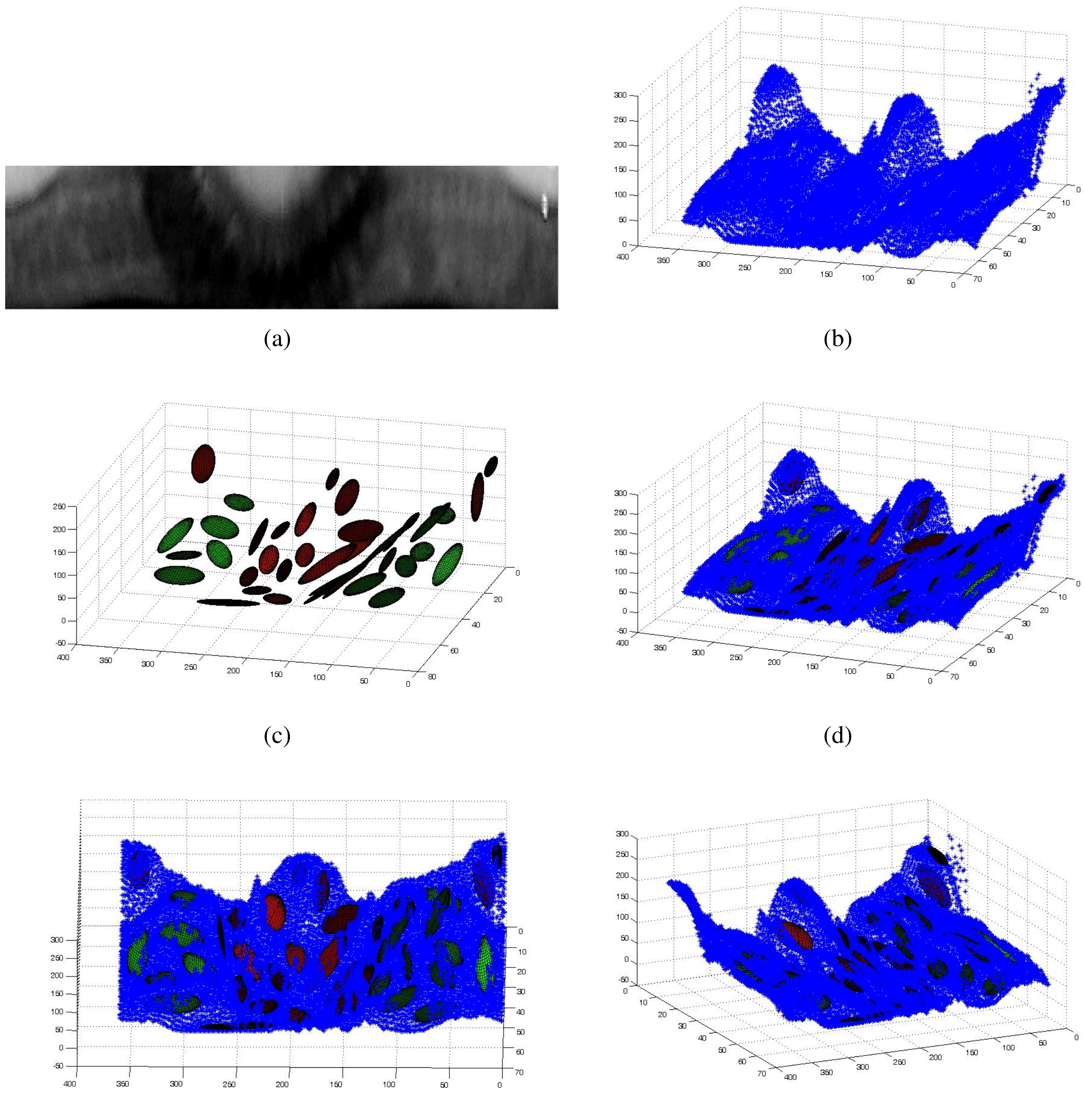
where $N$ is the number of training points, $V$ is the number of free parameters of the GMM, $C_{nz}$ is the number of Gaussian mixtures that have nonzero weight ($\alpha_c > 0$), $\theta$ is defined as in (3.4), and the last term is defined in (3.7).

By using (3.13) as a new objective function, the formula for estimating the prior distribution of the Gaussian mixture in the FJ algorithm becomes

$$\alpha_c^{i+1} = \frac{\max\left\{0,\, \left(\sum_{n=1}^{N} w_{n,c}\right) - \frac{V}{2}\right\}}{\sum_{j=1}^{C} \max\left\{0,\, \left(\sum_{n=1}^{N} w_{n,c}\right) - \frac{V}{2}\right\}} \tag{3.14}$$

where $w_{n,c}$ is the probability that the $n^{th}$ observation is generated from the $c^{th}$ Gaussian mixtures, defined as

$$w_{n,c} = \frac{\alpha_c^i p\left(x_n | c;\, \theta^i\right)}{\sum_{j=1}^{C} \alpha_c^i p\left(x_n | c;\, \theta^i\right)} \tag{3.15}$$



(a)

(b)

(c)

(d)

(e)

(f)

Figure 3.2: Visualization of GMM trained on single iris texture. (a) Example image for training; (b) Example iris texture viewed in 3D, where z coordinate is the pixel intensity value; (c) 3D GMMs trained with FJ algorithm. Mixtures with red color represent GMMs for occlusion; mixtures with green color represent GMMs for authentic iris texture; (d)-(f) Plotting original iris texture data together with trained GMM, in 3D view. From (d)-(f), We can see that trained GMMs fit the training data very well.

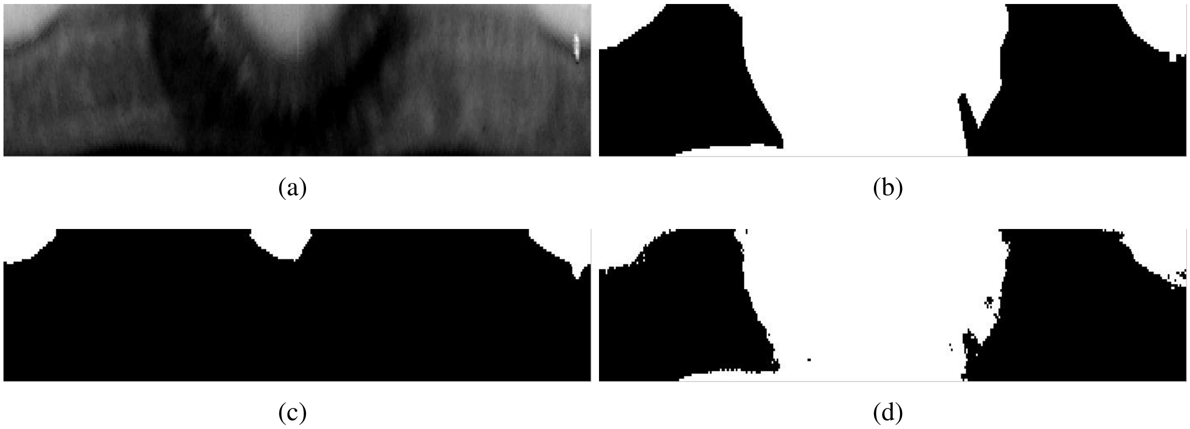

(a)
(b)
(c)
(d)

Figure 3.3: Comparison of the iris mask estimated by different algorithms. (a) the input iris texture image in the polar coordinate; (b) the "perfect" mask, generated by manual labor; (c) the mask generated by a rule-based method; (d) the mask generated by FJ-GMM algorithm.

Table 3.1: Statistics about ICE1 and ICE2

| Dataset | ICE1 | ICE2 |
|---|---|---|
| Number of classes | 124 | 120 |
| Number of Total Images | 1425 | 1528 |
| Minimal Number of Images per class | 1 | 1 |
| Maximal Number of Images per class | 31 | 31 |

The formula for estimating parameter $\mu_c$ and $\Sigma_c$ is the same as in traditional EM algorithm:

$$\mu_c^{i+1} = \frac{\sum_{n=1}^{N} x_n w_{n,c}}{\sum_{n=1}^{N} w_{n,c}} \tag{3.16}$$

$$\Sigma_c^{i+1} = \frac{\sum_{n=1}^{N} w_{n,c} \left(x_n - \mu_c^{i+1}\right) \left(x_n - \mu_c^{i+1}\right)^T}{\sum_{n=1}^{N} w_{n,c}} \tag{3.17}$$

## 3.4 Experiments and Results

### 3.4.1 Database Description

The database we used in our experiments is the Iris Challenge Evaluation (ICE) database, published by NIST in 2005, as described in [17]. The ICE database consists of two subsets. They are called ICE1 and ICE2, respectively. The details of the ICE1 and ICE2 subsets are given in Table 3.1.

### 3.4.2 GMM Trained on Single Image

We first tried our proposed method on a single iris image to see how the trained GMM would fit into the training data. We took the image 245241.giff from the ICE2 database to be our training sample. After manually segmenting the iris and performing iris normalization, we got Figure 3.2(a). Because we used the intensity value of every pixel as the Z coordinate of each pixel in three-dimensional space, we were able to plot the iris texture map in Figure 3.2(a) in a 3D perspective, as shown in Figure 3.2(b). We also manually created a mask for Figure 3.2(a) to indicate which part was authentic iris texture to use this information (true iris vs. occlusion) as the class labels for each pixel. Manually labelled iris masks were created, and each point was designated a class label, which was then used to train two GMM models; the first one modeled the distribution of iris texture, and the second one modeled the occluded regions. We plotted the trained GMMs in Figure 3.2(c). In Figure 3.2(c), GMMs with the red color represent GMMs trained for occlusion, and GMMs with the green color are models for iris texture. The brighter the color, the higher the prior probability it has among all GMMs.

In Figure 3.2(d)-(f), we can see that the trained GMMs approximate very well the original distribution of points in this three-dimensional space.The red GMMs modeled the noisy parts, while the green GMMs modeled the authentic iris texture area. This confirmed that the GMMs were able to fit for a single image.

If we use Figure 3.2(a) as our test image and use the trained GMM to perform the classification for every pixel on this image, plotting the results of the classification back into a two dimensional array, with 0 indicating the authentic iris and 1 indicating the occlusion region, we can visualize the mask generated by the GMM. We show the comparison of the iris mask estimated by different algorithms in Figure 3.3. The input iris texture image in the polar coordinate is shown in Figure 3.3(a), while the ground truth of the mask shown in Figure 3.3(b), is manually labeled. The mask generated by the proposed method is shown in Figure 3.3(d), compared with another mask generated by a rule-based method, shown in Figure 3.3(c). As we can see, the result generated by the proposed algorithm is much better than what we can get using the rule-based method.

### 3.4.3 Feature Set Exploration

The experiment depicted in sub-section 3.4.2 produced good results. However, the feature set seems overly simplistic. Besides only using the image intensity, local mean and standard deviation, we hypothesized that other features must exist that can better describe the difference between the true iris texture and the occlusion artifacts. Therefore, we conducted an experiment to look for a new set of features to capture texture discrimination.

In this experiment, the feature sets we tested included the following:

- X, Y coordinate of the location of the pixel and pixel intensity (learning the spatial geometry)
- Mean and standard deviation in a local 3x3 neighborhood
- Response intensity after the image is filtered by Sobel edge filter
- Response intensity after the image is filtered by Gaussian

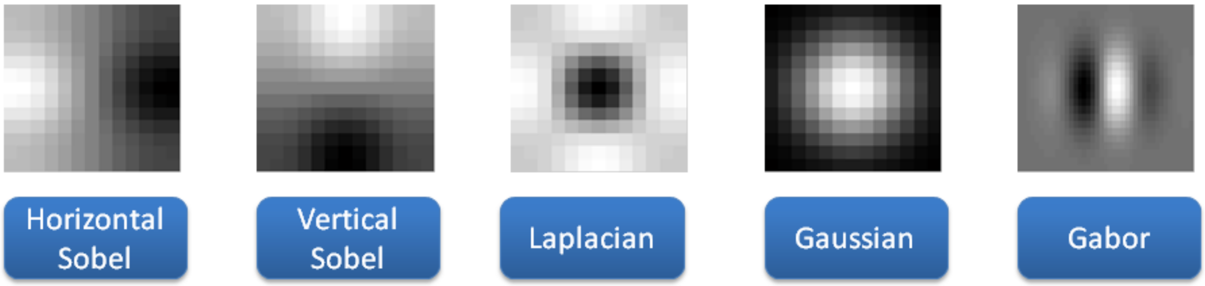


Figure 3.4: The feature sets (textons) used in the experiment of feature set exploration.

- Response intensity after the image is filtered by Laplacian of Gaussian (LOG)
- Response intensity after the image is filtered by Gabor filter

Figure 3.4 gives a visual illustration of all of the feature sets we used. In order to organize the experimental process and results, we gave a code name to each experiment, in order to better distinguish what we tried in each experiment. The code names and the features are listed below:

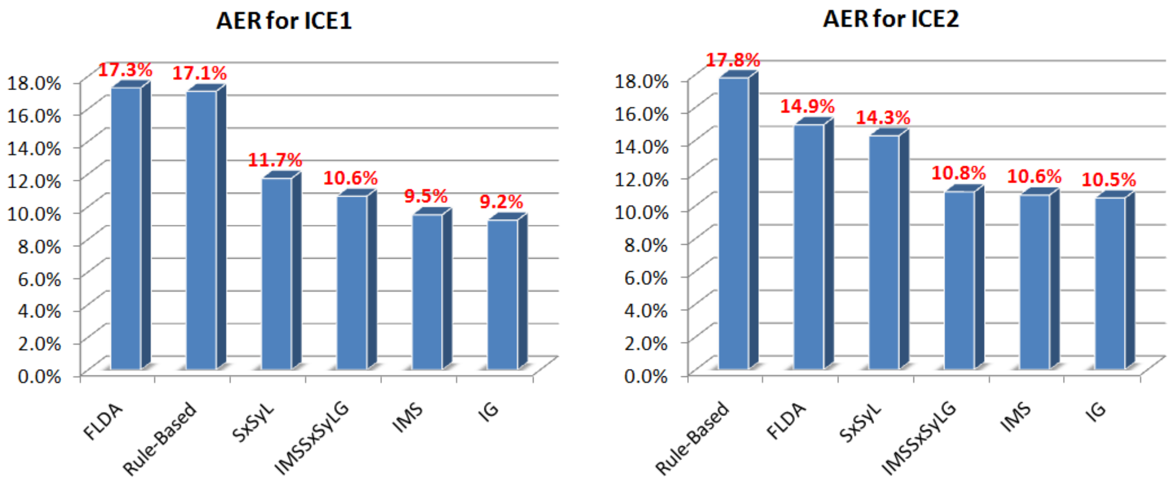
1. IMS: Intensity of the pixel, Mean and Standard deviation within 3x3 neighborhood.
2. SxSyL: Response intensity after the image is filtered by Sobel edge filter (both horizontally and vertically), and by Laplacian of Gaussian
3. IG: Intensity of the pixel and response intensity after the image is filtered by Gabor filter
4. IMSSxSyLG: Combination of all of the three above

Note that all of the above experiments included the (X,Y) coordinates of pixel locations as their features.

Additionally to the proposed method, we also used two existing methods which are very popular in the iris research community to create iris masks. The results from these two methods serve as the baseline for comparison purposes. By comparing the baseline results with our proposed method, we can verify the effectiveness and accuracy gained by the proposed approach. The first baseline method we use is the rule-based method which is similar to what is described in Kong and Zhang's work [22]. It basically detects whether there is a strong variance of pixel intensity in a local window and uses it as a feature to perform classification. It can be illustrated in four steps:

1. Normalize the image so that the energy of pixel intensity sums up to one.
2. Create a new image, which we will call the "mean image," where each pixel computes the mean value of the local 5x5 window.
3. Compute the global mean and standard deviation of all pixel intensity of the image of mean.
4. For every pixel on the mean image, if the difference between its intensity value and the global mean is less than twice the global standard deviation, then classify it as iris texture. Otherwise, classify it as an occluded region.

The second baseline method that we tested is the Fisher-Linear Discriminant Analysis (FLDA)-based occlusion detection method which is described in [12]. We performed our experiments



(a) (b)

Figure 3.5: The Average Error Rate (AER) for the mask generated by different algorithms, in the feature set exploration experiment.

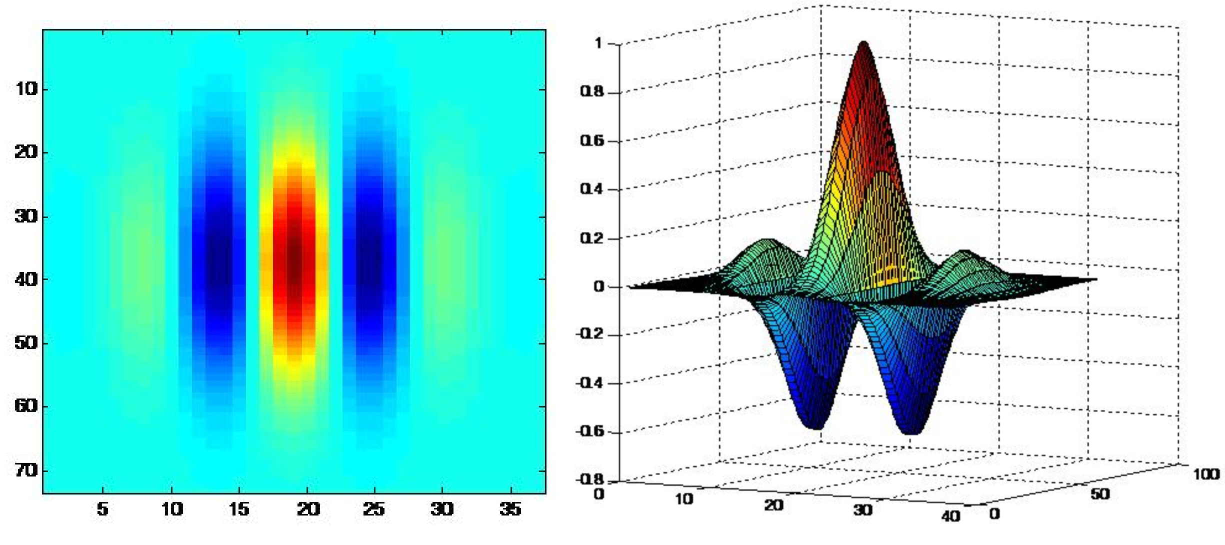
on the NIST ICE database, on both ICE1 and ICE2 subsets where we performed segmentation and generated masks for every image in ICE manually to get the ground truth for every mask corresponding to each iris. For each iris class in ICE, we picked one image as training data and left all the other images as test data. For each testing image, we computed the average error rate (AER) of the masks estimated by different algorithms. AER can be computed as:

$$AER = \frac{\sum_{i=1}^{N} e_i}{N} \tag{3.18}$$

$$e_i = \frac{NP(mask_{alg} \otimes mask_{gt})}{W \times H} \tag{3.19}$$

where (W, H) is the size of the mask, $mask_{alg}$ and $mask_{gt}$ is the mask generated by specified algorithm and human labor, respectively; $N$ is the total number of testing images; $\otimes$ is the pixel-wise XOR operator that can be used to compute the difference between two occlusion masks; $NP()$ is the function that counts the number of pixels in the images which are not zeros. The results are shown in Figure 3.5.

From the results shown in Figure 3.5, we can see that our proposed method outperforms the two baseline methods. For the feature set selection, we found that the feature set IG performs better than other combinations. Therefore, the image response generated by Gabor filtering is much more discriminative than those generated from other filters (Sobel, Gaussian and Laplacian of Gaussian).



(a)

(b)

Figure 3.6: Example of a Gabor filter. (a) 2D view. (b) 3D view.

### 3.4.4 Gabor Filter Banks Optimization by Simulated Annealing

Observing the experimental results shown in sub-section 3.4.3, we conclude that among the most popular image and feature extraction filters used in general field of image processing, the Gabor filter is the best to extract the salient features for the goal of classification between iris texture and occlusion artifacts. However, we had to decide which Gabor filter to use, i.e., what modulation frequency, orientation and scale. In this section, we detail how we optimize the Gabor filter banks, so the accuracy of the iris mask can be further enhanced. To achieve this goal, we employed an iterative nonlinear optimization approach.

#### 3.4.4.1 Gabor Filter Formulation

The mathematical equation of the generic Gabor filter is given below:

$$g(x, y; \lambda, \theta, \psi, \sigma, \gamma) = e^{-\frac{x'^2+\gamma^2 y'^2}{2\sigma^2}} cos\left(2\pi\frac{x'}{\lambda} + \psi\right) \tag{3.20}$$

where $x' = xcos\theta + ysin\theta$, and $y' = -xsin\theta + ycos\theta$. In this equation, $\lambda$ represents the wavelength of the wavelet, $\theta$ is the in-plane rotational angle of the filter, $\psi$ is the phase offset between the peak and the valley of the wavelet, $\gamma$ is the aspect ratio between the horizontal and vertical direction before the in-plane rotation, and it specifies the ellipticity of the support of the Gabor function; $\sigma$ specifies the variance of the global envelope (which corresponds to the frequency bandwidth of the bandpass in the frequency domain). Figure 3.6 shows the appearance of the Gabor filter in both 2D and 3D view.

#### 3.4.4.2 Simulated Annealing

The Gabor filters have five parameters that need to be optimally selected to achieve the best discrimination. Thus we formulate this selection task into an optimization problem. In mathematics, our goal is to discover $\hat{g}$ so that

$$\hat{g} = \underset{\{\lambda,\theta,\psi,\sigma,\gamma\}}{\arg\min} \; AER(g(x, y; \lambda, \theta, \psi, \sigma, \gamma)) \tag{3.21}$$

where $AER()$ is a function that returns the Average Error Rate of the iris mask, given that we are using the Gabor filter $g()$ to extract features.

It is not easy for us to foresee what the shape of the objective function looks like in this five dimensional parameter space. However, it is very likely to be highly nonlinear with possible local minima and maxima in the error function. Simple methods like gradient descent may work but are not guaranteed to find the global optima parameters. We propose to use Simulated Annealing approach to optimize the parameters because this approach is known to be able to overcome the local optima problem and find the global optimal solutions.

Simulated Annealing (SA) is a stochastic global optimization method which can (1) process cost functions which have an arbitrary degree of nonlinearities and discontinuities; (2) process arbitrary boundary conditions and constraints imposed on those cost functions; (3) be implemented quite easily with minimal effort of coding compared to other non-linear optimization methods; (4) statistically guarantee the finding of an optimal solution [31, 32, 33, 34, 35].

In SA, each point $\mu$ of the search space is analogous to a state of some physical system, and the function $E(\mu)$ to be minimized is analogous to the internal energy of the system in that state. The goal is to bring the system from an arbitrary initial state to a state with minimum possible energy. At each step, the SA heuristic considers some neighbor $\mu'$ of the current state $\mu$, and probabilistically decides between moving the system to state $\mu'$ or staying in state $\mu$. The probabilities are chosen so that the system ultimately tends to move to states of lower energy. Typically, this step is repeated until the system reaches a state that is good enough for the application, or until a given computation budget has been exhausted. The probability of accepting the new state $\mu'$ is determined by an acceptance probability function $P(e, e', T)$, which depends on the energies $e = E(\mu)$ and $e' = E(\mu')$, as well as the time-varying parameter $T$, which is used to model the temperature in the real annealing process in metallurgy.

The algorithmic steps of SA can be described as the following:

1. Initialize the value for the initial parameter $\mu_0$ and the corresponding cost function value $e_{best} = E(\mu_0)$
2. Initialize the best parameter $\hat{\mu} = \mu_0$
3. Set the starting temperature $T$, maximal number of iteration $N$, and the target value of the cost function $E$
4. While the iteration number $n < N$ and best cost function value $e_{best} > E$
    (a) Select new parameter $\mu_n$ from the neighborhood of the previous parameter $\mu_{n-1}$
    (b) Compute its cost $e_n$

   (c) If $e_n < e_{best}$ then accept this new parameter, which means we set $\hat{\mu} = \mu_n$ and $e_{best} = e_n$
   (d) If $P(e_{n-1}, e_n, T)$ is greater than some random number, accept this new parameter too.
   (e) Decrease the temperature $T$ and set $n = n + 1$

5. When the iteration reaches its criteria, return $\hat{\mu}$.

In our experiments, because the final goal of automatic iris mask estimation is to generate an accurate mask in order to improve iris recognition rate, we choose to use iris recognition performance as our objective function that we want to maximize. That means, we are not using Eq. (3.21) as our cost function. Instead, we are using this formula:

$$\hat{g} = \underset{\{\lambda,\theta,\psi,\sigma,\gamma\}}{\arg\min} \; FRR_{0.1}(g(x, y; \lambda, \theta, \psi, \sigma, \gamma)) \tag{3.22}$$

where $FRR_{0.1}$ stands for the False Reject Rate (FRR) when False Accept Rate (FAR) equals 0.1%.

#### 3.4.4.3 Exploration of Optimal Gabor Filter Banks for Occlusion Estimation

Our goal is not simply to find the best Gabor filter. Instead, we would like to discover a set of Gabor filters, or a set of Gabor Filter Banks (GFB), that maximizes the performance of our proposed algorithm. GFB can be defined as

$$GFB_N = \{\hat{g}_1, \hat{g}_2, ..., \hat{g}_N\} \tag{3.23}$$

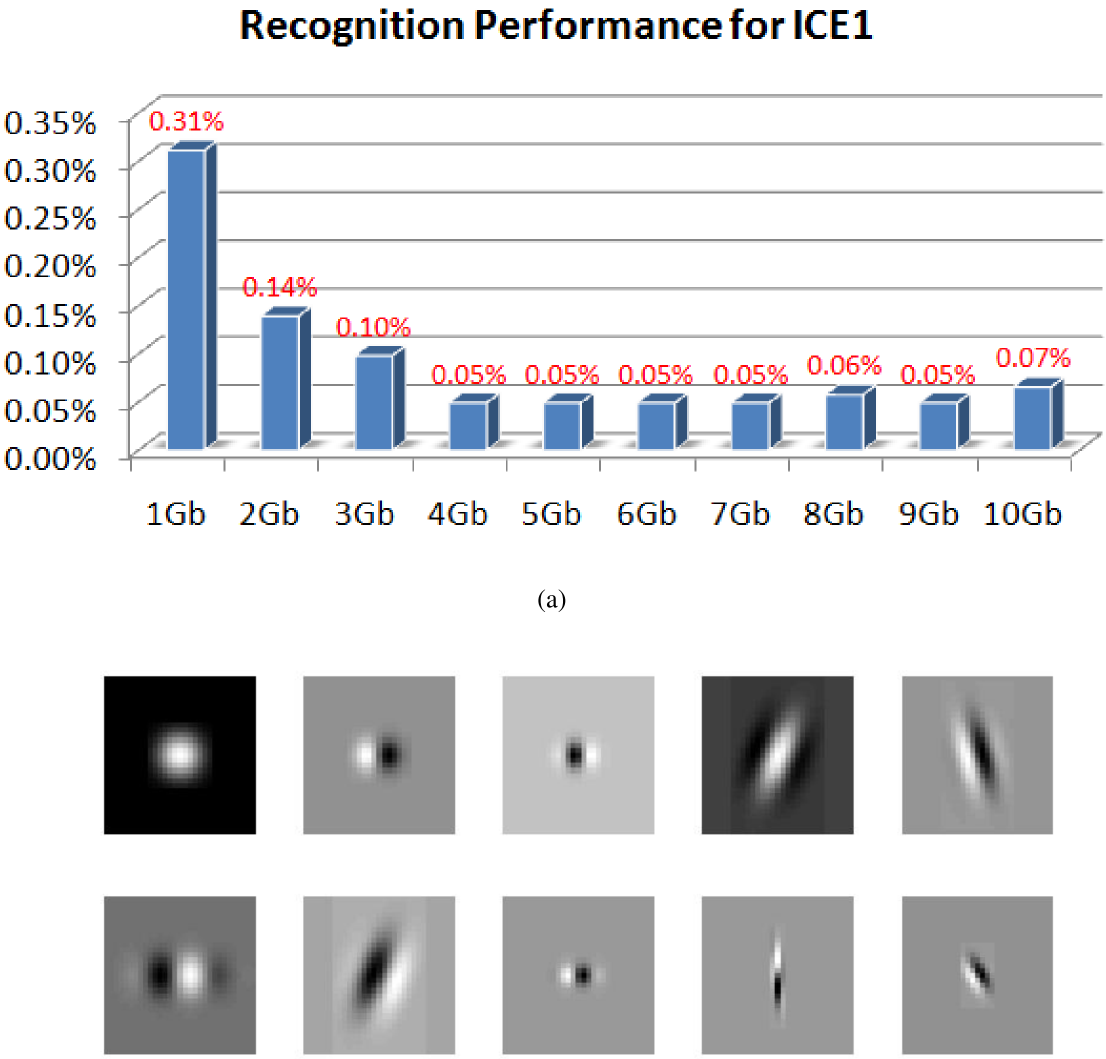
where $N$ denotes how many Gabor filters we want for the GFB.

We use an iterative procedure to discover the optimal GFB, and expand it one at a time. Our training procedure can be described as:

1. Estimate $\hat{g}_1$ by using Eq. (3.22), and set $GFB_1 = \{\hat{g}_1\}$.
2. Set N as the maximal number of Gabor filter we would like to discover.
3. When $i < N$, repeat the following steps:
   (a) Estimate $\hat{g}_{i+1}$ by $\hat{g}_{i+1} = \arg\min_{g_j} \; FRR_{0.1}(g_j(x, y; \lambda, \theta, \psi, \sigma, \gamma)|GFB_i)$
   (b) Set $GFB_{i+1} = GFB_i \cup \hat{g}_{i+1} = \{\hat{g}_1, \hat{g}_2, ..., \hat{g}_{i+1}\}$
4. Return $GFB_N$.

Figure 3.7 and 3.8 show the progressive results of the Gabor Filter Bank optimization, for the ICE1 and ICE2 databases, respectively. In sub-figure (a), the x-axis shows the number of Gabor filters in the GFB, and the y-axis shows the corresponding iris recognition performance (measured by the FRR at a fixed FAR of 0.1%). Sub-figure (b) shows the ten optimized Gabor filters that we obtained in each database after the optimization step.

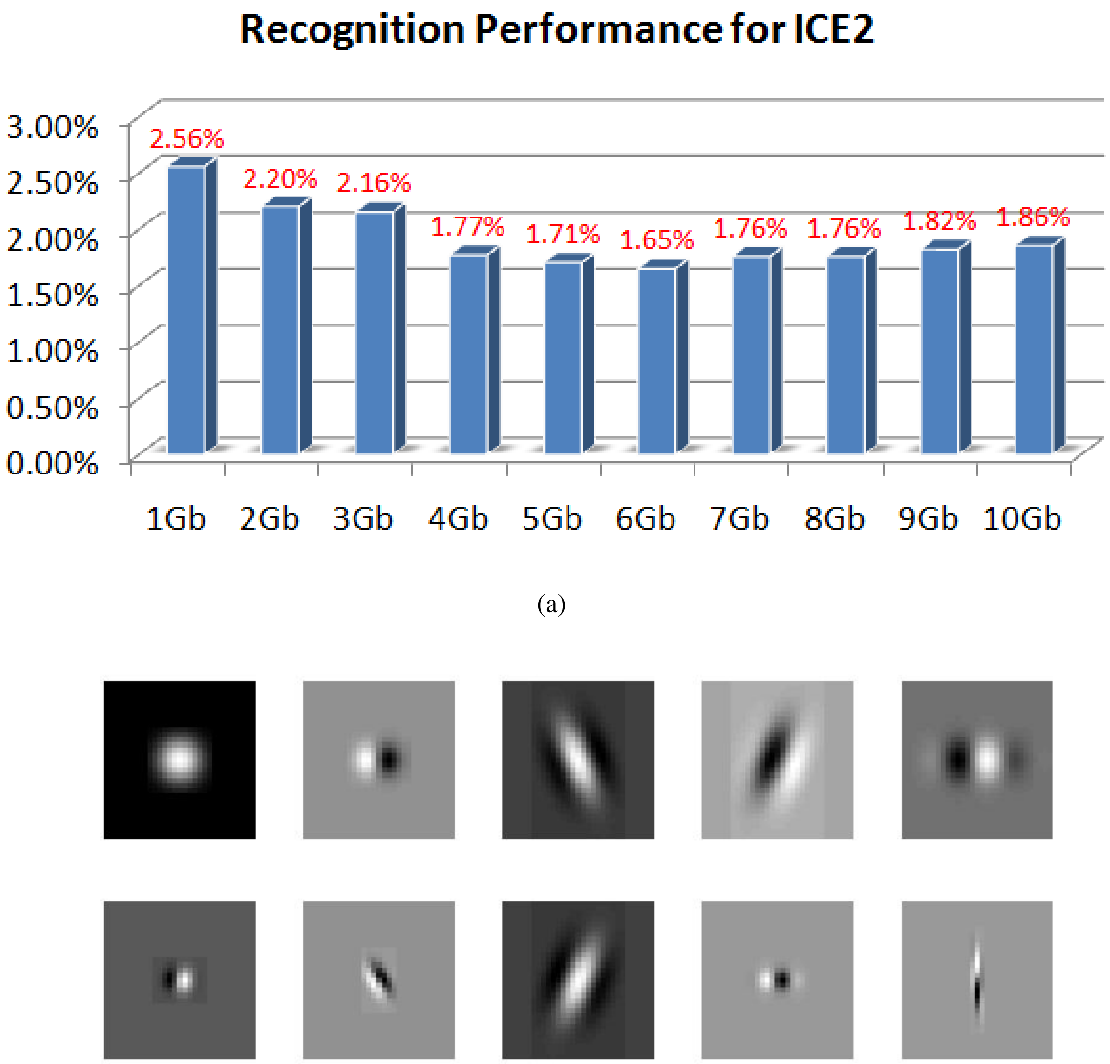
In order to show more details about the process of GFB optimization, we also plot the ROC curves for each of these ten experiments for both ICE1 and ICE2. Figure 3.9(a) shows the ROC curves of the large-scale iris recognition performance for the ICE1 database, when we use the best one, two, three and four Gabor filters that we obtained as the feature set. Figure 3.9(b) and



(a)

(b)

Figure 3.7: Progressive results for GFB optimization for ICE1. (a) Large-scale iris recognition performance measured with False Reject Rate (FRR) at 0.1% False Accept Rate (FAR), when using different number of Gabor Filters (from one to ten) for feature extraction; (b) the discovered 10 Gabor filters.



(a)

(b)

Figure 3.8: Progressive results for GFB optimization for ICE2. (a) Large-scale iris recognition performance measured with False Reject Rate (FRR) at 0.1% False Accept Rate (FAR), when using different number of Gabor Filters (from one to ten) for feature extraction; (b) the discovered 10 Gabor filters.

3.10(a) shows same result when we use four to seven Gabors, and then seven to ten Gabors, respectively.

Finally, in order to compare the results of GFB optimization with the baseline system, as well as the perfect system (which uses manually created masks), we show the ROC curves for all above cases in Figure 3.10(b).

Figure 3.11 and 3.12 shows the same results for the ICE2 database.

## 3.5 Discussion

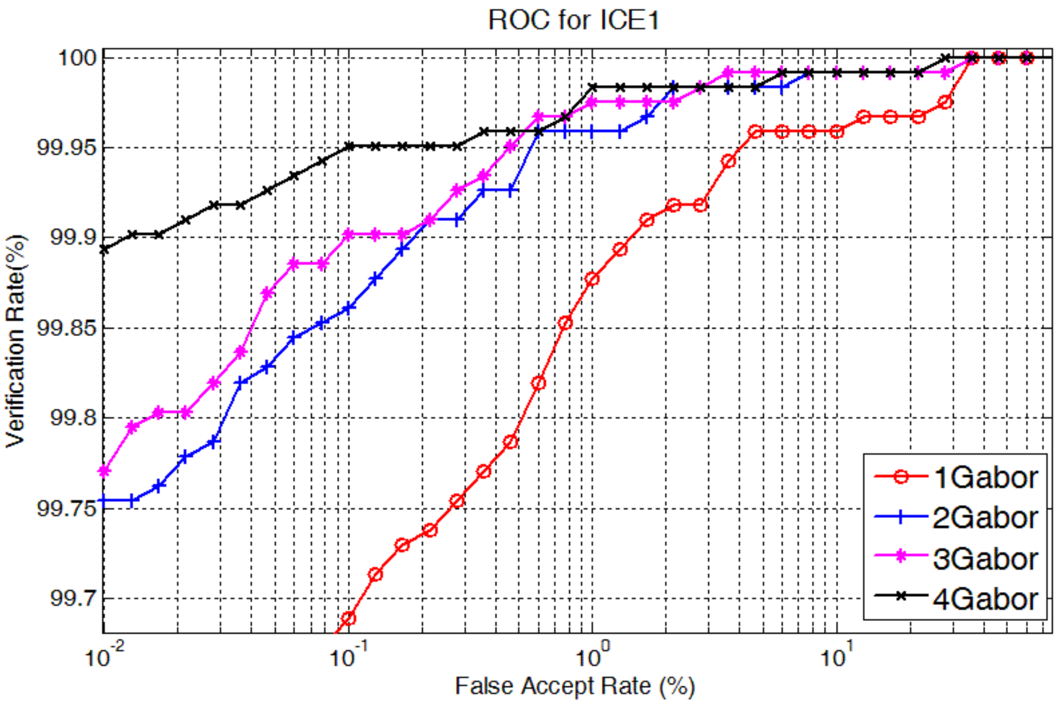
### 3.5.1 Average Error Rate for Occlusion Estimation

First, in terms of the accuracy of the generated iris mask, Figure 3.5 shows that in both the ICE1 and ICE2 database, our proposed method, no matter which feature set we use, is better than the two baseline systems (rule-based method and FLDA-based method). This result shows that the proposed approach is useful and able to create iris masks which are much more similar to manually created masks.

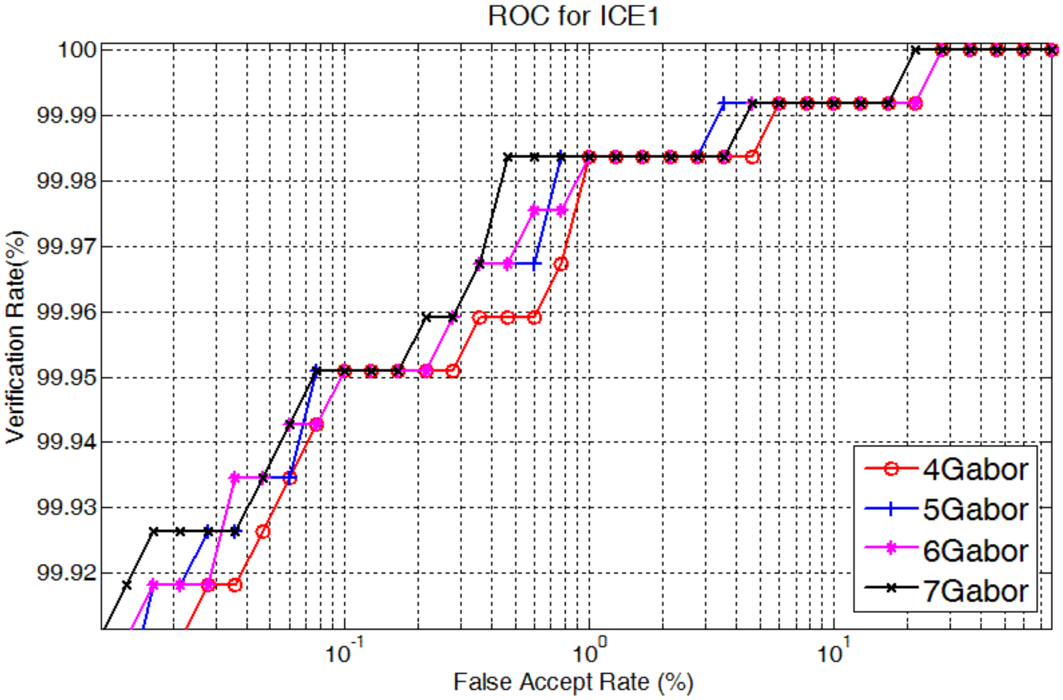
Second, from Figure 3.5 we found that feature IG seems to give the best result among all the feature combinations. Therefore, in sub-section 3.4.4 we start a series of experiments of GFB optimization. Using this GFB, we can further decrease the AER (for both the ICE1 and ICE2 databases). Figure 3.13 shows the final AER we get after GFB optimization, compared with the simplistic feature (IMS) as well as with the two baseline methods (FLDA and Rule-based methods). From Figure 3.13, we can see that our proposed method combined with the best feature can greatly improve the accuracy of the iris mask. Figure 3.14 shows the example iris masks generated by the proposed algorithm, with optimized GFB, for both ICE1 and ICE2 database.

### 3.5.2 Insight from Gabor Filter Bank Optimization

There are some insights we can draw from the the optimization process of the GFB features. For the ICE1 database, as the number of Gabor filters increases, we find that the recognition performance also increases. However the improvement of performance seems to saturate at the point when seven Gabor filters are used. When more than seven Gabor filters are used, the ROC curve begins to drop down, as shown in Figure 3.10(a). We thus conclude that, though multiple Gabor filters can give the most discriminative feature for occlusion estimation, there is a limit. When the number of features exceeds the optimal value, the less effective feature will confuse the generative model and degrade the performance. Only the good features can train better generative models. This same phenomenon can be observed in the case of the ICE2 database, as shown in Figure 3.11 and 3.12. For ICE2, the optimal number of Gabor filters is six, thus our observations and conclusions are correlated on different datasets.

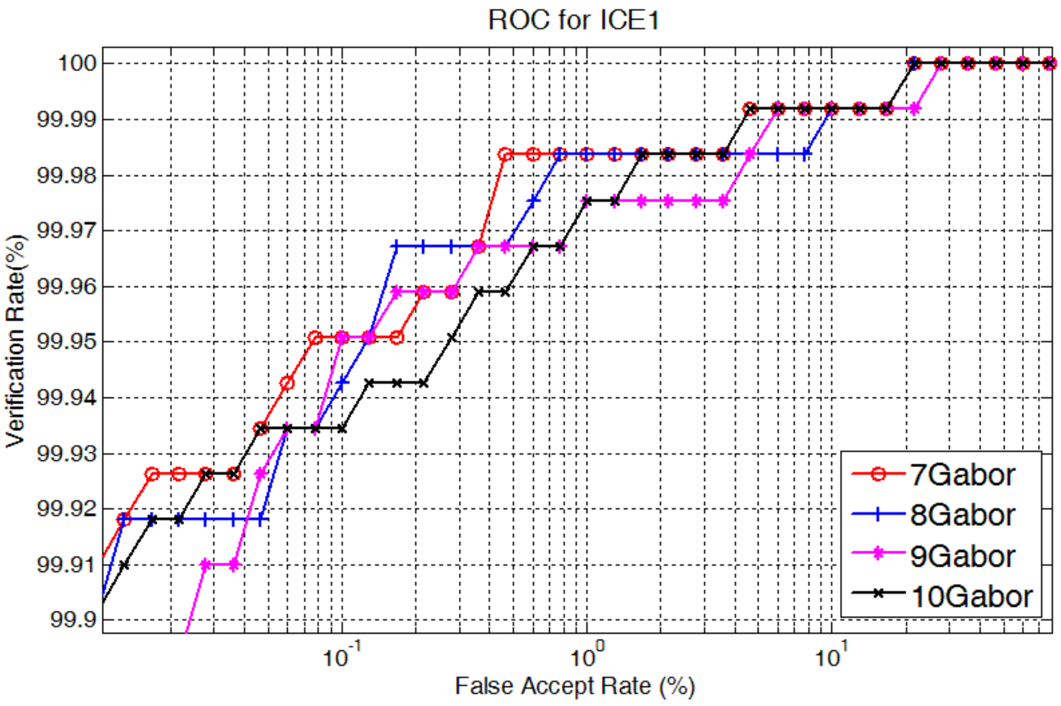


(a)

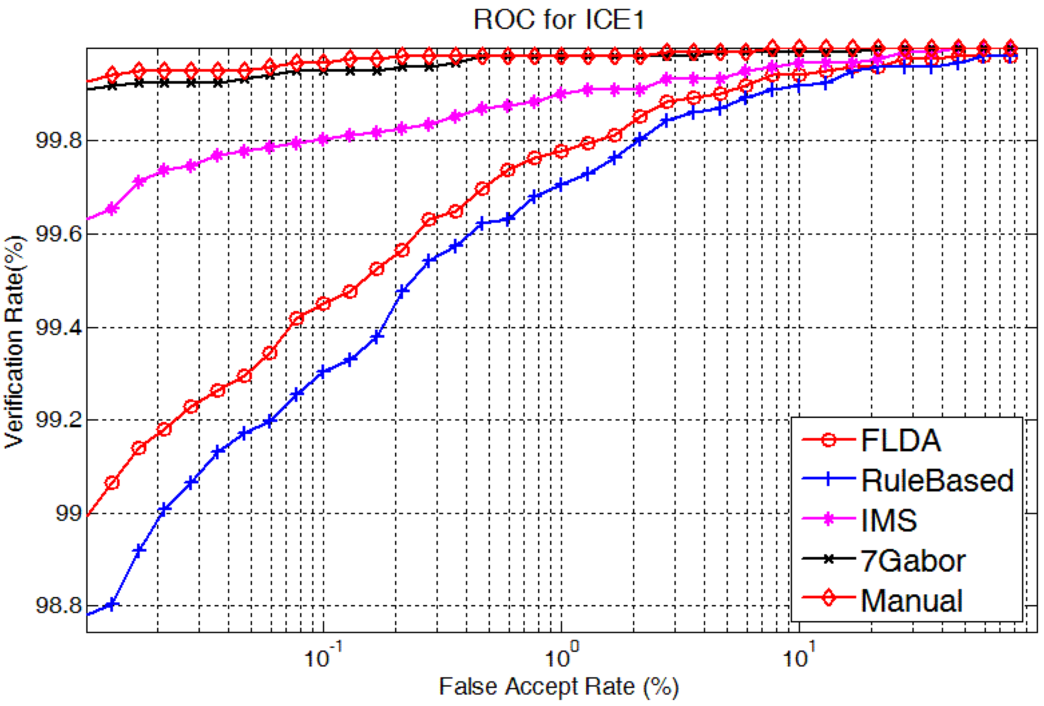


(b)

Figure 3.9: ROC curves that shows the how the recognition rate changes with the number of filters in GFB, with ICE1 database. (a) ROC curves of one to four Gabor filters (b) ROC curves of four to seven Gabor filters.

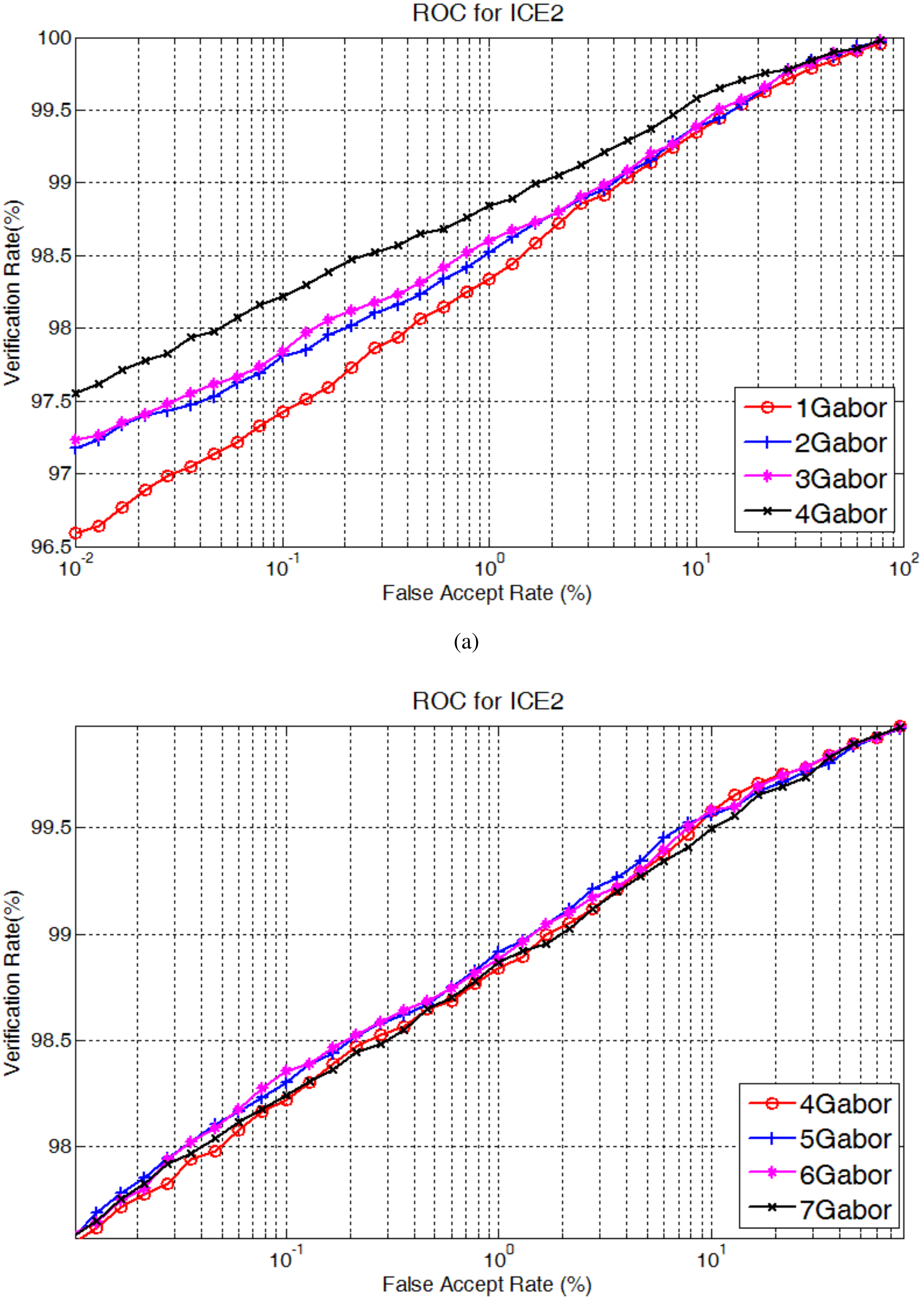


(a)



(b)

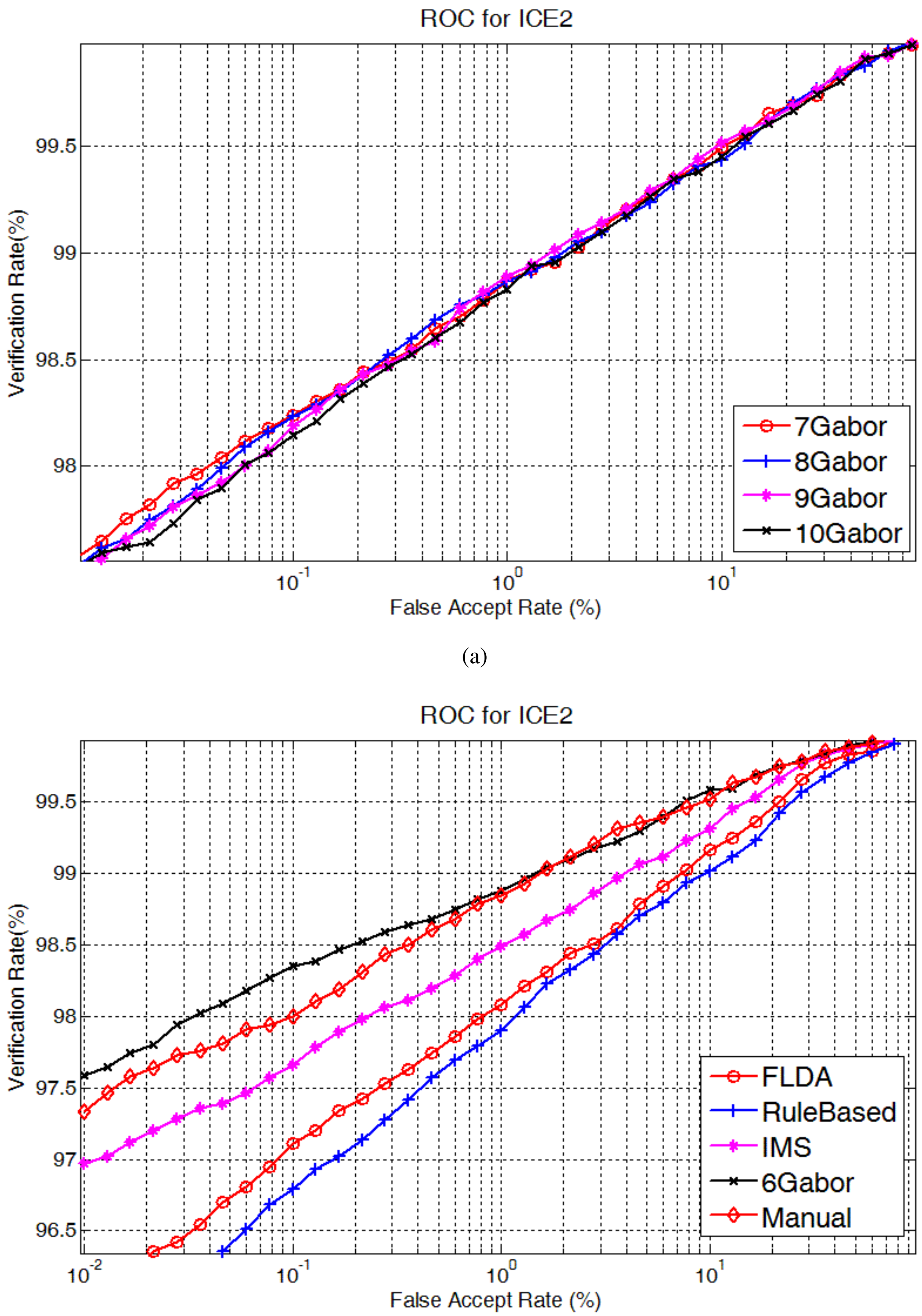
Figure 3.10: ROC curves that shows the how the recognition rate changes with the number of filters in GFB, with ICE1 database. (a) ROC curves of seven to ten Gabor filters (b) ROC curves comparison of the baseline, manual and the proposed method. The two baselines are FLDA and Rule-based method. The best performance of the proposed method is when we use 7 Gabor filters.



(a)

(b)

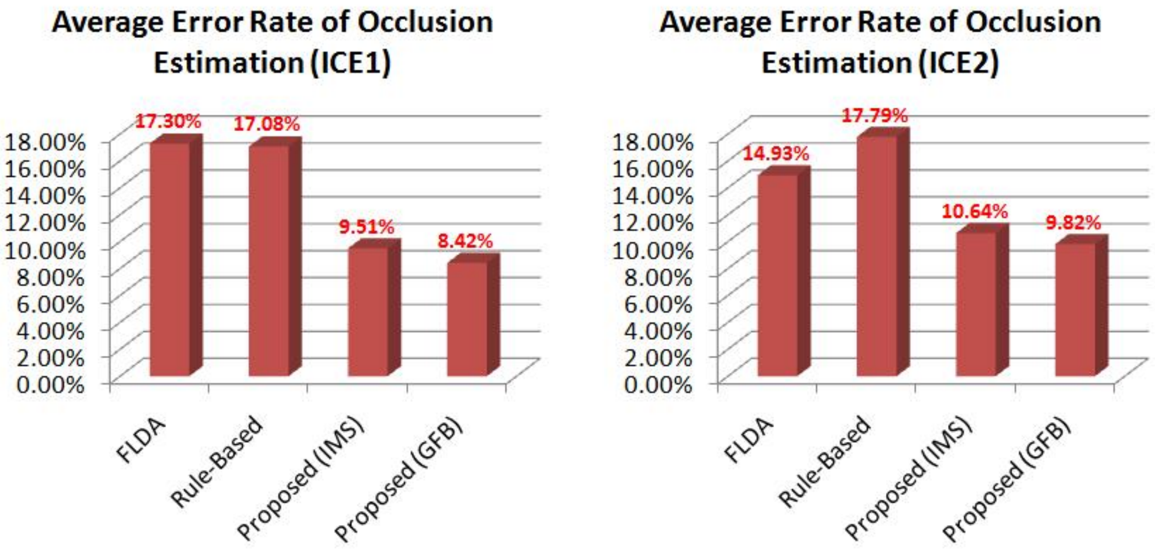
Figure 3.11: ROC curves that shows the how the recognition rate changes with the number of filters in GFB, with ICE2 database. (a) ROC curves of one to four Gabor filters (b) ROC curves of four to seven Gabor filters



(a)

(b)

Figure 3.12: ROC curves that shows the how the recognition rate changes with the number of filters in GFB, with ICE2 database. (a) ROC curves of seven to ten Gabor filters (b) ROC curves comparison of the baseline, manual and the proposed method. The two baselines are FLDA and Rule-based method. The best performance of the proposed method is when we use 6 Gabor filters.



(a) (b)

Figure 3.13: Average Error Rate (AER) of the iris occlusion estimation with different algorithm, which include (1) FLDA (2) Rule-based method (3) proposed method with IMS feature set (4) proposed method with optimized GFB. Sub-figure (a) and (b) show the results for ICE1 database and ICE2 database, respectively.

### 3.5.3 Performance Enhancement shown in ROC Curves

We can see how much the recognition performance is enhanced by our proposed algorithm, with the best feature set obtained through the SA optimization process, in Figure 3.10(b) and 3.12(b). In these two Figures, there are ROC curves that are based on five different masks. The ROC based on the masks that are generated from FLDA and rule-based methods, shows the baseline with which to be compared. The ROC based on the masks that are generated by the proposed algorithm, with the IMS feature set, is the performance with our proposed algorithm, before we perform GFB optimization. The ROC based on the manually labeled iris masks gave the best achievable performance if we manually identified the regions of occlusion. The ROC based on the masks that are generated by the proposed algorithm, with GFB optimization, is the ultimate performance we can get by the process of optimization.

From both Figure 3.10(b) and 3.12(b), we can see that without the optimization process, and only using simplistic features (image intensity, mean and standard deviation), we can greatly enhance the recognition performance. For ICE1, after GFB optimization, we can improve the recognition performance to a level that is very close to the manually-created mask.

For ICE2, the results are even more encouraging. After GFB optimization, the recognition performance can exceed the performance based on manually-created masks. This observed result suggests two things: 1. The masks generated by the proposed algorithm must be different from the manually created masks (if they are almost the same, both ROC curves should be very similar). 2. In order to have higher recognition performance, the masks generated by the pro-

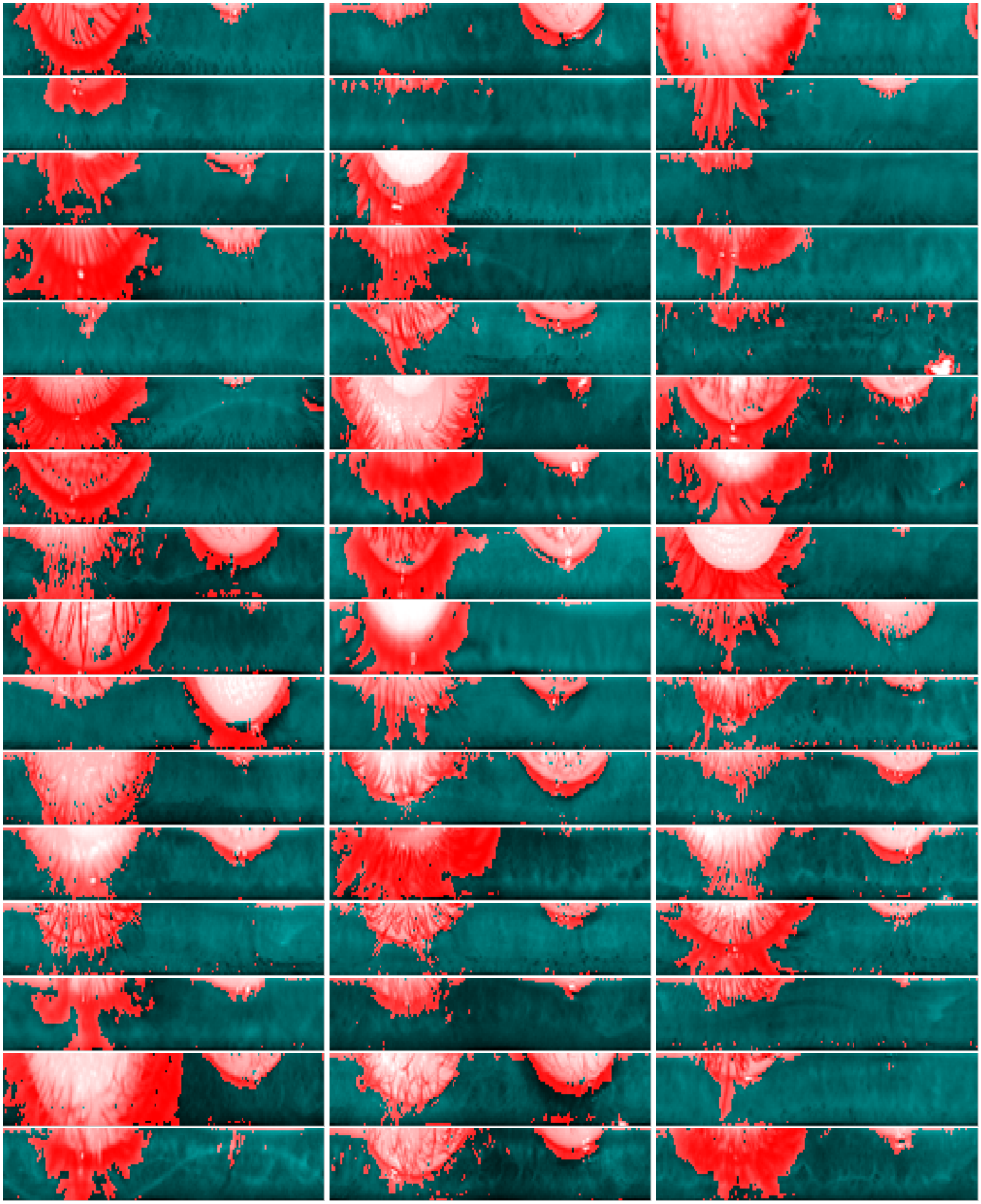


Figure 3.14: Iris occlusion estimated by the proposed algorithm. Red region denotes the estimated occlusion and dark green region denotes the estimated iris texture. Row 1~8 show the results from ICE1 (7 Gabor filters) and Row 9~16 show the results from ICE2 (6 Gabor filters).

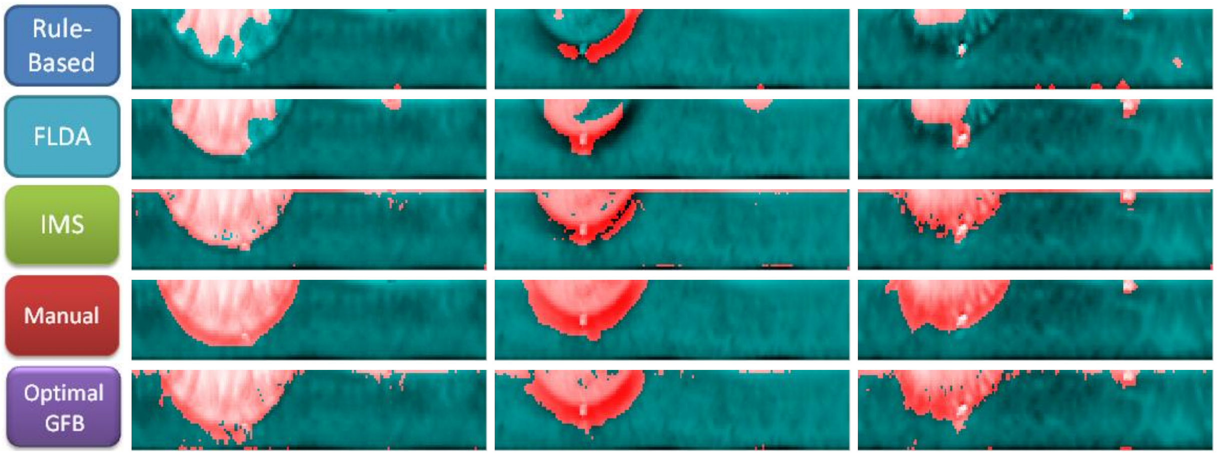


Figure 3.15: Comparison of the iris masks that are generated by different algorithms. Each column shows the iris masks that are generated by (1) Rule-based (2) FLDA (3) proposed method with IMS feature (4) manual (5) proposed method with GFB optimization. From these three examples we can see that GMM trained with optimal GFB can mask out the regions that contain less discriminative information for iris recognition.

posed algorithm must mask out some regions which contain less discriminative features for iris matching.

This observation and implication can be observed from Figure 3.15. In this figure, we show three examples of how iris masks “evolve” or “improve” with different algorithms. Each column represents the same input iris image being processed by different algorithms. The five algorithms we compared are (1) rule-based (2) FLDA (3) proposed method with IMS feature (4) manually labeled ground-truth data (5) proposed method with GFB optimization.

We can see that the proposed method with the IMS feature greatly improves the accuracy of the masks, compared to the two baseline methods. However, masks generated manually are slightly better than proposed method with IMS features, which explains the gap between the two ROC curves. Finally, the proposed method with GFB optimization further modifies the manual masks by eliminating the iris regions that have little or no discriminative information, e.g., the regions at the top of the images. If we wrap the polar domain iris image back to Cartesian domain, the top region of the polar domain image corresponds to the boundary between sclera and iris. Slightly inaccurate iris segmentation will include the sclera region into the polar domain image, causing the decrease of the recognition performance. Therefore, by masking out this region, the recognition rate can be further enhanced.

## 3.6 Conclusions

Extracting robust iris masks is one of the key factors to achieving high iris recognition rates. This is something that many researchers in this field have neglected. How to estimate iris occlusion and create accurate iris masks robustly and efficiently is key to high performance iris recognition. In this work, we have demonstrated that our approach for the iris occlusion estimation is accurate, efficient, and effective. It can detect the occluded region in an iris, and the detection result (the iris mask) is very similar to a manually-created mask, as shown in our empirical analysis. It is effective because we only use one training image from each class to get such a satisfactory result for generating iris texture masks. Experiments have also shown that the iris masks created by the proposed method can help to greatly improve iris recognition performance, and, in some cases, the performance can be even higher than the manually labeled masks.

This leads us to another interesting topic. If the recognition performance based on the manual masks can be further enhanced, then regions on authentic iris texture which are not very useful for recognition purpose must exist. If we can find these regions and mask them out, just like what our proposed algorithm did for the ICE2 database, the recognition performance can be further improved. How to determine these regions, and how to do it efficiently are two important issues. In chapter 4, we explore solutions for these questions and, hopefully, mask out more regions which are not discriminative.

# Chapter 4

# Automatic Iris Mask Refinement for Improved Iris Recognition Performance

Estimating artifacts on an iris image in the polar domain is an important problem for any iris recognition system which strives for a high recognition rate. In literature, there are many algorithms that estimate iris occlusion in either Cartesian or polar coordinates. In chapter 3, we proposed a novel learning-based iris occlusion estimation algorithm. Our goal in this chapter is not to propose another method to compete with existing ones. Rather, it is to propose a new algorithm which can take an iris mask estimated by any existing algorithm and refine it into a much more accurate mask. In this way, the proposed method could work side-by-side with any other existing algorithm and improve iris recognition performance. Experimental results show that the proposed method can improve iris recognition rate significantly compared to the performance of a system using un-refined iris masks.

## 4.1 Introduction

As discussed in chapter 3, iris occlusion estimation is an important problem in iris recognition systems. If all of the occluded regions in an iris image can be accurately masked, the iris recognition rate will be greatly improved. On the contrary, if the iris masks are not accurate and there are many occluded regions which are not identified by the iris masks, the iris recognition performance will be degraded. These facts have been shown in the section 3.4. A good illustration of an iris image in the polar domain and its corresponding mask is shown in Figure 3.1.

There are two approaches to estimate accurate iris masks from images. The first is to estimate the occluded region in the original images in Cartesian coordinates and then transform the mask into polar coordinates. The second is to estimate the occlusion from the image in the polar domain directly. In this chapter, we propose the latter approach. The advantage of estimating occluded region in the polar domain is similar to the benefit we get by performing iris normalization. In the Cartesian domain, there are many factors that vary all the time and are very difficult to model. Modeling the occluded region in the polar domain simplifies the problem and also increases the computational efficiency.

We propose a learning-based method which refines existing estimations of the occluded region on iris images in the polar domain by maximizing the mutual information on the bit-level. This method is simple, easy to implement, and computationally efficient. Experiments also show the masks refined by the proposed method can enhance the recognition performance compared to the original masks. All of the previous work related to iris occlusion estimation has been reviewed in section 3.2. Therefore, we will not repeat the literature review in this chapter. We describe the proposed method in section 4.2. Experimental settings and results are illustrated in section 4.3. Discussion and Conclusions are presented in section 4.4.

## 4.2 Methodology

The primary focus of this chapter is on the problem of refining existing iris masks with a learning-based method instead of iris mask generation from raw input iris images. Assuming there are at least two training images for each class of iris, and there are roughly estimated iris masks for both of them in the polar domain, by maximizing mutual information between every pair of images in the training data, we are able to refine the iris masks of each iris image. In this section, the proposed algorithm is demonstrated using only two training images, but the same procedure generalizes in situations with more images.

Suppose these two iris images in the polar domain are denoted as $X$ and $Y$, and their corresponding masks are $M_X$ and $M_Y$. The mutual information between these two masks can be defined as (4.1)

$$I(M_X; M_Y) = \sum_{y \ni M_Y} \sum_{x \ni M_X} P(x,y) log \frac{P(x,y)}{P(x)P(y)} \tag{4.1}$$

where $x$ and $y$ represents each pixel in the mask $M_X$ and $M_Y$ in the polar domain. Intuitively, the regions in $X$ which have high resemblance to $Y$ are the regions which are more resistant to noise and image distortion. Therefore, we propose to recover the refined mask $RM$ by maximizing the mutual information between $X$ and $Y$:

$$\begin{aligned} RM &= \arg\max_X I(X; M_Y) & (4.2) \\ &= \arg\max_X \sum_{y \ni M_Y} \sum_{x \ni X} P(x,y) log \frac{P(x,y)}{P(x)P(y)} \end{aligned}$$

Assuming all bits on the polar plane are equally likely to be discriminative regions, $P(x)$ has a flat distribution across the whole plane. Then (4.2) can be rewritten as

$$RM = \arg\max_X \sum_{y \ni M_Y} \sum_{x \ni X} P(x,y) log \frac{P(x,y)}{P(y)} \tag{4.3}$$

By definition of conditional probability, we can rewrite (4.3) into (4.4)

$$RM = \arg\max_X \sum_{y \ni M_Y} \sum_{x \ni X} P(x|y)P(y) log P(x|y) \tag{4.4}$$

The probability $P(y)$ is modeled by the given roughly estimated mask $M_Y$, set $P(y) = 1$ if and only if the pixel value of $y$ equals one on $M_Y$ and $P(y) = 0$ if and only if pixel value of $y$ equals zero on $M_Y$. For $P(x|y)$, it is modeled with the similarity scores between $X$ and $Y$, given the mask of $M_Y$. Specifically, we set $P(x|y) = 1$ if the pixel at location $x$ on image $X$ matches the same location on image $Y$, given the roughly estimated mask $M_Y$, otherwise, set $P(x|y) = 0$. In other words, we model $P(x|y)$ as a binomial distribution.

## 4.3 Preliminary Experiments

### 4.3.1 Iris Databases Description

Experiments are performed on the NIST ICE database, as described in section 3.4.1. We used both subsets of ICE, which is ICE1 and ICE2.

For every iris class in both datasets, we partitioned the images into training and test sets with an equal number of images for both sets. When there is only one image for a class, it is included in training data but not the test data. We estimated and refined iris masks on the training set, used training set images (partially or totally) as gallery images, and matched them against the probe images, which are the test data.

### 4.3.2 Preliminary Estimation of Iris Masks

We experimented with three different methods for roughly estimating initial iris masks, which had the possibility of being inaccurate and error-prone, and then used the proposed method to enhance the performance of the masks. The three methods are described in following sub-sections:

- Rule-based method

  The first method we tried is a simple rule-based method. The rule-based method we used here is similar to the method described in [22]. Basically, it detects whether there is strong variance of pixel intensity on a local window and uses it as a feature to perform classification. It can be illustrated in four steps:

  1. Normalize the image so that the energy of pixel intensity sums to one.
  2. Create a local mean image by computing the mean value of the 5x5 window centered at each pixel.
  3. Compute the global mean and standard deviation of all pixel intensity of the image of the mean.
  4. For every pixel on the local mean image, if the difference between its intensity value and the global mean is less than twice the global standard deviation, classify it as the iris texture. Otherwise, classify it as an occluded region.

- FLDA-based method

  The second method we tried is an FLDA-based method, as described in [12]. It can be illustrated in following steps:

1. Extract features from the normalized iris image. The extracted features include:
    (a) the mean intensity value in a small neighborhood of the pixel
    (b) the standard deviation of the intensity values in the same neighborhood
    (c) the percentage of pixels whose intensity is greater than one standard deviation above the mean of the entire iris plane
    (d) the shortest Euclidean distance to the centers of the upper and lower eyelids.
2. Use the FLDA method to reduce dimensionality of the features from four to one.
3. Set a threshold for the feature value to classify a pixel into either the "authentic iris" or "occluded region" class.

- Active-contour based method

  Daugman recently proposed a new iris segmentation algorithm, based on the active contour technique (Snake), as shown in [21]. With the new segmentation scheme and other improvements in score normalization, eyelash detection and off-axis eye normalization, he reported a very high accuracy on the ICE database. Therefore, we also implement the iris mask estimation algorithm by using snake, in order to re-evaluate it fairly and compare it with other methods. The snake method we used here is the GVF Snake algorithm, as described in [36].

### 4.3.3 Refining Iris Masks

After the initial estimation of the iris mask is computed, the initial iris masks are refined with the proposed method. Since the separation of training data and test data is a random process, to make the experimental results fair and reasonable, the whole training and testing process is repeated ten times. This will eliminate any chance factors that bias the performance due to the selection of training data. The experimental results reported in section 4.3.4 are derived by accumulating all Hamming Distance from the ten-fold cross validation for all authentic and impostor comparisons.

For each experimental setting (one specific initial mask estimation algorithm on one specific database), we ran three sub-experiments and used ROC curves to compare their individual performance. First, we benchmarked the performance of the initial mask estimation algorithm, without using the proposed method to refine the mask. This setting is called "baseline" in subsequent sections. Second, we refined the masks from the baseline method, randomly picked one training image per class to be the gallery image, and used all untrained images as probe images. This setting is called "Algorithm 1" in subsequent sections. Finally, we used every image of every class in the training data as gallery images, and use all the untrained images as probe images. This setting is called "Algorithm 2" in subsequent sections.

We split the proposed method into "Algorithm 1" and "Algorithm 2" because we want to push the algorithm to the limit to see how much improvement in matching accuracy it can achieve, even in the case where only one training image is selected to be the gallery image. We expect that the proposed algorithm will enhance the matching accuracy to a significant level so that, even if there is only one training image per class, the iris recognition performance will still be significantly improved. The use of "Algorithm 1" is an attempt to validate this expectation.

Table 4.1: Numeric performance benchmarks for the three different settings, averaged over ten iterations, on ICE1 dataset

| Experimental Settings | Baseline | | | |
|---|---|---|---|---|
| Initial Mask Estimation Method | EER | FRR@.1 | FRR@0 | FR |
| Rule-based | 1.9 | 3.1 | 9.3 | 2.4 |
| FLDA-based | 2.1 | 3.7 | 9.3 | 2.3 |
| Active Contour based | 2.0 | 3.4 | 10.8 | 2.4 |

| Experimental Settings | Algorithm 1 | | | |
|---|---|---|---|---|
| Initial Mask Estimation Method | EER | FRR@.1 | FRR@0 | FR |
| Rule-based | 0.8 | 1.1 | 2.3 | 4.3 |
| FLDA-based | 0.8 | 1.0 | 2.1 | 4.2 |
| Active Contour based | 0.8 | 1.3 | 2.9 | 4.1 |

| Experimental Settings | Algorithm 2 | | | |
|---|---|---|---|---|
| Initial Mask Estimation Method | EER | FRR@.1 | FRR@0 | FR |
| Rule-based | 0.7 | 1.0 | 3.1 | 4.4 |
| FLDA-based | 0.7 | 1.0 | 3.7 | 4.3 |
| Active Contour based | 0.7 | 1.3 | 3.4 | 4.2 |

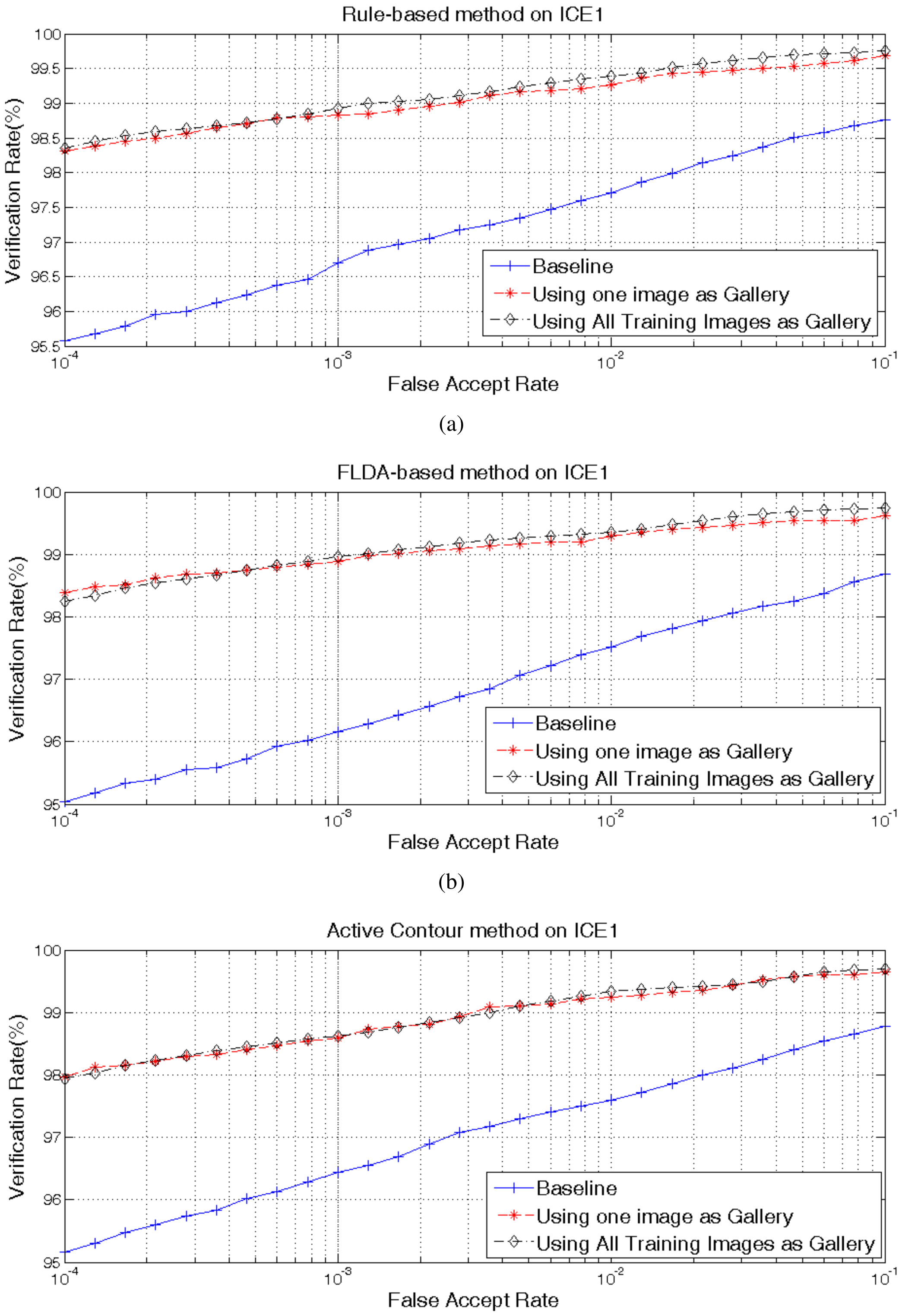
For iris feature extraction, we used Libor Masek's Matlab implementation of Daugman's algorithm, which can be freely downloaded, as stated in [16].

### 4.3.4 Results

As stated in section 4.3.3, we present the experimental results as ROC curves for different experimental settings. The results are presented in two parts: First, ROC curves are plotted to illustrate the performance of the baselines, Algorithm 1 and Algorithm 2, with respect to each initial iris mask estimation method on a given database. According to section 4.3.2, there are three different methods for estimating an initial iris mask and there are two databases (ICE1 and ICE2). Therefore, in the first part of this section, there are six ROC plots. They are shown in Figure 4.1 and 4.2.

In the second part of the section, we observe the effect of the proposed algorithm on the performance of all three initial occlusion estimation algorithms. This comparison would give us an idea as to whether the proposed algorithms (Algorithm 1 and 2) favor a particular initial estimation method. In addition, more insights can be obtained about how to choose an initial method wisely. The results are shown in Figure 4.3 and 4.4.

For every setting, besides ROC curves, we also measure some numeric performance benchmarks. The benchmarks are: (1) Equal Error Rate (EER); (2) False Reject Rate when False Accept Rate (FAR) equals to 0.1%, denoted as FRR@.1; (3) False Reject Rate when False Accept Rate (FAR) equals to 0%, denoted as FRR@0; (4) Fisher Ratios (FR). The results are shown in Table 4.1 and 4.2.

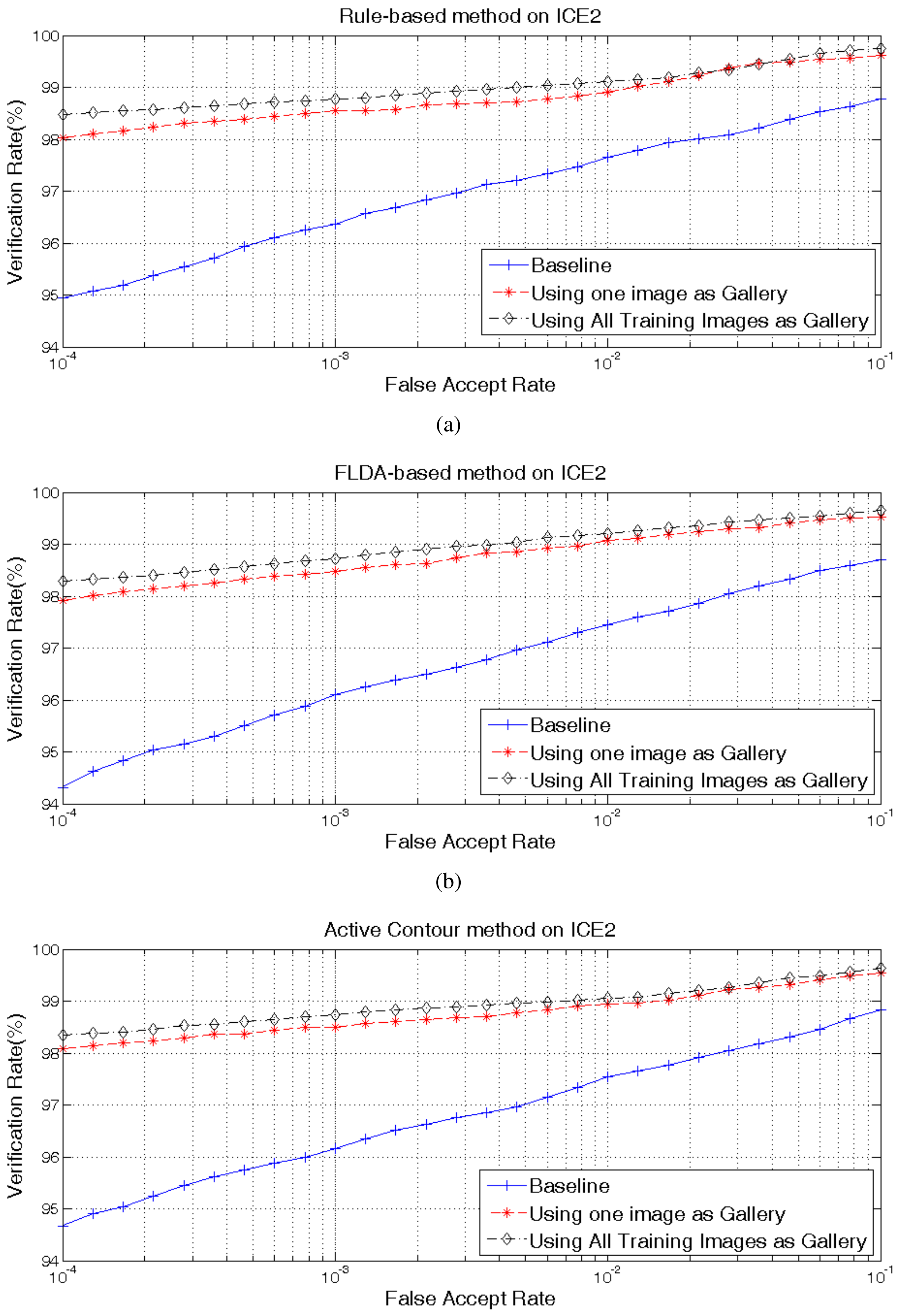


(a)

(b)

(c)

Figure 4.1: ROC curves for different iris mask estimation algorithm, on ICE1 database. (a) Rule-based iris mask estimation algorithm. (b) FLDA-based iris mask estimation algorithm. (c) Active Contour based iris mask estimation algorithm.

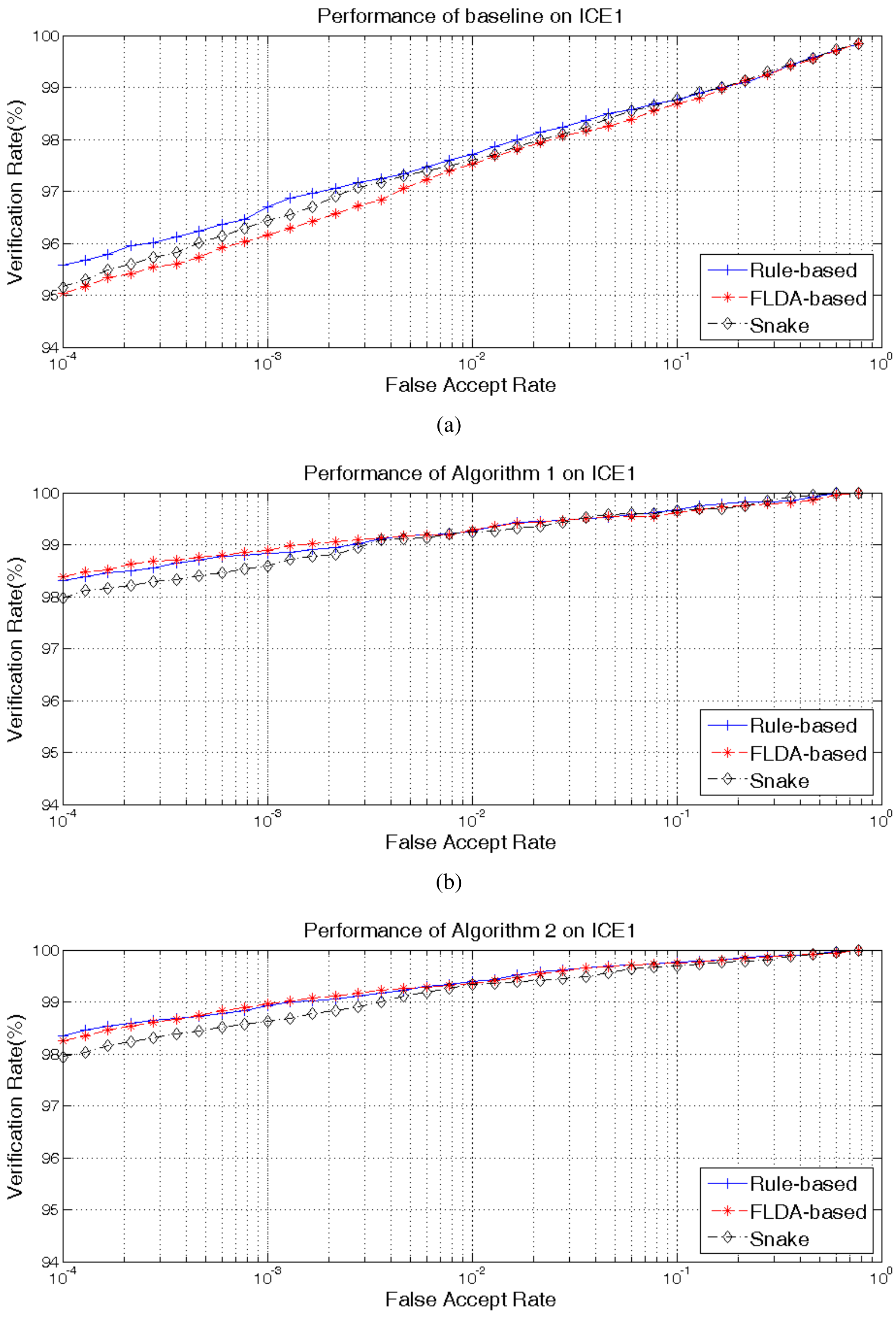


(a)

(b)

(c)

Figure 4.2: ROC curves for different iris mask estimation algorithm, on ICE2 database. (a) Rule-based iris mask estimation algorithm. (b) FLDA-based iris mask estimation algorithm. (c) Active Contour based iris mask estimation algorithm.

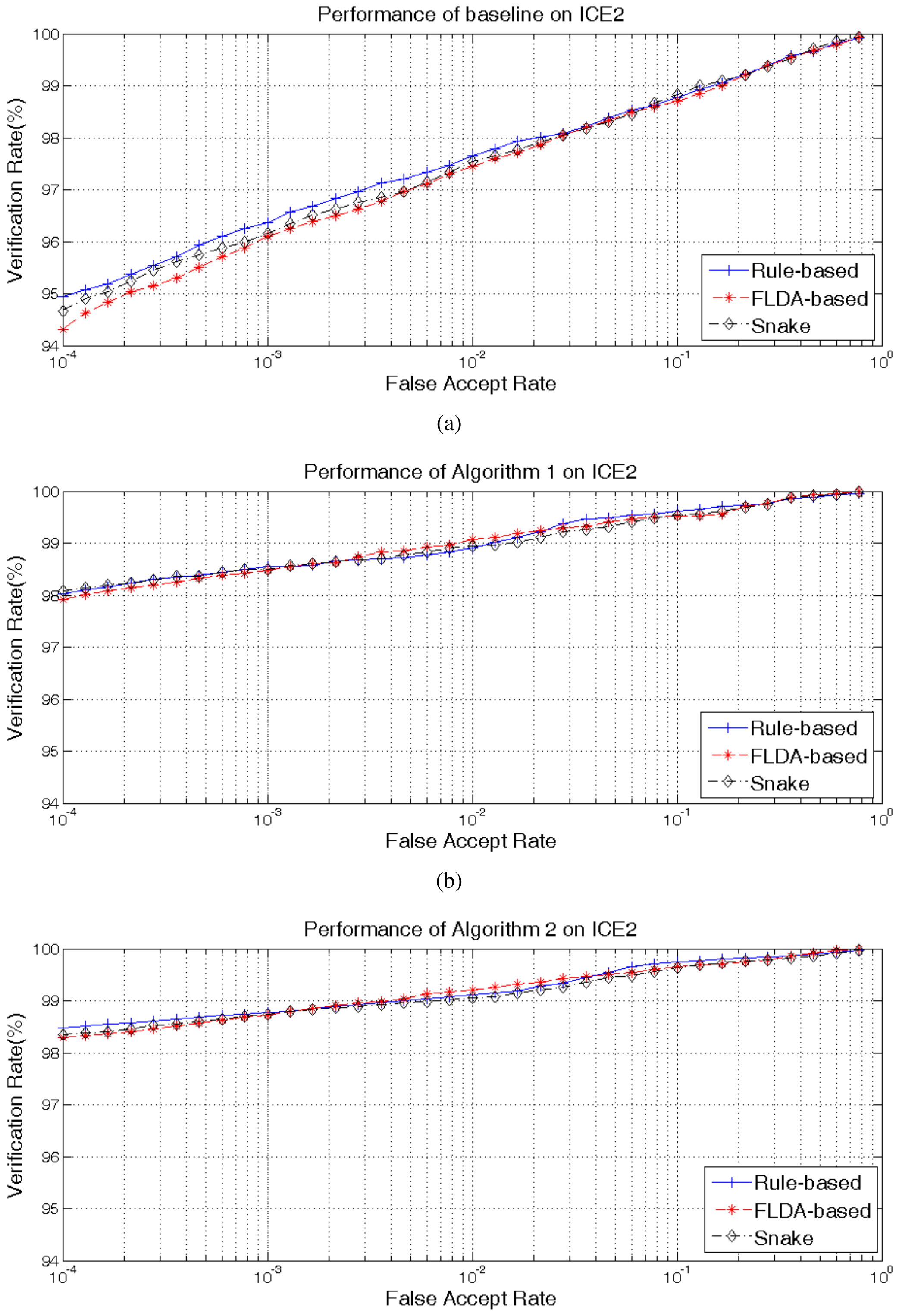


(a)

(b)

(c)

Figure 4.3: ROC curves for benchmark the performance of baseline, Algorithm 1 and 2, on ICE1 database. Within each plot, performance of three initial iris mask estimation methods is shown. (a) Baseline performance for all three methods. (b) Performance of all three methods, after improved by Algorithm 1. (c) Performance of all three methods, after improved by Algorithm 2.



(a)

(b)

(c)

Figure 4.4: ROC curves for benchmark the performance of baseline, Algorithm 1 and 2, on ICE2 database. Within each plot, performance of three initial iris mask estimation methods is shown. (a) Baseline performance for all three methods. (b) Performance of all three methods, after improved by Algorithm 1. (c) Performance of all three methods, after improved by Algorithm 2.

Table 4.2: Numeric performance benchmarks for the three different settings, averaged over ten iterations, on ICE2 dataset

| Experimental Settings | Baseline | | | |
|---|---|---|---|---|
| Initial Mask Estimation Method | EER | FRR@.1 | FRR@0 | FR |
| Rule-based | 2.0 | 3.4 | 9.3 | 2.3 |
| FLDA-based | 2.1 | 3.7 | 12.9 | 2.2 |
| Active Contour based | 2.1 | 3.6 | 10.7 | 2.3 |

| Experimental Settings | Algorithm 1 | | | |
|---|---|---|---|---|
| Initial Mask Estimation Method | EER | FRR@.1 | FRR@0 | FR |
| Rule-based | 1.1 | 1.5 | 2.3 | 4.4 |
| FLDA-based | 0.9 | 1.5 | 2.9 | 4.3 |
| Active Contour based | 1.1 | 1.4 | 2.4 | 4.3 |

| Experimental Settings | Algorithm 2 | | | |
|---|---|---|---|---|
| Initial Mask Estimation Method | EER | FRR@.1 | FRR@0 | FR |
| Rule-based | 0.9 | 1.2 | 2.3 | 4.6 |
| FLDA-based | 0.8 | 1.2 | 2.7 | 4.5 |
| Active Contour based | 1.0 | 1.2 | 2.3 | 4.4 |

## 4.4 Discussion and Conclusions

### 4.4.1 Effectiveness of the Proposed Algorithm

All ROC plots in Figure 4.1 and 4.2 confirm the effectiveness of the proposed algorithm. For every subplot in Figure 4.1 and 4.2, Algorithm 2 out-performs Algorithm 1, which in turn, out-performs the baseline. This result is consistent with our assumptions from before: Our proposed method improves the accuracy of the iris mask, which, in turn, improves the accuracy of the matching experiment. It is also observed that using more training images in the gallery set is better than using fewer images.

### 4.4.2 Algorithm Performance Comparison

ROC plots in Figure 4.3 and 4.4 also show the comparison of the accuracy enhancement through Algorithm 1 and Algorithm2. Although in the subplots in Figure 4.2, Algorithm 2 is constantly better than Algorithm 1, the performance of these too algorithms is similar and the difference between the two ROC curves is very small. This tells us that, even when we randomly pick only one image to be in the gallery set, after using our proposed algorithm to refine the iris mask, we can still achieve a high recognition rate. This is a significant advantage of the proposed algorithm, especially when dealing with a large-scale iris recognition task. For online iris recognition systems, where efficiency is critical, decreasing the number of gallery images as the number of classes increases significantly improves the matching speed.

### 4.4.3 Choice of Initial Iris Mask Estimation Algorithm

From the ROC plots in Figure 4.3 and 4.4, we get an insight about how to choose the initial iris mask estimation algorithm. First, without using the proposed algorithm to refine masks, in terms of baseline performance, Figure 4.3(a) and 4.4(a) show that among these three initial estimation methods, the rule-based method performs the best, and the FLDA-based method performs the worst. But after applying the proposed method, the performance becomes similar among all three cases, as shown in Figure 4.3 (b)-(c) and Figure 4.4 (b)-(c). In other words, no matter which initial estimation algorithm we started with, after applying the proposed refinement algorithm, the iris masks can be improved to approximately the same level of accuracy. This gives us the freedom to choose whatever algorithm is most convenient. This is also an advantage of the proposed algorithm.

### 4.4.4 Conclusion

Experimental results have shown that our proposed iris mask refinement algorithm is effective and useful. It can refine the iris mask to a much higher quality and improve the iris recognition rate significantly. Moreover, the proposed method is easy to implement, and can work side-by-side with any existing iris mask estimation algorithm. The proposed algorithm can be implemented in existing iris recognition systems as an additional layer between iris mask generation and matching, making it suitable to be used in a real-world situation without the need of large scale modification of the original system.

One limitation of our proposed method is that it requires at least two training images for the same iris class in order to estimate mutual information. This limitation should be easily overcome because modern iris recognition systems usually take more than one iris image during the enrollment stage. In fact, most of the newly released iris databases contain more than one image per class.

# Chapter 5

# Image Restoration and Enhancement for Iris Images from the Long-Range Iris Acquisition System

## 5.1 Introduction

As stated in Chapter 1, the IOM system has the ability to capture a subject's iris image from a large stand-off distance, while the subject is walking through the portal. Here we notice that for a less constrained iris capturing device like the IOM, the optical conditions required to capture a quality image are difficult to obtain. Image capture issues in less constrained systems arise from the following:

1. Motion blurring due to subject movement.
2. Uneven illumination of the target objects caused by the locations of the infrared illuminators.
3. In some cases, the images are taken while the subject is located outside of the focus zone; therefore, the images may suffer from out-of-focus blur.

On the other hand, images taken by a constrained device, such as SecuriMetrics PIER [37], have no such problem.

Figure 5.1 shows an iris image captured by the IOM system, compared with one captured by SecuriMetrics PIER device. From the pictures shown, it can be seen that there is a considerable difference in picture quality between the IOM images and the PIER images. Irises texture is revealed clearly in PIER images, while in IOM images, the motion blur, out-of-focus blur and uneven illumination reduce quality of the images. Iris textures cannot be seen clearly in those images.

In order to overcome this problem, we propose to use the technique of image Super-Resolution (SR) to perform image restoration and enhancement. SR is a technique which estimates high-resolution (HR) images from low-resolution (LR) ones. Depending on the input and output constraints, SR can be divided into one of the following three categories:

1. Single Input, Single Output (SISO): it is the case where only one image is used as input,

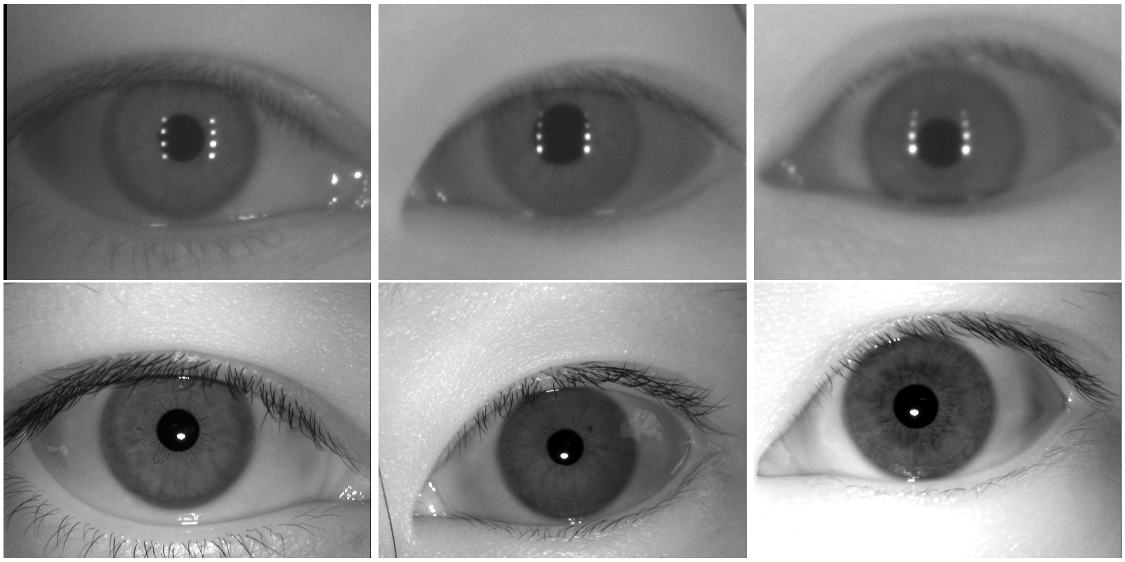


Figure 5.1: Comparison of image quality between IOM images and PIER images. First row shows images captured by IOM system. Second row shows images of exactly the same eye, captured by SecuriMetrics PIER device.

and only one output image is acquired, as shown in [38, 39, 40, 41, 42].

2. Multiple Input, Single Output (MISO): it is the case where several LR images are used as input, and one HR image is generated as output, as shown in [43, 44, 45, 46, 47, 48, 49].
3. Multiple Input, Multiple Output (MIMO): it is the case where the input of SR is a LR video and the output is HR video, as shown in [50, 51].

In our case, we would like to enhance the quality of iris images acquired by the long-range iris acquisition system. Since an advantage of such system is the ability to capture several images per subject, the suitable SR in this case is to use multiple images to produce a single restored and enhanced output image, i.e., MISO. In the section 5.2, an image degradation model which is widely used in the SR research community is presented. After that, in section 5.3, we introduce some of the famous SR algorithms in the MISO category. In section 5.4, we introduce our proposed SR algorithm and describe all of the details in every functional unit. Experimental results of both the existing and the proposed algorithms are also presented to show the effectiveness of our proposed algorithms in section 5.5. In order to explore the limit of the ability of our proposed algorithm in dealing with focus blur and noises, we design two experiments to simulate the possible degradation process in a quantitative manner. The results are shown in section 5.6. Lastly, We summarize the findings and conclude in section 5.7.

## 5.2 Image Degradation Model for Super-Resolution

In the literature, the process of image degradation can be modeled by the original image convolved with a Point Spread Function (PSF) of the camera (for modeling of defocus blur), followed by a convolution with a motion operator (for modeling of motion blur) and then mixed with Gaussian white noise [52]. This image degradation model is illustrated in Figure 5.2.

This process can be described in matrix form as

$$Y = DMGX + N \tag{5.1}$$

where $X$ denotes the original high quality image, represented as a column vector.

Suppose the original high quality image is of size $L_1N_1 \times L_2N_2$, then $X$ is a column vector of size $L_1N_1L_2N_2 \times 1$, arranged in lexicographical order. $Y$ denotes the observed LR image. Suppose the observed LR image is of size $N_1 \times N_2$, then $Y$ is a column vector of size $N_1N_2 \times 1$, arranged in lexicographical order. $L_1$ and $L_2$ are the down-sampling factors for the horizontal and vertical directions between the original HR image and observed LR image. $G$ is the matrix operator used to model camera PSF. Camera PSF may introduce optical blur or affine warping between the input and output image. Since the size of the input and output images are the same, $G$ is of size $L_1N_1L_2N_2 \times L_1N_1L_2N_2$. $M$ represents a $L_1N_1L_2N_2 \times L_1N_1L_2N_2$ motion blur matrix, which models the motion blur effect between the input and output images. $D$ is a down-sampling matrix, of size $N_1N_2 \times L_1N_1L_2N_2$. $N$ is to model the additive noise. The size of $N$ is $N_1N_2 \times 1$.

## 5.3 Previous Work

To the best of the author's knowledge, there has been no previous work done on the application of the super-resolution technique for the goal of enhancing the quality of iris images. Therefore, in this section, we will briefly review the existing work of generic image super-resolution.

Generally speaking, the image super-resolution algorithm can be divided into three stages, as illustrated in Figure 5.3.

The first stage in Figure 5.3 is image registration. In this stage, one of the input images is chosen to be the template image. Image alignment algorithms are applied on all of the other input images to measure the most likely rotation and translation between them. After image alignment is completed, it is possible to determine how much rotation and translation each input image has with respect to the template image. The rotation and translation in opposite direction is applied in order to correct all of the other images. During image reconstruction, we fuse the information from input images into one final image. Finally, during the post-processing stage, we apply image processing techniques to further remove noise and blur. In the following sub-sections, we review some of the most popular algorithms for each stage.

### 5.3.1 Image Registration

Keren et al. proposed a sub-pixel image registration method in the late 1980s [48]that formulated the image registration problem in matrix form in the spatial domain. Assuming the relative rotational and translational shift between two input images is small, Taylor series expansion can be applied to compute the estimated image after the relative shift. The residual error can be computed by the difference between the second image and the estimated image. By taking the partial derivative of the residual error, one can find the minimal value of the error and recover the rotational and translational parameters; this gives the minimal residual error.

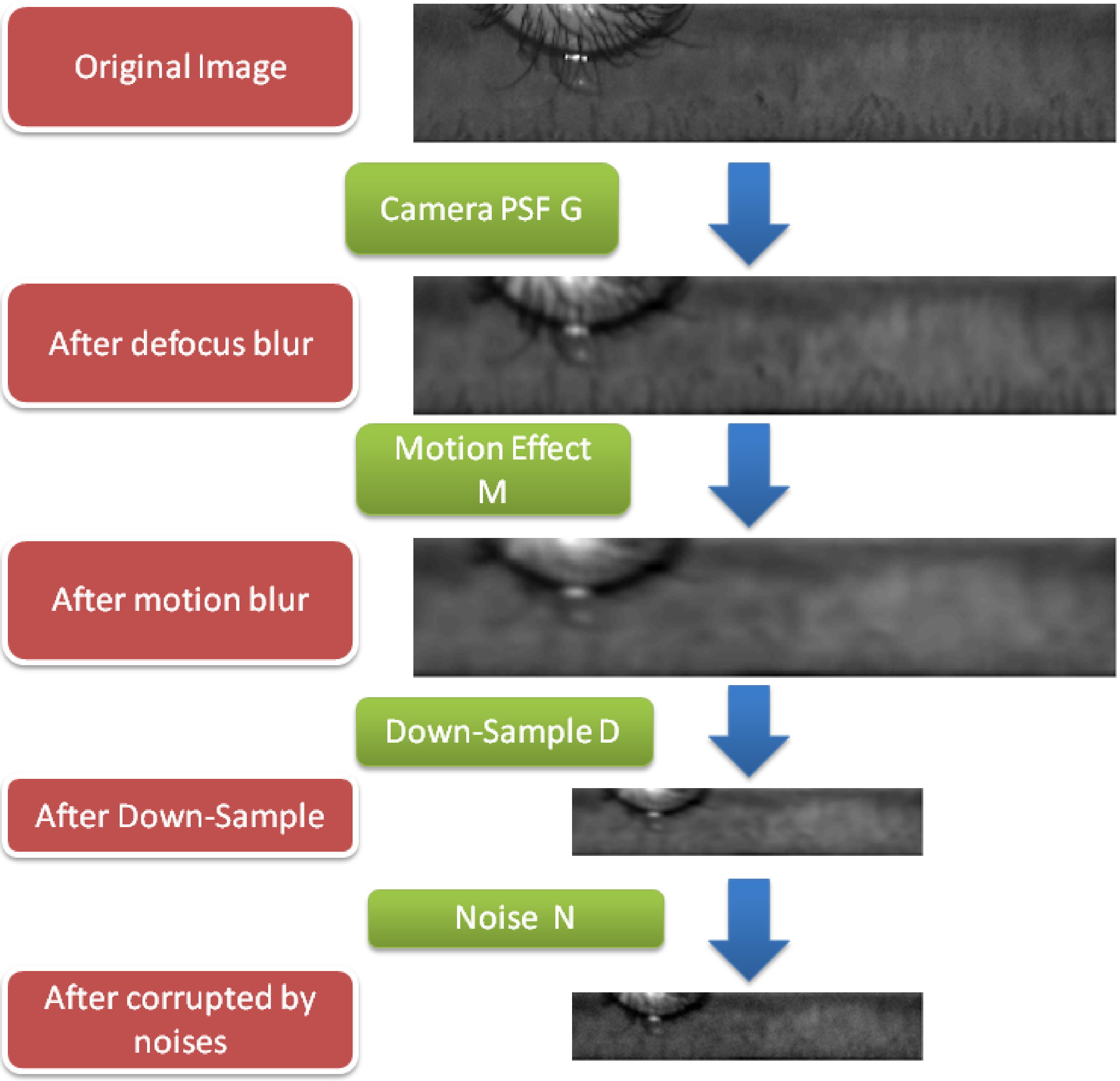


Figure 5.2: Image degradation modeling for Super-Resolution reconstruction.

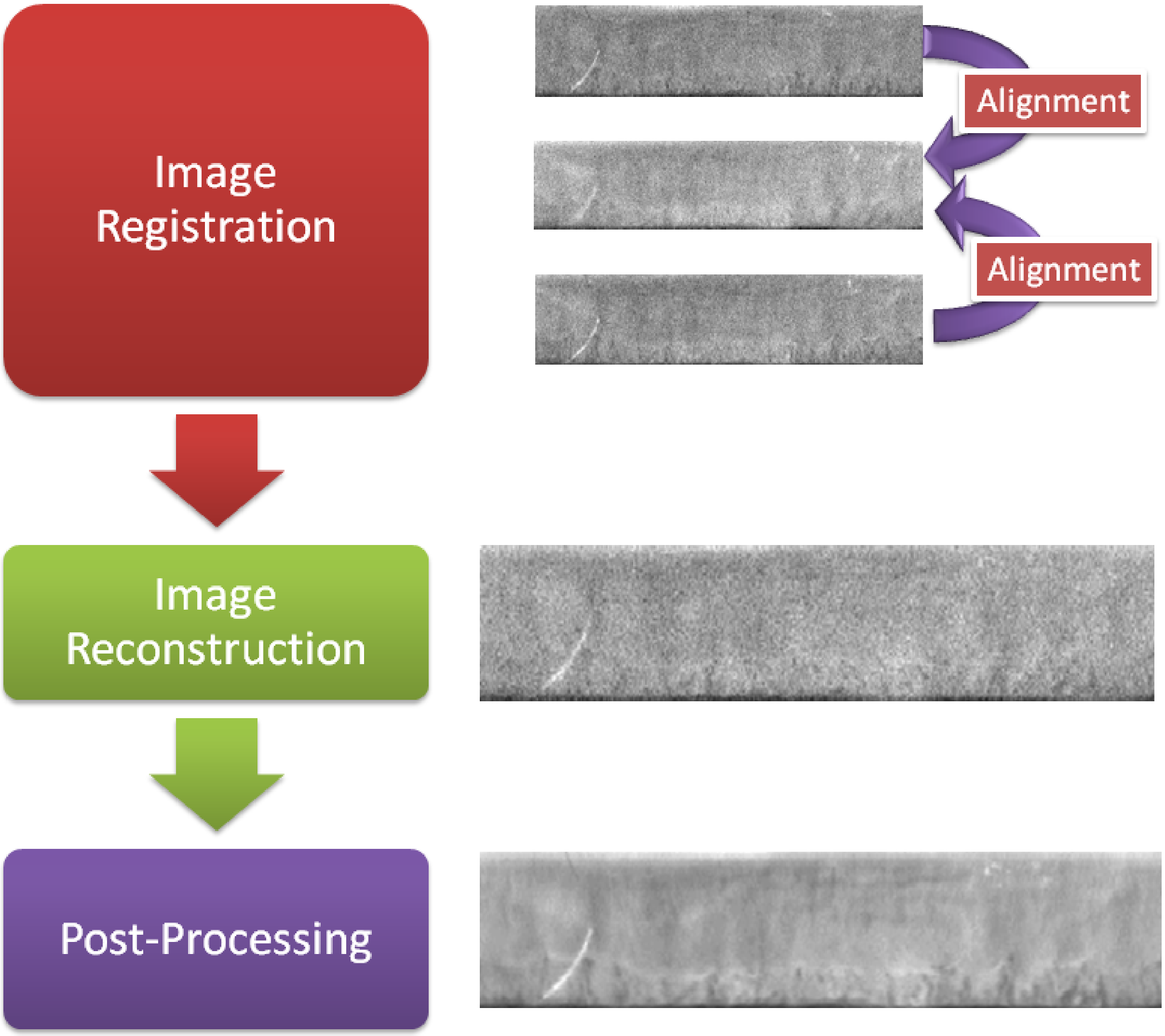


Figure 5.3: Flow chart of a typical image super-resolution algorithm.

The advantage of this approach is that nine out of the twelve parameters can be computed beforehand. This reduces the computational cost of the registration. In order to speed up the process further, the authors suggest using a Gaussian pyramid in the registration process [53]. A Gaussian pyramid is a coarse-to-fine image analysis tool. In this scheme, the original image of size $N \times N$ is convolved with a Gaussian filter and sub-sampled to give an $\frac{N}{2} \times \frac{N}{2}$ image. This process repeats until the image is reduced to one pixel. Keren et al. suggest a recursive process to compute the motion parameters for an image. Their process starts by computing the parameters for a very small image and using those results to interpolate the a slightly larger image, which are corrected and used to initialize an even larger image. This process is performed until the desired image size is reached. This algorithm works because, at smaller levels, big translations appear small. This process continues until the original image size is reached.

Reddy and Chatterji proposed another image registration method in frequency domain [54]. Their method is based on FFT analysis. Assuming that the relative rotational and translational shift between two input images is known, it is possible to write an equation to describe the relation between the original image and its translated version. After applying FFT to both sides of the equation, the relation between these two images can be described in the Fourier domain. The translational parameters can be recovered by computing the inverse Fourier transform of the cross-power spectrum, and the rotational parameters can be recovered by performing Cartesian to Polar coordinate transformation on the magnitudes of FFT spectrum and finding the most likely relative shift between them.

Vandewalle et al. proposed another frequency domain-based image registration method in 2006 [45]. Their method is similar to Reddy and Chatterji's method, except with a different parameter estimation scheme. In [54], Reddy and Chatterji propose using Cartesian to Polar coordinate transformation on the magnitudes of FFT spectrum in order to recover the rotational parameters. However, this transformation is time-consuming and error-prone. Instead, Vandewalle et al. proposed to compute a new function $h(\alpha)$ to describe the summation of the magnitude of the frequency spectrum for the given angle $\alpha$. When computing the function $h(\alpha)$, Vandewalle et al. suggested ignoring the low-frequency region in order to make this method more robust. After $h(\alpha)$ is computed, by applying 1D cross-correlation, we are able to find the most likely translational shift between $h_1(\alpha)$ and $h_2(\alpha)$, recovering the rotational parameters. To recover the translational parameter, the authors apply plane fitting through the phase difference using a least squares method that makes the solution robust to noise.

### 5.3.2 Image Reconstruction

After the relative translational and rotational shift between LR frames are measured, we have to reconstruct the HR image based on LR images. In this section we briefly review some of the most commonly used image reconstruction methods.

One of the most intuitive ways to fuse several LR images into one HR image is the nonuniform image interpolation method. The procedure of nonuniform image interpolation simply places the sample points from all input images back at their corresponding location on the template image and uses a function to approximate the values where there are no available sample points. The approximation function can be spline, bilinear, bicubic or any other higher order function. The only concern for the function we use is the trade-off between the speed and the accuracy.

Irani and Peleg proposed a Super-Resolution reconstruction method based on a process of iterative back projection [49]. First, the motion parameters of all the LR images have to be precisely estimated. They used a method which is very similar to the method in [48] to estimate motion parameters. After the parameters are estimated, reconstruction is completed by an iterative process. The assumption is that if the motion parameters are all accurate, and we know the estimated parameters of the image degradation process, then we will be able to project the degraded LR images from the HR image. Hence, the HR image is an initial estimation of the SR image. By minimizing the residual error between the real LR input images and the "back-projected" LR images, we are able to reconstruct the HR image so that it approximates the ground truth closely. The process is done in an iterative manner. In each iteration, the residual error becomes smaller and the SR image will gradually approximate to the real HR image.

In 2001, Zomet et al. proposed a method called "Robust Super-Resolution" for image reconstruction [46]. Their method is very similar to Irani and Peleg's method. However, Zomet et al. proposed a novel idea about how to iteratively update the estimated SR image. Instead of using the sum of the residual error as the derivative, they proposed to use the median value to replace the sum. For a symmetric distribution, a median can approximate the mean quite accurately, given a sufficient set of samples; in the case of distant outliers, the median is much more robust than the mean.

## 5.4 Proposed Method

All of the above SR reconstruction methods are global methods. Here, by using the term "global", it means that in every algorithmic step, the operations are performed on the holistic image. The image registration is performed in order to find the global relative shift between two LR images. The image reconstruction is performed to fuse the information globally from multiple LR images.

Instead of using a global SR algorithm, in this thesis, we propose to use local patch-based SR to perform image SR on iris images in the polar domain. The main difference between local patch-based SR and global SR algorithm is that we first break the whole image into subsets of rectangular areas, which are referred as "local patches" in the text. The image registration and image fusion are performed on each local patch set, not on the global image. The final global image is composed by combining each super-resolved local patch into one image.

By using local patches to perform Super-Resolution on the iris image in the polar domain, it is possible to avoid local deformation. As shown in chapter 3 and 4, the iris images in the polar domain are obtained by unwrapping the iris region from the Cartesian coordinate. This Cartesian to Polar coordinate transformation is extremely sensitive to the accuracy of both the iris and pupillary boundary recovered by iris segmentation algorithm. Even for the same iris image, different segmentation algorithms will recover different iris boundaries. Also, for the same iris image, there may exist multiple iris and pupillary boundaries that can be considered "correct" segmentations The parameter values of different "correct" segmentations may only be slightly different from each other, but may result in an observable local deformation after unwrapping the iris from Cartesian to polar coordinates.

The local deformation phenomenon of iris images in the polar domain is widely known in the iris recognition community. In [12, 55, 56], the authors propose a local patch-based matching algorithm which reports an extremely high recognition rate compared to Daugman's algorithm. If the local patch-based recognition algorithm can outperform global image-based recognition algorithms, it is very reasonable to apply local patch-based philosophy to the SR problem for iris images.

### 5.4.1 Key Components of the Proposed Algorithm

The key components of the proposed Local Path-based Super-Resolution algorithm (LPSR) can be summarized as the following:

1. Quality assessment
2. Global alignment
3. Up-sampling and patch binning
4. Local alignment
5. Local patch reconstruction
6. Patch stitching
7. Post-processing

We explain the details of each stage in the following paragraphs.

### 5.4.2 Iris Image Quality Assessment

The first step in the LPSR algorithm is to assess the image quality among the input LR images and find out which image is of the best quality image. After the best quality image is located, we can use it as the template image, and perform all other operations to fuse information from other images into this template image.

We propose two different kinds of image quality assessment algorithms. They are based on different philosophies, but both are reasonable. We illustrate each of them in the following paragraphs.

The first method we tried is to use 2D-FFT spectrum to assess image quality. FFT spectrum analysis is a widely used method to analyze the frequency component in a given signal. Since an image can be seen as a two-dimensional discrete signal, we can use 2D-FFT to represent the frequency component of an image. Images that contain sharp edges and details should exhibit more high frequency content in the 2D-FFT spectrum. If an image is blurred and sharp edge information is lost, there will be less high-frequency content. Therefore, the basic idea is that, by comparing the magnitude of the ratio between high frequency and low frequency components, we are able to assess how good the quality of the image is. The higher the ratio, the better the quality of the image.

The flow chart of quality assessment of the 2D-FFT method is presented in Figure 5.4, with the flow chart shown on the left side and an example image for each stage shown on the right side. First, given an input image (a), the 2D-FFT spectrum is computed, shown in (b). Note

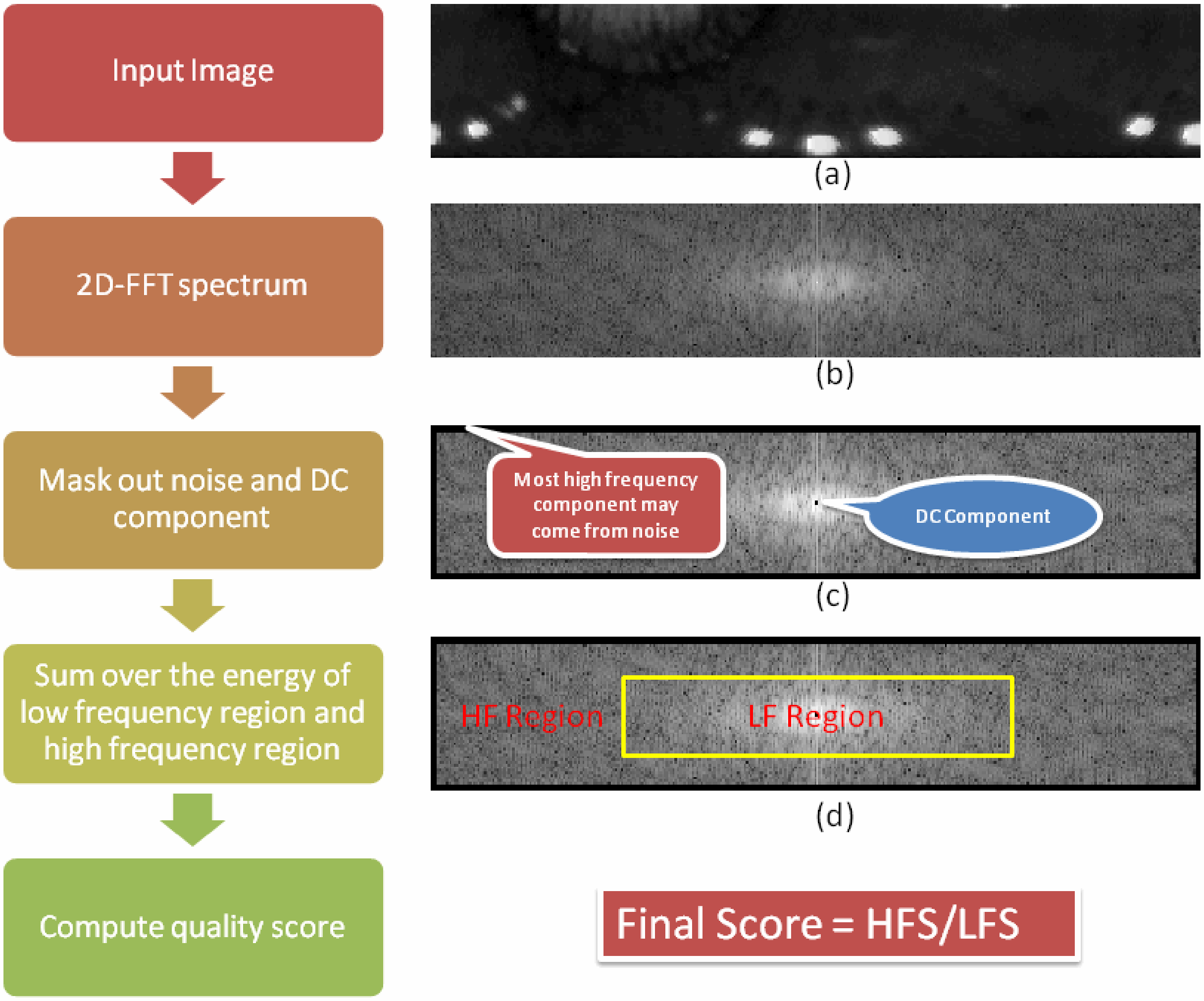

Input Image
2D-FFT spectrum
Mask out noise and DC component
Sum over the energy of low frequency region and high frequency region
Compute quality score
(a)
(b)
Most high frequency component may come from noise
DC Component
(c)
HF Region
LF Region
(d)
Final Score = HFS/LFS

Figure 5.4: Flow chart of quality assessment with 2D-FFT

that in this 2D-FFT plot, origin is located at the center of the plot. The X-axis and Y-axis represent the frequency scale in the x and y direction. The intensity denotes the magnitude of the frequency component in its own frequency band. In Figure 5.4(b), we can see that the center region is brighter than the neighboring regions, which shows that there are more low frequency components than high frequency ones.

After the 2D-FFT spectrum is computed, two regions should be masked out. The first is the pixels located at the center of the spectrum. These pixels denote the magnitude of the DC component in the image, which is the average intensity over the whole image. Since the goal of this method is to assess the ratio between the low frequency and high frequency component, it should not be influenced by the average intensity of the image. Therefore, we should fill this region with zeros so we do not consider the DC component.

Second, since the noise in images is represented as high frequency components in the 2D-FFT spectrum, the region of very high frequency is also not trustworthy and should be masked out. By masking out the very high frequency components, the performance of this method should be more stable and robust to noise-contaminated images.

After masking out the DC component and the very high frequency components in the 2D-FFT plot, the spectrum is divided into two regions: the Low-Frequency (LF) region and High-Frequency (HF) region. Since the origin of the 2D-FFT plot is at the center, this region is the LF region and the outer region is the HF region. We accumulate the energy over the LF region and the HF region separately, getting a score for the LF region (LFS) and the HF region (HFS) separately.

Finally the quality score is computed by dividing HFS with LFS.

The second method we proposed for quality assessment is based on mutual Hamming Distance (HD) score. The HD score computation is a standard method used for measuring the distance between two iris images within a normalized range between zero and one. The lower the HD, the greater the similarity between two iris images. Usually, if two iris images are from the same class, and both of them are very sharp and clear, the HD would be low. However, if the images are blurred or noisy, even if they are from the same class, the HD would be high.

The use of HD in iris matching reveals the possibility of using it in quality assessment. In this thesis, we propose an iris quality assessment and alignment method based on the HD, which has never been seen in literature before. Given N input iris images, we can compute the HD between every possible pair of images, and organize all the HD in the form of a similarity matrix. A similarity matrix is a matrix $S$ of size $N \times N$ where each element $S(i, j)$ denotes the HD between image $i$ and $j$. Therefore, $S$ is a symmetric matrix and $S = S^T$.

After the Similarity matrix is computed, we can compute the average of each row and get a column vector $C$. In this column vector, each element $C_i$ represents the average HD between the image $i$ and all other images. If an iris image is clear and of good quality, it is supposed to have a lower HD when being matched to all other images. On the contrary, if an iris image is blurred and noisy, the HD would be high. Therefore, by locating the smallest value in $C$, we would be able to locate the iris image of the best quality, and use this image as our template for all subsequent stages. The process of quality assessment by mutual HD is illustrated in Figure 5.5.

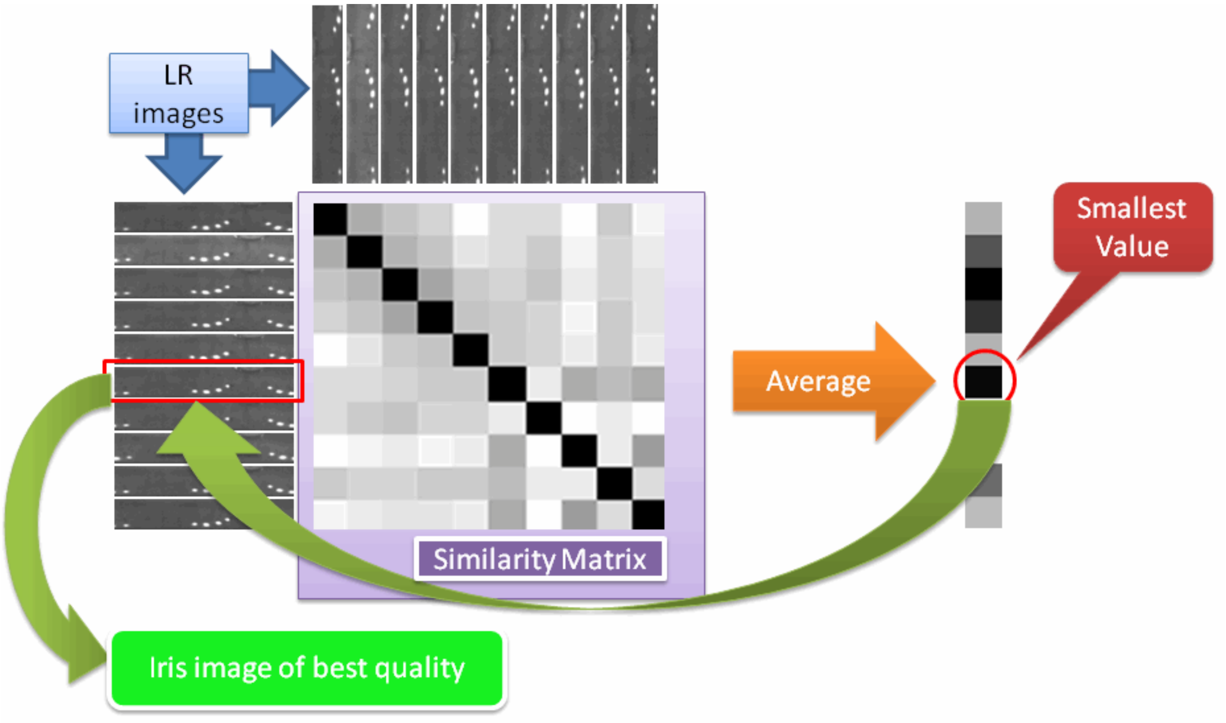


Figure 5.5: Illustration of image quality assessment by comparing mutual HD score.

### 5.4.3 Global Alignment

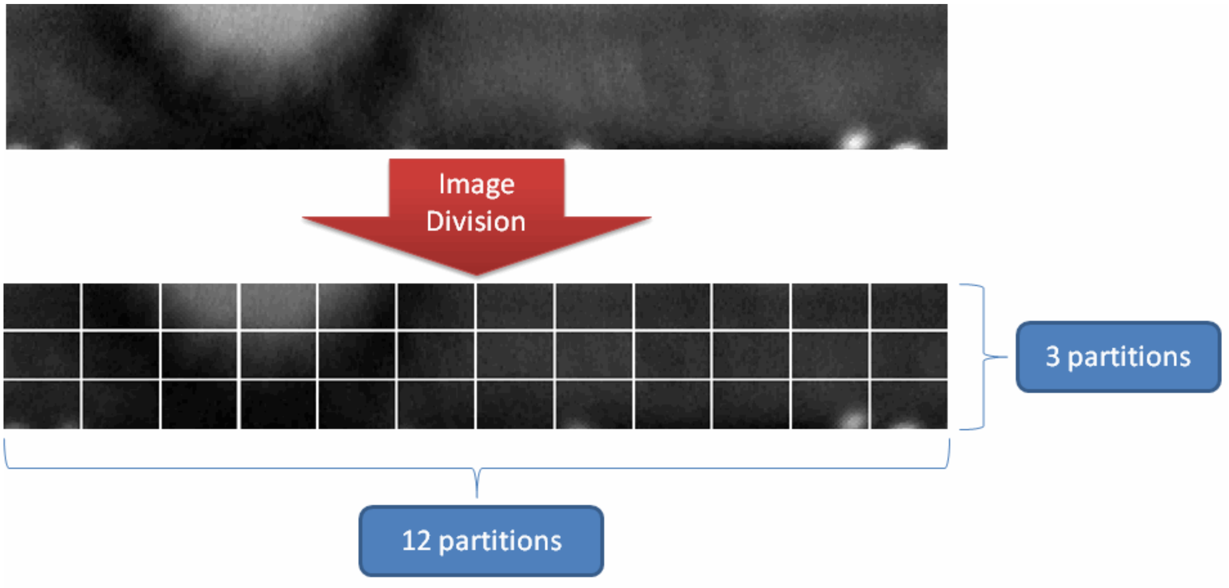
After we have located the image of best quality, but before we start the SR process using local patches, it is a good idea to align the input LR images globally. There are chances that the input LR images may have relatively large translational shifts between them. By aligning LR images globally, we can minimize the possibility of erroneous alignment in the local patch domain, thereby enhancing the success rate of local alignment.

The reference image for global alignment is the template image, which is the image of best quality. In section 5.4.2, at the same time when computing HD between each pair of LR images, we can also keep track of the relative translational shift that gives the best HD. This optimal relative translational shift information tells us by how many pixels we should shift one image in order for it to align to another image. In the stage of global alignment, we shift each input LR image with exactly the optimal translational shift with respect to the template iris image, but in the opposite direction. In this way, all input LR images would be aligned to the template image.

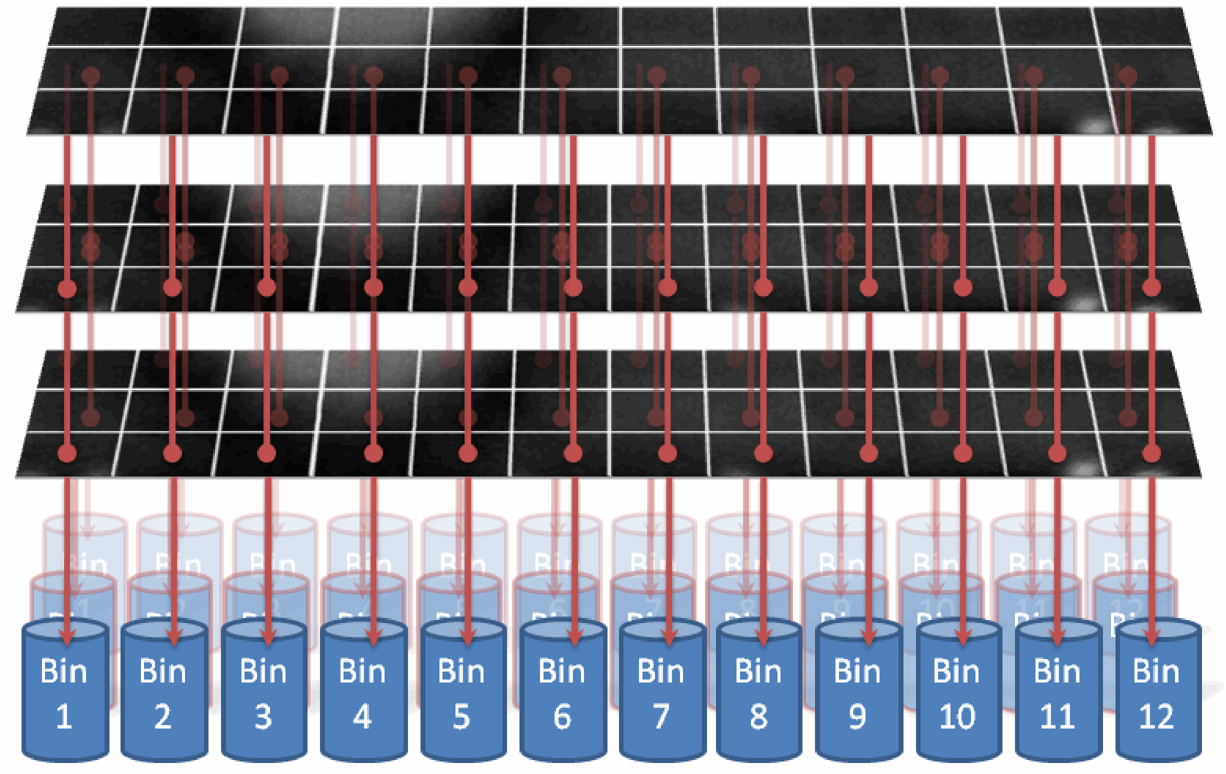
### 5.4.4 Iris Patch Configuration

After input LR images are globally aligned, they are up-sampled to double the original size, both horizontally and vertically. The LPSR algorithm aligns and fuses local patches in the up-sampled image, and then reduce the super-resolved image back to its original size. In the following experiments, the size of LR images is $360 \times 60$, and the size after up-sampling is $720 \times 120$.

After the input LR images are up-sampled, they are divided into local patch sets. In the



(a)



(b)

Figure 5.6: Illustration of image division and local patches reorganization. (a) Every input LR image is divided into 36 local patches. (b) Patches that come from different mother image but belong to the same location would be stored in the same "location bin".

following experiments, we divide the image into 3 partitions in the y direction and 12 partitions in the x direction. Therefore, the total number of patches for one image is $3 \times 12 = 36$. Since the size of the up-sampled image is $720 \times 120$, the size of each patch is $60 \times 40$. An example of the image division process is illustrated in Figure 5.6(a).

### 5.4.5 Local Alignment

After the input LR images are divided, the local patches are re-ordered based on the patch location. In section 5.4.4, it is mentioned that there are 36 patches in all per LR image, which means there are 36 locations. The input image patches are reorganized and grouped according to their corresponding locations, as shown in Figure 5.6(b).

For local patches that belongs to the same bin, we would like to align them locally. As stated before, an iris image in the polar domain suffers from the effect of local deformation. Therefore, the goal of local alignment is to recover the relative translational parameters between every pair of patches. Once this parameter is recovered, we are able to fuse the information from multiple patches during the local patch reconstruction stage and at the correct location where they should be fused.

The algorithm we used in this stage to align local patches is Normalized Cross Correlation (NCC). NCC is an important technique in image processing and has been widely used in pattern recognition, object and target tracking, robotics, and biomedical image processing [57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67]. Traditionally, cross correlation between two signals can be computed in the frequency domain, but this is not true for NCC. In this thesis, the NCC algorithm implementation follows the technique proposed by Lewis in 1995 [68, 69]. We briefly review and describe this technique in the following paragraphs.

For two images (target image and scene image) $t(x, y)$ and $f(x, y)$, the Euclidean distance between them when one of them is shifted $(u, v)$ in the spatial domain is

$$d^2_{f,t}(u, v) = \sum_{x,y} [f(x, y) - t(x - u, y - v)]^2 \tag{5.2}$$

We can expand $d^2_{f,t}(u, v)$ so that

$$d^2_{f,t}(u, v) = \sum_{x,y} \left[f^2(x, y) - 2f(x, y)t(x - u, y - v) + t^2(x - u, y - v)\right] \tag{5.3}$$

If $t(x, y)$ represents the target image, then $\sum_{x,y} t^2(x - u, y - v)$ is constant. Also, if the term $\sum f^2(x, y)$ is approximately constant then the remaining cross-correlation term

$$c(u, v) = \sum_{x,y} f(x, y)t(x - u, y - v) \tag{5.4}$$

is a measure of the similarity between the target image and the scene.

If we used Eq. 5.4 to compute similarity, there are several disadvantages. First, if the image energy $\sum f^2(x, y)$ varies with position, the approximation would fail. Second, the range of $c(u, v)$ is dependent on the size of the feature. Third, Eq. 5.4 is not invariant to changes in image intensity. Therefore, if there is illumination variation between the target and the scene image, the similarity score computed by Eq. 5.4 would be different.

Instead, if we compute the correlation coefficient by normalizing the target and scene feature vectors to unit length, it would become a cosine-like correlation coefficient

$$\gamma(u,v) = \frac{\sum_{x,y}\left[f(x,y)-\bar{f}_{u,v}\right]\left[t(x-u,y-v)-\bar{t}\right]}{\sqrt{\sum_{x,y}\left[f(x,y)-\bar{f}_{u,v}\right]^2\sum_{x,y}\left[t(x-u,y-v)-\bar{t}\right]^2}} \tag{5.5}$$

where $\bar{t}$ is the mean of the feature and $\bar{f_{u,v}}$ is the mean of $f(x,y)$ in the region under the feature. Eq. 5.5 is the definition of NCC.

Now, consider the numerator in Eq. 5.5, and assume that we have pre-computed the image $f'(x,y) \triangleq f(x,y)-\bar{f}_{u,v}$ and $t'(x,y) \triangleq t(x,y)-\bar{t}$ in which the mean value has already been removed:

$$N(u,v) = \sum_{x,y} f'(x,y)t'(x-u,y-v) \tag{5.6}$$

We can use FFT to compute Eq. 5.6 efficiently because it is basically convolving the image $f'$ with the reversed target $t'(-x,-y)$. The FFT computation of $N(u,v)$ is

$$N(u,v) = F^{-1}\{F(f')F^*(t')\} \tag{5.7}$$

where $F(\cdot)$ is the Fourier transform. The complex conjugate operator would reverse the feature via Fourier transform property $F(f^*(-x)) = F^*(\omega)$.

Now, let us consider the denominator of Eq. 5.5. The length of the feature vector can be precomputed, and the feature itself can be pre-normalized into length one. But the problem lies on the term $\sum_{x,y}[f(x,y)-\bar{f}_{u,v}]^2$. The image mean and local energy must be computed at each $(u,v)$, resulting in a higher number of computations.

A more efficient method to compute this term is by using integral (the running sum) of the image and image square over the search area, i.e.,

$$\begin{aligned} s(u,v) &= f(u,v)+s(u-1,v)+s(u,v-1)-s(u-1,v-1) \\ s^2(u,v) &= f^2(u,v)+s^2(u-1,v)+s^2(u,v-1)-s^2(u-1,v-1) \end{aligned} \tag{5.8}$$

with $s(u,v) = s^2(u,v) = 0$ when either $u,v < 0$. The energy of the image under the feature positioned at $(u,v)$ is

$$\begin{aligned} e_f(u,v) &= s^2(u+N-1,v+N-1)-s^2(u-1,v+N-1) \\ &\quad -s^2(u+N-1,v-1)+s^2(u-1,v-1) \end{aligned} \tag{5.9}$$

and similarly for the image sum under the feature.

The problematic term $\sum_{x,y}[f(x,y)-\bar{f}_{u,v}]^2$ can now be computed with very few operations since it expands into an expression involving only the image sum and sum squared under the feature. The construction of the table requires approximately $3MN$, where $(M,N)$ is the size of $f$.

### 5.4.6 Local Patch Reconstruction

After local patches are aligned with each other, the next step is to fuse the information from multiple patches and generate one single HR patch. In this stage, there are two key factors we have to consider. The first factor deals with the components which can be fused, and the benefit in using them to reconstruct the HR local patch. The second important factor is how to fuse pixel information from multiple patches. In this section, more details about these two key factors are discussed.

#### 5.4.6.1 Components for Local Patch Reconstruction

The goal of local patch reconstruction is to fuse pixel intensity information from multiple local patches and generate one final patch which is supposed to be clear, clean, and sharp. In order to achieve this goal, we have to consider how to sharpen the final image so that the edge information is more evident in the final patch when compared to the original patches.

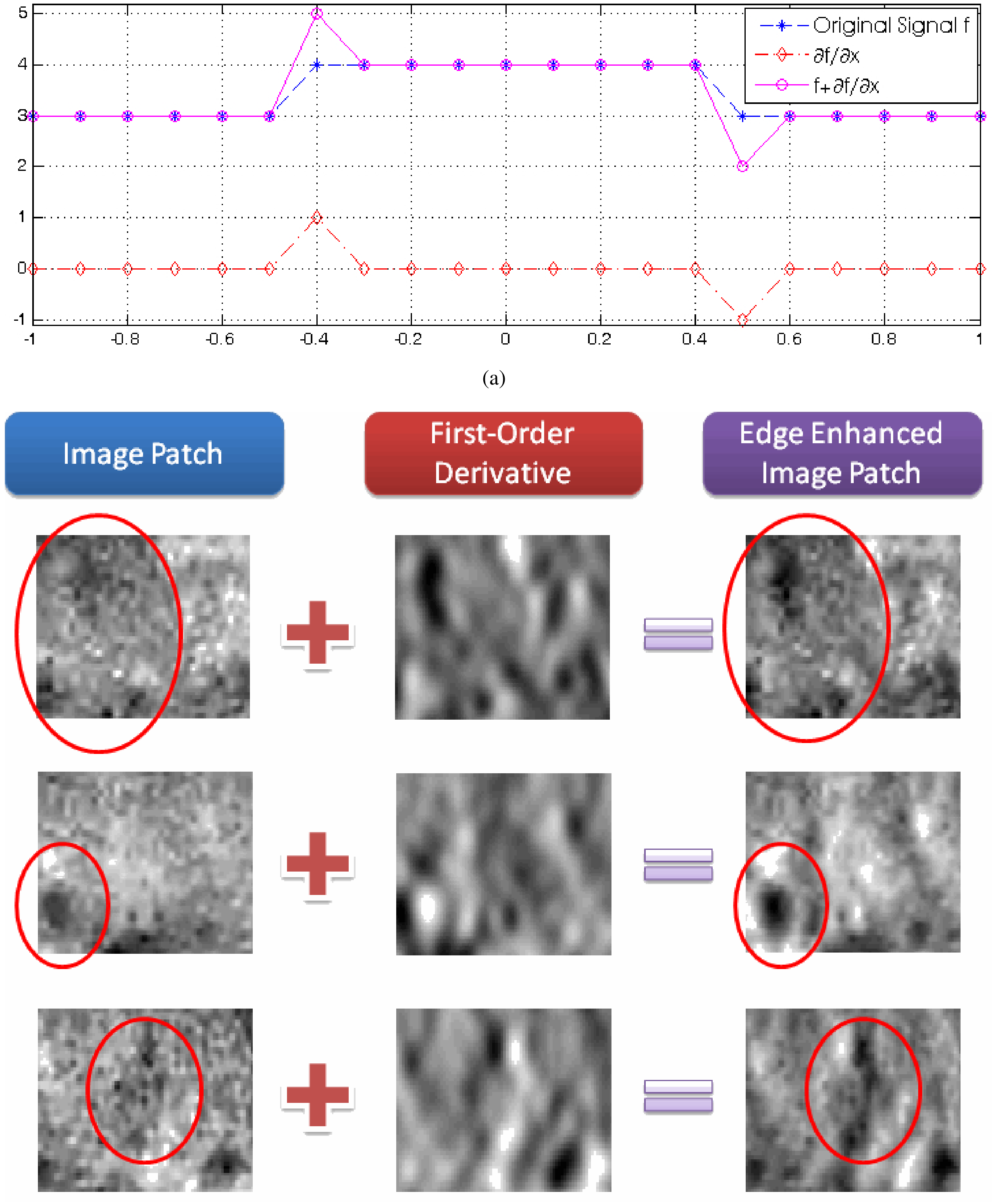
We know that the edge in an image is the location where there is discontinuity in image pixel intensity. If we treat the image as a signal, denoted as $f(x)$, the locations where the discontinuities happen can be found by taking the first-order derivative of the signal $\frac{\partial f}{\partial x}$, and locating the positions of non-zeros ($\frac{\partial f}{\partial x} \neq 0$). This is because $\frac{\partial f}{\partial x}$ returns the difference of the adjacent points in $f(x)$, at the places where $\frac{\partial f}{\partial x} = 0$, $f(x)$ does not change at all.

It is an interesting idea to consider what we would get if we perform linear combination between $f(x)$ and its derivative $\frac{\partial f}{\partial x}$. Let $g_1(x)$ denotes $f(x) + \frac{\partial f}{\partial x}$. At the places where $\frac{\partial f}{\partial x} = 0$, $g_1(x) = f(x)$, which means at the places where there is a constant value in original image $f(x)$, the new image $g_1(x)$ has the same value as $f(x)$. At the places where edges occur ($\frac{\partial f}{\partial x} \neq 0$), if the signal $f(x)$ is increasing, $\frac{\partial f}{\partial x} > 0$, $g_1(x)$ would have a higher value then $f(x)$. On the contrary, at the places where $f(x)$ is decreasing, $\frac{\partial f}{\partial x} < 0$, $g_1(x)$ would have a lower value then $f(x)$. In this way, $g_1$ is an edge-enhanced version of $f$, because both increasing and decreasing edges are enhanced.

This idea can be illustrated by the examples shown in Figure 5.7. Figure 5.7(a) shows an example of the 1D signal. Suppose the original signal $f(x)$ has an edge at $x = -0.4$ and $x = 0.5$. Red dash-dot line shows $\frac{\partial f}{\partial x}$, which is zero everywhere, except at the two positions where edges occur. If we sum up $f(x)$ and $\frac{\partial f}{\partial x}$, the pink curve is what we get. Comparing it with $f(x)$, we can see the new curve has a stronger pixel difference at the places where edges occur.

Figure 5.7(b) shows examples in 2D signals (images). We can see that by adding the first-order derivative of the local patch to itself, it generates images that have stronger edges. Compare the regions circled in red ellipses.

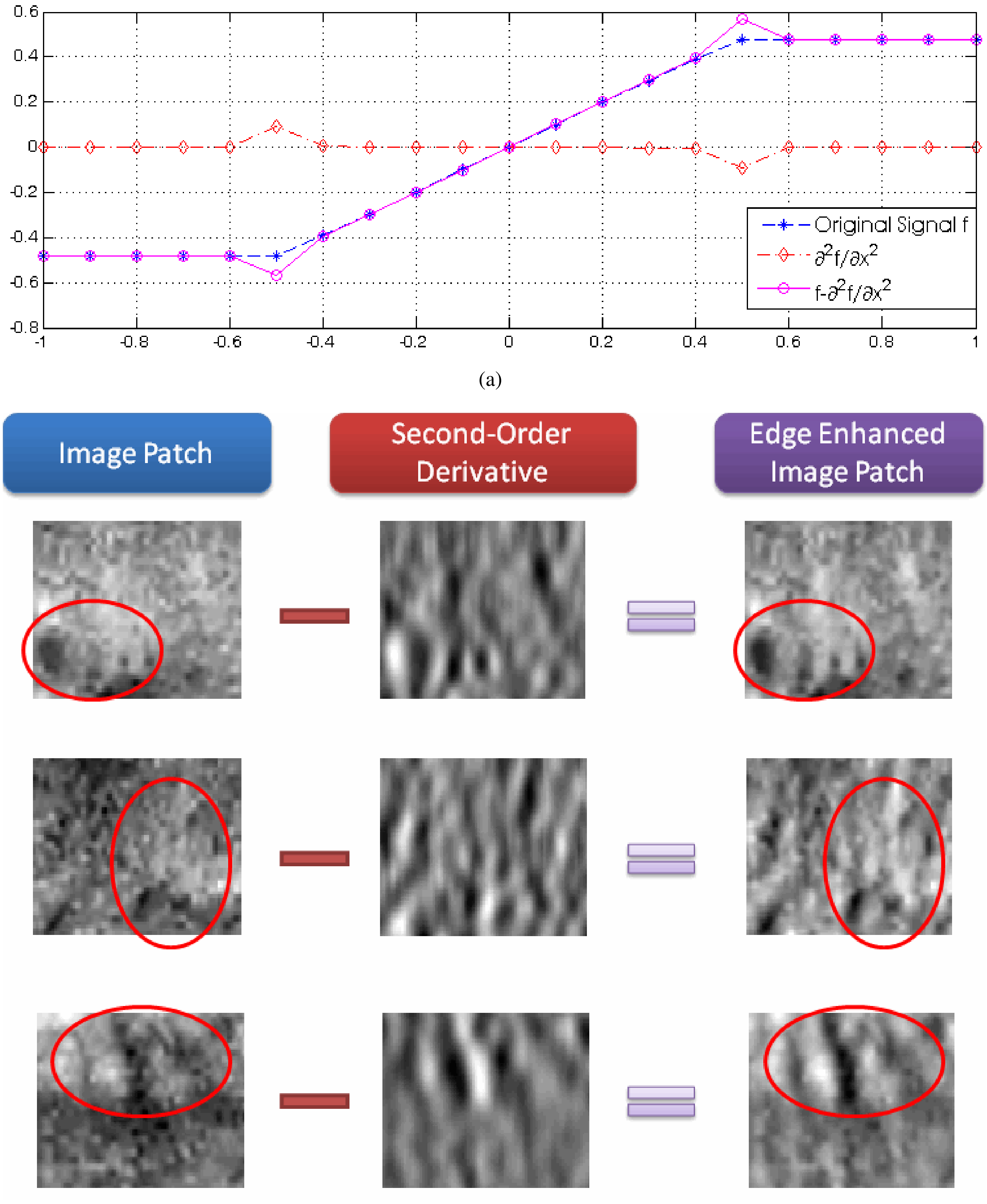
Alternatively, we can take the second-order derivative of the original signal (denoted as $\frac{\partial^2 f}{\partial x^2}$) and add it back to the original image. Figure 5.8 shows some examples. Suppose original signal $f(x)$ has a gradually increasing edge between $x = -0.5$ and $x = 0.5$. This edge is not so strong compared to $f(x)$ in Figure 5.7(a). However, if we took the second-order derivative of $f(x)$, the red curve is what we get. It clearly shows the two places where edges occur. If we compute a new signal $g_2(x) = f(x) - \frac{\partial^2 f}{\partial x^2}$, $g_2(x)$ would be an edge-enhanced version of $f(x)$, as shown in the pink curve in Figure 5.8(a).



(a)

(b)

Figure 5.7: Effect of edge enhancement by combining original signal with its first-order derivative. (a) 1D example (b) 2D example



(a)

(b)

Figure 5.8: Effect of edge enhancement by combining original signal with its second-order derivative. (a) 1D example (b) 2D example

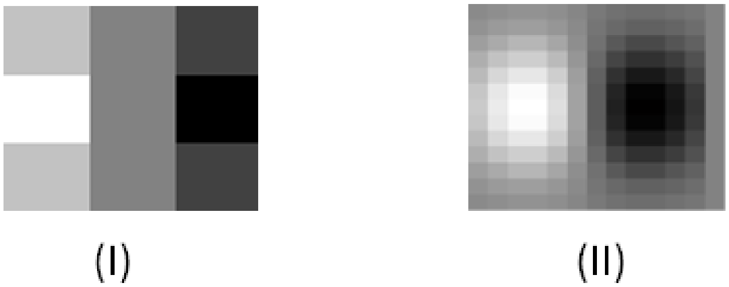


Figure 5.9: Two different types of approximations to image gradient used in experiments. (I) Sobel filter in x direction (II) Derivative of Gaussian

Figure 5.8(b) shows examples in 2D signal (images). We can see that, by adding second-order derivative of the local patch to itself, it generates images that have stronger edges. Compare the regions circled in red ellipses.

In short, the components that we can add into the final reconstructed image in order to further enhance the edge include the following:

1. Local patches of image itself.
2. Local patches of first-order derivative of the image patches.
3. Local patches of second-order derivative of the image patches.

Practically speaking, when we want to compute the first-order derivative of the image, usually we use image filter to achieve this goal. One commonly used filter is the Sobel filter [14]. Another choice is to use derivative of Gaussians. For the experiments described later in section 5.5, we tried two different kinds of derivative approximations, as shown in Figure 5.9.

#### 5.4.6.2 Local Patch Weighting Scheme

The next important issue is how to fuse the pixel information from multiple patches. The most intuitive way of doing this is to compute the average value (mean) of the pixel values from multiple patches. This scheme can be called “uniform weighting”. The probability density function of the uniform weighting scheme is shown in Figure 5.10(a).

Alternatively, since the quality of the input LR images vary, an improved way of doing this is to consider the quality index and use it to weigh the pixel value of different patches.

In this thesis, a method called Exponential Decay Weighting by Image Quality (EDWIQ) is proposed. The basic idea is to weigh the pixel value of different patches differently, according to their quality. The quality of the input LR images can be determined by the method mentioned in section 5.4.2. After the rank of the image quality has been determined, it will be mapped to an exponential decaying function, from which the weighting coefficients can be derived.

Specifically speaking, suppose there are $N$ input LR images. After the image quality has been determined, we re-arrange the order of the input image, so the first image is the best quality image and the last image is the worst. Then in the EDWIQ scheme, we make the weighting factor $w_i$ fit an exponential decaying function. Suppose the target exponential decaying function is $e^x$ with x-axis reversed and $x \in [1, 8]$, as shown in Figure 5.10(c). Suppose we make the weighting

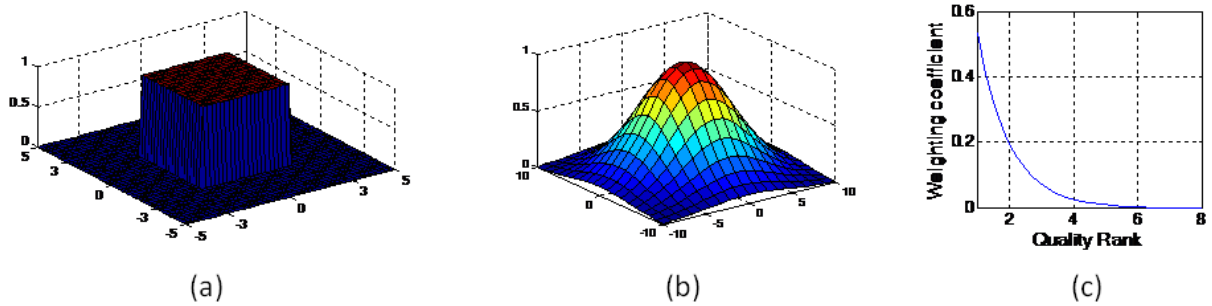


Figure 5.10: Proposed patch weighting function. (a) uniform weighting (b) Gaussian Translational Belief Model (GTBM) (c) Exponential Decay Weighting by Image Quality (EDWIQ)

of rank 1 image as $e^8$ and the weighting of rank $N$ image as $e^1$. Then in general, the weighting factor $w_i$ for image at rank $i$ is equal to

$$w_i = \frac{e^{1+\frac{7(N-i)}{N-1}}}{\sum_{i=1}^{N} e^{1+\frac{7(N-i)}{N-1}}} \tag{5.10}$$

The denominator in Eq. 5.10 is a normalization factor so that $\sum_{i=1}^{N} w_i = 1$.

Generally speaking, $x \in [M_1, M_2]$, as long as $M_2 > M_1$ and $|M_2 > M_1|$ are reasonably large. The reason for choosing $x \in [1, 8]$ is a decision we arrive at after parameter tuning. Also, in the database collected for the experiment, the average number of IOM images in one session is eight, which justifies the decision.

Another possible weighting scheme is to weigh by the possibility of local distortion of the patches. As mentioned in section 5.4, iris images in the polar domain suffer from local distortion. However, since all of the input images have been aligned globally, in the most likely case, the local distortion between local patches should be small. If the local distortion is too big, there might be something wrong in the preprocessing stage (wrong iris detection or segmentation); therefore, we should weigh this patch much lower than the others.

We propose a Gaussian Translational Belief Model (GTBM) which models the probability of the local patch as a 2D Gaussian distribution based on 2D relative shift. Assume the relative shift between two patches, discovered by using the method described in section 5.4.5, is $(\triangle x, \triangle y)$. Then the probability of this patch is modeled as

$$P_{\triangle x, \triangle y} = \frac{1}{C} e^{-\frac{\triangle x^2}{2\sigma_x^2}} e^{-\frac{\triangle y^2}{2\sigma_y^2}} \tag{5.11}$$

where $C$ is a normalization factor, and $\sigma_x$ and $\sigma_y$ are parameters that control the width of the 2D Gaussian in the x and y direction. Because we assume the translational shifts in x and y direction are independent to each other, they are modeled separately and combined by the product rule. When $\sigma$ is large, the Gaussian envelope appears wider, and it means that we tolerate more local distortion, and vice versa.

An example of the 3D plot of probability density function of GTBM is shown in Figure 5.10(b).

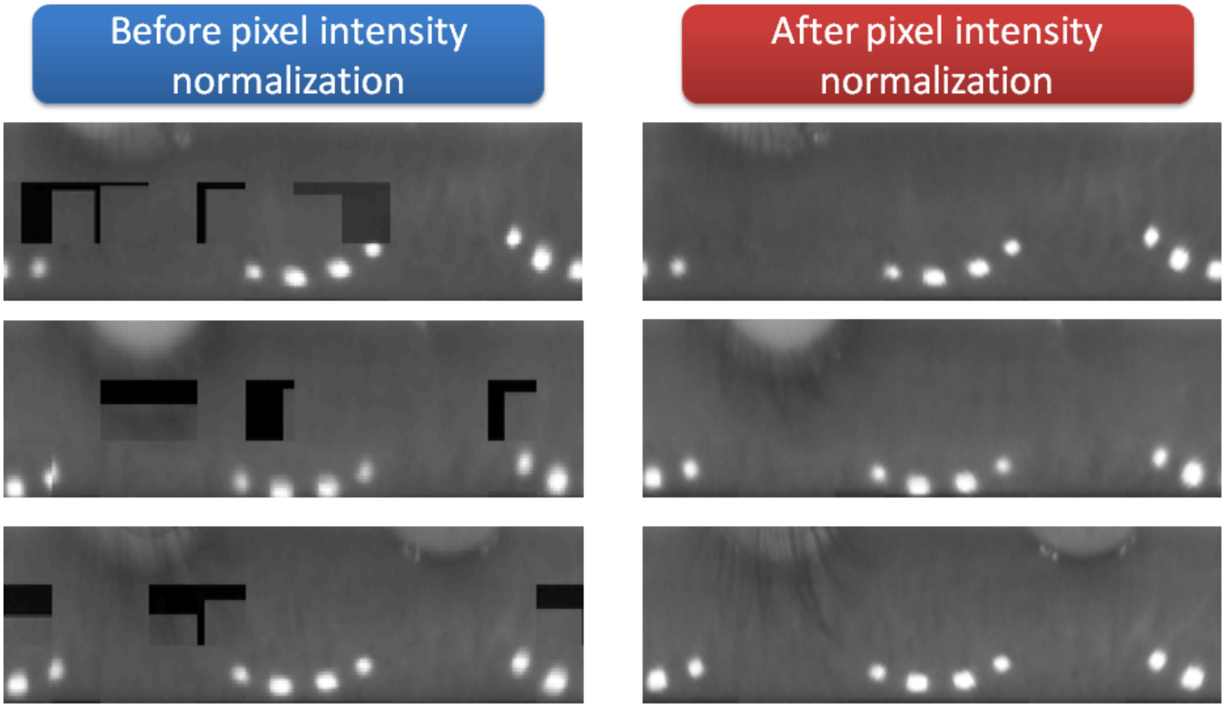

Before pixel intensity normalization
After pixel intensity normalization

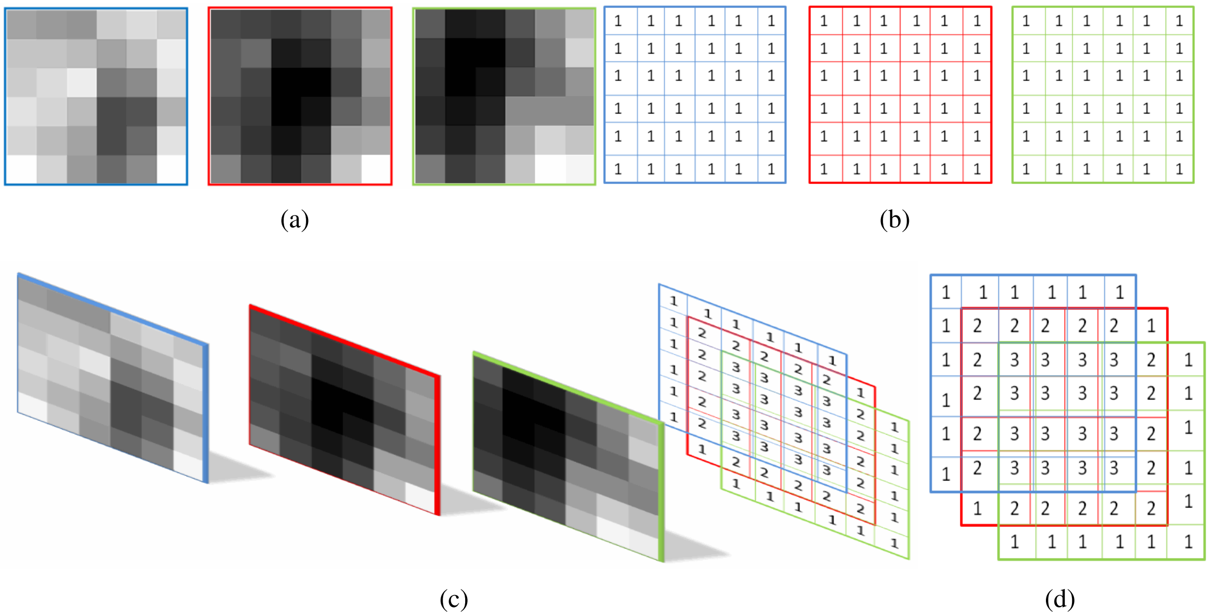
Figure 5.11: Effects of pixel intensity normalization in the stage of patch stitching.

### 5.4.7 Local Iris Patch Stitching

In the stage of patch stitching, the goal is to combine the local reconstructed patches back into one global image. One severe problem we face when doing Super-Resolution in local patch-based method, is the mismatch of image energy across the patch boundaries. This is because every input LR image may have different global intensities. When the input LR images are divided into patches, the average intensity between each patch may vary more. The problem becomes even worse when there are local distortions between images in the same local patch set. If we shift the distorted local patch and add it back into the template patch, the boundary between them will look very obvious in the final image.

A few examples are shown in Figure 5.11. On the left are those SR images produced without considering the problem of pixel intensity normalization. There are many block-like boundaries in the images, and the average intensity of problematic patches is quite different from adjacent patches, resulting in a very bad final stitched image. On the contrary, if we take steps to perform pixel intensity normalization, the problem of blocky effects will be solved. The final fused images look very natural and the average intensity of each patch matched with the neighboring patches very well.

In this thesis, we normalize the pixel intensity by constructing a pixel-wise accumulator. This process can be illustrated by Figure 5.12. Suppose we have three input local patches, as shown in Figure 5.12(a). Assume the reference template is the patch in the middle, with the red border. For each input local patch, a pixel accumulation plane is created, as shown in Figure 5.12(b). A pixel accumulation plane is a 2D matrix, where the values at a given position keep track of



(a)

(b)

(c)

(d)

Figure 5.12: Illustration of pixel intensity normalization by pixel counting. (a) Input local patches. Assume the one with red border is the template. (b) Initialization of pixel accumulation plane (c) Aligning local patches (blue and green ones) with respect to the template patch, and pixel accumulation planes are aligned as well. (d) Final grid values for pixel accumulation plane.

how many pixels have been added into this position. Therefore, every location in this plane is a "counter" for that location. The initial size of the pixel accumulation plane is exactly the same as the size of the input patches, but the size will expand later after the alignment is complete. We find the relative positions between all three patches by performing local alignment. Then, we align the three pixel accumulation plane accordingly, as shown in Figure 5.12(c). The final pixel accumulation plane consists of the area which is the union of the three, and the value at each location is the sum of all value that intersect with it, as shown in Figure 5.12(d).

After we compute the final pixel accumulation plane, the last step is to divide the fused patch with the pixel counter element-wise. Assume the fused patch is $f(x, y)$, and the final pixel accumulation plane is $a(x, y)$, the final fused patch is derived by

$$p(x, y) = \frac{f(x, y)}{a(x, y)}, \ \forall x, y \tag{5.12}$$

In this way, pixel values of the final fused patch will be normalized and the block effects would be eliminated.

Note that this method can also be applied to the case when we use GTBM and EDWIQ to give a fractional weighting to local patches. In that case, the value accumulated in each pixel counter is not an integer. Instead, it is a real number which keeps track of how much "weighting" has been added to this location. Therefore, the method of using the pixel accumulation plane can apply to all three patch weighting schemes mentioned in section 5.4.6.2.

### 5.4.8 Post-Processing

After the local patches have been correctly stitched together, we can apply some post-processing techniques to further remove noises in the final image. The post-processing technique we have tried in the experiments include the following: (1) Gaussian Low-Pass Filtering (GLPF) (2) Gamma correction (3) Median filtering (4) Histogram specification

#### 5.4.8.1 Gaussian Low-Pass Filter

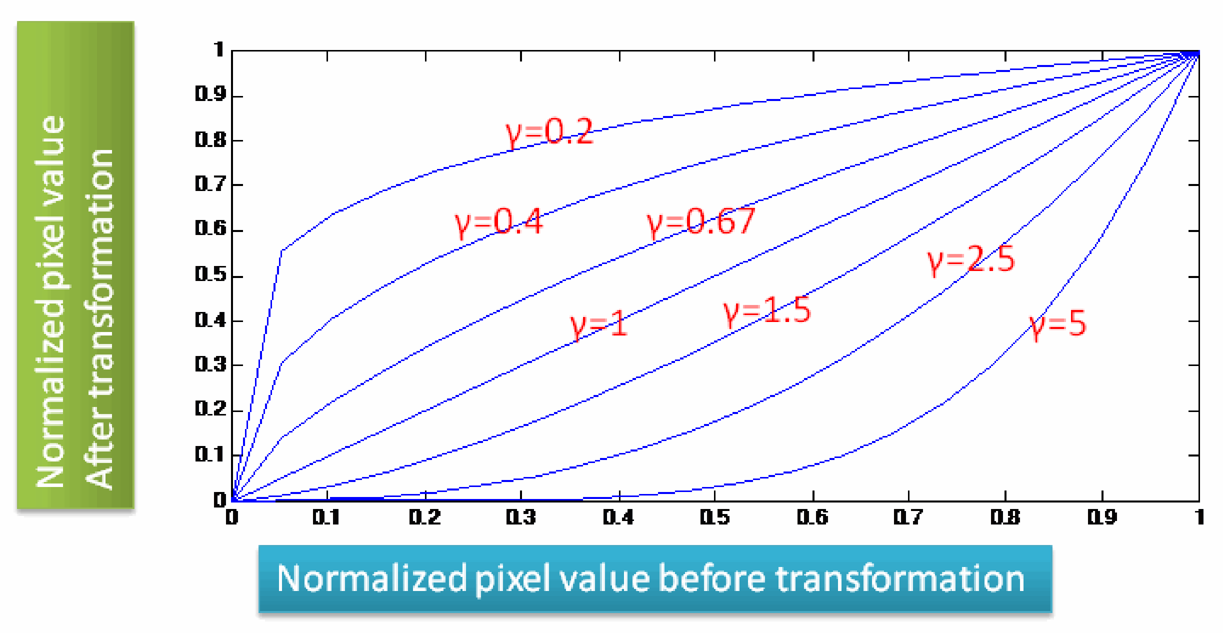
The Gaussian Low-Pass Filter (GLPF) is one of the most simplistic post-processing methods for removing noise. Although it is simple, it has been proved to be useful in many practical situations. The frequency response of a GLPF can be described as

$$H(u, v) = e^{-D^2(u,v)/2\sigma^2} \tag{5.13}$$

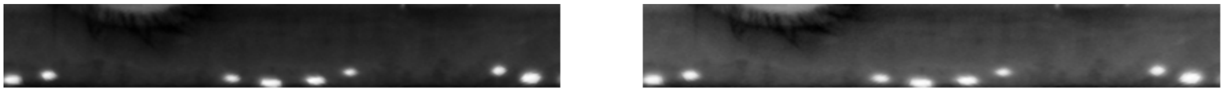
where $D(u, v)$ is the distance from the center of the frequency spectrum, and parameter $\sigma$ decides the cut-off frequency of the GLPF. The cut-off frequency should not be too large, otherwise details of the reconstructed image will be smoothed out. In the experiment, the size of the GLPF used is 3x3, with $\sigma = 0.5$.

#### 5.4.8.2 Gamma Correction

Gamma correction is a nonlinear transformation of pixel intensity value. It can be described as



(a)



(b)

Figure 5.13: (a) The relation between input and output pixel value of Gamma correction, based on different $\gamma$ (b) The effect of Gamma correction. The image on the left is the original input image, the image on the right is the output after Gamma corrected.

$$s = cr^{\gamma} \tag{5.14}$$

where $r$ is the original pixel value, $s$ is the new pixel value, $c$ and $\gamma$ are positive constants. Sometimes Eq. 5.14 will be written as $s = c(r + \epsilon)^{\gamma}$ to account for an offset.

Gamma correction provides a warping effect on the pixel intensity. Figure 5.13(a) shows this warping effect for different values of $\gamma$. Figure 5.13(b) shows a visual example of how Gamma correction can make the iris image brighter such that more details are revealed.

#### 5.4.8.3 Median Filter

Median filter is a very useful tool in removing salt-and-pepper noise. Assume the input image is $g(x, y)$, and given a local window $S$ whose size is pre-determined, the filtered image $f(x, y)$ can be expressed as

$$f(x, y) = \underset{(s,t)\in S}{median}\{g(s, t)\} \tag{5.15}$$

However, just like GLPF, when using the median filter, the window size should be carefully tuned. If the window size is too large, many details in the iris texture may be lost. In the experiment described in section 5.5, we use a median filter of size 3x3.

#### 5.4.8.4 Histogram Specification

Histogram specification is an image processing technique that maps the histogram of pixels of an input image to the histogram of pixels of a target image. It is different from the technique of "histogram equalization", which transforms the histogram of pixels of input image to an uniform distribution. However, histogram specification is based on histogram equalization. In the following paragraphs, the mathematical equations and the procedures of histogram specification are briefly reviewed.

Assume we have an input image $R$, and the pixel value contained in $R$ is denoted by a variable $r$. Suppose $R$ is of size MxN, and the range of pixel intensity value is within $[0, L - 1]$, then we can treat $r$ as a random variable, which lies in $[0, L - 1]$, and we have $MN$ observations of $r$. Now, let us consider a transformation for this random variable

$$s = T(r), \quad 0 \leq r \leq L - 1 \tag{5.16}$$

This transformation will transform the pixel value of input image $R$ into another set of pixel values, and producing a new image $S$. Assume $T(r)$ satisfies two conditions:

1. $T(r)$ is monotonically increasing function in the interval $0 \leq r \leq L - 1$
2. $0 \leq T(r) \leq L - 1$ for $0 \leq r \leq L - 1$

Then $T(r)$ is a reasonable histogram transformation function. This is because after histogram transformation, we would like to preserve the order of the pixel intensity value, so that if region A is brighter than region B in $R$, after histogram transformation, region A is still brighter than region B in $S$. Therefore, condition 1 is important. Condition 2 tells us that, after histogram transformation, the pixel intensity range of $S$ should still be confined within the original allowable range, which is also a basic requirement.

Since we treat $r$ as a random variable, we can compute the Probability Density Function (PDF) of $r$ from the input image $R$. Assume the PDF of $r$ is $p_r(r)$. Since $s$ is a transformation of $r$, it can also be treated as a random variable. Let $p_s(s)$ denotes the PDF of $s$. Now, let us consider a particular form of $T(r)$, as defined in Eq. 5.17

$$s = T(r) = (L-1)\int_0^r p_r(w)dw \tag{5.17}$$

The right side of Eq. 5.17 is the Cumulative Distribution Function (CDF) of $r$. Because PDFs are always positive and CDF is the integral of the PDF, $T(r)$ will be monotonically increasing, which satisfies condition 1. Also, when the right side of Eq. 5.17 reaches its maximum, which means $r = L-1$, the integral of a PDF within whole range is one, therefore, $s = L-1$. So $T(r)$ also satisfies condition 2.

If we take derivative of $s$ in Eq. 5.17, we get

$$\frac{ds}{dr} = \frac{dT(r)}{dr} = (L-1)\frac{d}{dr}\int_0^r p_r(w)dw = (L-1)p_r(r) \tag{5.18}$$

From the basic probability theory, we know the relation between the PDF of a transformed random variable and the PDF of original variable is

$$\begin{aligned} p_s(s) &= p_r(r)|\frac{dr}{ds}| \\ &= p_r(r)|\frac{1}{(L-1)p_r(r)}| \\ &= \frac{1}{L-1} \end{aligned} \tag{5.19}$$

Eq. 5.19 tells us that the PDF of $s$ is a constant, it does not change with respect to the value of $s$. Therefore, the histogram of the output image $S$ is an uniform distribution. Therefore, the transformation stated in Eq. 5.17 achieves "histogram equalization" of the input image $R$.

Now, let us add one more image into the example. Suppose we have another image $Z$, and random variable $z$ is used to model the distribution of the pixel intensity in $Z$. Assume the PDF of $z$ is $p_z(z)$. Now, consider another transformation function $G$:

$$G(z) = (L-1)\int_0^z p_z(t)dt \tag{5.20}$$

Compare Eq. 5.17 with Eq. 5.20, one can quickly see that $G(z)$ is the histogram equalization transform for $z$. Therefore, theoretically speaking, the new variable derived by transformation $G(z)$ would have exactly the same distribution of $s$. If $G(z)$ can transform $Z$ into a histogram equalized image $S$, then $G^{-1}(s)$ can transform a histogram equalized image $S$ into the target image $Z$.

In summary, assume the input image is $R$ and the target image is $Z$, the procedure for histogram specification is:

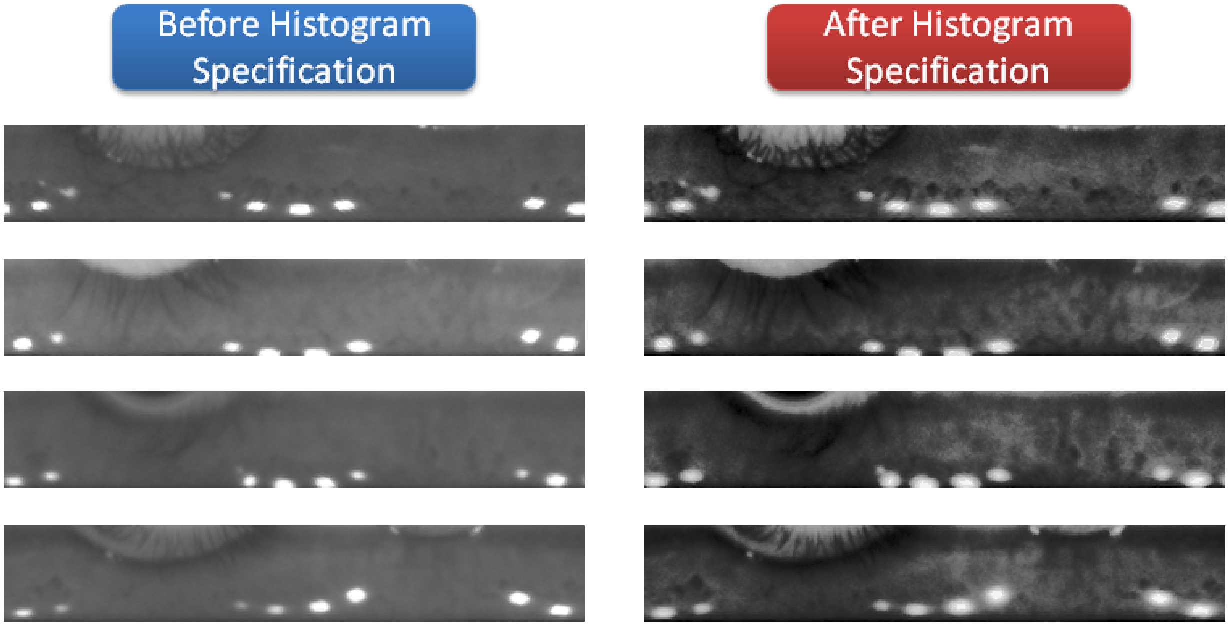


Figure 5.14: The effect of histogram specification.

1. Compute $p_r(r)$ from the input image $R$, and use Eq. 5.17 to obtain the mapping table from $r$ to $s$.
2. Compute $p_z(z)$ from the target image $Z$ and use Eq. 5.20 to obtain the mapping table from $z$ to $s$.
3. Compute the inverse transformation $z = G^{-1}(s)$ by inverting the table derived in step 2.
4. For every pixel in the input image $R$, first transform it to $s$ by using the look-up table in step 1, then transform it to $z$ by using the look-up table in step 3. Then the final image is the image whose histogram has been matched to the histogram of the target image $Z$.

Figure 5.14 shows a few example images before and after histogram specification. From those examples, we can clearly see that histogram specification is very useful and effective in enhancing the details of the original images.

## 5.5 Large-Scale Experiments

### 5.5.1 Database Description

In order to measure the iris recognition performance based on the proposed SR algorithm, experiments have to be performed on databases which contain both HR and LR iris images for the same iris class. So far there does not exist any such database which is publicly available to the iris recognition community. Therefore, we collected an iris database at Carnegie Mellon University during March and April in 2009. The iris images are captured by three different kinds of iris acquisition devices: IOM [6], LG IrisAccess 4000 [4] and SecuriMetrics PIER 2.3 [37]. The iris

Table 5.1: Details of iris database collected by IOM, LG IrisAccess 4000 and SecuriMetrics PIER 2.3

| Database Properties | IOM | LG | PIER |
|---|---|---|---|
| Number of iris classes | 111 | | |
| Size of the picture | 640x480 | | |
| Number of sessions | 3 | 1 | 1 |
| Maximal number of images per session | 28 | 4 | 3 |
| Minimal number of images per session | 3 | 4 | 3 |
| Average number of images per session | 8 | 4 | 3 |
| Total number of images | 2682 | 444 | 333 |

images captured by the IOM system are low-quality images. On the contrary, images captured by the LG IrisAccess 4000 and the SecuriMetrics PIER are high-quality.

For the LG IrisAccess 4000, the iris images are captured by default programs that come with the device. The default setting of the program captures two images of each of the subjects' eye. For every subject, we use the program to record the iris images twice. Therefore, for every iris class, there are four images recorded by the LG IrisAccess 4000.

For the PIER, the default enrollment program is used to record the iris images. For every iris class, three images are recorded by the PIER.

For the IOM, when the subjects walk through the IOM portal, the number of iris images recorded varies, depending on the subject's height and how fast they are moving through the portal. Often, only one of the subject's eyes is recorded. Therefore, the number of iris images recorded by the IOM is indefinite.

We asked every subject to walk through the IOM portal three times. Each such trial can be called a "session". Therefore, for every subject, iris images of three sessions are recorded. This allows us to explore the relation of the image series between different sessions for the same iris class.

The details of the iris database collected by the IOM, the LG and the PIER is listed in Table 5.1.

Figure 5.15 shows some examples of IOM images. Figure 5.16 shows some examples of LG images. Figure 5.17 shows some examples of PIER images.

### 5.5.2 Super-Resolution Methods Compared in the Experiments

In section 5.3, a few previous SR algorithms were introduced. In summary, there are three registration methods proposed by the following: (1) Keren et al., (2) Reddy and Chatterji (3) Vandewalle et al. For the image reconstruction methods, there are also three different existing methods: (1) nonuniform image interpolation (2) iterative back projection (3) robust Super-Resolution. Later in the experiments, all possible combinations of the image registration and reconstruction methods are used to compare the SR performance to the proposed methods.

Table 5.2 shows all nine possible combinations of the existing SR algorithm and their corresponding code names. Later in the experiment, the code name is used to refer to each algorithm.

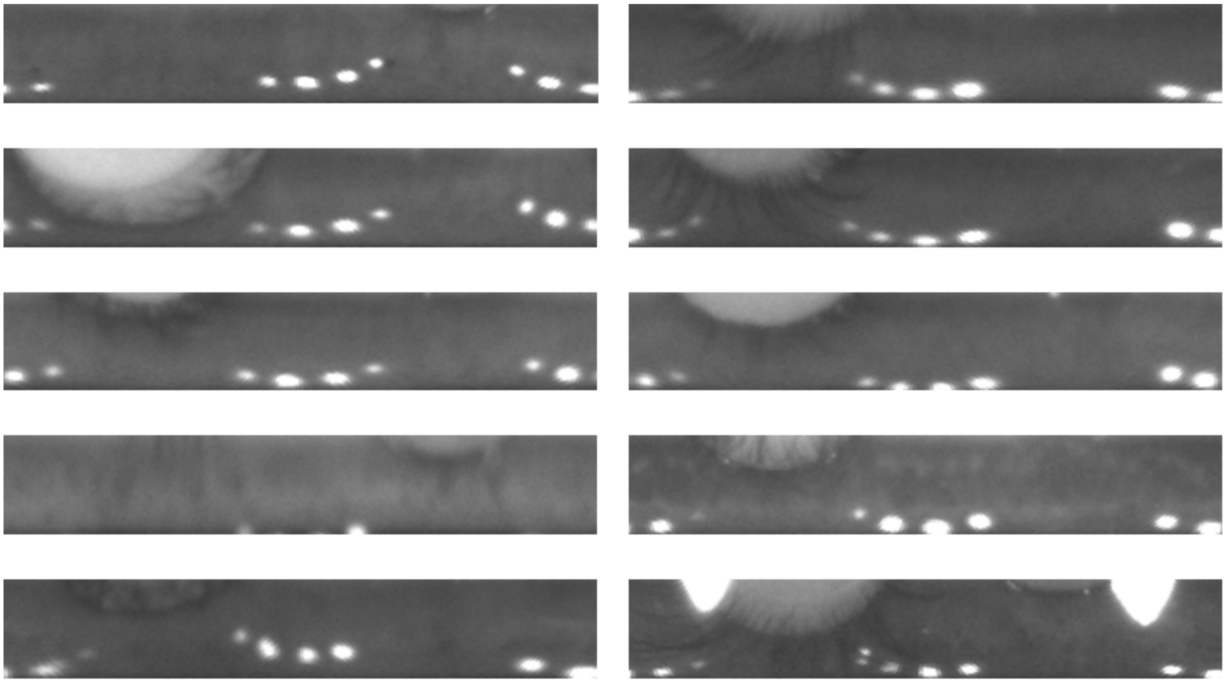


Figure 5.15: Examples of IOM images.

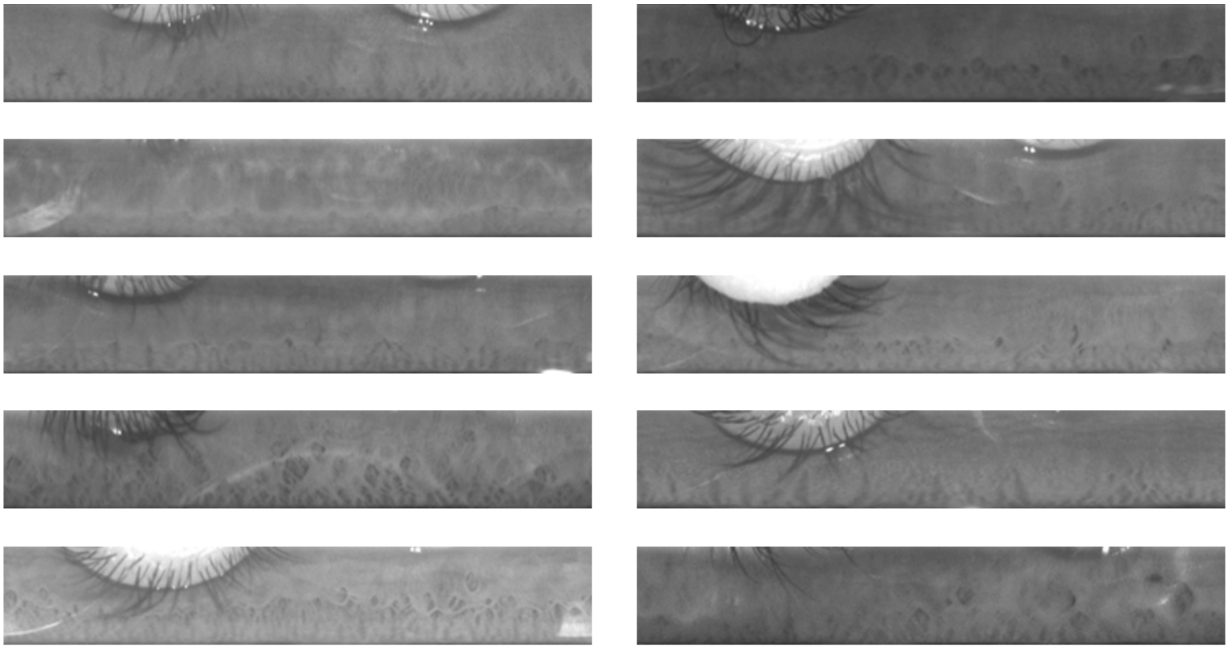


Figure 5.16: Examples of LG images.

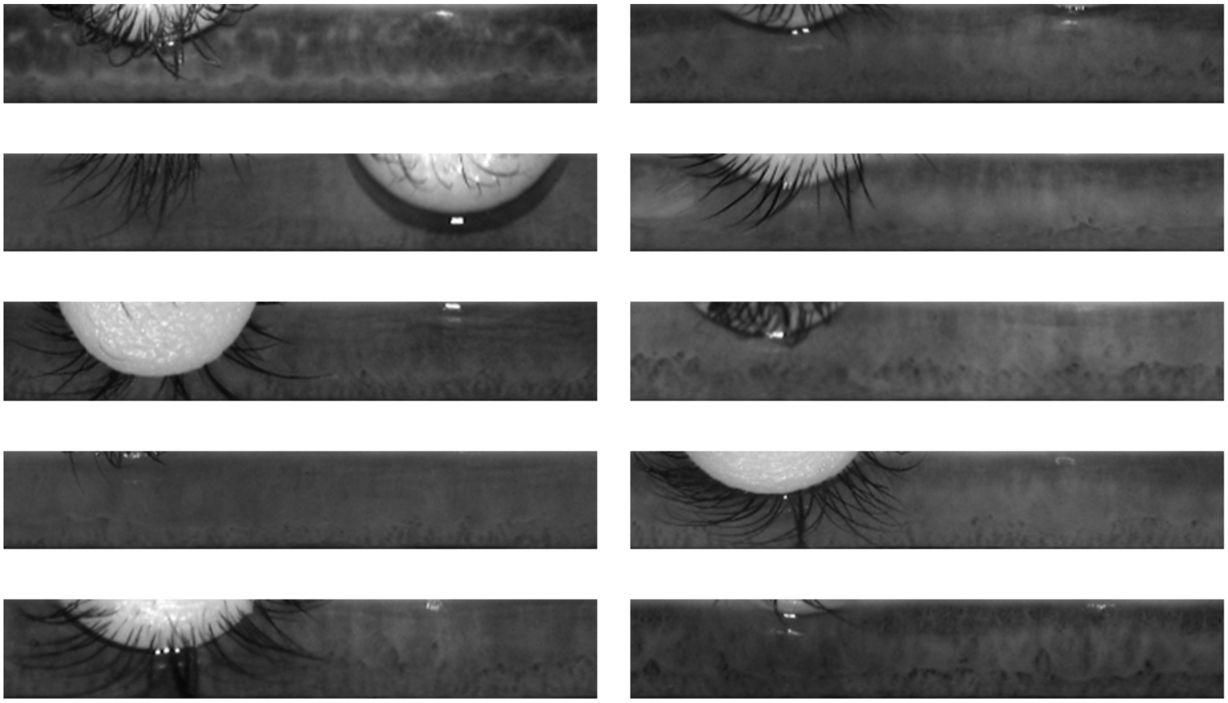


Figure 5.17: Examples of PIER images.

Table 5.2: Code names for the existing SR algorithms. There are totally nine possible combinations.

| Registration\Reconstruction | interpolation | back projection | robust Super-Resolution |
|---|---|---|---|
| Keren et al. | KE+IN | KE+BP | KE+RS |
| Reddy and Chatterji | RC+IN | RC+BP | RC+RS |
| Vandewalle et al. | VA+IN | VA+BP | VA+RS |

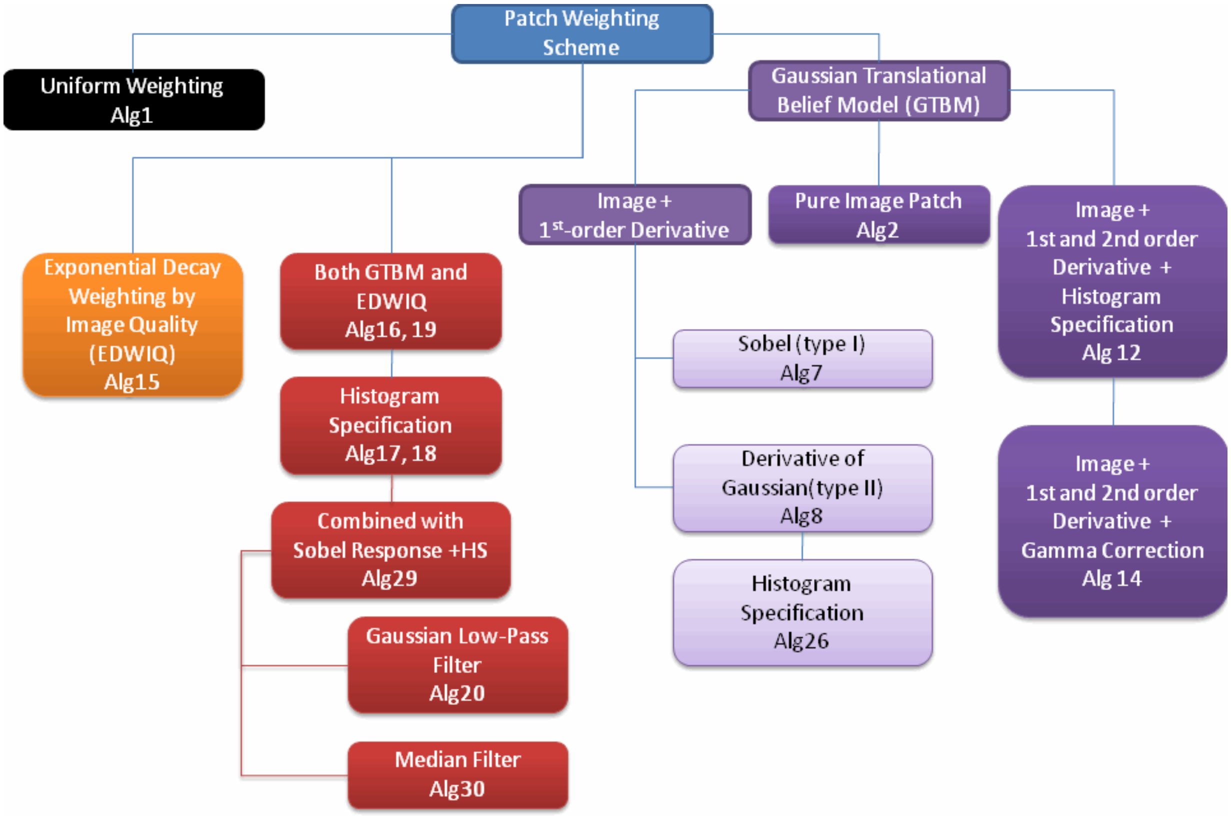

Patch Weighting Scheme
Uniform Weighting Alg1
Gaussian Translational Belief Model (GTBM)
Image + 1st-order Derivative
Pure Image Patch Alg2
Image + 1st and 2nd order Derivative + Histogram Specification Alg 12
Exponential Decay Weighting by Image Quality (EDWIQ) Alg15
Both GTBM and EDWIQ Alg16, 19
Sobel (type I) Alg7
Histogram Specification Alg17, 18
Derivative of Gaussian(type II) Alg8
Image + 1st and 2nd order Derivative + Gamma Correction Alg 14
Combined with Sobel Response +HS Alg29
Histogram Specification Alg26
Gaussian Low-Pass Filter Alg20
Median Filter Alg30

Figure 5.18: Relational chart of the proposed local patch-based SR algorithms. There are totally 15 combinations.

For the proposed local patch-based SR algorithms, given all of the choices that one can choose freely in each of the stages stated in section 5.4, there are, in total, 15 different combinations of local patch-based SR algorithms tested in the experiments. The relation between them is illustrated in Figure 5.18. They will be referred to as LPSR#, where # is the numbering given to that algorithm. In summary, the characteristics of each algorithm are the following:

- LPSR1: use uniform weighting to weigh the local patches. Use only the image patches to do reconstruction.
- LPSR2: use GTBM to weigh the local patches. Use only the image patches to do reconstruction.
- LPSR7: use GTBM to weigh the local patches. Use both image patches and the first-order derivative of the patches to do reconstruction. The first-order derivative of the patches is computed by convolving image with Sobel filter (type I).
- LPSR8: use GTBM to weigh the local patches. Use both image patches and the first-order derivative of the patches to do reconstruction. The first-order derivative of the patches are computed by convolving images with derivative of Gaussians (type II).
- LPSR26: same as 8, but the final reconstructed image is transformed by using histogram specification.
- LPSR12: use GTBM to weigh the local patches. The local patches are reconstructed by using the image patches, the first-order and the second-order derivative of the patches. The first-order derivative of the patches are computed by convolving images with derivative of Gaussians (type II). The final reconstructed image is transformed by using histogram specification.
- LPSR14: use GTBM to weigh the local patches. The local patches are reconstructed by using the image patches, the first-order and the second-order derivative of the patches. The first-order derivative of the patches are computed by convolving images with derivative of Gaussians (type II). The final reconstructed image is transformed by using Gamma correction.
- LPSR15: use EDWIQ to weigh the local patches. Use only the image patches to do reconstruction.
- LPSR16: use both GTBM and EDWIQ to weigh the local patches. Use only the image patches to do reconstruction.
- LPSR17: use both GTBM and EDWIQ to weigh the local patches. Use only the image patches to do reconstruction. The final reconstructed image is transformed by using histogram specification.
- LPSR18: same as 17, but the parameter of GTBM has been fine-tuned according to a smaller subset of data.
- LPSR19: same as 16, but the parameter of GTBM has been fine-tuned according to a smaller subset of data.
- LPSR29: use both GTBM and EDWIQ to weigh the local patches. Use both image patches and the first-order derivative of the patches to do reconstruction. The final reconstructed

image is transformed by using histogram specification.

- LPSR20: same as 29, but the the final reconstructed image is further transformed by using GLPF.
- LPSR30: same as 29, but the the final reconstructed image is further filtered by median filter.

### 5.5.3 Performance Evaluation -Scenario 1

The goal of the experiments is to measure the recognition performance using the proposed SR algorithm and compare it to the original IOM system. In order to fairly measure the difference of the performance, we have to understand how the regular IOM system operates. Here, the operation of a regular IOM system refers to how a regular IOM system enrolls a new iris class into the database and how it perform identification during the test stage. It will be illustrated in the following paragraph.

#### 5.5.3.1 Baseline for Scenario 1: IOM vs. PIER

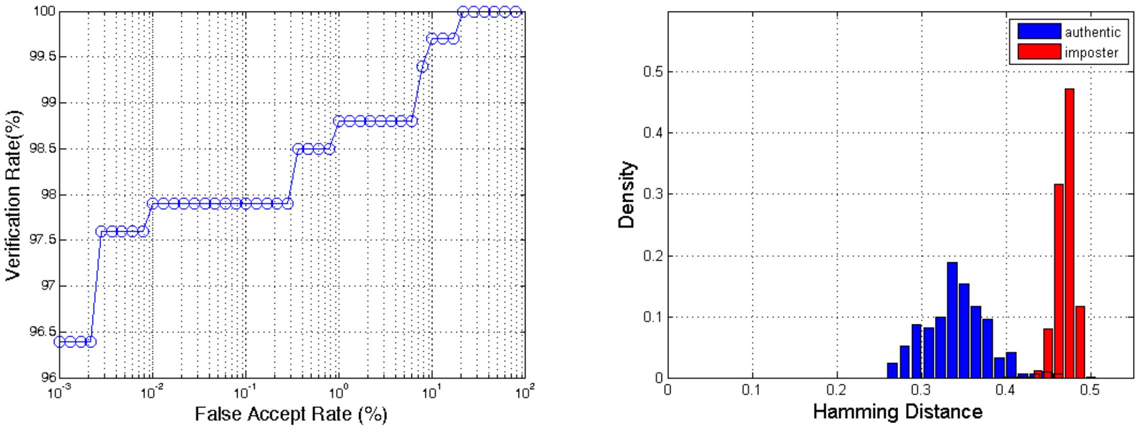
The baseline system describes how a regular IOM system works when no SR algorithm is applied. In almost all biometric recognition systems, the recognition performance can be measured if the global similarity matrix $\Sigma$ can be computed. The way to compute the global similarity matrix $\Sigma$ is to compute the similarity score, or the distance score, between the gallery class $\gamma$ and the probe class $\pi$, and record the score in the element $\Sigma(\gamma, \pi)$.

For a baseline system, suppose we use all three images captured by the PIER device to enroll the subject into the database of gallery images. During the test (identification) stage, when a subject is walking through the IOM portal, multiple iris images are recorded by the IOM device. Assume for the probe class $\pi$, there are totally $p$ IOM images recorded, and for each gallery class $\gamma$, there are $g$ images. In our case, $g = 3$ because we only have three PIER images for each class.

For each gallery class $\gamma$ and probe class $\pi$, a local similarity matrix $L$ is formed by computing the similarity score between every possible pair from the gallery images and the probe images. Then the largest value (or the smallest value, if distance score is used) of this local similarity matrix is the best score we can get to match gallery class $\gamma$ with the probe class $\pi$, when no SR algorithm is used. In mathematics, it can be described as following:

$$\Sigma(\gamma, \pi) = min(L(i, j)),\ \forall 1 \leq i \leq g,\ 1 \leq j \leq p \tag{5.21}$$

Figure 5.19 shows the iris recognition performance of the baseline system, which is IOM vs. PIER. We call this configuration as scenario 1 because later in section 5.5.4 another possible configuration is introduced. Figure 5.19(a) shows the ROC curve of baseline 1, and Figure 5.19(b) shows the histogram distribution of the HD.



(a) (b)

Figure 5.19: Iris recognition performance of the default IOM setting, which is IOM vs. PIER (scenario 1). (a) ROC curve (b) Histogram distribution of HD of the authentic and imposter comparison.

#### 5.5.3.2 Proposed System Structure for Enhancing Image Quality by SR Algorithms in Scenario 1: SR vs. PIER

In order to use the SR algorithm to enhance the iris recognition performance, the way we compute the local similarity matrix needs to be changed. For probe class $\pi$, we can apply the proposed SR algorithm to the input probe images captured by the IOM system, generating a single high quality image $S_\pi$. Now the new local similarity matrix $\Lambda$ becomes a vector of size $g \times 1$ because we only need to match $S_\pi$ with each gallery images. In mathematics, it can be described as following:

$$\Sigma(\gamma, \pi) = min(\Lambda(i, 1)), \; \forall 1 \leq i \leq g \tag{5.22}$$

Because we are matching a super-resolved image with a PIER image, we can call this setting as “SR vs. PIER”.

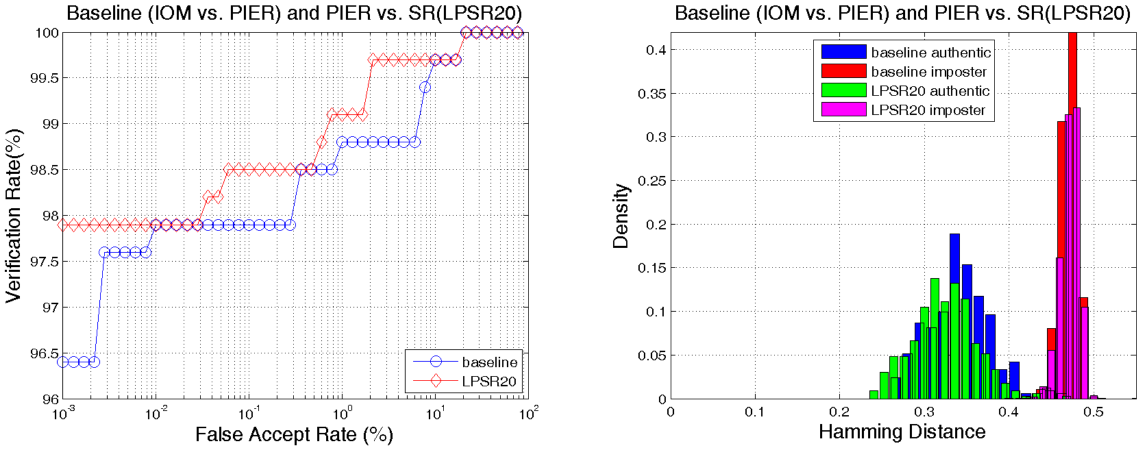
#### 5.5.3.3 Performance Evaluation of SR-enhanced System (SRvsPIER) for all SR Methods

As described in section 5.5.2, there are 9 existing SR methods and 15 proposed local patch-based SR methods. So there are totally 24 SR methods being compared in this section. Figure A.1-A.8 shows all of the plots for the 24 SR methods in Appendix A. Among them we can see that the LPSR20 achieves the lowest error rate. The ROC curves and histogram distribution of the Hamming distance of the LPSR20 versus baseline are shown in Figure 5.20.

For every experiment, besides ROC curves and HD histogram distribution, four important quantitative metrics are also recorded, they are the following: Equal Error Rate (EER), False Reject Rate at False Accept Rate=0.1% (FRR@FAR=0.1), False Reject Rate at False Accept Rate=0% (FRR@FAR=0) and Fisher Ratio (FR) of the HD histogram distribution. They are all listed in Table 5.3. This table is sorted according to the performance. The performance is

Table 5.3: EER, FRR@FAR=0.1, FRR@FAR=0 and FR for all of the SR experiment (SRvsPIER) in scenario 1.

| Method | FRR@FAR=0.1 | EER | FRR@FAR=0 | FR |
|---|---|---|---|---|
| LPSR20 | 1.50% | 0.90% | 2.10% | 2.932 |
| LPSR18 | 1.50% | 0.90% | 2.10% | 2.863 |
| LPSR17 | 1.50% | 0.90% | 2.10% | 2.804 |
| LPSR30 | 1.50% | 0.90% | 2.40% | 2.943 |
| LPSR29 | 1.50% | 0.90% | 2.40% | 2.878 |
| LPSR12 | 1.50% | 1.20% | 1.80% | 2.865 |
| LPSR2 | 1.50% | 1.20% | 2.10% | 2.873 |
| LPSR8 | 1.50% | 1.20% | 2.10% | 2.839 |
| LPSR16 | 1.50% | 1.20% | 2.10% | 2.721 |
| LPSR26 | 1.50% | 1.50% | 2.10% | 2.874 |
| LPSR14 | 1.50% | 1.50% | 2.10% | 2.83 |
| LPSR15 | 1.80% | 1.20% | 2.10% | 2.805 |
| LPSR19 | 1.80% | 1.20% | 2.10% | 2.793 |
| LPSR7 | 1.80% | 1.50% | 2.10% | 2.793 |
| LPSR1 | 2.10% | 1.20% | 3.60% | 2.588 |
| baseline | 2.11% | 1.20% | 3.61% | 2.74 |
| VA+IN | 2.40% | 1.20% | 4.81% | 2.717 |
| VA+RS | 2.40% | 1.50% | 2.40% | 2.721 |
| KE+RS | 3.00% | 1.20% | 5.11% | 2.51 |
| KE+IN | 4.81% | 1.80% | 10.21% | 2.293 |
| VA+BP | 10.51% | 7.81% | 12.01% | 1.881 |
| KE+BP | 10.81% | 7.21% | 13.81% | 1.832 |
| RC+RS | 21.62% | 8.41% | 31.53% | 1.443 |
| RC+IN | 29.13% | 8.71% | 37.84% | 1.337 |
| RC+BP | 30.63% | 13.51% | 41.74% | 1.19 |



(a) (b)

Figure 5.20: Large-scale iris recognition performance of SR-enhanced System, under the system structure SR vs. PIER. Under this setting, LPSR20 performs best. (a) ROC curves of baseline (IOM vs. PIER) and LPSR20-enhanced system. (b) Histogram distribution of Hamming distance of baseline (IOM vs. PIER) and LPSR20-enhanced system.

evaluated according to the following priorities: (1) FRR@FAR=0.1 (the lower the better) (2) EER (the lower the better) (3) FRR@FAR=0 (the lower the better) (4) FR (the higher the better).

### 5.5.4 Performance Evaluation -Scenario 2

Section 5.5.3.1 described the default setting for the IOM system. However, there are other possibilities. In this section, we offer another possible baseline setup and the possible solution for embedding SR methods to enhance performance in such setting.

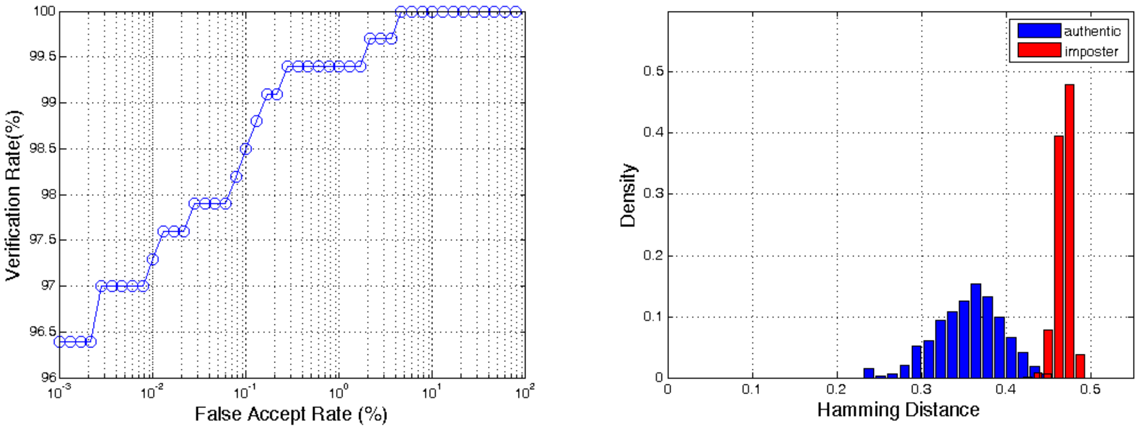
#### 5.5.4.1 Baseline for Scenario 2: IOM vs. LG

Since we collect LG images in our database, we can also use LG images to enroll new iris classes. In this baseline, everything is the same as described in section 5.5.3.1, except we replace PIER images with LG images during the enrollment and testing stages. This setting can be called “IOM vs. LG”.

Figure 5.21 shows the iris recognition performance of the baseline 2, which is IOM vs. LG. Figure 5.21(a) shows the ROC curve of baseline 2, and Figure 5.21(b) shows the histogram distribution of the HD of baseline 2.

#### 5.5.4.2 Proposed System Structure for Enhancing Image Quality by SR Algorithms in Scenario 2: SR vs. LG

We can apply the same system as described in section 5.5.3.2. In summary, we can super-resolve a high quality image $S_\pi$ from several low quality IOM images of probe class $\pi$, and compute the



(a) (b)

Figure 5.21: Iris recognition performance of the baseline 2, which is IOM vs. LG (scenario 2). (a) ROC curve (b) Histogram distribution of HD of the authentic and imposter comparison.

new local similarity matrix $\Lambda$ based on $S_\pi$. The computation of the global similarity matrix $\Sigma$ is still the same, as described in Eq. (5.22). This scheme can be called as "SR vs. LG".

#### 5.5.4.3 Performance Evaluation of SR-enhanced System (SRvsLG) for all SR Methods

We used both the ROC curve and HD histogram distribution plot to illustrate the performance and compared it with the baseline performance. All nine existing SR methods and 15 proposed local patch-based SR methods are plotted, just like what we did in section 5.5.3.3. So there are totally 24 SR methods being compared in this section. Figure B.1-B.8 shows all of the plots for the 24 SR methods in Appendix B. Table 5.4 shows the four quantitative measures (FRR@FAR=0.1, EER, FRR@FAR=0, and FR) for all of the 24 SR methods, as well as the baseline system, which is IOM vs. LG. Among them we can see that LPSR30 achieves the lowest error rate. The ROC curves and histogram distribution of the Hamming distance of the LPSR30 versus the baseline are shown in Figure 5.22.

### 5.5.5 Performance Evaluation -Scenario 3

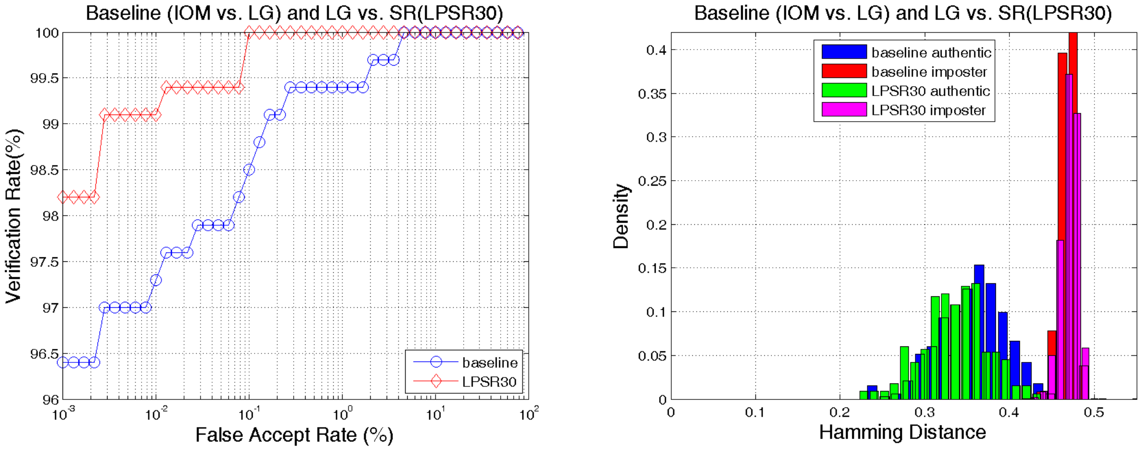
Besides the system architecture proposed in section 5.5.3.2 (SR vs. PIER) and 5.5.4.2 (SR vs. LG), we would also like to propose a different system architecture to see if we can further improve the overall performance. These two new system architectures can be called "SR vs. IOM" and "SR vs. SR". We describe each of them in section 5.5.5 and 5.5.6, respectively.

#### 5.5.5.1 Proposed System Structure in Scenario 3: SR vs. IOM

One of our ultimate goals in this study is to explore the possibility of whether we could use super-resolved high quality images for iris database enrollment so that we do not have to use the PIER device to capture images anymore. Using the IOM system to acquire iris images is

Table 5.4: EER, FRR@FAR=0.1, FRR@FAR=0 and FR for all of the SR experiment (SRvsLG) in scenario 2.

| Method | FRR@FAR=0.1 | EER | FRR@FAR=0 | FR |
|---|---|---|---|---|
| LPSR30 | 0.00% | 0.00% | 1.80% | 2.818 |
| LPSR26 | 0.00% | 0.00% | 1.80% | 2.762 |
| LPSR12 | 0.30% | 0.00% | 1.80% | 2.769 |
| LPSR17 | 0.30% | 0.00% | 1.80% | 2.686 |
| LPSR29 | 0.30% | 0.00% | 2.10% | 2.749 |
| LPSR20 | 0.30% | 0.30% | 1.80% | 2.803 |
| LPSR2 | 0.30% | 0.30% | 2.70% | 2.757 |
| LPSR8 | 0.30% | 0.30% | 2.70% | 2.707 |
| LPSR7 | 0.30% | 0.30% | 2.70% | 2.676 |
| LPSR18 | 0.60% | 0.00% | 1.20% | 2.755 |
| LPSR14 | 0.60% | 0.30% | 2.70% | 2.708 |
| LPSR15 | 0.60% | 0.30% | 2.70% | 2.673 |
| LPSR19 | 0.60% | 0.30% | 2.70% | 2.662 |
| LPSR16 | 0.60% | 0.30% | 3.00% | 2.584 |
| LPSR1 | 0.60% | 0.30% | 4.20% | 2.552 |
| VA+RS | 0.90% | 0.30% | 3.30% | 2.679 |
| KE+RS | 1.20% | 0.60% | 5.41% | 2.443 |
| baseline | 1.50% | 0.60% | 3.60% | 2.35 |
| VA+IN | 1.50% | 0.30% | 3.60% | 2.635 |
| KE+IN | 3.60% | 0.60% | 10.81% | 2.23 |
| VA+BP | 8.71% | 6.91% | 11.41% | 1.892 |
| KE+BP | 9.01% | 6.61% | 11.11% | 1.828 |
| RC+RS | 18.92% | 7.51% | 39.64% | 1.491 |
| RC+IN | 26.73% | 9.31% | 37.84% | 1.312 |
| RC+BP | 29.13% | 11.71% | 51.95% | 1.19 |



(a) (b)

Figure 5.22: Large-scale iris recognition performance of SR-enhanced System, under the system structure SR vs. LG. Under this setting, LPSR30 performs best. (a) ROC curves of baseline (IOM vs. LG) and LPSR30-enhanced system. (b) Histogram distribution of Hamming distance of baseline (IOM vs. LG) and LPSR30-enhanced system.

much easier and time-saving than most of the traditional iris capturing device. Therefore, if we can achieve the goal of enrolling iris data by using SR images, it will improve the usability and user-friendliness of the system.

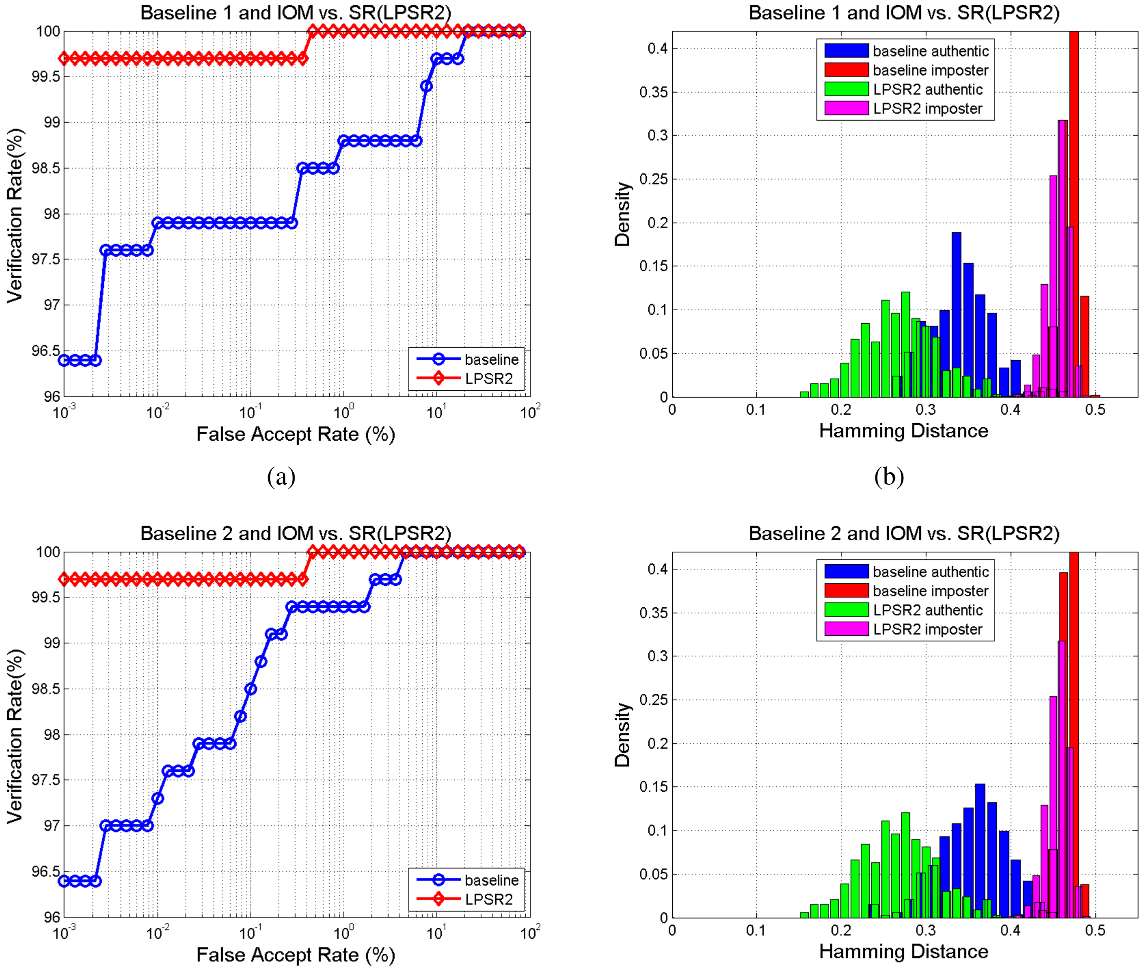
The new system structure can be called “SR vs. IOM”. During the data enrollment, for gallery class $\gamma$, instead of using PIER or LG images, we use super-resolved image $S_\gamma$(computed by super-resolving IOM images of class $\gamma$) to enroll. During the testing stage, the local similarity matrix $\Lambda$ (between gallery class $\gamma$ and probe class $\pi$) is computed by matching $S_\gamma$ with all IOM images of the probe class $\pi$. Assume there are $p$ IOM images for probe class $\pi$. In mathematics, it can be described as the following:

$$\Sigma(\gamma, \pi) = min(\Lambda(1, j)), \ \forall 1 \leq j \leq p \tag{5.23}$$

Under the system structure of “SR vs. IOM”, we perform large-scale iris recognition experiments for all of the 24 algorithms (9 existing algorithms + 15 proposed algorithms). For comparison purposes, we can compare the performance, in the format of either ROC curves or HD histogram distribution, to either baseline 1 (IOM vs. PIER) or baseline 2 (IOM vs. LG). All of the experimental result are shown in figures in Appendix C. Table 5.5 shows the four quantitative measures (FRR@FAR=0.1, EER, FRR@FAR=0, and FR) for all of the experiments, as well as both baseline systems. Among them we can see that the LPSR2 achieves the lowest error rate. The ROC curves and histogram distribution of the Hamming distance of the LPSR2 versus both baselines are shown in Figure 5.23.

Table 5.5: EER, FRR@FAR=0.1, FRR@FAR=0 and FR for all of the SR experiment (SRvsIOM) in scenario 3, compared with baseline 1 (PIER vs. IOM) and baseline 2 (LG vs. IOM).

| Method | FRR@FAR=0.1 | EER | FRR@FAR=0 | FR |
|---|---|---|---|---|
| LPSR2 | 0.3% | 0.3% | 0.3% | 3.223 |
| LPSR12 | 0.3% | 0.3% | 0.3% | 3.156 |
| LPSR26 | 0.3% | 0.3% | 0.3% | 3.121 |
| LPSR7 | 0.3% | 0.3% | 0.6% | 3.132 |
| LPSR30 | 0.3% | 0.3% | 0.6% | 3.121 |
| LPSR18 | 0.3% | 0.3% | 0.6% | 3.094 |
| LPSR20 | 0.3% | 0.3% | 0.6% | 3.08 |
| LPSR29 | 0.3% | 0.3% | 0.6% | 3.036 |
| LPSR14 | 0.3% | 0.3% | 0.9% | 3.2 |
| LPSR8 | 0.3% | 0.3% | 0.9% | 3.183 |
| LPSR15 | 0.3% | 0.3% | 0.9% | 3.153 |
| LPSR19 | 0.3% | 0.3% | 0.9% | 3.127 |
| LPSR1 | 0.3% | 0.3% | 0.9% | 3.122 |
| LPSR16 | 0.3% | 0.3% | 1.2% | 3.039 |
| LPSR17 | 0.3% | 0.3% | 1.2% | 3.01 |
| VA+RS | 0.3% | 0.3% | 1.5% | 3.125 |
| KE+RS | 0.3% | 0.3% | 1.5% | 2.968 |
| VA+IN | 0.3% | 0.3% | 1.8% | 3.107 |
| KE+IN | 0.3% | 0.3% | 3.6% | 2.776 |
| baseline2 | 1.5% | 0.6% | 3.6% | 2.35 |
| baseline1 | 2.1% | 1.2% | 3.6% | 2.74 |
| VA+BP | 2.1% | 1.5% | 2.4% | 2.659 |
| KE+BP | 2.7% | 1.8% | 3.3% | 2.573 |
| RC+RS | 8.4% | 2.1% | 13.5% | 2.08 |
| RC+IN | 8.7% | 2.4% | 17.1% | 1.964 |
| RC+BP | 11.7% | 4.5% | 19.2% | 1.829 |



(a)

(b)

(c)

(d)

Figure 5.23: Large-scale iris recognition performance of SR-enhanced System, under the system structure SR vs. IOM. Under this setting, LPSR2 performs best. (a) ROC curves of baseline 1 (IOM vs. PIER) and LPSR2-enhanced system. (b) Histogram distribution of Hamming distance of baseline 1 (IOM vs. PIER) and LPSR2-enhanced system. (c) ROC curves of baseline 2 (IOM vs. LG) and LPSR2-enhanced system. (d) Histogram distribution of Hamming distance of baseline 2 (IOM vs. LG) and LPSR2-enhanced system.

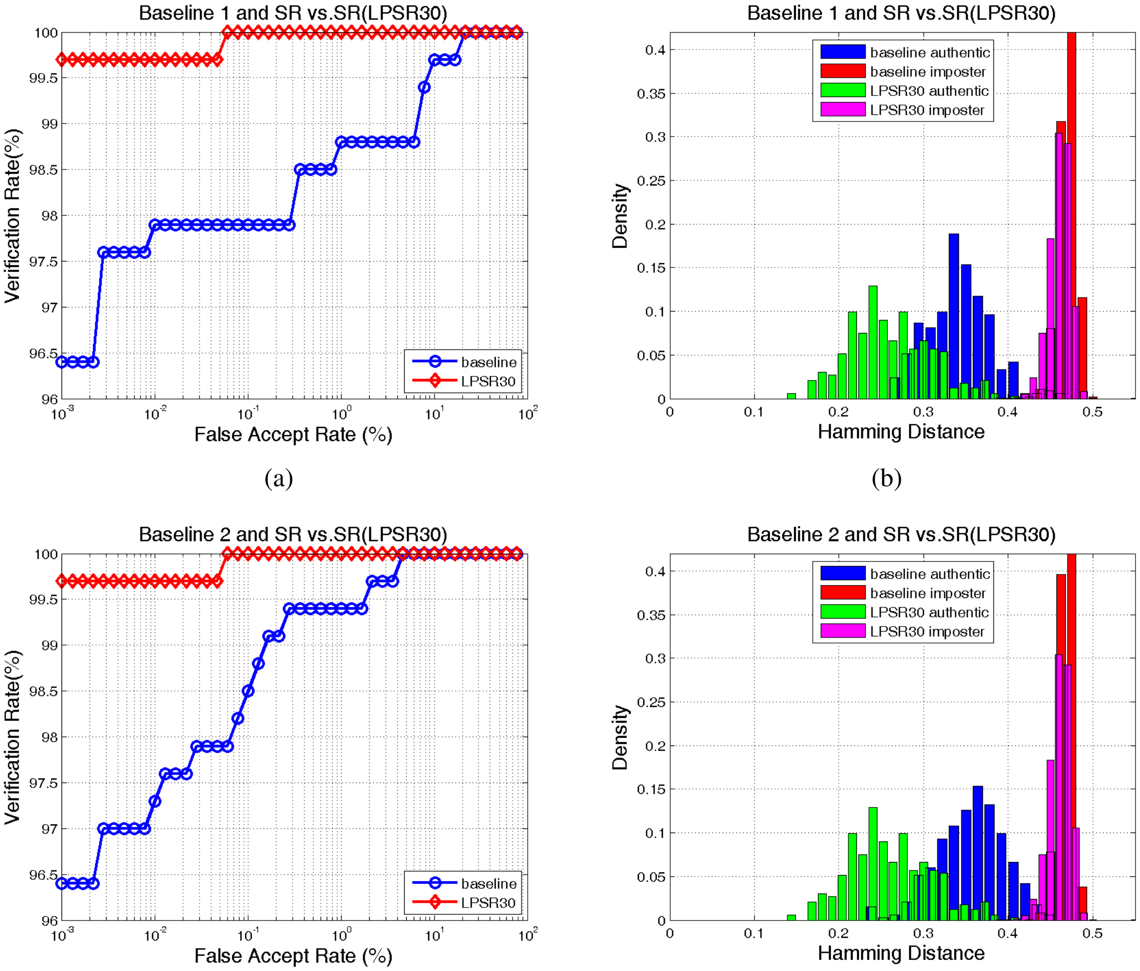
Table 5.6: EER, FRR@FAR=0.1, FRR@FAR=0 and FR for all of the SR experiment (SRvsSR) in scenario 4, compared with baseline 1 (PIER vs. IOM) and baseline 2 (LG vs. IOM).

| Method | FRR@FAR=0.1 | EER | FRR@FAR=0 | FR |
|---|---|---|---|---|
| LPSR30 | 0.0% | 0.0% | 0.3% | 3.288 |
| LPSR14 | 0.0% | 0.0% | 0.6% | 3.349 |
| LPSR12 | 0.0% | 0.0% | 0.6% | 3.315 |
| LPSR26 | 0.0% | 0.0% | 0.6% | 3.236 |
| LPSR18 | 0.0% | 0.0% | 0.6% | 3.148 |
| LPSR29 | 0.0% | 0.0% | 0.6% | 3.119 |
| LPSR17 | 0.0% | 0.0% | 0.6% | 3.084 |
| LPSR20 | 0.3% | 0.0% | 0.3% | 3.191 |
| LPSR2 | 0.3% | 0.0% | 0.6% | 3.428 |
| LPSR8 | 0.3% | 0.0% | 0.6% | 3.333 |
| LPSR7 | 0.3% | 0.0% | 0.6% | 3.24 |
| VA+RS | 0.3% | 0.3% | 0.3% | 3.048 |
| LPSR15 | 0.3% | 0.3% | 0.6% | 3.205 |
| LPSR19 | 0.3% | 0.3% | 0.9% | 3.162 |
| LPSR16 | 0.3% | 0.3% | 0.9% | 3.132 |
| KE+RS | 0.3% | 0.3% | 1.8% | 2.766 |
| VA+IN | 0.6% | 0.3% | 1.8% | 3.143 |
| LPSR1 | 0.6% | 0.3% | 1.8% | 3.073 |
| baseline2 | 1.5% | 0.6% | 3.6% | 2.35 |
| baseline1 | 2.1% | 1.2% | 3.6% | 2.74 |
| KE+IN | 2.4% | 0.9% | 10.8% | 2.501 |
| KE+BP | 9.0% | 7.2% | 10.5% | 1.91 |
| VA+BP | 9.0% | 7.5% | 10.8% | 1.948 |
| RC+RS | 24.0% | 10.5% | 40.2% | 1.353 |
| RC+BP | 32.4% | 15.6% | 41.4% | 1.117 |
| RC+IN | 34.8% | 10.5% | 46.2% | 1.219 |

### 5.5.6 Performance Evaluation -Scenario 4

Another system structure is worth exploring, too. In section 5.5.5.1, we enroll the iris data with SR images and match them with the IOM images. If the SR algorithm is also applied during the testing stage, we can generate one single high-quality iris image from several probe images. Then the local similarity matrix $\Lambda$, between gallery class $\gamma$ and probe class $\pi$, can be simplified to a scalar value, derived by computing the the similarity score $\xi_{\gamma,\pi}$(or distance score) between $S_\gamma$ and $S_\pi$. The global similarity matrix can be computed as $\Sigma(\gamma,\pi) = \xi_{\gamma,\pi}$. This structure can be called as “SR vs SR”.

Similar to sub-section 5.5.5.1, for the system structure of “SR vs. SR”, we also perform a large-scale experiment of all 24 SR algorithms. All of the results are shown in figures in Appendix D. Table 5.6 shows the four quantitative measures (FRR@FAR=0.1, EER, FRR@FAR=0,



(a)

(b)

(c)

(d)

Figure 5.24: Large-scale iris recognition performance of SR-enhanced System, under the system structure SR vs. SR. Under this setting, LPSR30 performs best. (a) ROC curves of baseline 1 (IOM vs. PIER) and LPSR30-enhanced system. (b) Histogram distribution of Hamming distance of baseline 1 (IOM vs. PIER) and LPSR30-enhanced system. (c) ROC curves of baseline 2 (IOM vs. LG) and LPSR30-enhanced system. (d) Histogram distribution of Hamming distance of baseline 2 (IOM vs. LG) and LPSR30-enhanced system.

and FR) for all 24 SR-enhanced systems, as well as the both baseline systems. Among them we can see that the LPSR30 achieves the lowest error rate. The ROC curves and histogram distribution of the Hamming distance of the LPSR30 versus both baselines are shown in Figure 5.24.

## 5.6 Exploration of the Limit of the Power of the Proposed SR Algorithm

We want to explore the limit of the power of our proposed SR algorithm. As stated in section 5.2, the two major factors that degrade the images are noise and defocus blur. In this section, we describe how we design experiments to simulate the degradation process and measure the performance of our proposed SR algorithm under various degradation levels.

### 5.6.1 The Limit of the Ability of Proposed SR Algorithm in Handling Noise Contamination

We designed experiments to detect how well our proposed SR algorithms could handle noise contamination, in a quantitative manner. As stated in section 5.5, we have the PIER database which contains crystal clear iris images. Assume there is no noise in the PIER database. For every iris class, we take all three PIER images of that class as gallery image set, and add Gaussian white noise with a specific Signal-to-Noise Ratio (SNR). We then use the noise contaminated PIER images as probe set. We match the probe set with the gallery set, and the smallest HD is the best HD we can get with presence of the noise with this predefined SNR, for this particular class. We repeat this procedure for all classes, for a wide range of SNR values. The SNR range we used is from 1 to 40. This experiment test the ability of our iris matcher, and the results tell us how the HD distribution varies with the SNR levels. These results can be called the baseline.

For our proposed SR algorithm, for every iris class, again, we add Gaussian white noise to all three PIER images. Then, we apply our proposed SR algorithm to these three noisy PIER images and get a super-resolved image. Matching this SR image with all three images in the gallery set, we get three HDs. The smallest value among them is the best score we can get by using SR algorithm to improve iris quality. We repeat this procedure for all classes, for a wide range of SNR values. We use the LPSR30 algorithm in this experiment because it is the most delicate algorithm among all, and seems to have the best performance in section 5.5.4.3.

The results of this experiment is the HD of authentic comparison, with various SNR levels. In order to show more details of the HD distribution for every SNR level, we choose to use the box plot to present the result. The main component of the box plot contains a rectangular box with a red line in the middle. The red line indicates the median of the distribution, while the lower and upper edge of the box indicates the lower and higher quartile of the data. The distance between the higher and lower quartile is called the Inter-Quartile Range (IQR). The upper and lower black edge outside the box is called the “whisker”, which indicates the location that extends 1.5 IQR from the higher and lower quartile. Samples that fall outside the upper and lower whisker are defined as outliers and are denoted as red + sign. Figure 5.25 shows the effect of noises on the

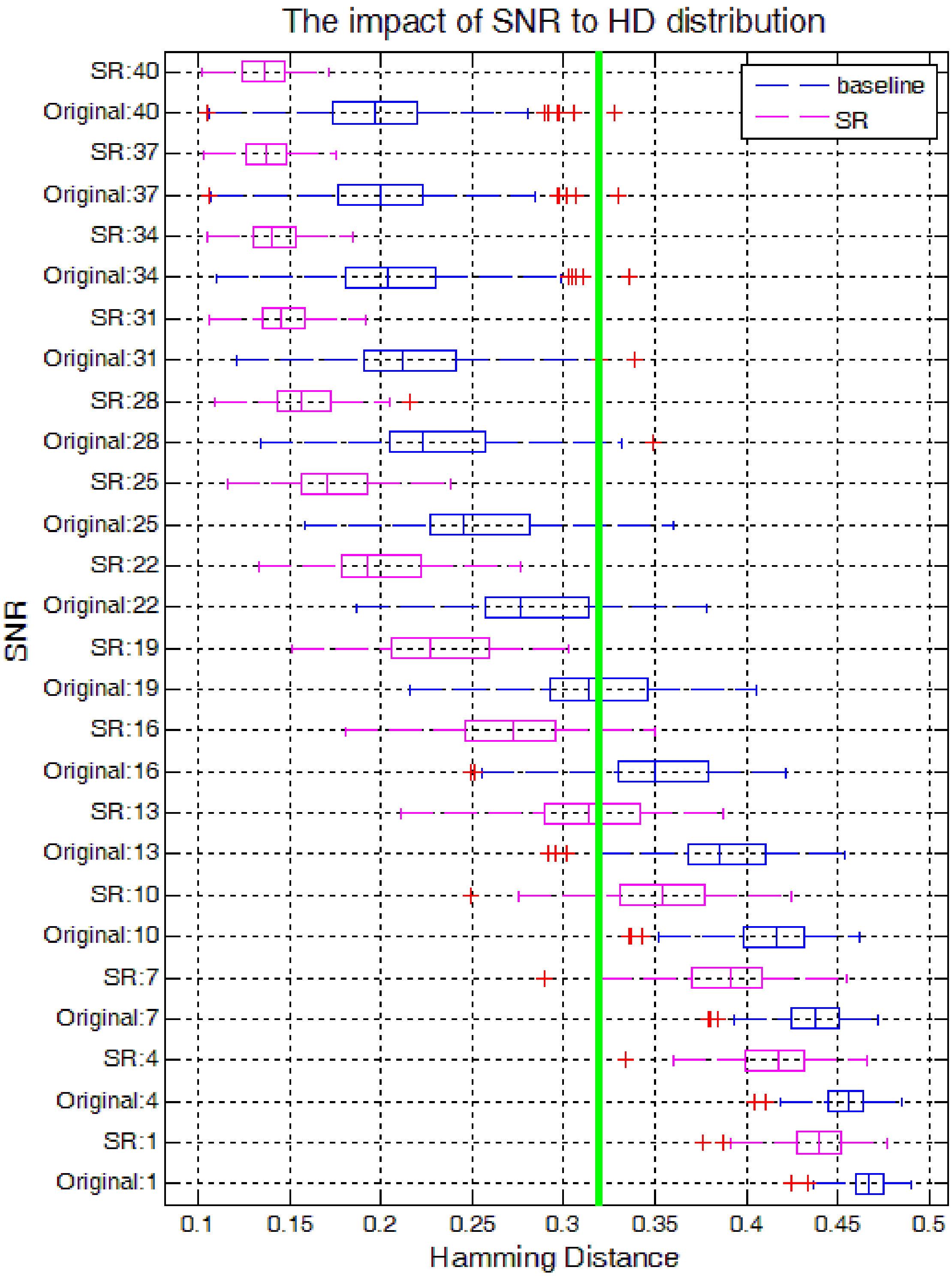


Figure 5.25: The quantitative measurement of the ability of proposed SR algorithm (LPSR30) to deal with various noise levels. The HD distributions are plotted with respect to different noise levels. For every measured SNR level, both results from “original” (baseline) and “SR” are plotted with box plot.

HD distribution on the baseline system and the SR-enhanced system.

### 5.6.2 The Limit of the Ability of Proposed SR Algorithm in Handling Defocus Blur

We also designed experiments to detect how well our proposed SR algorithms can handle image degradation caused by defocus blur in a quantitative manner. As in section 5.6.1, we use the PIER database to perform our experiments. For every iris class in the PIER database, we take all three PIER images of that class as the gallery image set, and we perform defocus blur on all three images, by filtering them with Gaussian filters. Gaussian filters, with different sizes and standard deviations (denoted as $\sigma$), can simulate the effect of defocus blur in optical devices. Using the blurred images as probe set, and matching them with the gallery set, we can get a local similarity matrix. The smallest number in the local similarity matrix is the best HD we can get for this particular class. Accumulating all the best HD values for every class in the PIER database, we can get a full distribution of the HD under this particular defocus blur. These results are called the baseline.

For our proposed SR algorithm, for every iris class, again, we filtered all three PIER images with Gaussian filters. Then, we applied our proposed SR algorithm on these three blurred PIER images to get a super-resolved image. Matching this SR image with all three images in the gallery set, we get three HDs. The smallest value among them is the best score we can get by using the SR algorithm to improve iris quality. We repeat this procedure for all classes, for various Gaussian filters (from small sigma to large sigma). We use the LPSR30 algorithm in this experiment as well.

The HD distribution of the baseline system, along with that of the SR-enhanced system, is plotted with respect to different $\sigma$ values, in Figure 5.26.

## 5.7 Conclusion

### 5.7.1 The Necessity of Image Enhancement for Long-Range Iris Recognition System

The quality of the iris pictures taken by a long-range iris acquisition device is highly constrained by optical conditions. For most of the available iris cameras on the market, the iris images are captured by infrared (IR) optical devices. This makes the goal of capturing high quality iris images from a long distance, in a less constrained setting become much more difficult. This is because of the difficulty in providing enough and evenly distributed IR illumination across the subjects' faces, while they are allowed to move freely without restriction.

The second problem of long-range iris acquisition is also related to the less constrained environment and the user's mobility. Because of these two factors, it is difficult to predict the user's position or speed when the iris images are taken, giving rise to the possibility of both motion blur and defocus blur in the outcome images.

Figure 5.29 and 5.30 shows six example sets of images. Images located in the same column denote the same set. In every column, the bottom image is the super-resolved image after ap-

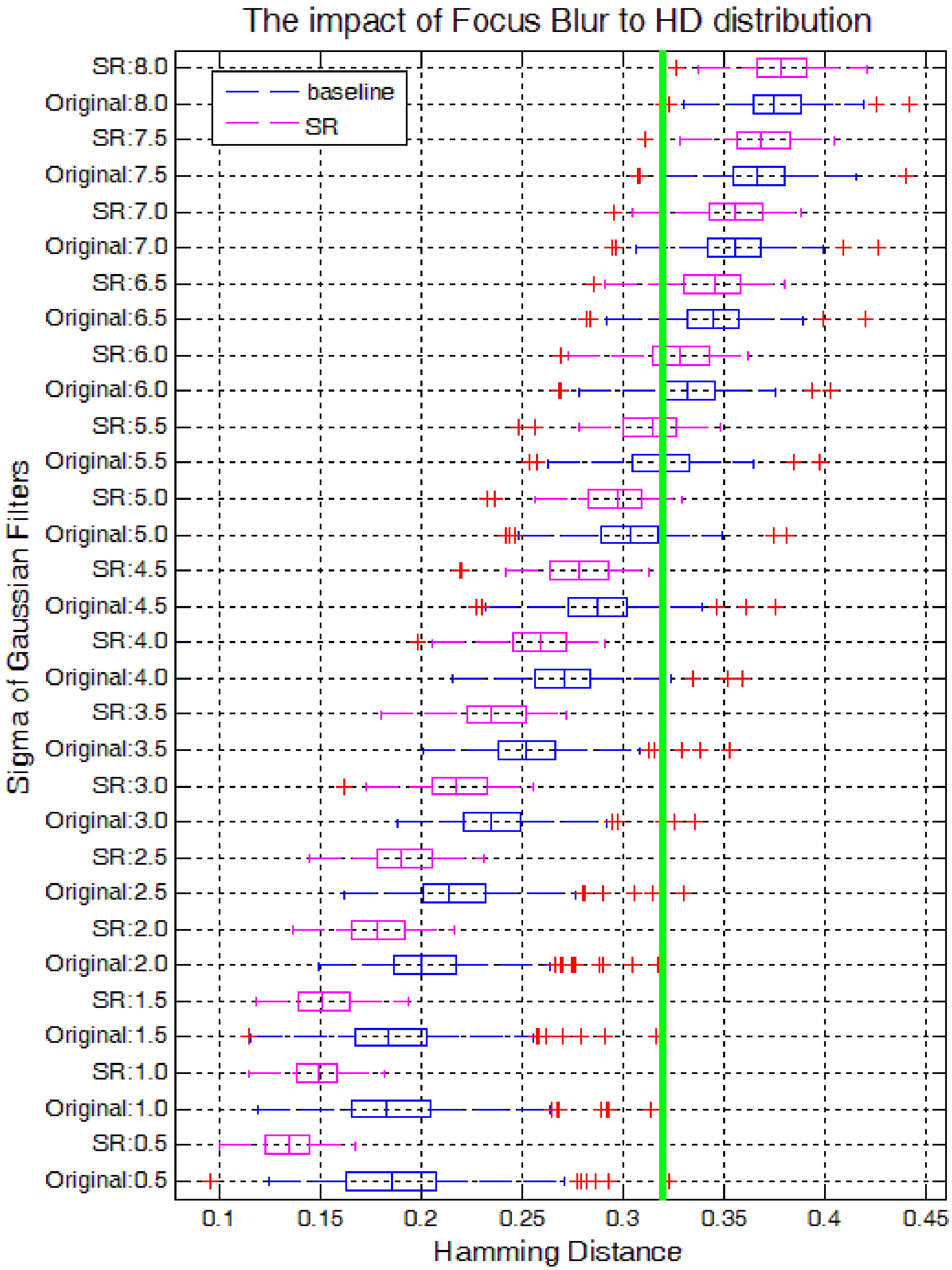


Figure 5.26: The quantitative measurement of the ability of proposed SR algorithm (LPSR30) to deal with various levels of defocus blur. The blurring effect is simulated by filtering the images with Gaussian filters with various sigma values. The HD distributions are plotted with respect to the sigma values of Gaussian filters. For every measured blurring level, both results from "original" (baseline) and "SR" are plotted with box plot.

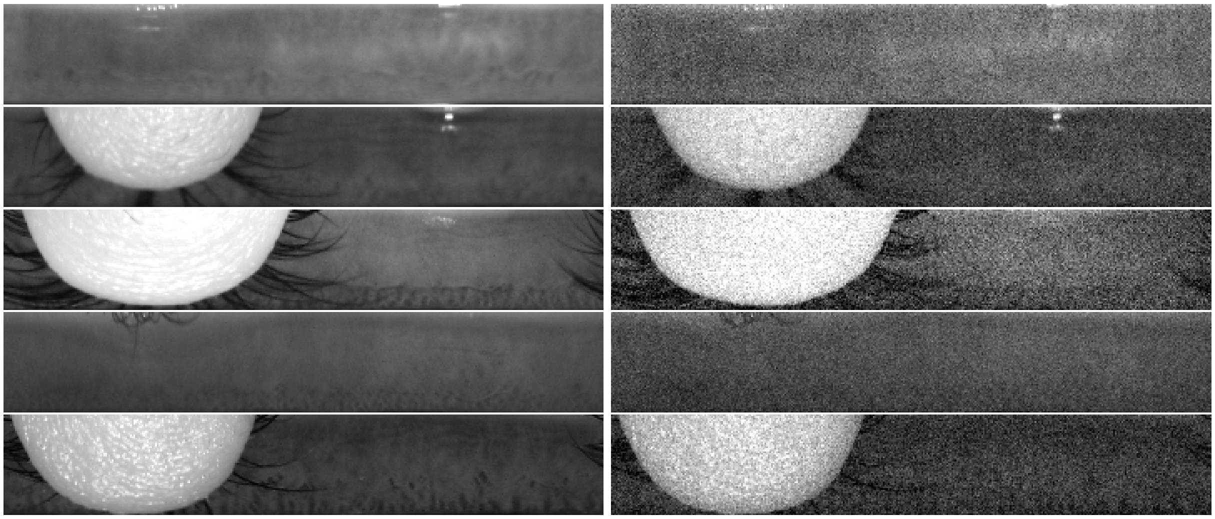


Figure 5.27: Examples of original PIER images and degraded images after they are contaminated with Gaussian white noises. Left: original clear images. Right: the degraded image after contaminated with Gaussian white noises, when SNR=16.

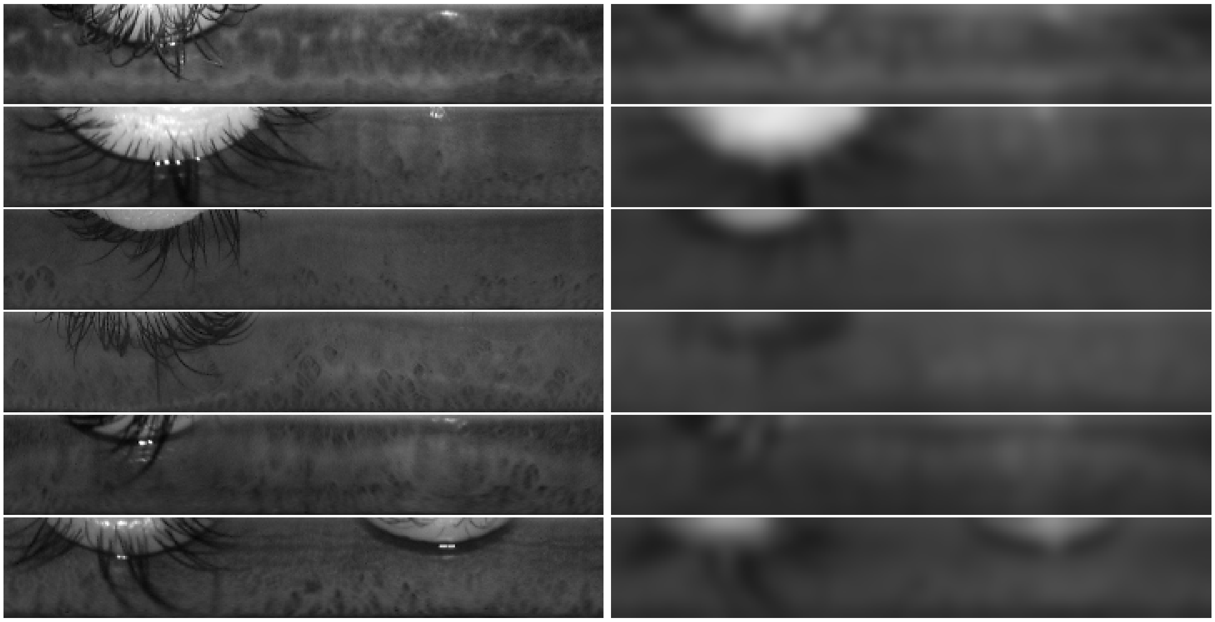


Figure 5.28: Examples of original PIER images and degraded images after they are filtered with Gaussian filters to simulate defocus blur effect. Left: original clear images. Right: the degraded image after filtered with Gaussian filter, with $\sigma$=5.

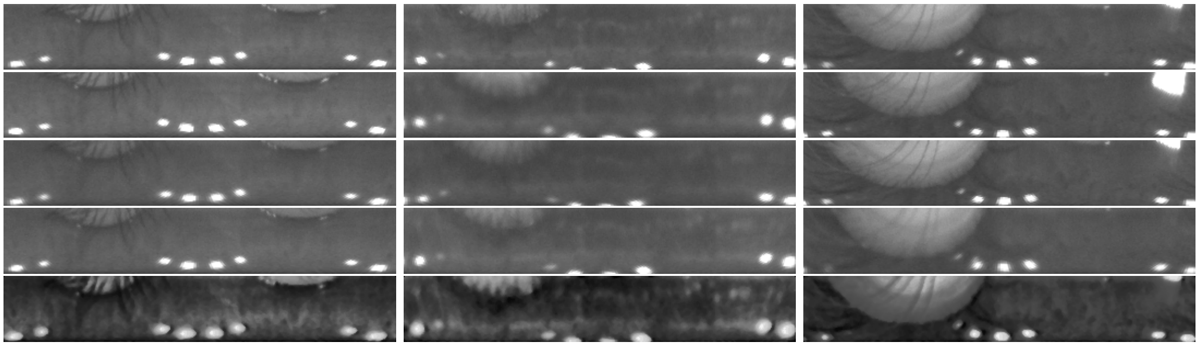


Figure 5.29: Example of the source images and the super-resolved image (with LPSR30 algorithm). Each column show an example of the four input source images and the final super-resolved image (at the bottom).

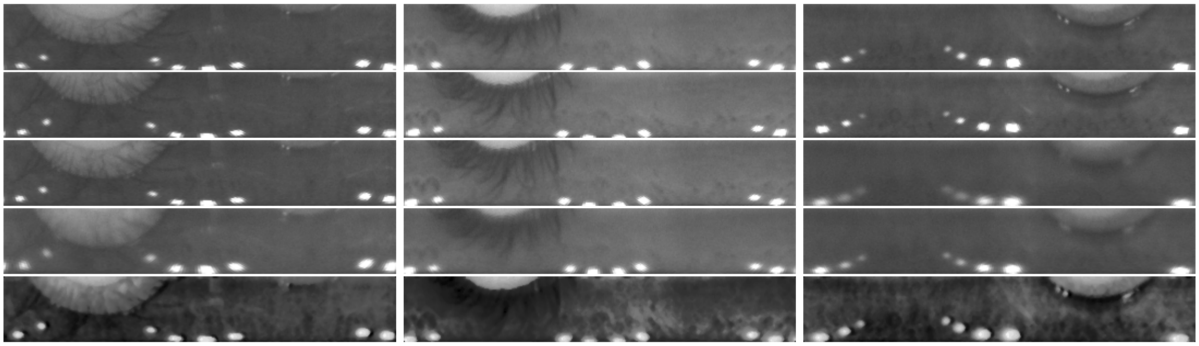


Figure 5.30: Example of the source images and the super-resolved image (with LPSR30 algorithm). Each column show an example of the four input source images and the final super-resolved image (at the bottom).

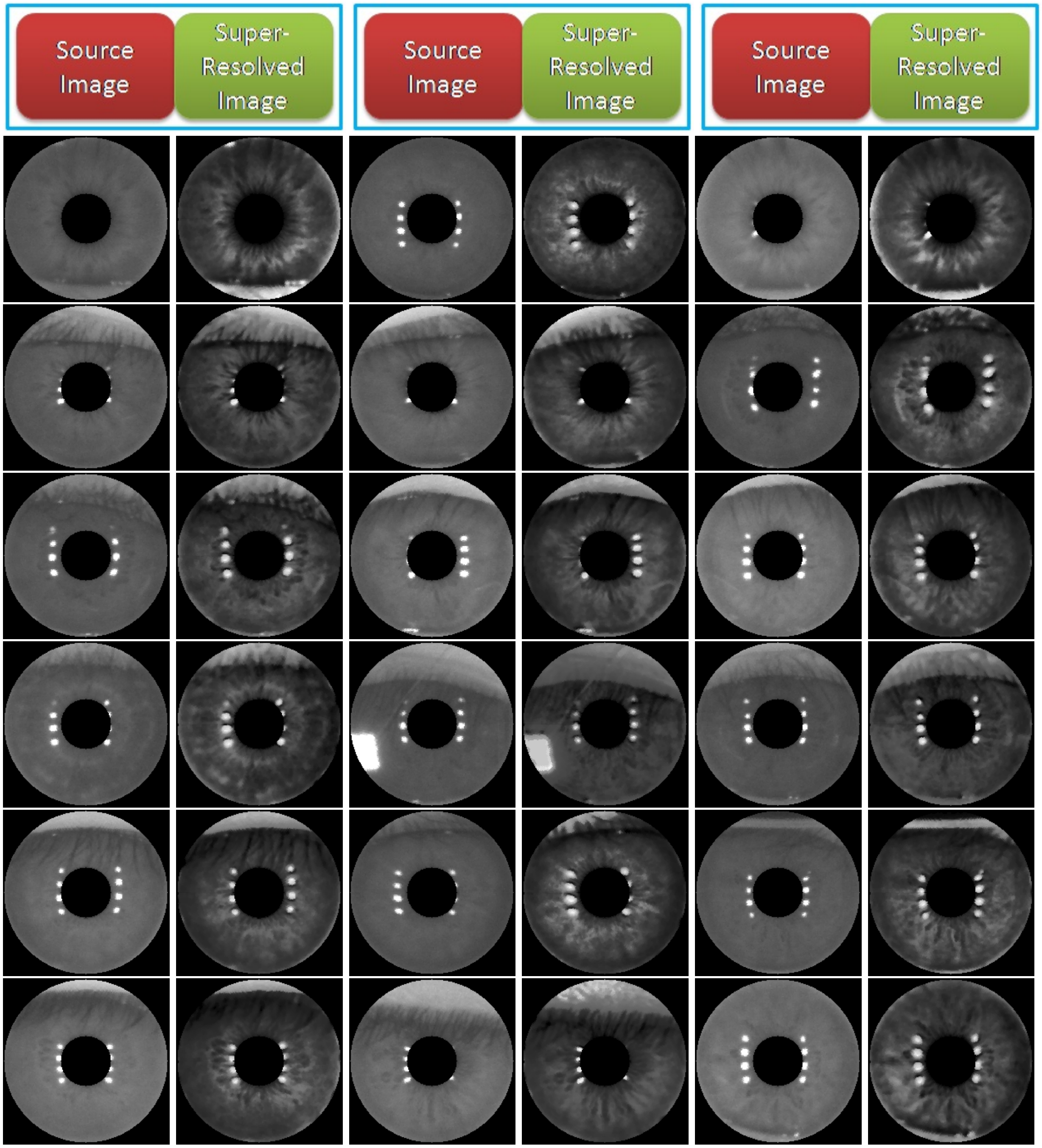


Figure 5.31: Comparison between original IOM images and their super-resolved counter part, shown in Cartesian coordinate. Column 1, 3 and 5: original IOM images in Cartesian coordinate. Column 2, 4 and 6: the super-resolved image by LPSR30 algorithm. The super-resolved images are much more clear and many details of iris texture are restored by the proposed algorithm.

plying the LPSR30 algorithm on source IOM images. Figure 5.31 shows original IOM images and their super-resolved counter parts, in the Cartesian coordinate. We can see that the proposed algorithm can successfully reveal and enhance the details of the iris texture, so the iris features are more prominent and can be used to improve the iris recognition rate in later stages.

### 5.7.2 Novelty of the Proposed Method

Most existing super-resolution and image restoration algorithms are global methods, as described in section 5.3. However, for image restoration for iris images, it seems like our method, which is mainly based on local patches, is more suitable and effective because local deformation is an easily observable phenomenon for almost all iris images in the polar domain. As stated in section 5.4, iris images in the polar domain suffer from local deformation because of the variation in iris and pupillary boundary estimation, variations of pupil size, and the off-axis viewing angle of the subjects. Therefore, if we only apply the super-resolution algorithm in the global sense, the details of the local iris texture may not be recovered completely, which can cause the final image to be still blurry and noisy.

By applying the proposed Local Patch-based Super-Resolution (LPSR) algorithm, as shown in our large-scale experimental results in section 5.5, it is demonstrated that LPSR method can successfully enhance the iris recognition performance, compared to global super-resolution method, which conforms with the observation of local deformation of iris texture.

### 5.7.3 Exploring the Limitation of the Proposed Method in Simulated Image Degradation

By reading and analyzing the plots in Figure 5.25 and 5.26, we can learn the limits of our proposed LPSR algorithm, in terms of the percentage of the authentic comparison score that can be kept under a reasonable range, with respect to different levels of noise and defocus blur. In literature, the best threshold of the HD which achieves the optimal compromise between False Accept Rate (FAR) and False Reject Rate (FRR) is 0.32 [2, 20]. Therefore, in this study, we also use HD=0.32 as our threshold to test the limit of our algorithm.

Figure 5.25 shows HD distribution of authentic comparison, under different levels of noise contamination. The HD distribution is plotted by pairs, and each pair contains the result based on the matching with the original degraded images (marked with blue color) and the super-resolved images (marked with purple color). The green line indicates the optimal HD threshold, which is 0.32. Figure 5.26 use the same format to plot the experimental result for different levels of defocus blur.

The first observable result in Figure 5.25 and 5.26 is that for each level of noise contamination and defocus blur, the HD distribution of authentic comparisons using super-resolved images are globally shifted towards zero, compared with using the original degraded images. Note that this observation holds true for all distributions, no matter if they are under or above the threshold (the green line). This result shows our proposed LPSR algorithm can successfully increase the iris recognition performance, in both noise-contaminated images and blurred images.

The second result we observe is that, for noise-contaminated images (Figure 5.25), the majority of the HD scores are under the optimal threshold (HD=0.32), as long as the SNR is higher or equal to 16 dB. For images with defocus blur (Figure 5.26), the majority of HD scores are under the optimal threshold (HD=0.32), as long as the $\sigma$ value, the standard deviation, of the spatial Gaussian filters is under 5. Figure 5.27 and 5.28 show example iris images that are degraded by Gaussian white noise and the defocus blur, at SNR=16dB and $\sigma = 5$, respectively. It is amazing that under such bad imaging condition, the proposed LPSR algorithm is still able to enhance the image quality, so the recognition performance can be maintained.

### 5.7.4 Performance of the Proposed Method in Large-Scale Iris Recognition

In section 5.5, we show the results of iris recognition performance on large-scale database. The experimental results can be summarized as the following:

1. The proposed LPSR algorithms outperform all existing SR algorithms, and the baseline system.
2. In the setting of “SR vs. PIER”, LPSR20 performs best.
3. In the setting of “SR vs. LG”, LPSR30 performs best.
4. In the setting of “SR vs. IOM”, LPSR2 performs best.
5. In the setting of “SR vs. SR”, LPSR30 performs best.

Overall, the system structured “SR vs. SR” achieves the best recognition performance. This result has significant meaning. It means that we can enroll iris images by using super-resolved images, instead of using the PIER or LG images. This means that the image quality of the super-resolved image is good enough for large-scale experiments. Also, from a system perspective, it means that we can acquire iris images by using only one iris acquisition device, in a less constrained environment. This greatly mitigates the difficulty of the iris acquisition process and greatly improves the overall usability.

Anti-terrorism can benefit from the proposed system structure, too. We can combine the long-range iris acquisition system with a face detector/tracker; when there is a terrorist suspect detected by the face detector, we can immediately acquire the iris image from a remote location and enroll it into the nationwide iris database. Or if the iris database is already established, we can immediately match the iris images of the suspect with the database from a remote location, providing more time for early screening and alert.

# Chapter 6

# Conclusions and Future Work

The goal of this dissertation is to enhance the iris recognition performance of the long-range iris acquisition system by capitalizing on the video stream of irises. This goal can be achieved through improvements in every stage of the system: iris segmentation, iris occlusion estimation, iris mask refinement, and multiple image fusion. All of the previous chapters have explained the proposed methods for each stage, and the experimental results show that these proposed methods are indeed effective, efficient, and robust. This dissertation is the first of its kind to tackle multi-frame iris fusion (super-resolution) for image enhancement and robust long range iris recognition.

## 6.1 Conclusions

This sub-section summarizes and discusses the findings from the previous chapters. These include findings specific to each stage of the iris recognition system, as well as findings about the nature of the iris texture in general.

- For iris segmentation on iris images acquired by the long-range iris acquisition system:
  - We found that the randomly distributed specular reflections are indeed a challenging problem. The traditional iris segmentation algorithms are mostly based on strong circular edge feature detection. For example, the correlation-based method looks for strong response from partial circular contour filters, and the Hough Transform method is heavily based on edge map too. In such cases, eliminating specularity through image processing algorithms and decreasing the reliance on edge information becomes the key to successful iris segmentation.
  - We found that the median filter is suitable to eliminate specular reflection in iris images. Due to the size of the specular reflections, which are usually relatively small, median filtering does not destroy the contour of the pupil or the iris. However, inpainting a proper intensity value before median filtering is important. Median filtering cannot render smooth images if incorrect values are inpainted into the specularity.
  - Median filtering softens the edge information in images. If we still rely on continuous circular edge information to perform segmentation, the success rate decreases

significantly. That is the reason why the existing segmentation algorithms fail even when we pre-process the iris image with the median filter. Instead, we should use sparse samples on the hypothesis boundaries to estimate them.

  - Our proposed algorithm, which is based on specularity removal and sparse sampling on the boundary, has proved to be more robust and accurate compared to existing methods. It can deal with iris images which can be degraded due to one of the following reasons: (1) contaminated by multiple strong specular reflections (2) partially occluded by other artifacts, such as eyeglasses frames, eyelids or eyelashes (3) degraded by low intensity contrast and low ambient lighting (4) degraded by motion blur, defocus blur and noise. Our proposed algorithm is also more efficient than the Hough Transform method.

- For automatic iris mask generation:

  - In literature, there is virtually no solid and thorough research focused on mask generation. It was always treated as an easy problem during the iris segmentation stage. Usually people use simple features and pre-defined rules to create the iris masks. In this thesis, we formulate the problem from a pattern recognition perspective, and translate it into a pixel-wise two-class classification problem in the 2D plane.
  - We experimented with different features in order to see which image feature set works best for this problem. The features we tried are all described in section 3.4. We found that the feature set IG (pixel intensity and Gabor response) is the optimal feature combination for the GMM to learn discriminative features, so it can differentiate the iris texture from the occlusion. Later, we further optimized the Gabor Filter Bank(GFB) in order to discover the optimal GFB which has the most discriminative power for detecting true iris texture, for the purpose of enhancing the large-scale iris recognition rate.
  - The optimization method used in our experiment is Simulated Annealing (SA). SA has the ability to escape from the local minimum and discover the global optimal solution. We used SA to optimize the GFB with up to 10 filters, for both the ICE1 and ICE2 NIST databases. We empirically found that the performance reaches its peak when we use 6 or 7 Gabor filters. After that, the performance decreases.
  - The choice of using the GMM instead of other pattern recognition algorithms is appropriate for our problem and is supported by the experimental results. This is because the GMM has the ability to model the nature of the appearance of both iris texture and the occlusion artifacts. As shown in chapter 3, various objects occlude the iris, including eyelids, eyelashes, eyeglasses frames, and specular reflection. Therefore, by using a multi-modal approach, it is easier for us to model the nature of the artifacts.
  - From the system perspective, the GMM is a good choice with low computational operational complexity. Although GMMs may take a longer time during the learning phase, they are very fast and efficient during classification. The performance of iris recognition can be enhanced to levels similar to those of manually created masks.

Table 6.1: Large-scale iris recognition performance comparison, in terms of three different error metrics, as well as the average of them, among the four new system structures we proposed in chapter 5, compared to the baseline.

| New System Structure | PIER vs. IOM | SR vs. PIER | SR vs. LG | SR vs. IOM | SR vs. SR |
|---|---|---|---|---|---|
| FRR@FAR=0.1% | 2.1% | 1.5% | 0.0% | 0.3% | 0.0% |
| EER | 1.2% | 0.9% | 0.0% | 0.3% | 0.0% |
| FRR@FAR=0% | 3.6% | 2.1% | 1.8% | 0.3% | 0.3% |
| Average | 2.3% | 1.5% | 0.6% | 0.3% | 0.1% |

Table 6.2: Large-scale iris recognition performance comparison, in terms of three different error metrics, as well as the average of them, among the configuration “PIER vs. PIER”, “LG vs. LG”, and “SR vs. SR”.

| New System Structure | PIER vs. PIER | LG vs. LG | SR vs. SR |
|---|---|---|---|
| FRR@FAR=0.1% | 0.0% | 0.0% | 0.0% |
| EER | 0.0% | 0.0% | 0.0% |
| FRR@FAR=0% | 0.0% | 0.0% | 0.3% |
| Average | 0.0% | 0.0% | 0.1% |

- For automatic iris mask refinement:
  - Some parts of the iris region contain more complex tissue structure than other parts. This fact is easily observed by simply inspecting several iris images of different classes. The complex regions have higher response to Gabor filters. If we can successfully identify those regions and emphasize them during iris matching, the iris recognition performance improves. This hypothesis was verified in our experiments.
  - The final iris mask derived from the proposed algorithm is more conservative than the manually created masks. In other words, the refined iris mask is a super set of the manual masks, where only the discriminative region of the iris is highlighted as the true iris region. Our experiment proved that less is better; focusing on smaller but more discriminative regions gives higher recognition rates.
  - The performance when using a single image with a refined mask is relatively the same as using multiple training images. This is a huge advantage from a system perspective. This will greatly reduce the run-time of iris matching, which is key for a large-scale iris recognition, like searching in regional or national databases.
- For iris image enhancement by Super-Resolution algorithm:
  - Super-Resolution (SR) technology has been successfully applied in the fields of computer vision and image processing. However, in the field of iris image processing, to the best of the author’s knowledge, this thesis is the first in-depth study about how to apply SR on iris images in the polar domain, and how much performance we can gain from it.
  - We have proposed a novel local patch-based and quality-weighted SR algorithm. This

idea is inspired from the observation that iris images in the polar domain suffer from serious local deformation. Our experiments show that even the simplest form of our patch-based algorithm outperforms the global SR algorithm, which suggests that local deformation is an important factor to consider in order to restore high quality iris images.

- In this thesis, four new system structures have been proposed in order to utilize the SR in the traditional IOM system. In each of the four system structures, the proposed SR algorithm can enhance the system performance. Table 6.1 shows the system performance enhancement by each of the four system structures, in terms of the three error metrics. We found that the scenario of matching super-resolved irises "SR vs. SR" achieves the best performance. This scenario uses all images captured of the person passing through the IOM to synthesize a single iris which is used for enrollment and matching. Since the source images in the gallery are different from those in the probe, this result suggests that our proposed SR algorithm is robust and stable, and is able to restore the low quality IOM images from different sessions to a more stable and consistent representation. Table 6.2 shows the system performance of "SR vs. SR", compared with "PIER vs. PIER" and "LG vs. LG", which are two upper bounds of the recognition performance in our case. We can see that after applying our proposed SR algorithm, the system performance of "SR vs. SR" approaches very closely to that of "PIER vs. PIER" and "LG vs. LG".
- We also performed experiments to explore the limit of our proposed SR algorithm in terms of the noise and defocus blur it can handle. The experimental results show that our proposed SR algorithm can handle Gaussian white noise up to 16dB, and defocus blur up to $\sigma = 5$.
- From a system perspective, there is a trade-off between accuracy and speed. If we aim for speed during testing stage, we can choose the system structure of "SR vs. IOM", because the SR can be operated during enrollment. If we prefer system accuracy to the speed, we can choose the system structure of "SR vs. SR", which requires the SR operation in both enrollment and testing stages.
- From a system perspective, the success of the "SR vs. SR" mode suggests that we can enroll and match iris images using super-resolved IOM images. This has potential for many defense and national security applications in matching a potential terrorist from a distance. Similarly for law enforcement applications, the ability to identify a criminal without getting too close has significant life saving implications to the officers patrolling and guarding our streets.

## 6.2 Possible Future Work

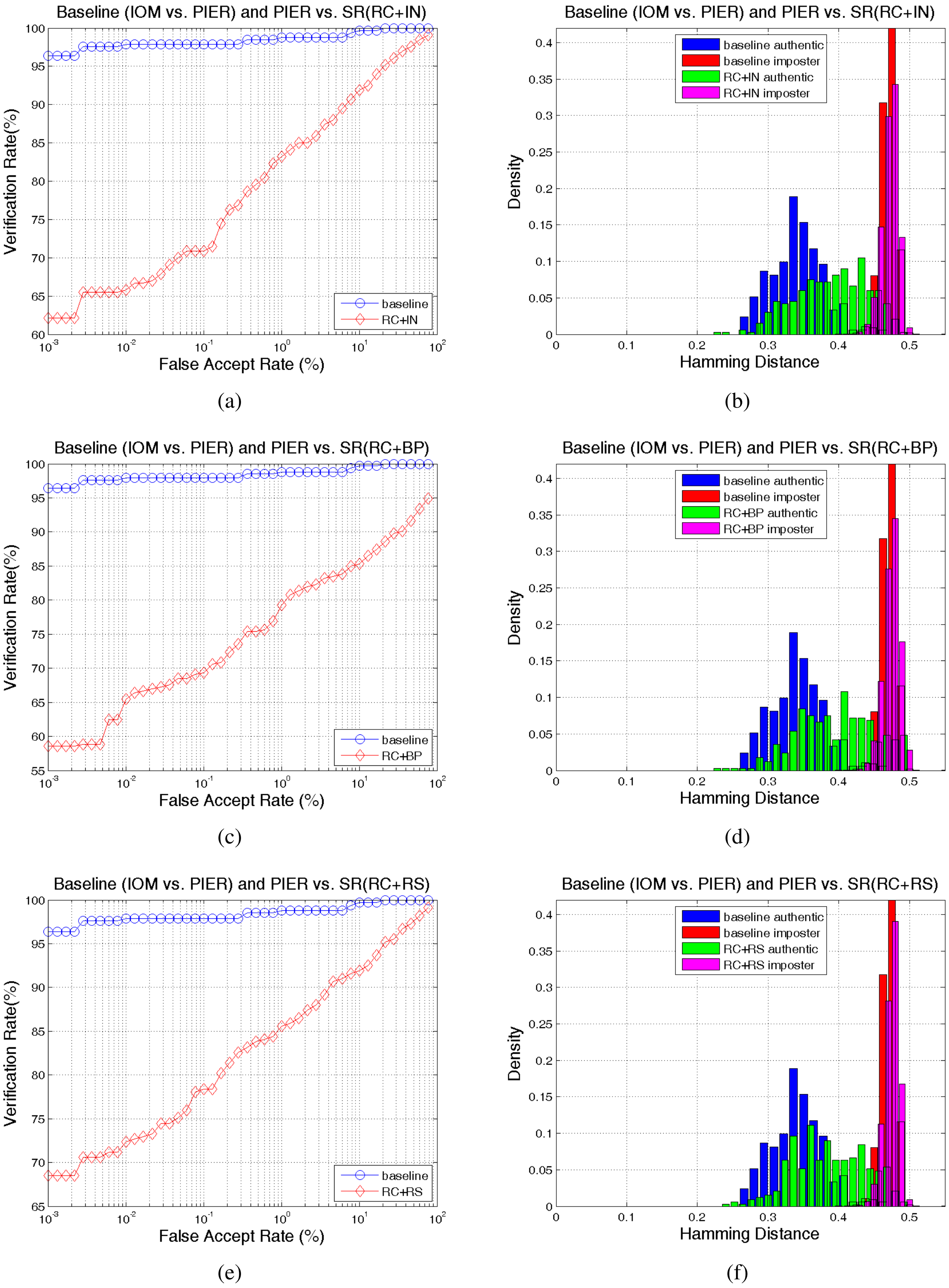
We plan to extend the work on automatic iris mask generation based on GMM of the iris texture to other generative machine learning methods such as the Probabilistic Latent Semantic Analysis (pLSA) and the Latent Dirichlet Allocation (LDA), which are two machine learning methods based on probabilistic graphical models [70]. They were originally used in information retrieval, but later widely applied to source separation, speech processing, machine translation, and computer vision.

From a system perspective, it is desirable to build a system which can acquire iris images from even farther away, without the constraint that users have to walk through a portal. Such a system would have to be equipped with a high resolution camera and a telephoto lens. The camera may need Pan-Tilt-Zoom (PTZ) capability to track the object with three degrees of freedom. The system would need to have fast and accurate face/eye tracking software in order to detect and track subjects efficiently. Additionally, tackling the illumination problem is challenging for long range acquisition due to eye safety considerations.

# Appendix A

# Experimental Figures for SR vs. PIER

All of the experimental results figures (ROC curves and histogram distribution of Hamming distance) of the system architecture SR vs. PIER are listed here. Under this architecture, the LPSR 20 achieves lowest recognition errors compared to other methods.



(a)

(b)

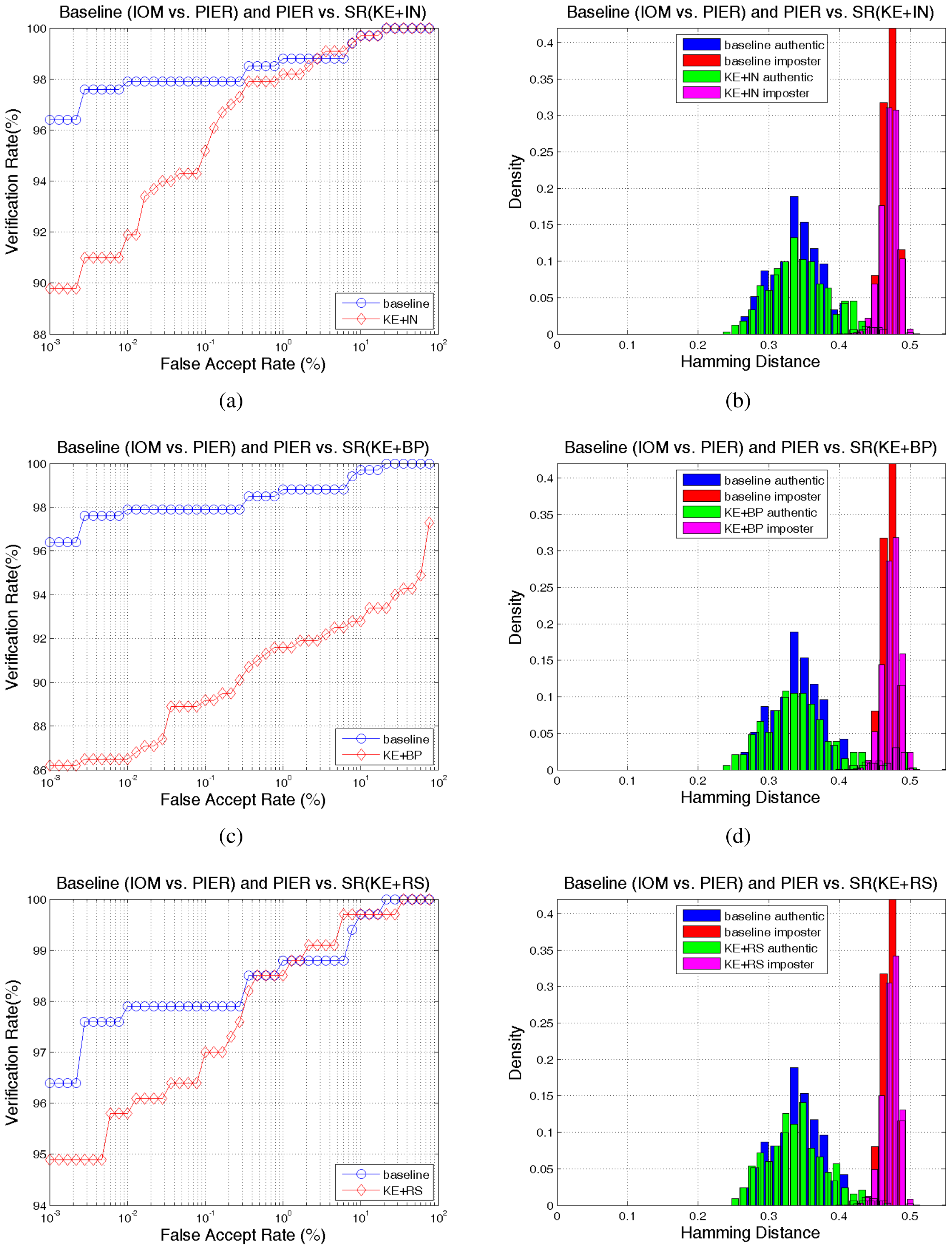
(c)

(d)

(e)

(f)

Figure A.1: Performance comparison: baseline scenario 1 vs. SR enhanced system (RC). (a) ROC curves of RC+IN; (b) HD histogram distribution of RC+IN; (c) ROC curves of RC+BP; (d) HD histogram distribution of RC+BP; (e) ROC curves of RC+RS; (f) HD histogram distribution of RC+RS



(a)

(b)

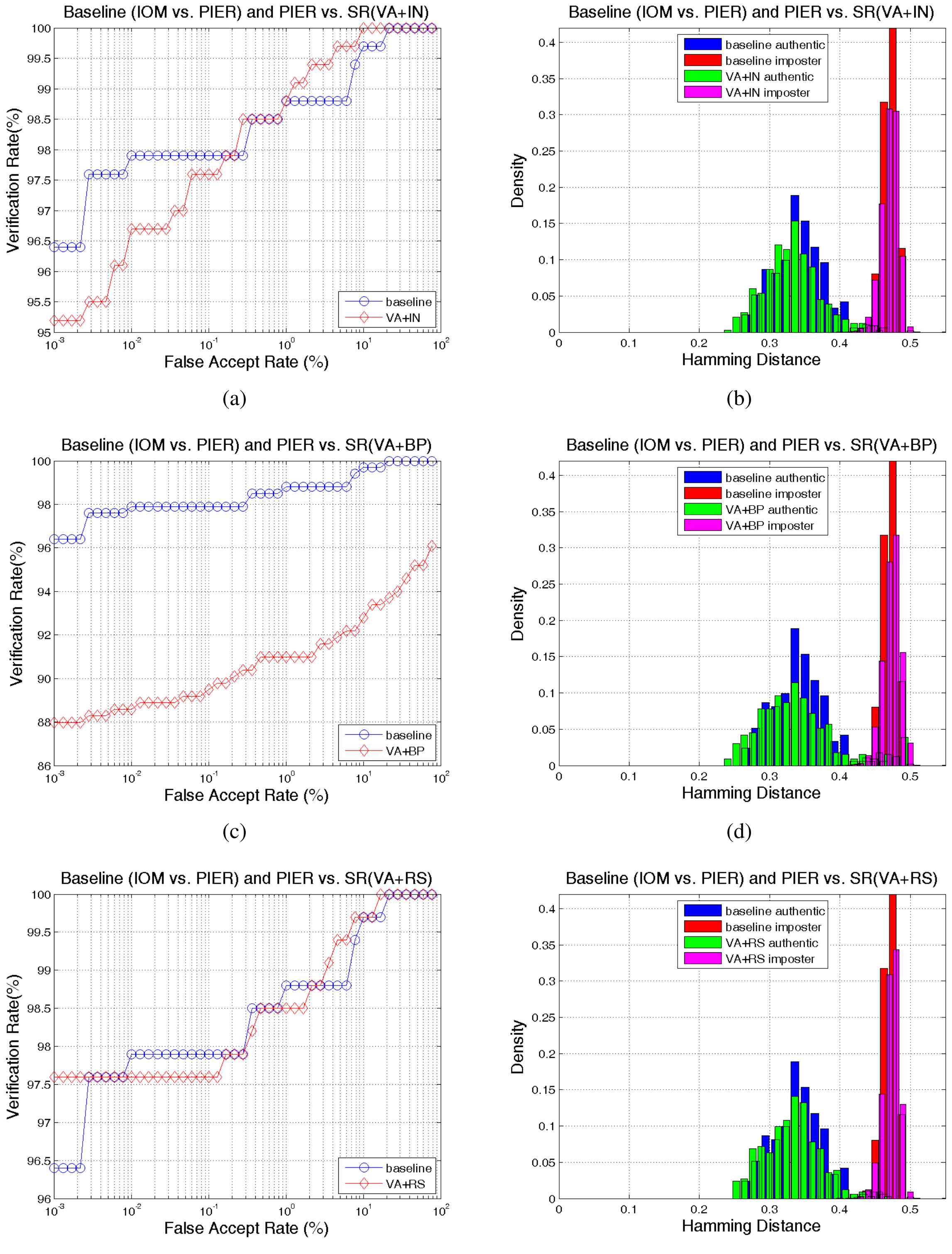
(c)

(d)

(e)

(f)

Figure A.2: Performance comparison: baseline scenario 1 vs. SR enhanced system (KE). (a) ROC curves of KE+IN; (b) HD histogram distribution of KE+IN; (c) ROC curves of KE+BP; (d) HD histogram distribution of KE+BP; (e) ROC curves of KE+RS; (f) HD histogram distribution of KE+RS



(a)

(b)

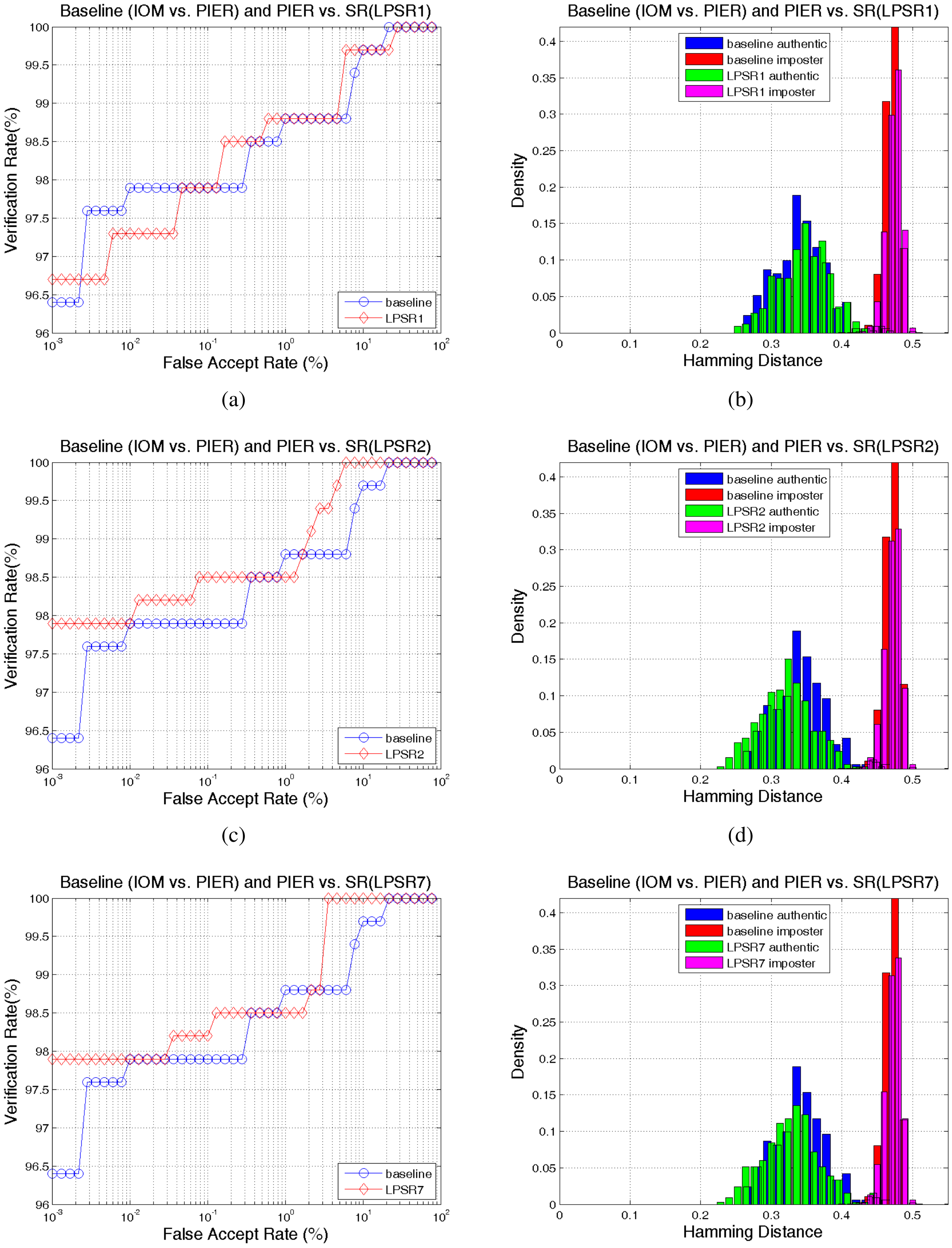
(c)

(d)

(e)

(f)

Figure A.3: Performance comparison: baseline scenario 1 vs. SR enhanced system (VA). (a) ROC curves of VA+IN; (b) HD histogram distribution of VA+IN; (c) ROC curves of VA+BP; (d) HD histogram distribution of VA+BP; (e) ROC curves of VA+RS; (f) HD histogram distribution of VA+RS

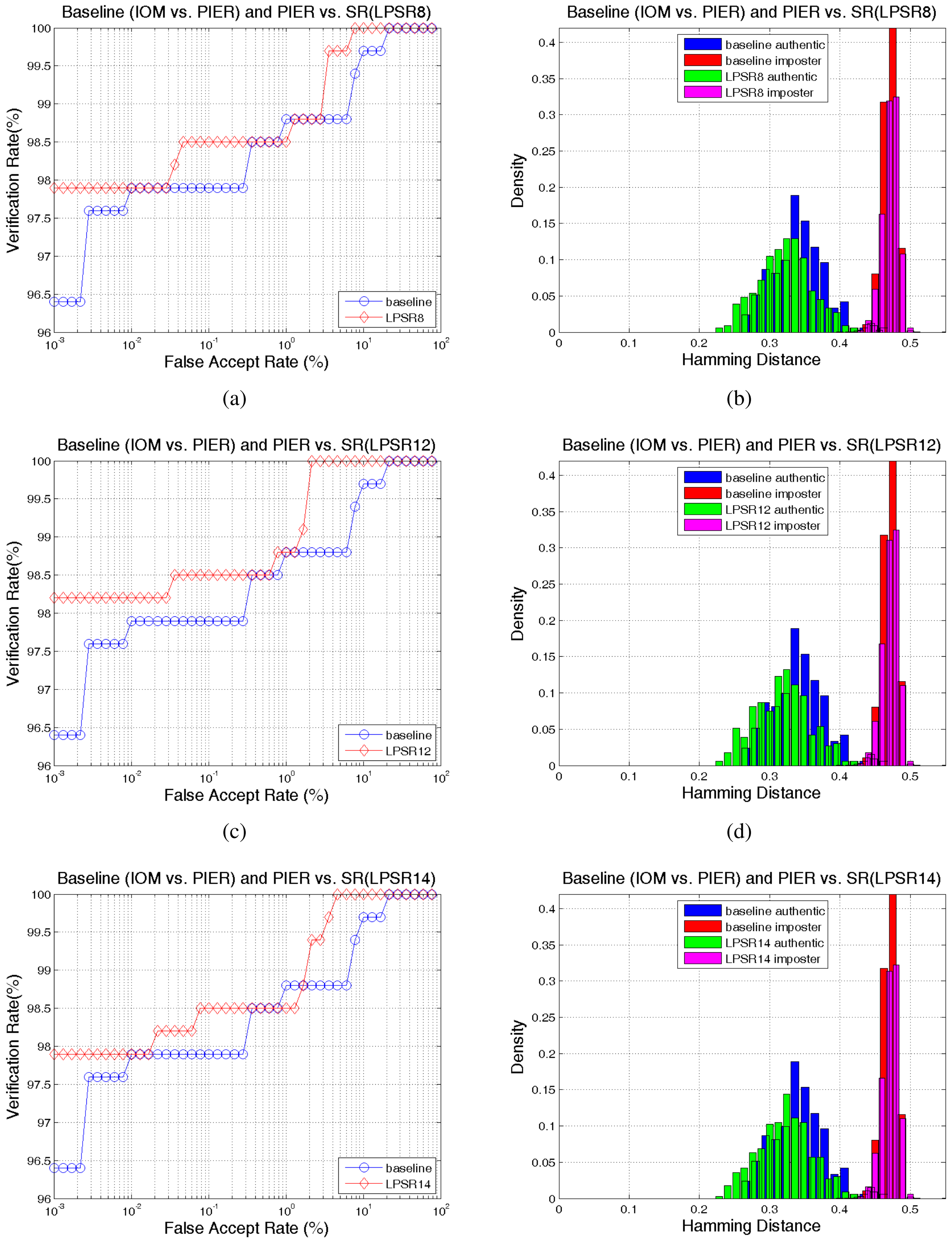


(a) (b)

(c) (d)

(e) (f)

Figure A.4: Performance comparison: baseline scenario 1 vs. SR enhanced system (LPSR1, 2 and 7). (a) ROC curves of LPSR1; (b) HD histogram distribution of LPSR1; (c) ROC curves of LPSR2; (d) HD histogram distribution of LPSR2; (e) ROC curves of LPSR7; (f) HD histogram distribution of LPSR7



(a)

(b)

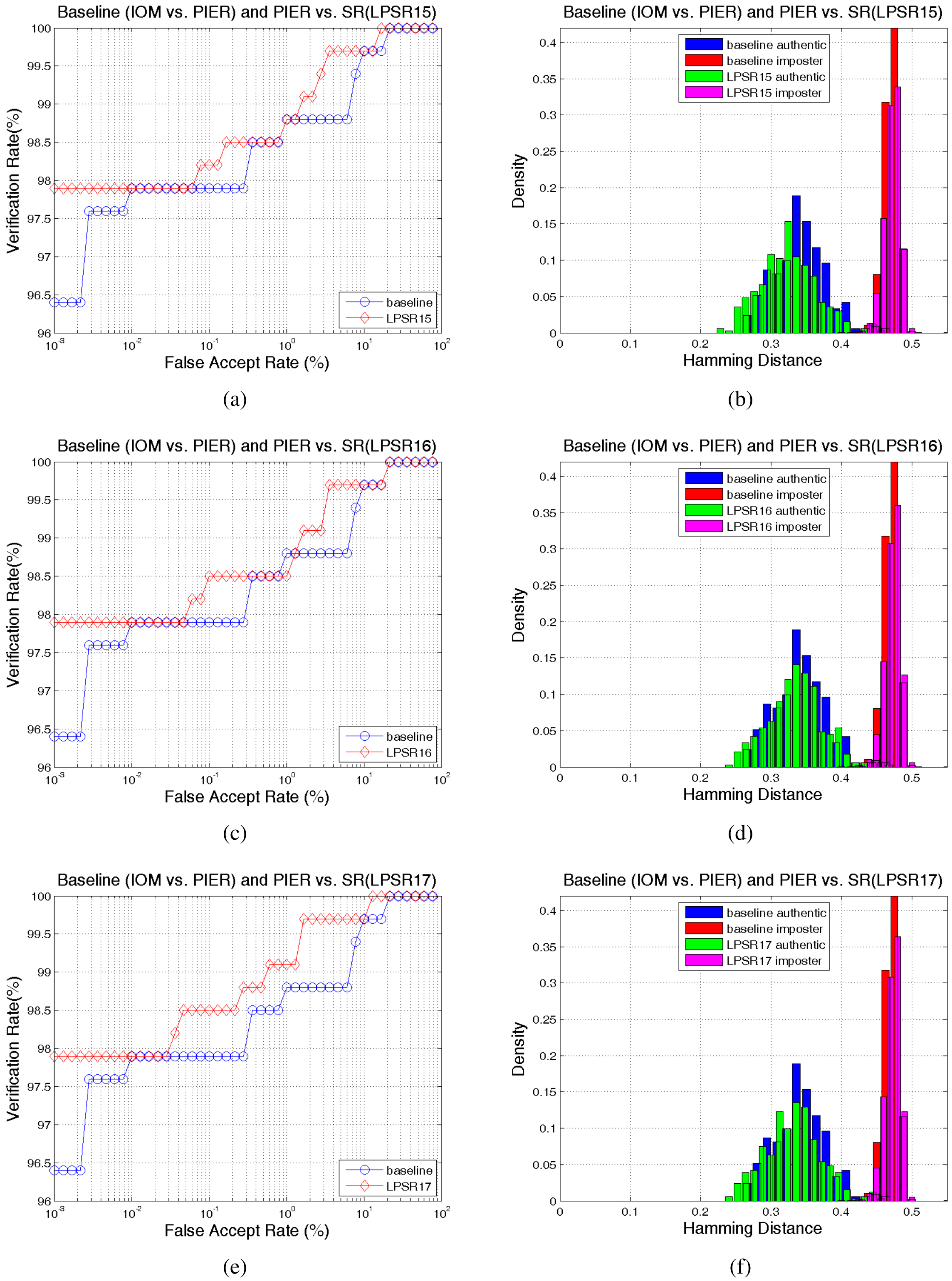
(c)

(d)

(e)

(f)

Figure A.5: Performance comparison: baseline scenario 1 vs. SR enhanced system (LPSR8, 12 and 14). (a) ROC curves of LPSR8; (b) HD histogram distribution of LPSR8; (c) ROC curves of LPSR12; (d) HD histogram distribution of LPSR12; (e) ROC curves of LPSR14; (f) HD histogram distribution of LPSR14



(a)

(b)

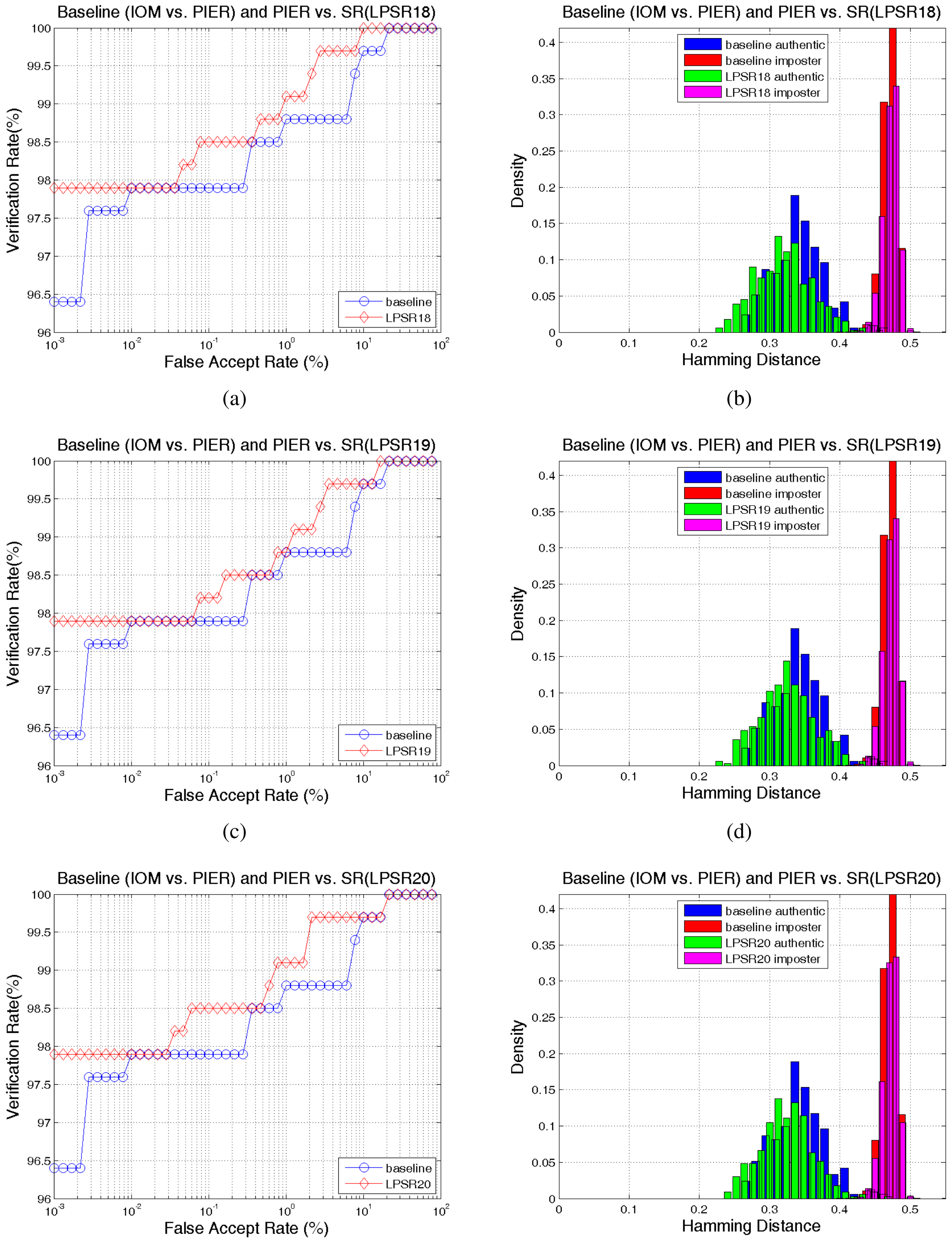
(c)

(d)

(e)

(f)

Figure A.6: Performance comparison: baseline scenario 1 vs. SR enhanced system (LPSR15, 16 and 17). (a) ROC curves of LPSR15; (b) HD histogram distribution of LPSR15; (c) ROC curves of LPSR16; (d) HD histogram distribution of LPSR16; (e) ROC curves of LPSR17; (f) HD histogram distribution of LPSR17



(a)

(b)

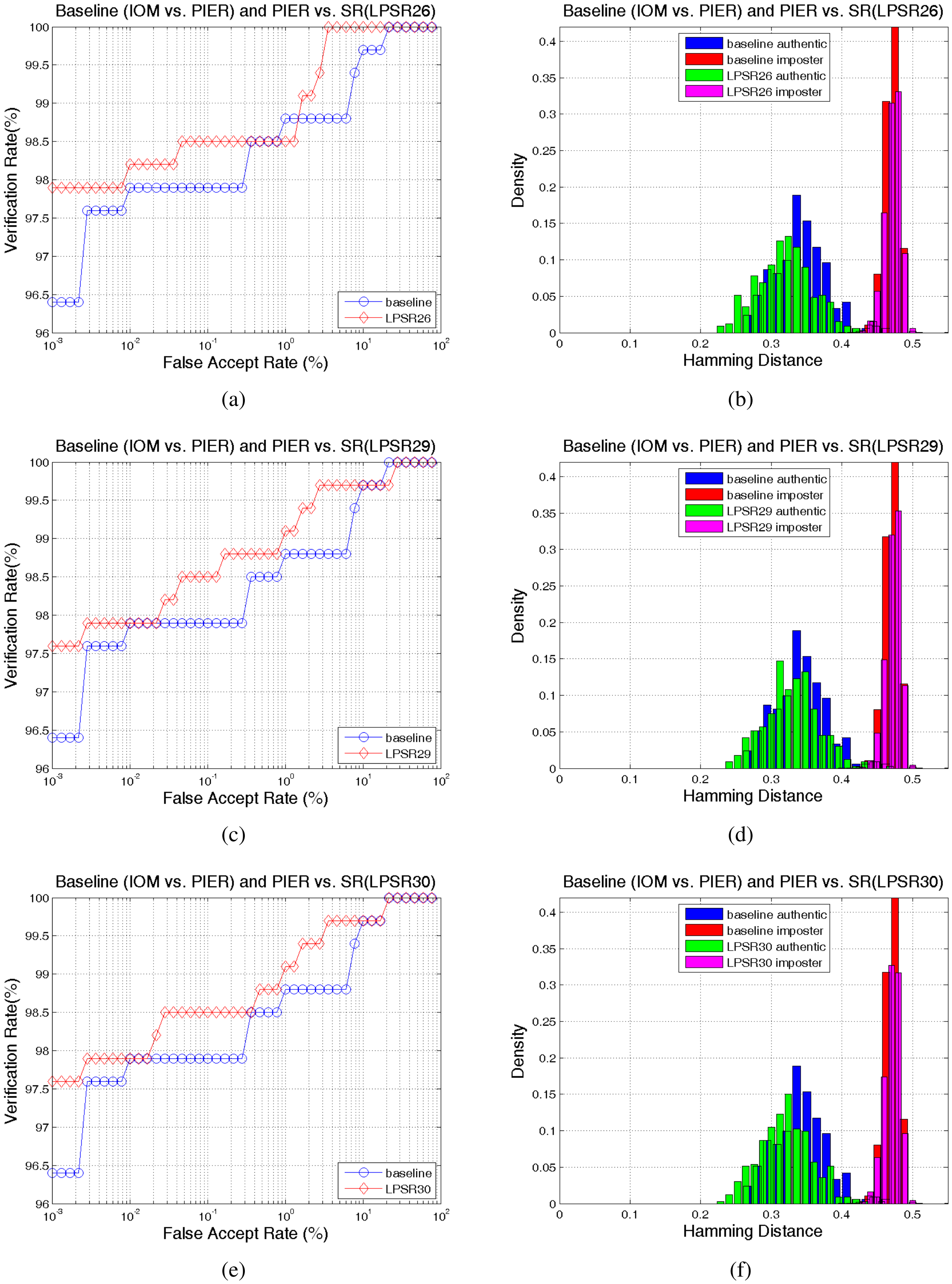
(c)

(d)

(e)

(f)

Figure A.7: Performance comparison: baseline scenario 1 vs. SR enhanced system (LPSR18, 19 and 20). (a) ROC curves of LPSR18; (b) HD histogram distribution of LPSR18; (c) ROC curves of LPSR19; (d) HD histogram distribution of LPSR19; (e) ROC curves of LPSR20; (f) HD histogram distribution of LPSR20



(a)

(b)

(c)

(d)

(e)

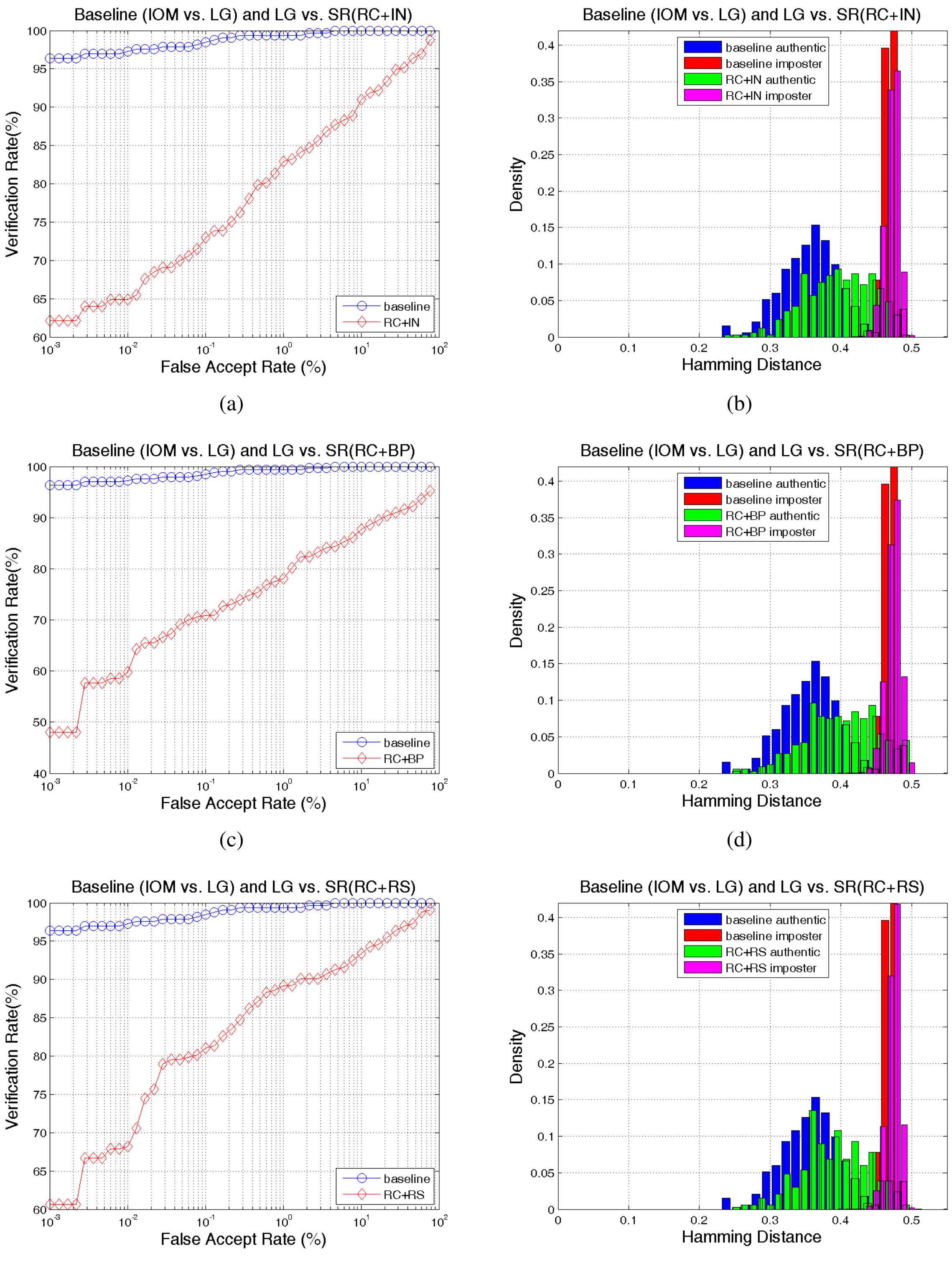
(f)

Figure A.8: Performance comparison: baseline scenario 1 vs. SR enhanced system (LPSR26, 29 and 30). (a) ROC curves of LPSR26; (b) HD histogram distribution of LPSR26; (c) ROC curves of LPSR29; (d) HD histogram distribution of LPSR29; (e) ROC curves of LPSR30; (f) HD histogram distribution of LPSR30

# Appendix B

# Experimental Figures for SR vs. LG

All of the experimental results figures (ROC curves and histogram distribution of Hamming distance) of the system architecture SR vs. LG are listed here. Under this architecture, the LPSR 30 achieves lowest recognition errors compared to other methods.

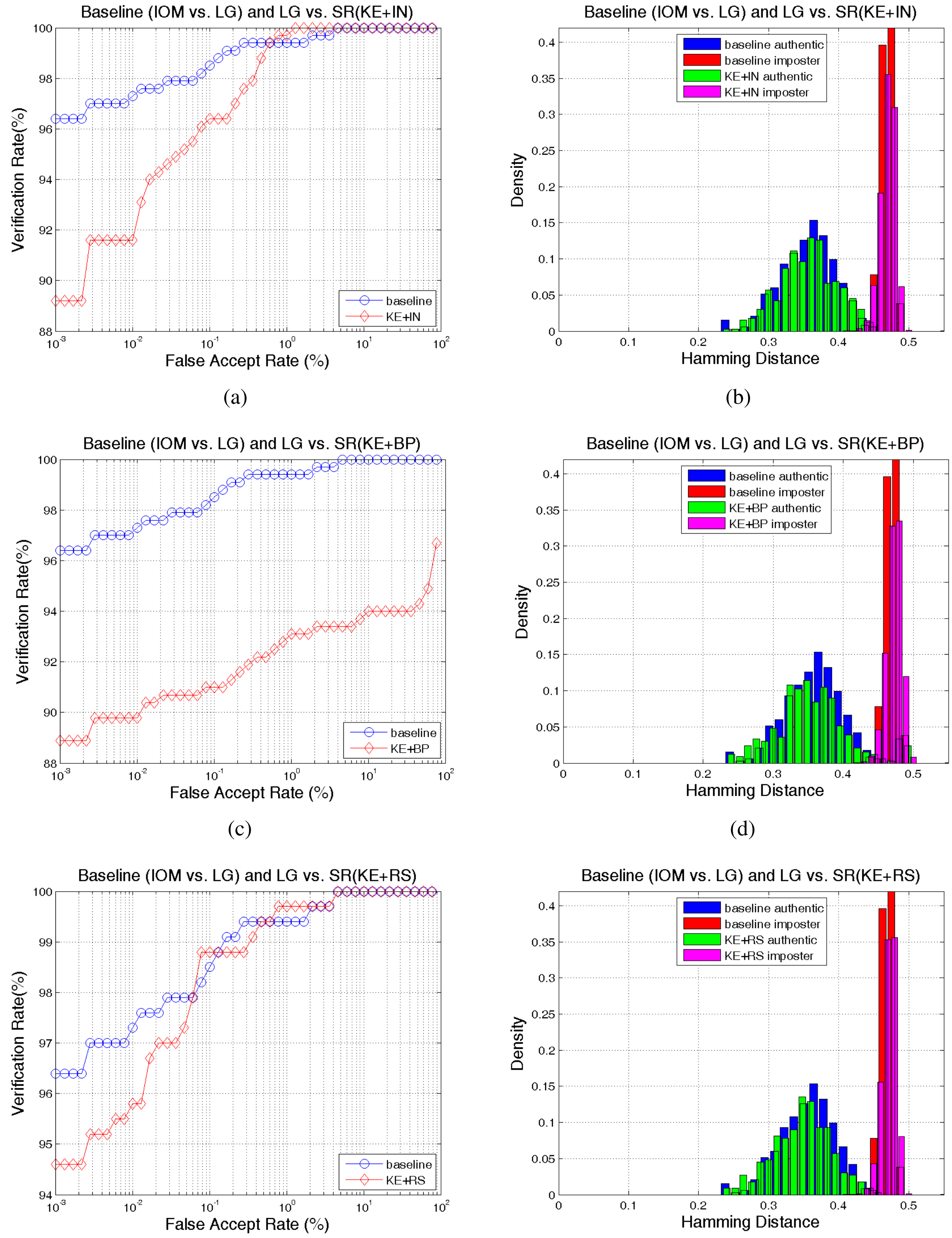


(a) (b)

(c) (d)

(e) (f)

Figure B.1: Performance comparison: baseline scenario 2 vs. SR enhanced system (RC). (a) ROC curves of RC+IN; (b) HD histogram distribution of RC+IN; (c) ROC curves of RC+BP; (d) HD histogram distribution of RC+BP; (e) ROC curves of RC+RS; (f) HD histogram distribution of RC+RS



(a)

(b)

(c)

(d)

(e)

(f)

Figure B.2: Performance comparison: baseline scenario 2 vs. SR enhanced system (KE). (a) ROC curves of KE+IN; (b) HD histogram distribution of KE+IN; (c) ROC curves of KE+BP; (d) HD histogram distribution of KE+BP; (e) ROC curves of KE+RS; (f) HD histogram distribution of KE+RS

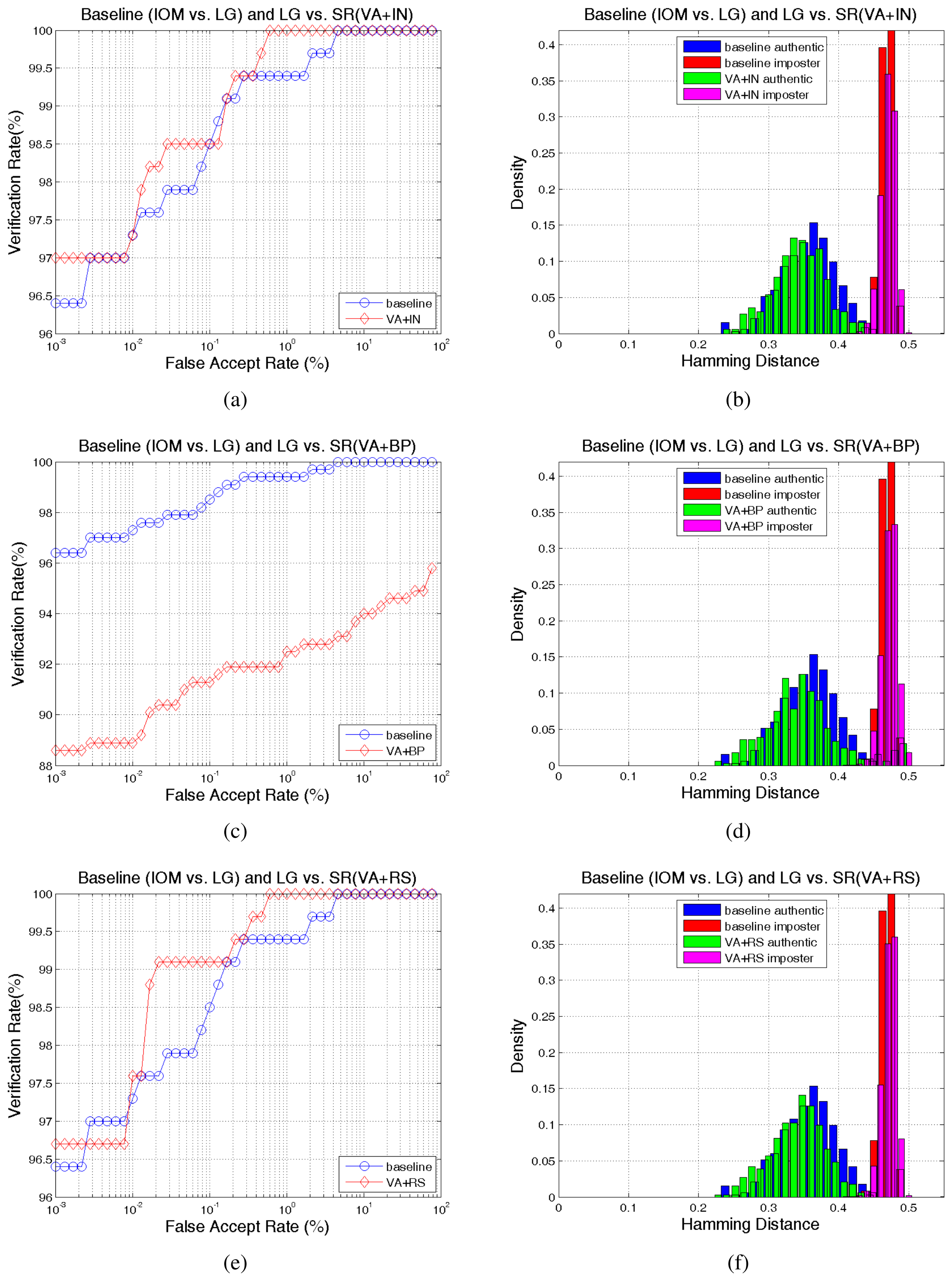
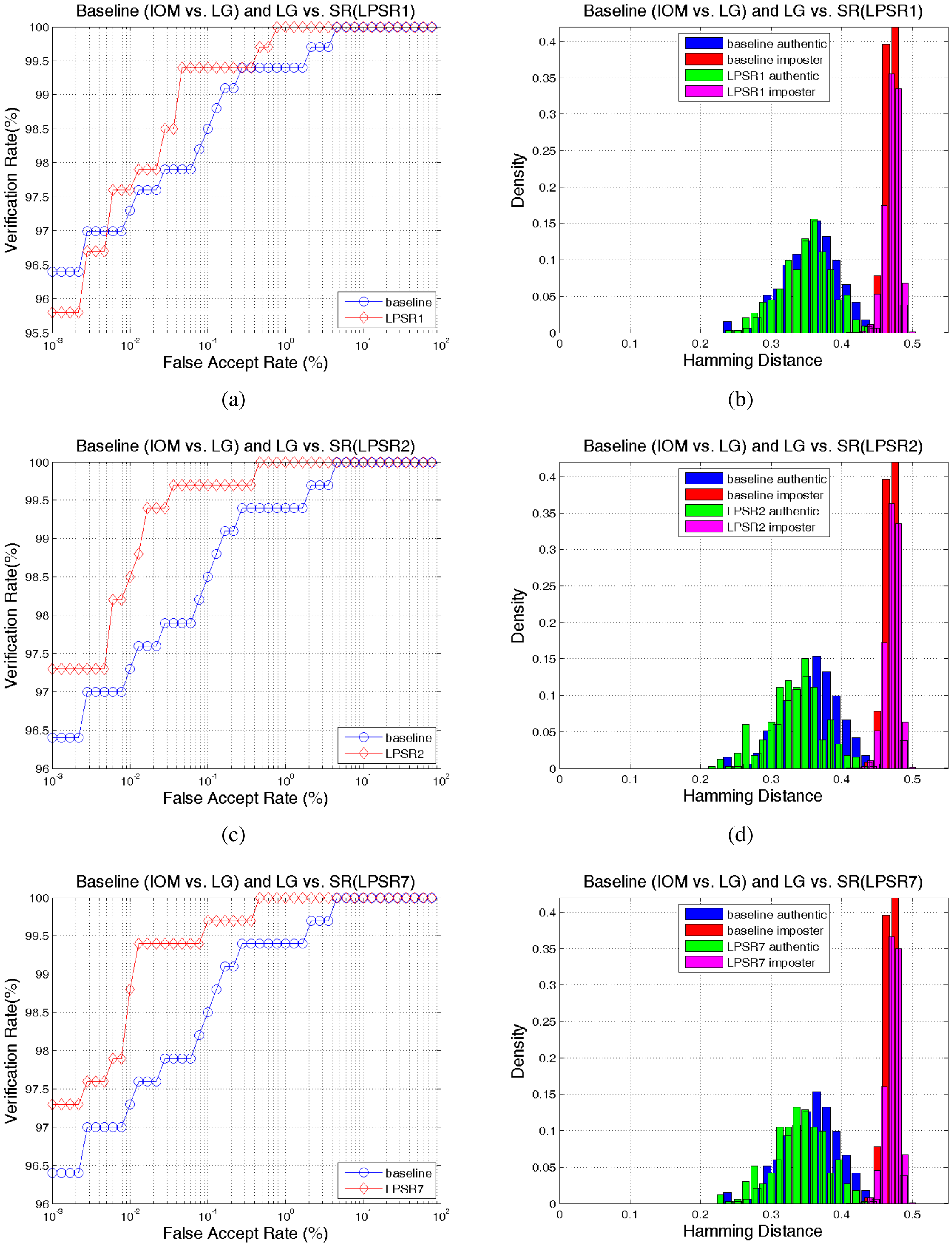


Figure B.3: Performance comparison: baseline scenario 2 vs. SR enhanced system (VA). (a) ROC curves of VA+IN; (b) HD histogram distribution of VA+IN; (c) ROC curves of VA+BP; (d) HD histogram distribution of VA+BP; (e) ROC curves of VA+RS; (f) HD histogram distribution of VA+RS



(e) (f)

Figure B.4: Performance comparison: baseline scenario 2 vs. SR enhanced system (LPSR1, 2 and 7). (a) ROC curves of LPSR1; (b) HD histogram distribution of LPSR1; (c) ROC curves of LPSR2; (d) HD histogram distribution of LPSR2; (e) ROC curves of LPSR7; (f) HD histogram distribution of LPSR7

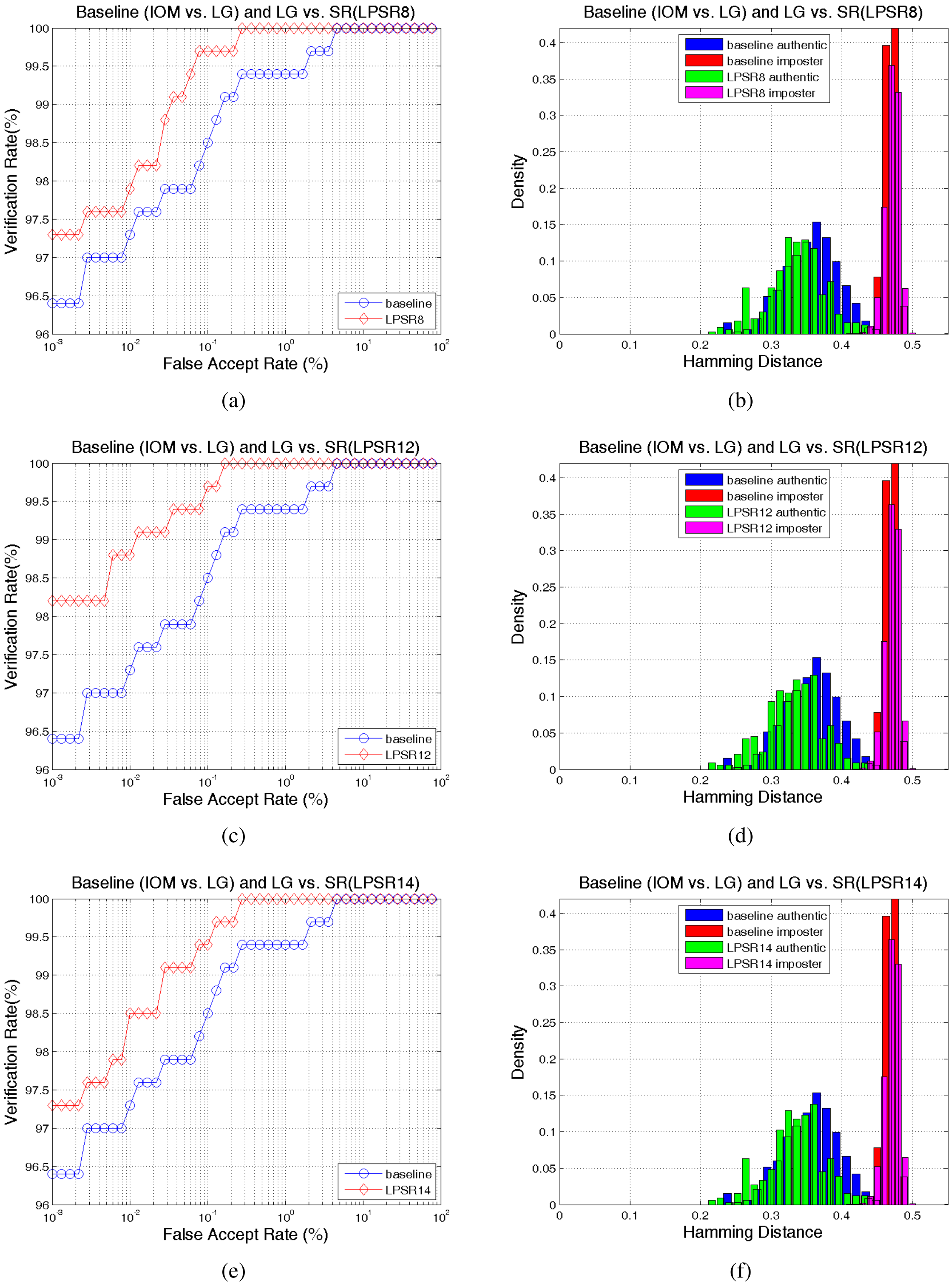
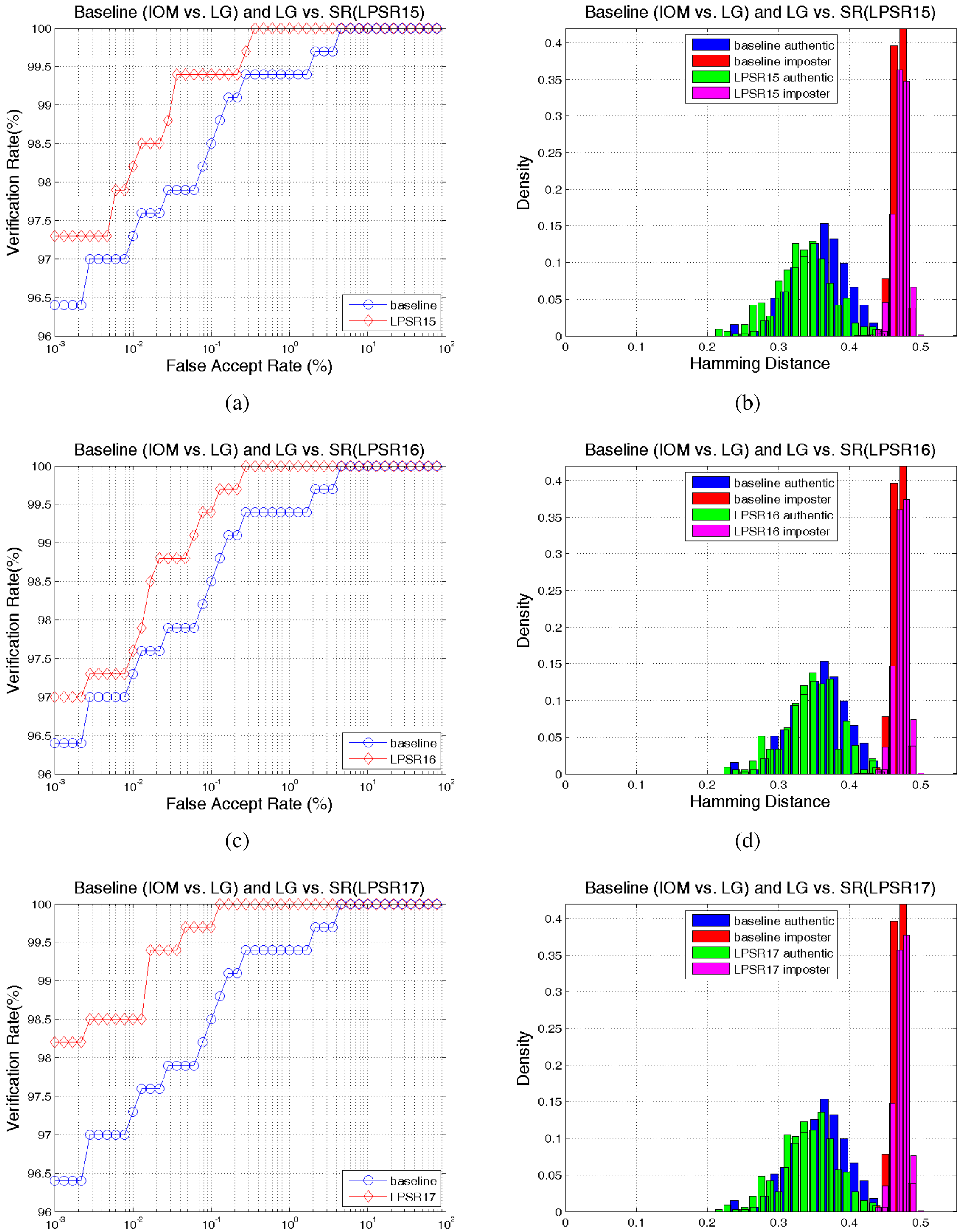


Figure B.5: Performance comparison: baseline scenario 2 vs. SR enhanced system (LPSR8, 12 and 14). (a) ROC curves of LPSR8; (b) HD histogram distribution of LPSR8; (c) ROC curves of LPSR12; (d) HD histogram distribution of LPSR12; (e) ROC curves of LPSR14; (f) HD histogram distribution of LPSR14

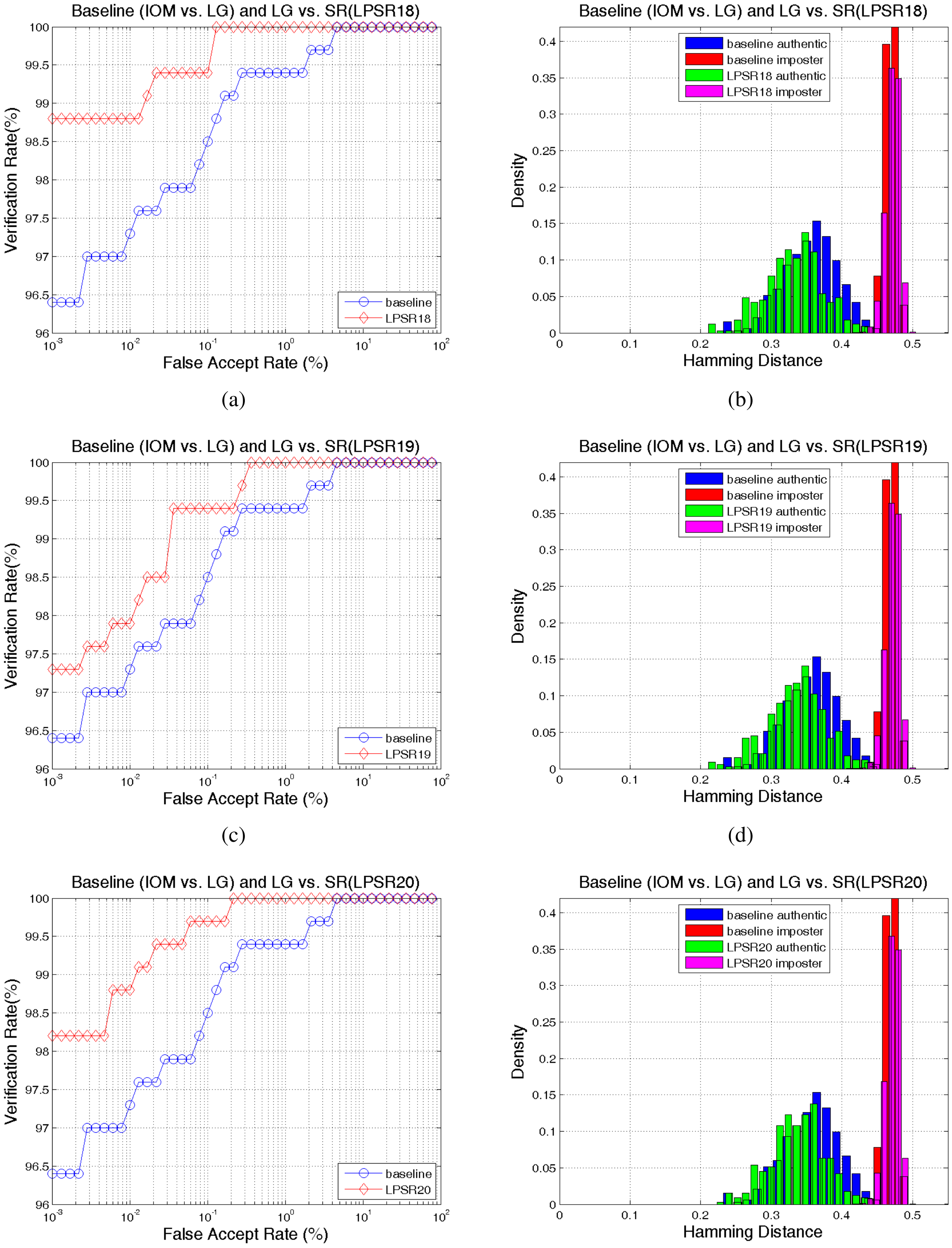


(a) (b)

(c) (d)

(e) (f)

Figure B.6: Performance comparison: baseline scenario 2 vs. SR enhanced system (LPSR15, 16 and 17). (a) ROC curves of LPSR15; (b) HD histogram distribution of LPSR15; (c) ROC curves of LPSR16; (d) HD histogram distribution of LPSR16; (e) ROC curves of LPSR17; (f) HD histogram distribution of LPSR17



(a)

(b)

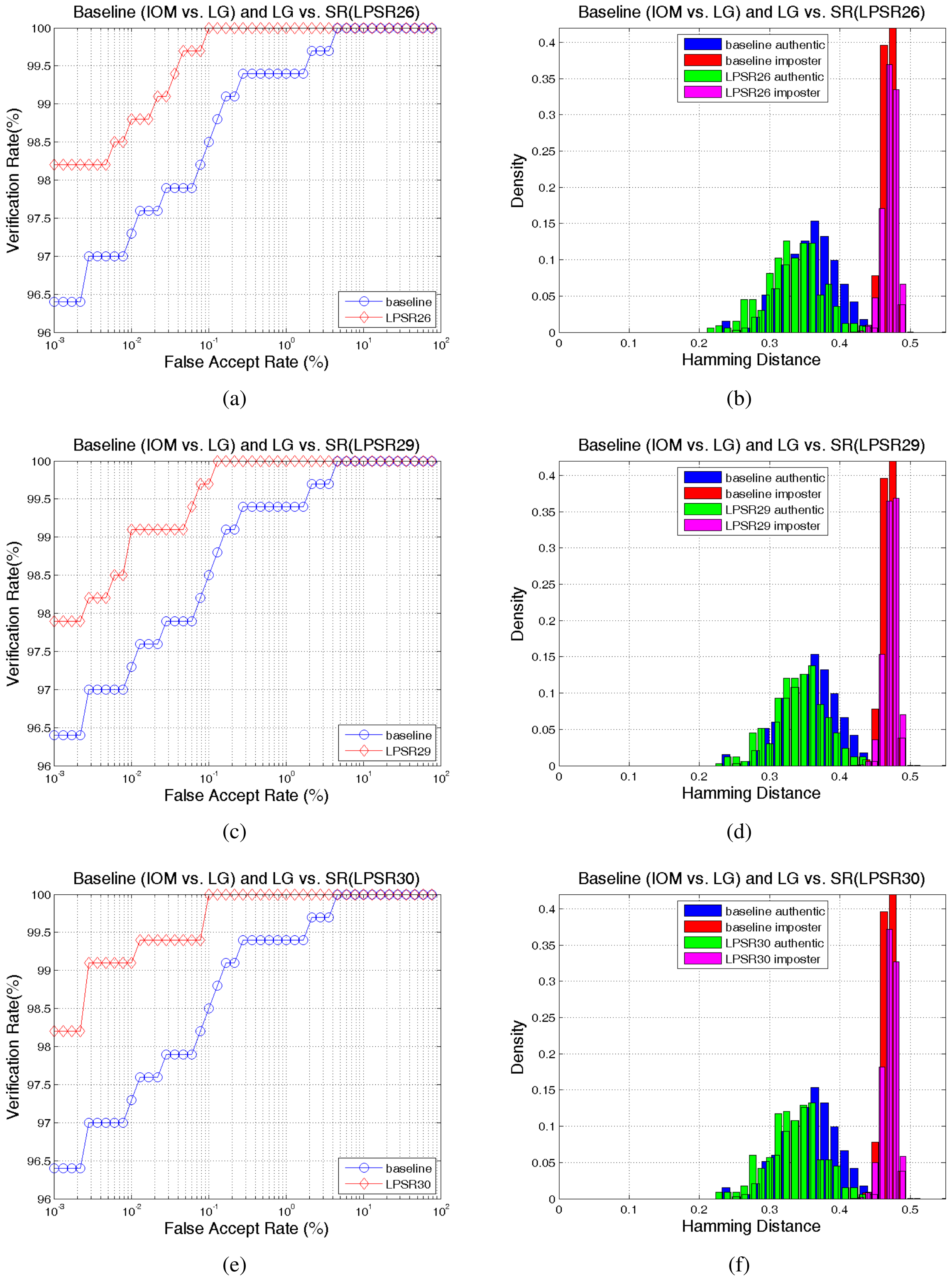
(c)

(d)

(e)

(f)

Figure B.7: Performance comparison: baseline scenario 2 vs. SR enhanced system (LPSR18, 19 and 20). (a) ROC curves of LPSR18; (b) HD histogram distribution of LPSR18; (c) ROC curves of LPSR19; (d) HD histogram distribution of LPSR19; (e) ROC curves of LPSR20; (f) HD histogram distribution of LPSR20



(a)

(b)

(c)

(d)

(e)

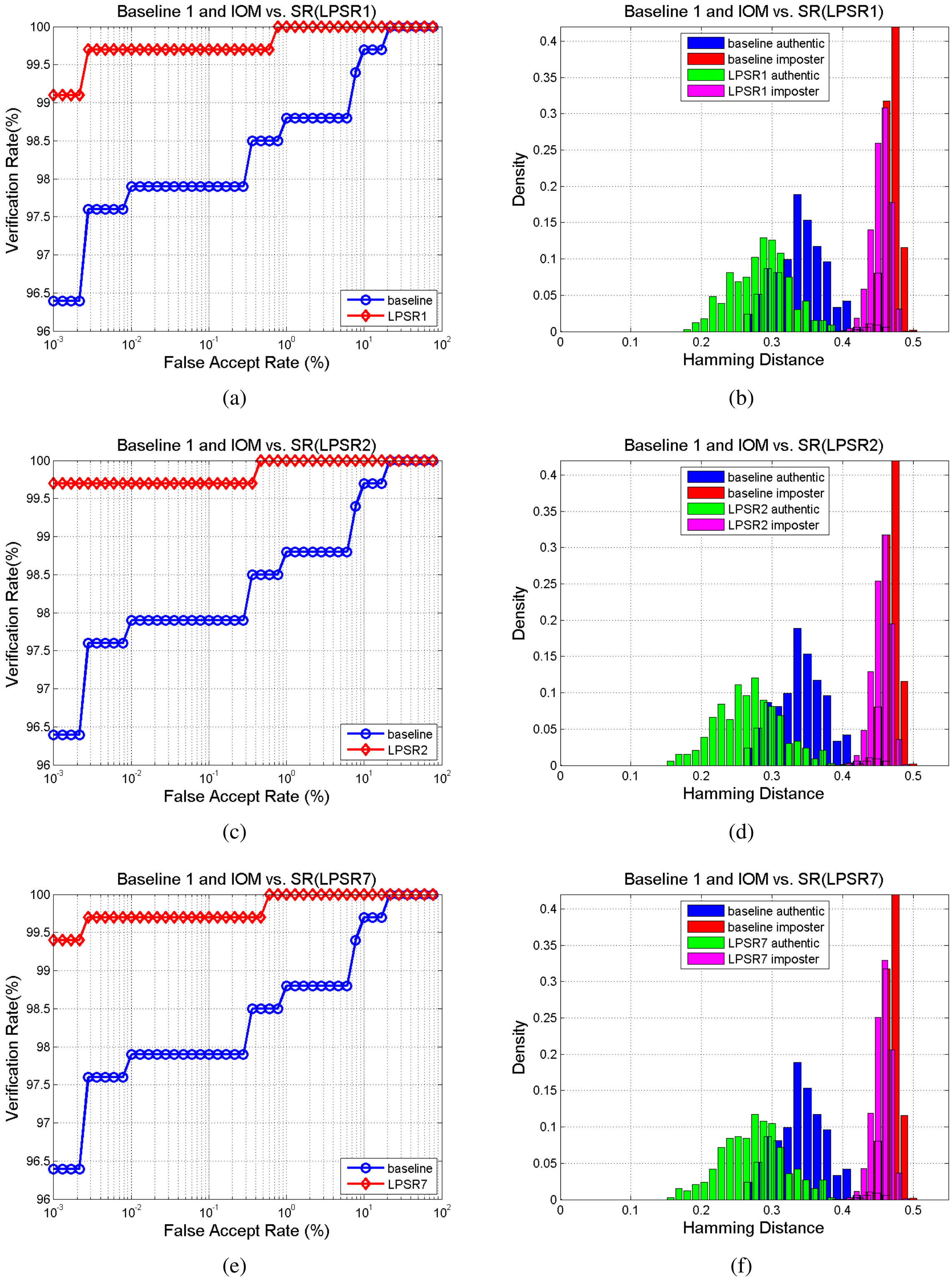
(f)

Figure B.8: Performance comparison: baseline scenario 2 vs. SR enhanced system (LPSR26, 29 and 30). (a) ROC curves of LPSR26; (b) HD histogram distribution of LPSR26; (c) ROC curves of LPSR29; (d) HD histogram distribution of LPSR29; (e) ROC curves of LPSR30; (f) HD histogram distribution of LPSR30

# Appendix C

# Experimental Figures for SR vs. IOM

All of the experimental results figures (ROC curves and histogram distribution of Hamming distance) of the system architecture SR vs. IOM are listed here. Under this architecture, the LPSR 2 achieves lowest recognition errors compared to other methods.



(a)

(b)

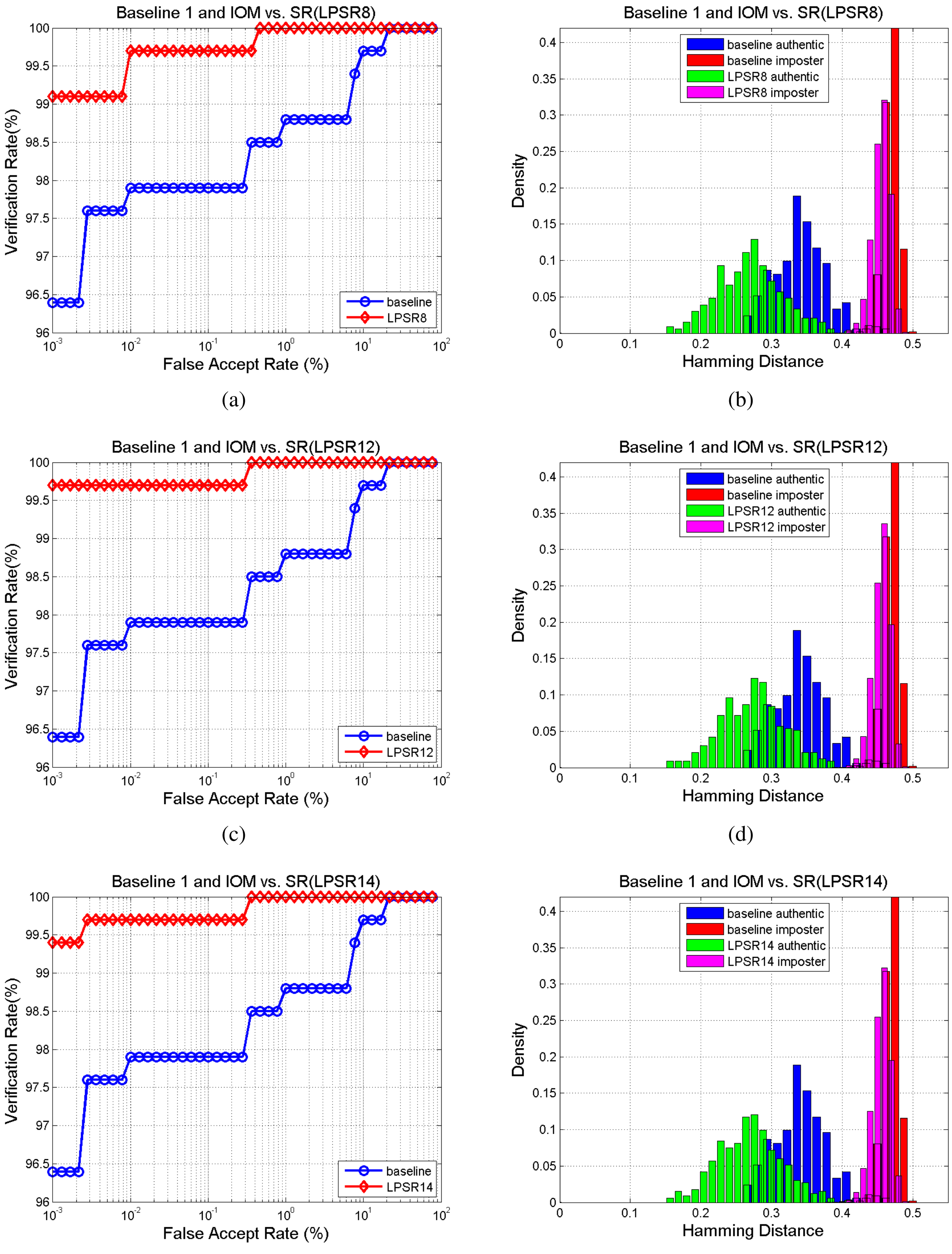
(c)

(d)

(e)

(f)

Figure C.1: Performance comparison: baseline scenario 1 (IOM vs. PIER) compared to configuration of "IOM vs. SR" (LPSR1, 2 and 7). (a) ROC curves of LPSR1; (b) HD histogram distribution of LPSR1; (c) ROC curves of LPSR2; (d) HD histogram distribution of LPSR2; (e) ROC curves of LPSR7; (f) HD histogram distribution of LPSR7



(a)

(b)

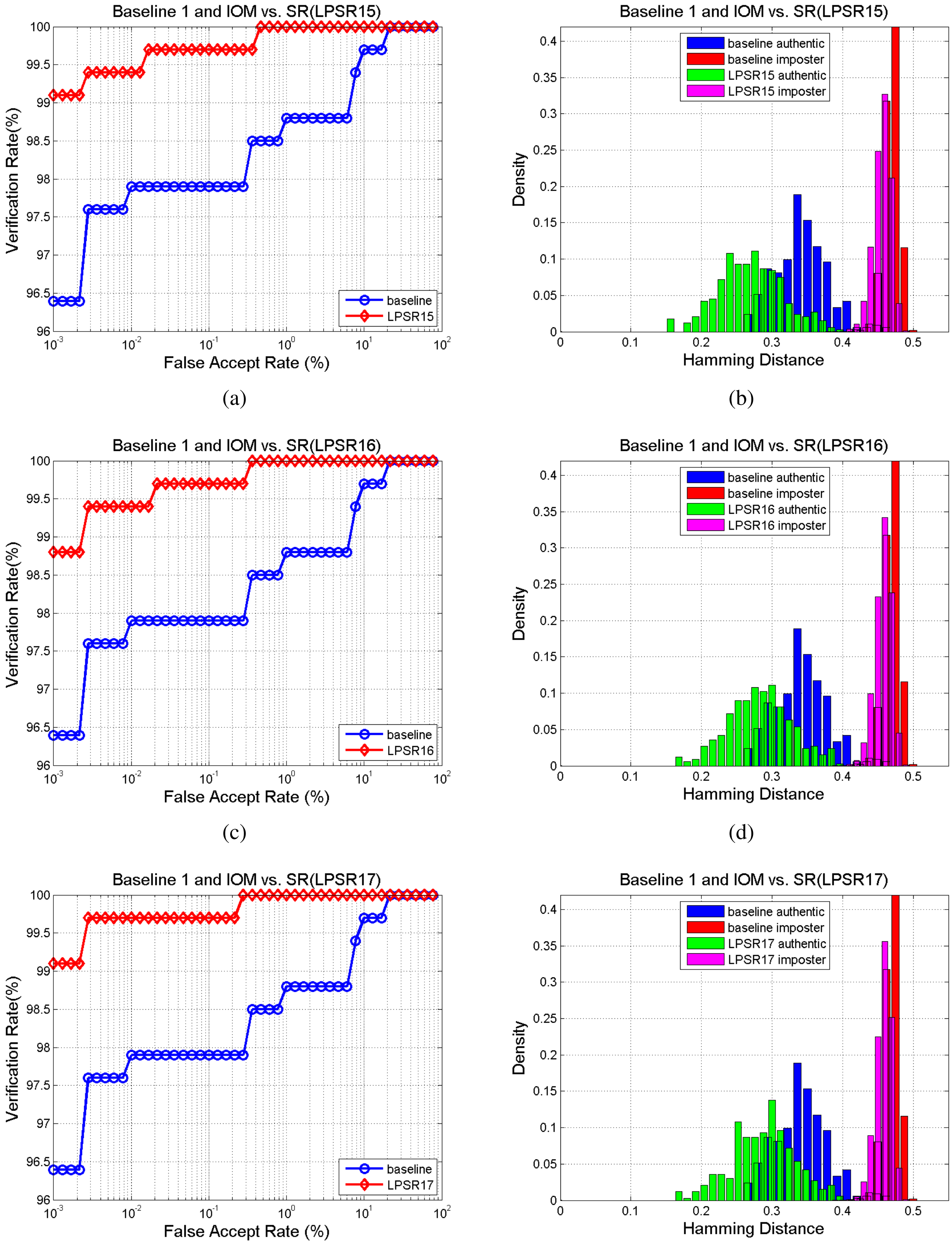
(c)

(d)

(e)

(f)

Figure C.2: Performance comparison: baseline scenario 1 (IOM vs. PIER) compared to configuration of "IOM vs. SR" (LPSR8, 12 and 14). (a) ROC curves of LPSR8; (b) HD histogram distribution of LPSR8; (c) ROC curves of LPSR12; (d) HD histogram distribution of LPSR12; (e) ROC curves of LPSR14; (f) HD histogram distribution of LPSR14



(a)

(b)

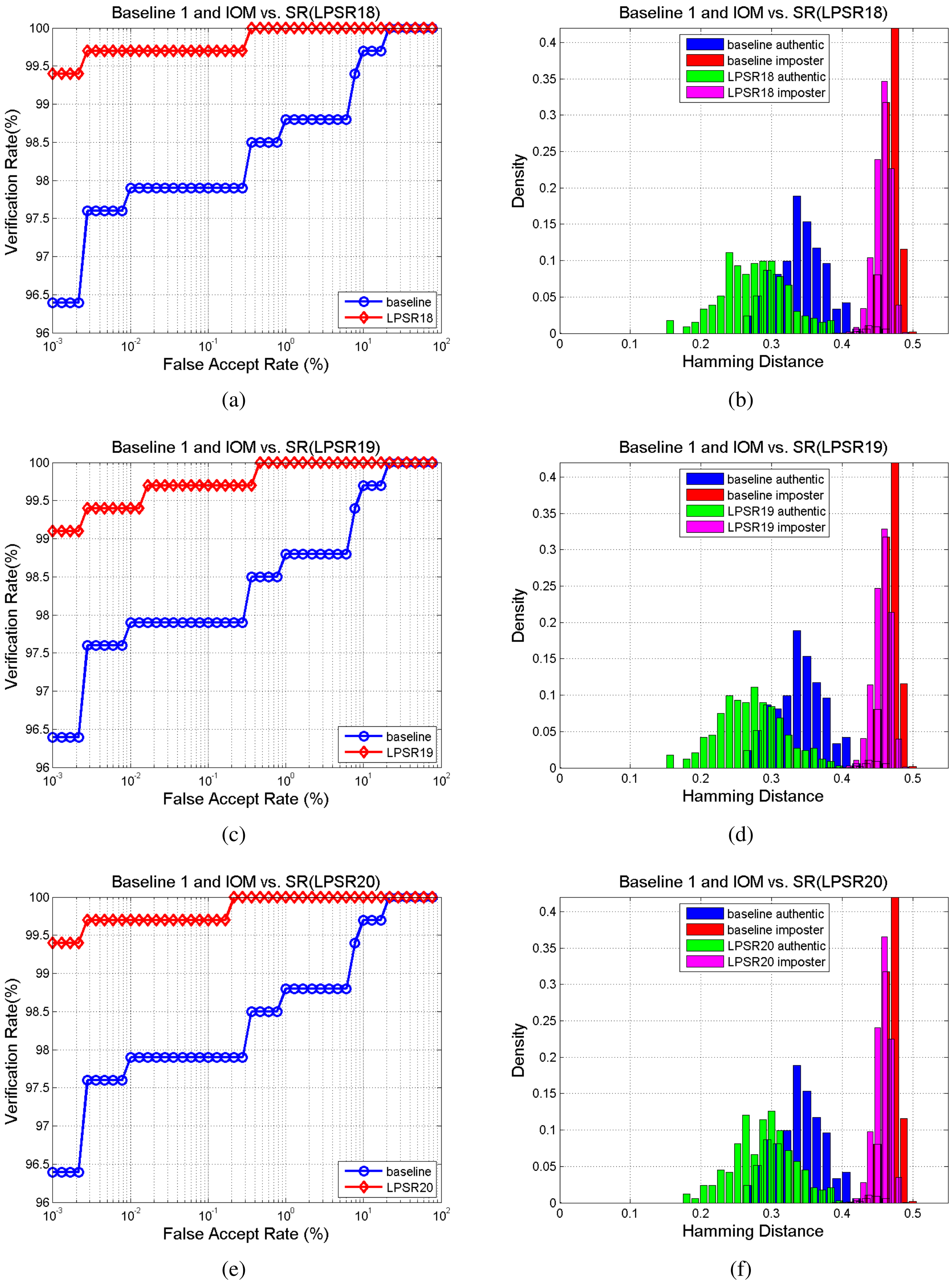
(c)

(d)

(e)

(f)

Figure C.3: Performance comparison: baseline scenario 1 (IOM vs. PIER) compared to configuration of "IOM vs. SR" (LPSR15, 16 and 17). (a) ROC curves of LPSR15; (b) HD histogram distribution of LPSR15; (c) ROC curves of LPSR16; (d) HD histogram distribution of LPSR16; (e) ROC curves of LPSR17; (f) HD histogram distribution of LPSR17



(a)

(b)

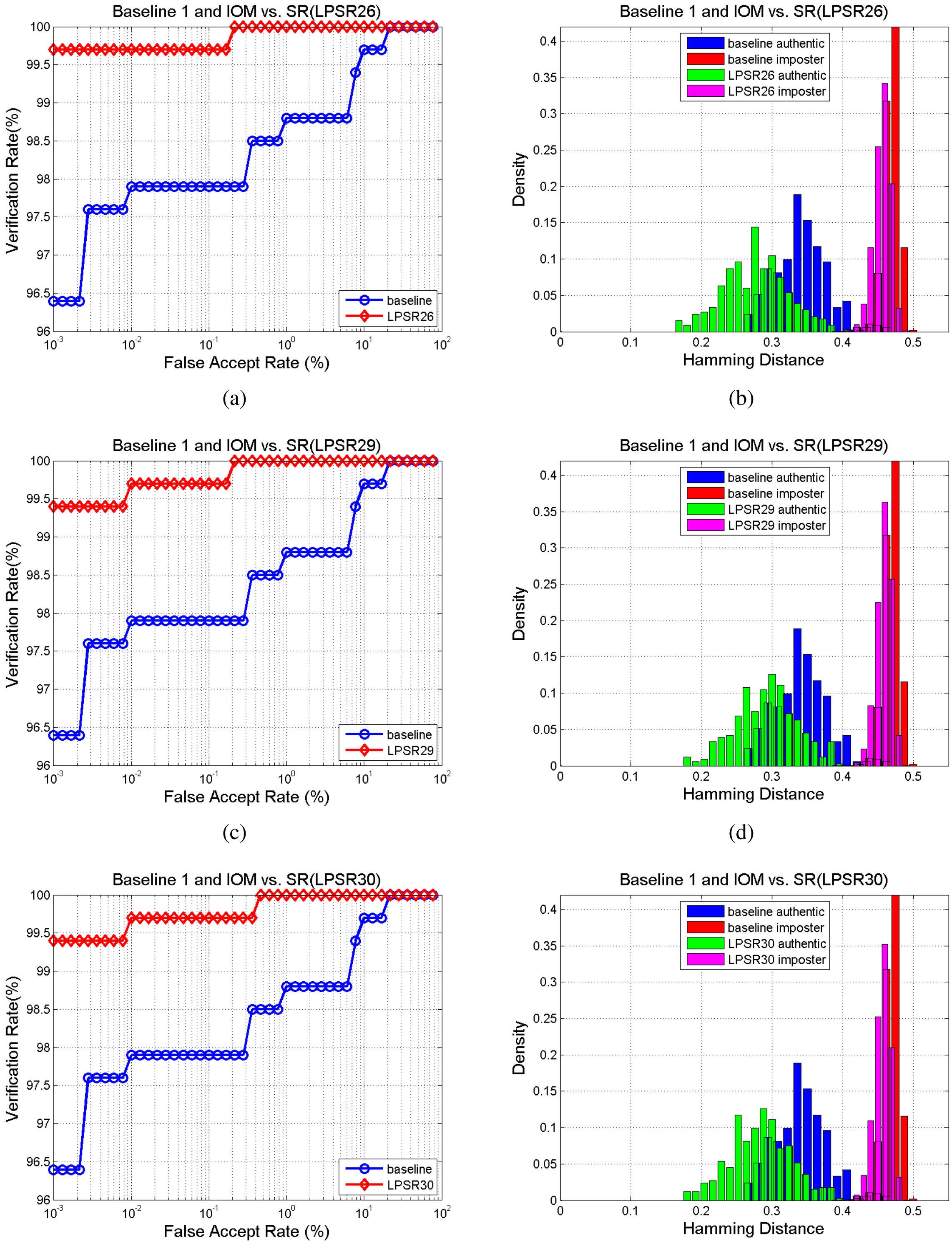
(c)

(d)

(e)

(f)

Figure C.4: Performance comparison: baseline scenario 1 (IOM vs. PIER) compared to configuration of "IOM vs. SR" (LPSR18, 19 and 20). (a) ROC curves of LPSR18; (b) HD histogram distribution of LPSR18; (c) ROC curves of LPSR19; (d) HD histogram distribution of LPSR19; (e) ROC curves of LPSR20; (f) HD histogram distribution of LPSR20

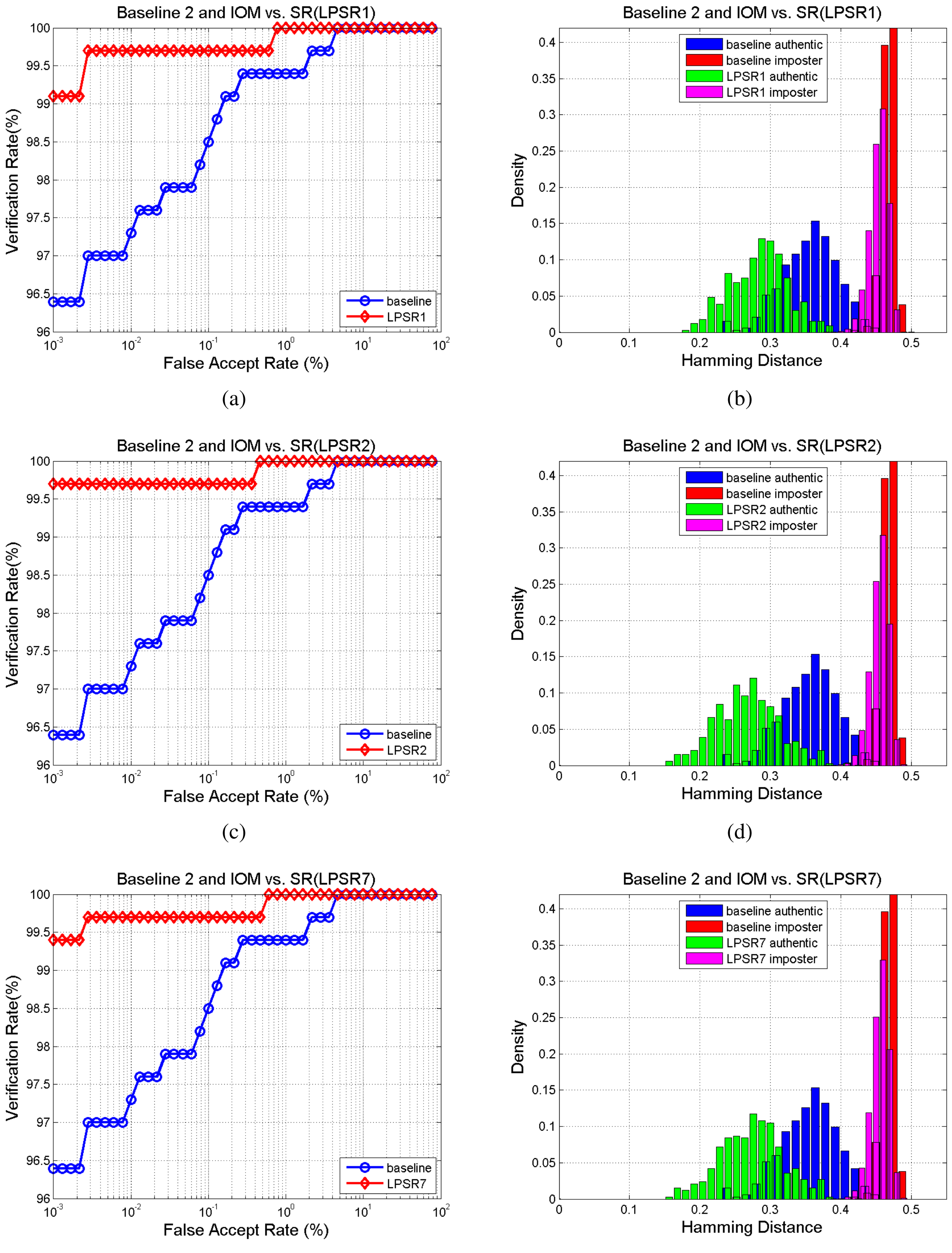


(a) (b)

(c) (d)

(e) (f)

Figure C.5: Performance comparison: baseline scenario 1 (IOM vs. PIER) compared to configuration of "IOM vs. SR" (LPSR26, 29 and 30). (a) ROC curves of LPSR26; (b) HD histogram distribution of LPSR26; (c) ROC curves of LPSR29; (d) HD histogram distribution of LPSR29; (e) ROC curves of LPSR30; (f) HD histogram distribution of LPSR30



(a)

(b)

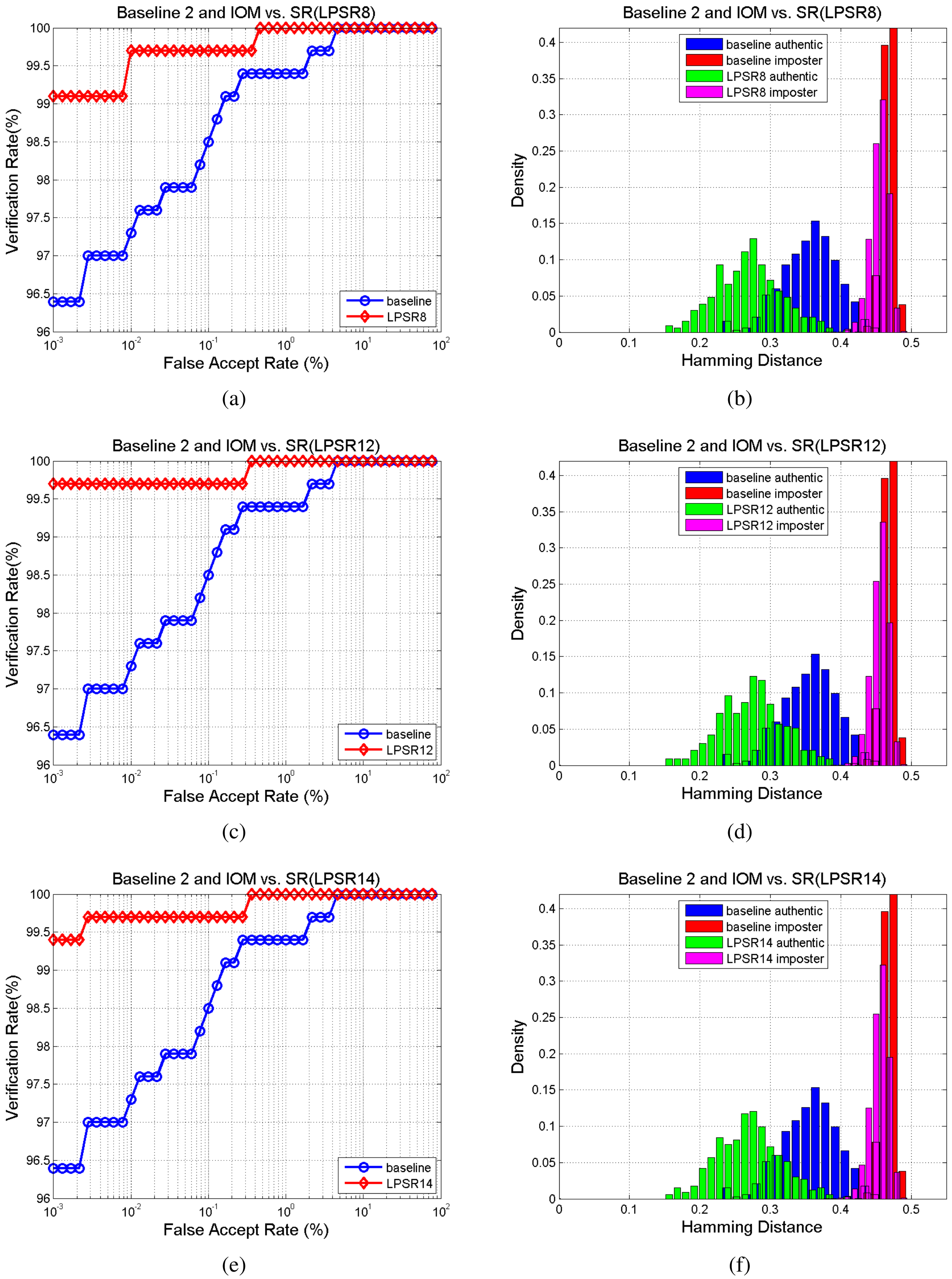
(c)

(d)

(e)

(f)

Figure C.6: Performance comparison: baseline scenario 2 (IOM vs. LG) compared to configuration of "IOM vs. SR" (LPSR1, 2 and 7). (a) ROC curves of LPSR1; (b) HD histogram distribution of LPSR1; (c) ROC curves of LPSR2; (d) HD histogram distribution of LPSR2; (e) ROC curves of LPSR7; (f) HD histogram distribution of LPSR7



(a)

(b)

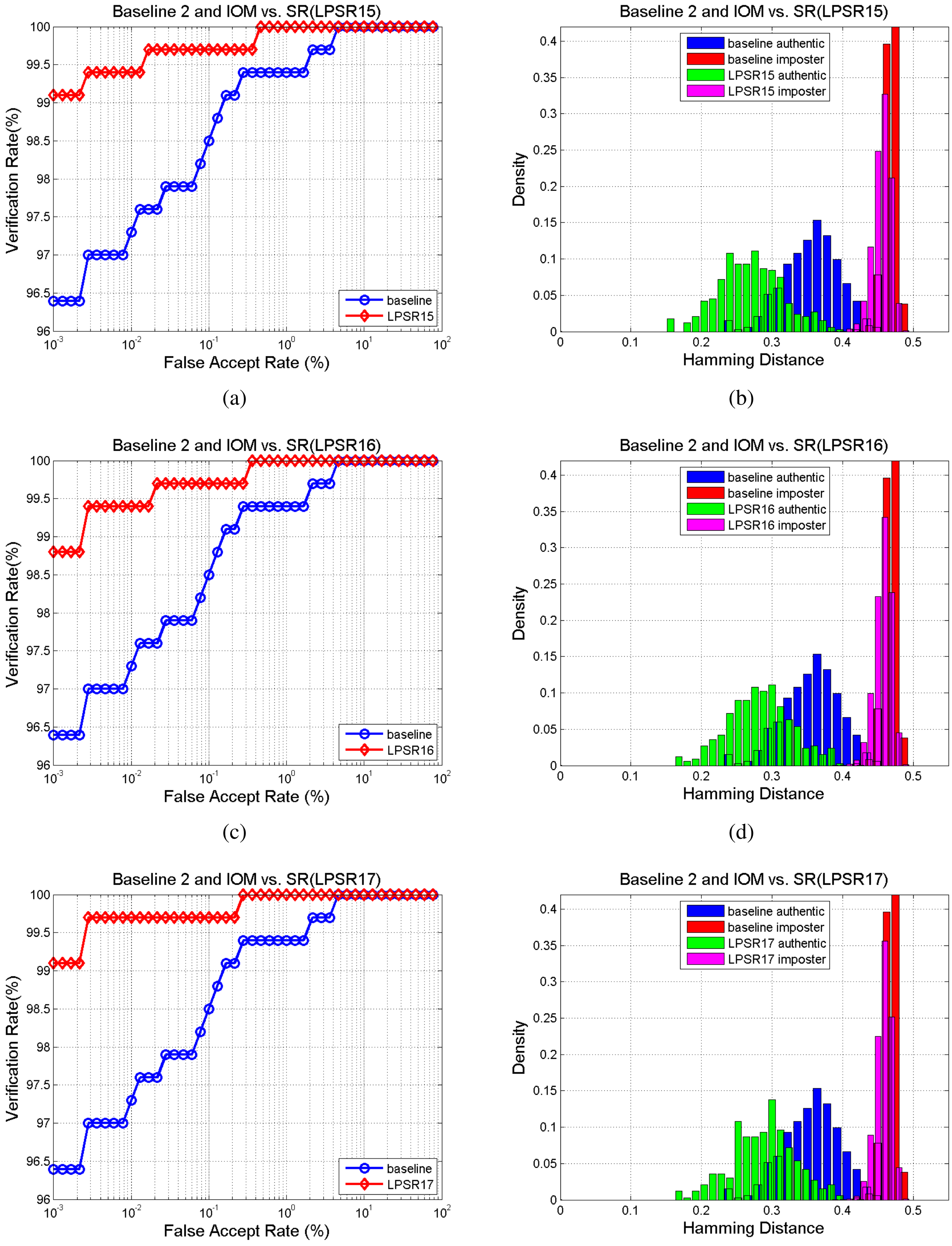
(c)

(d)

(e)

(f)

Figure C.7: Performance comparison: baseline scenario 2 (IOM vs. LG) compared to configuration of "IOM vs. SR" (LPSR8, 12 and 14). (a) ROC curves of LPSR8; (b) HD histogram distribution of LPSR8; (c) ROC curves of LPSR12; (d) HD histogram distribution of LPSR12; (e) ROC curves of LPSR14; (f) HD histogram distribution of LPSR14



(a)

(b)

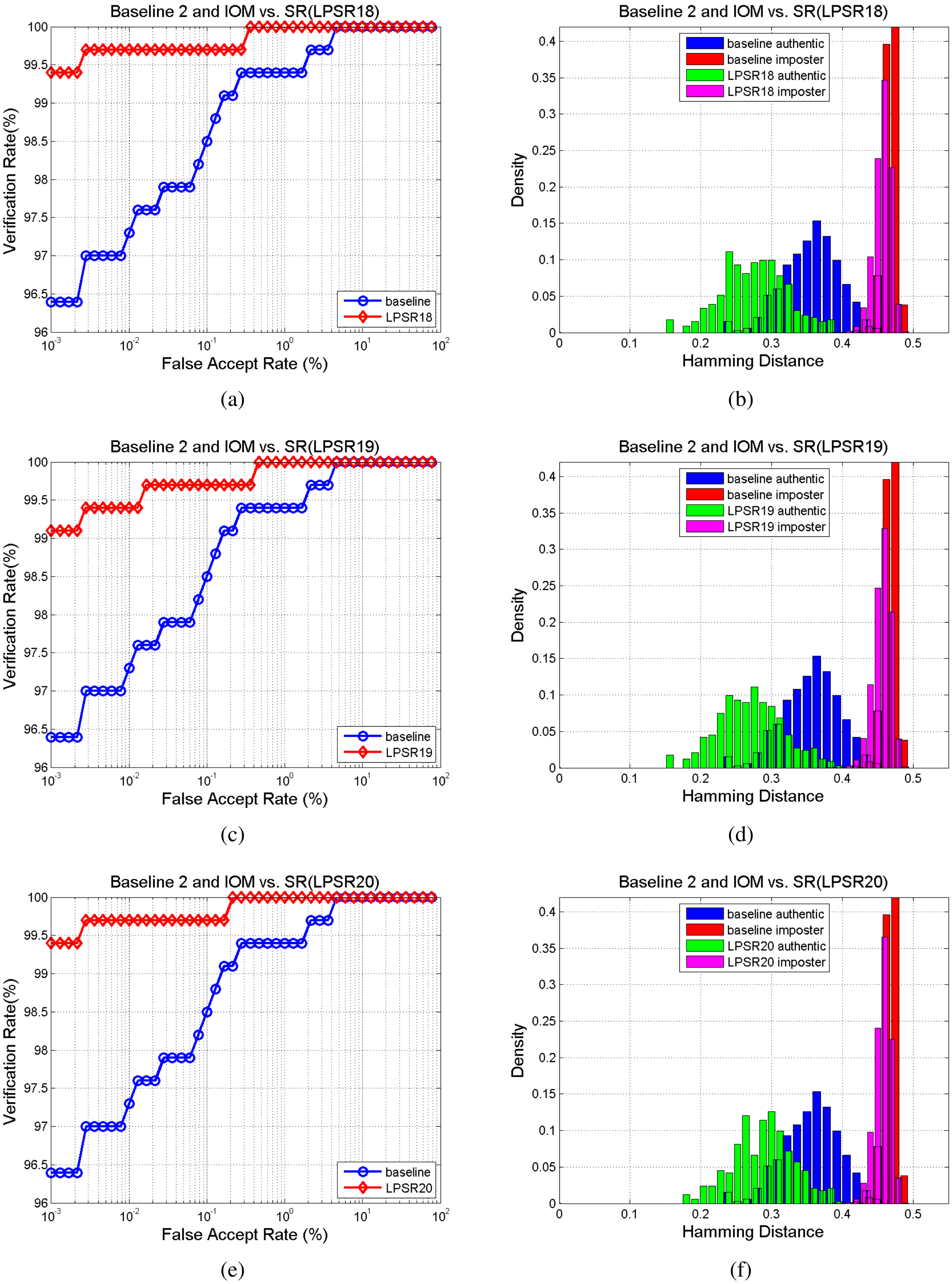
(c)

(d)

(e)

(f)

Figure C.8: Performance comparison: baseline scenario 2 (IOM vs. LG) compared to configuration of "IOM vs. SR" (LPSR15, 16 and 17). (a) ROC curves of LPSR15; (b) HD histogram distribution of LPSR15; (c) ROC curves of LPSR16; (d) HD histogram distribution of LPSR16; (e) ROC curves of LPSR17; (f) HD histogram distribution of LPSR17



(a)

(b)

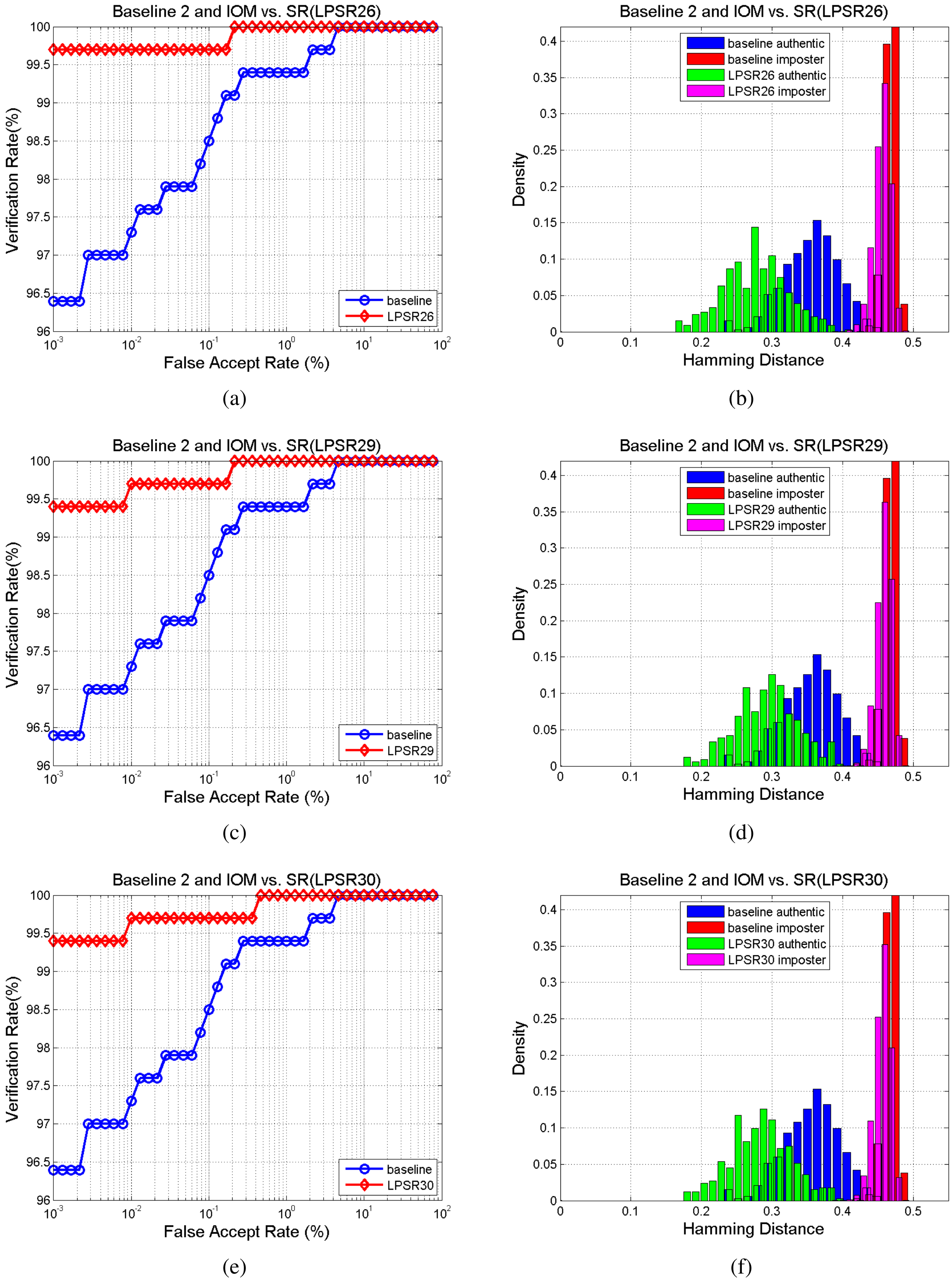
(c)

(d)

(e)

(f)

Figure C.9: Performance comparison: baseline scenario 2 (IOM vs. LG) compared to configuration of "IOM vs. SR" (LPSR18, 19 and 20). (a) ROC curves of LPSR18; (b) HD histogram distribution of LPSR18; (c) ROC curves of LPSR19; (d) HD histogram distribution of LPSR19; (e) ROC curves of LPSR20; (f) HD histogram distribution of LPSR20



(a)

(b)

(c)

(d)

(e)

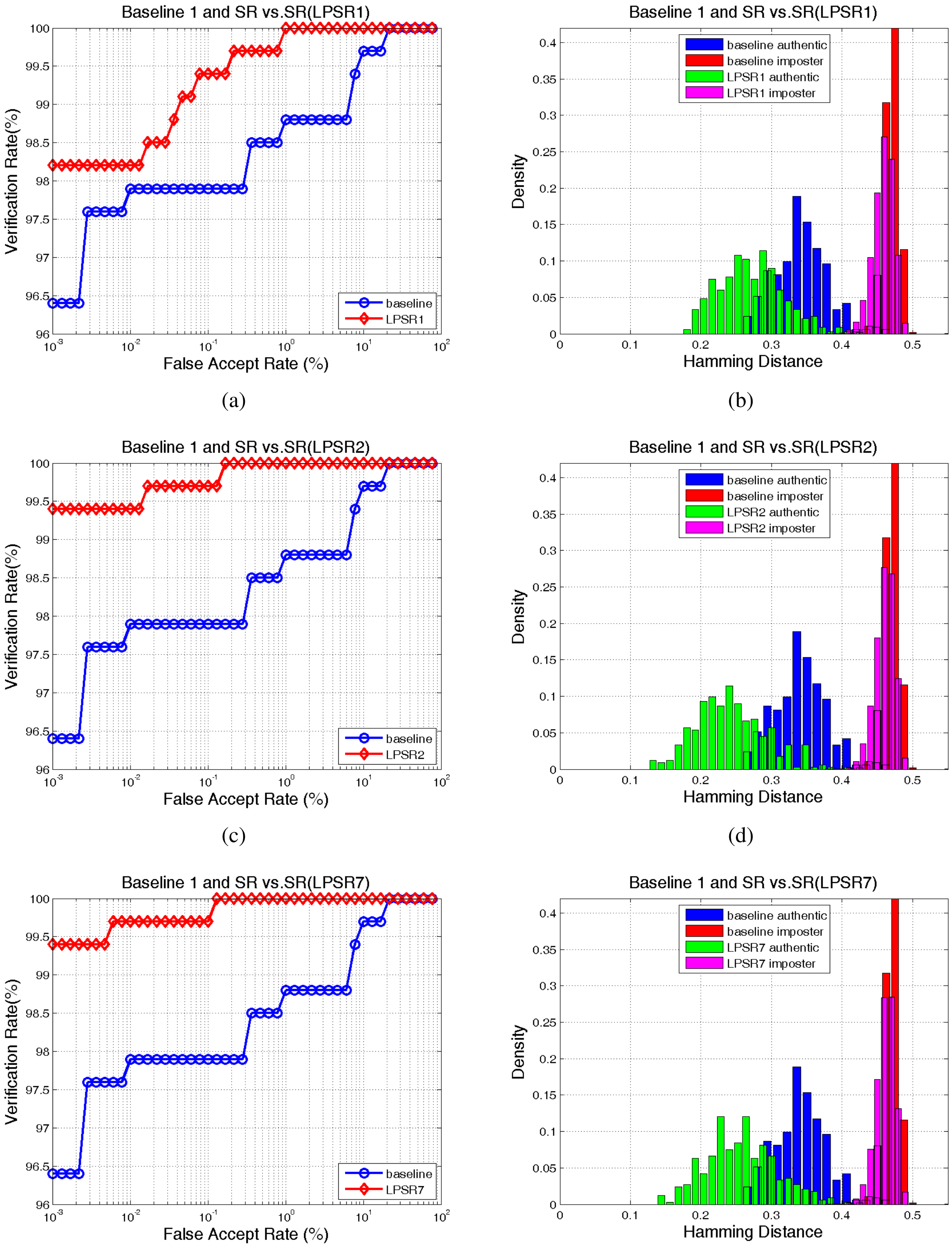
(f)

Figure C.10: Performance comparison: baseline scenario 2 (IOM vs. LG) compared to configuration of "IOM vs. SR" (LPSR26, 29 and 30). (a) ROC curves of LPSR26; (b) HD histogram distribution of LPSR26; (c) ROC curves of LPSR29; (d) HD histogram distribution of LPSR29; (e) ROC curves of LPSR30; (f) HD histogram distribution of LPSR30

# Appendix D

# Experimental Figures for SR vs. SR

All of the experimental results figures (ROC curves and histogram distribution of Hamming distance) of the system architecture SR vs. SR are listed here. Under this architecture, the LPSR 30 achieves lowest recognition errors compared to other methods.

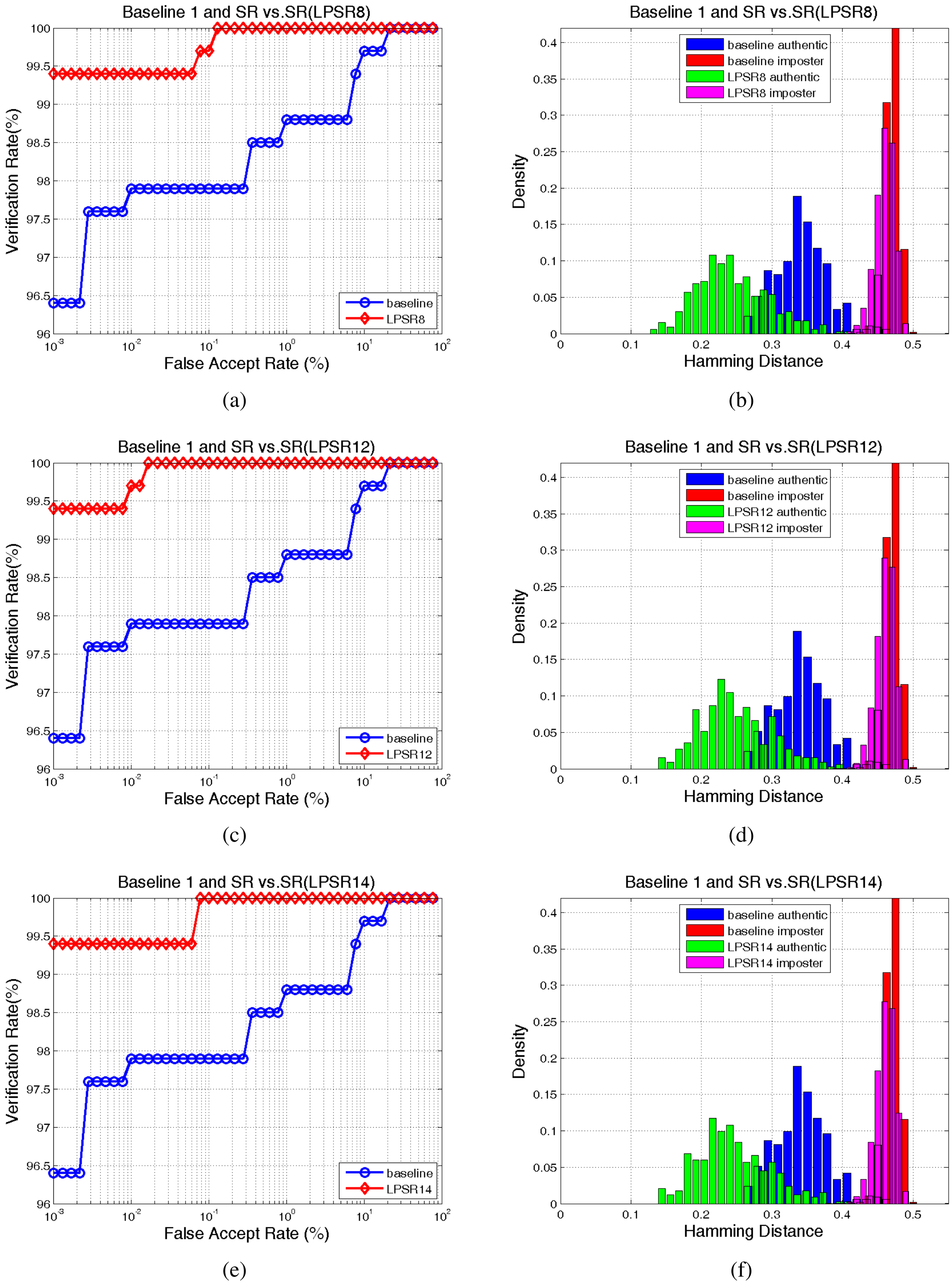


(a) (b)

(c) (d)

(e) (f)

Figure D.1: Performance comparison: baseline scenario 1 (IOM vs. PIER) compared to configuration of "SR vs. SR" (LPSR1, 2 and 7). (a) ROC curves of LPSR1; (b) HD histogram distribution of LPSR1; (c) ROC curves of LPSR2; (d) HD histogram distribution of LPSR2; (e) ROC curves of LPSR7; (f) HD histogram distribution of LPSR7



(a)

(b)

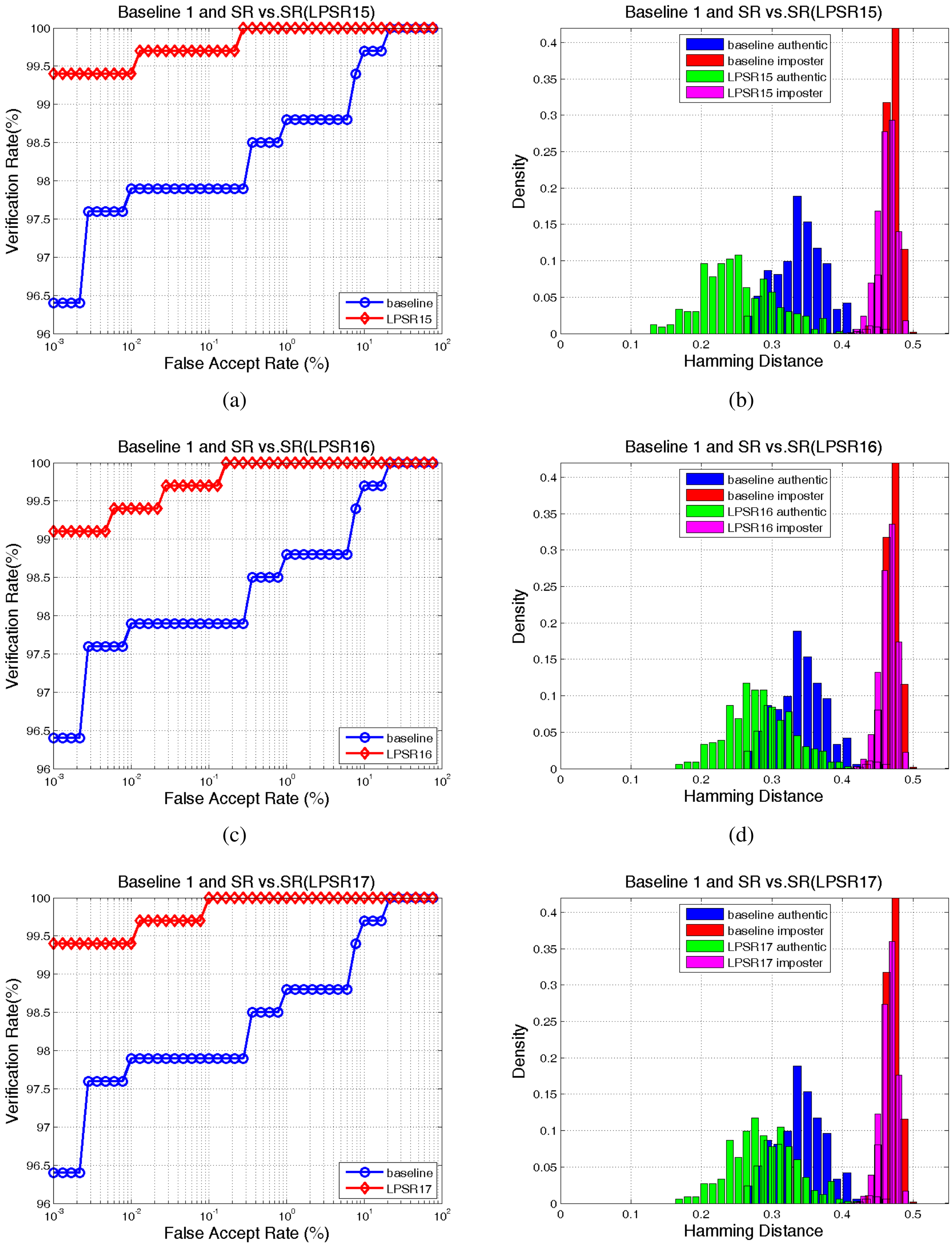
(c)

(d)

(e)

(f)

Figure D.2: Performance comparison: baseline scenario 1 (IOM vs. PIER) compared to configuration of "SR vs. SR" (LPSR8, 12 and 14). (a) ROC curves of LPSR8; (b) HD histogram distribution of LPSR8; (c) ROC curves of LPSR12; (d) HD histogram distribution of LPSR12; (e) ROC curves of LPSR14; (f) HD histogram distribution of LPSR14



(a)

(b)

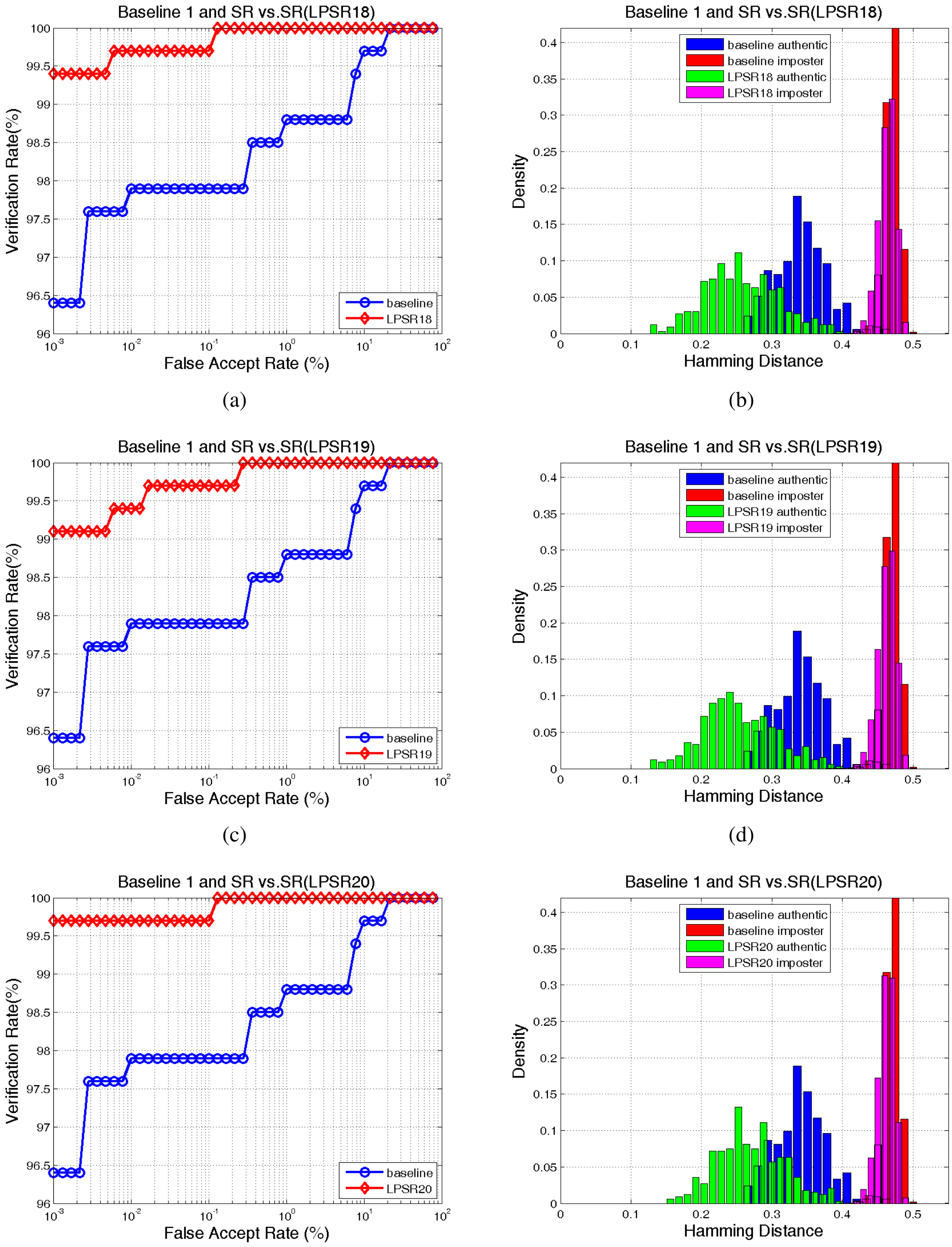
(c)

(d)

(e)

(f)

Figure D.3: Performance comparison: baseline scenario 1 (IOM vs. PIER) compared to configuration of "SR vs. SR" (LPSR15, 16 and 17). (a) ROC curves of LPSR15; (b) HD histogram distribution of LPSR15; (c) ROC curves of LPSR16; (d) HD histogram distribution of LPSR16; (e) ROC curves of LPSR17; (f) HD histogram distribution of LPSR17

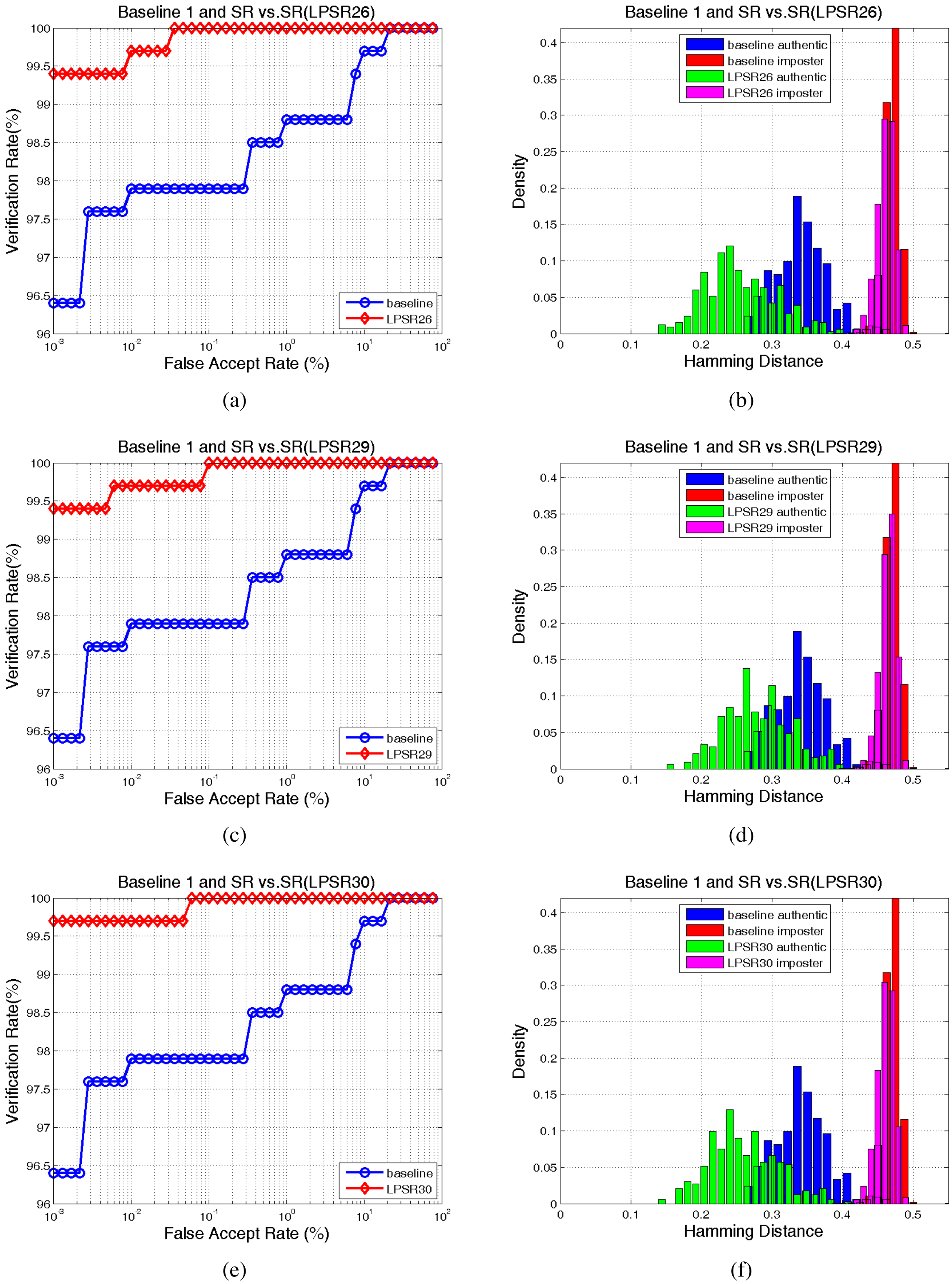


(a) (b)

(c) (d)

(e) (f)

Figure D.4: Performance comparison: baseline scenario 1 (IOM vs. PIER) compared to configuration of "SR vs. SR" (LPSR18, 19 and 20). (a) ROC curves of LPSR18; (b) HD histogram distribution of LPSR18; (c) ROC curves of LPSR19; (d) HD histogram distribution of LPSR19; (e) ROC curves of LPSR20; (f) HD histogram distribution of LPSR20

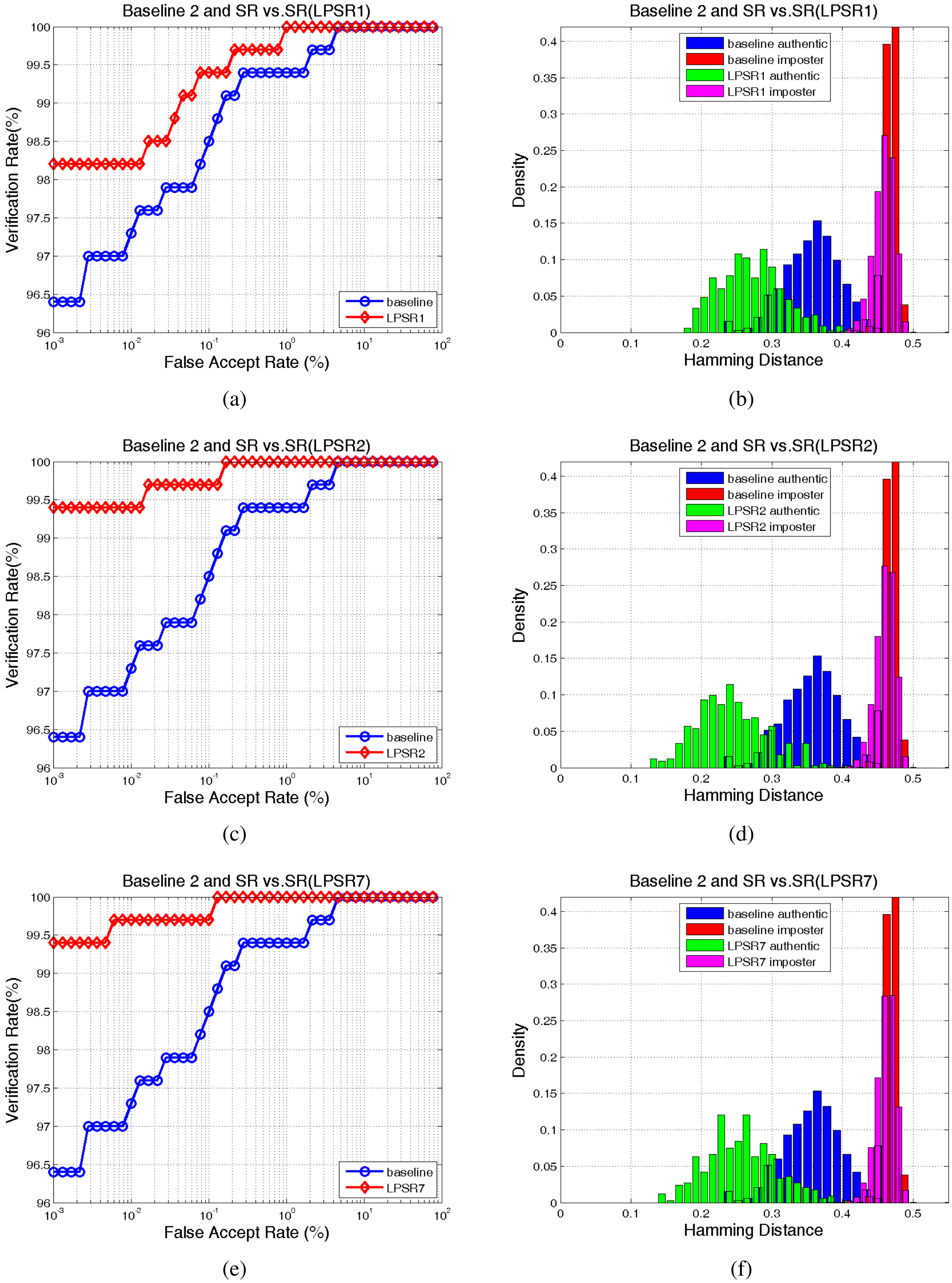


(a) (b)

(c) (d)

(e) (f)

Figure D.5: Performance comparison: baseline scenario 1 (IOM vs. PIER) compared to configuration of "SR vs. SR" (LPSR26, 29 and 30). (a) ROC curves of LPSR26; (b) HD histogram distribution of LPSR26; (c) ROC curves of LPSR29; (d) HD histogram distribution of LPSR29; (e) ROC curves of LPSR30; (f) HD histogram distribution of LPSR30

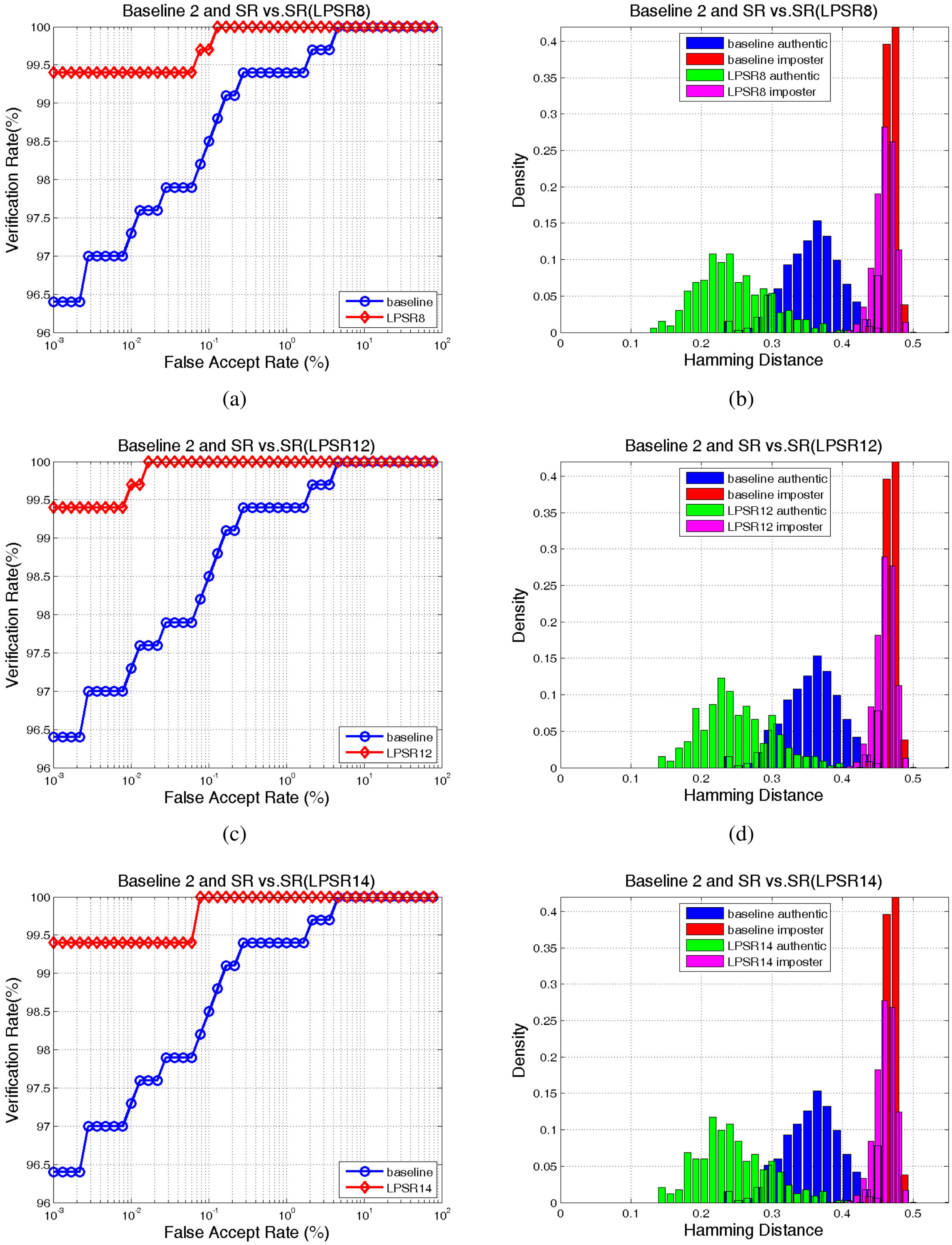


(a) (b)

(c) (d)

(e) (f)

Figure D.6: Performance comparison: baseline scenario 2 (IOM vs. LG) compared to configuration of "SR vs. SR" (LPSR1, 2 and 7). (a) ROC curves of LPSR1; (b) HD histogram distribution of LPSR1; (c) ROC curves of LPSR2; (d) HD histogram distribution of LPSR2; (e) ROC curves of LPSR7; (f) HD histogram distribution of LPSR7

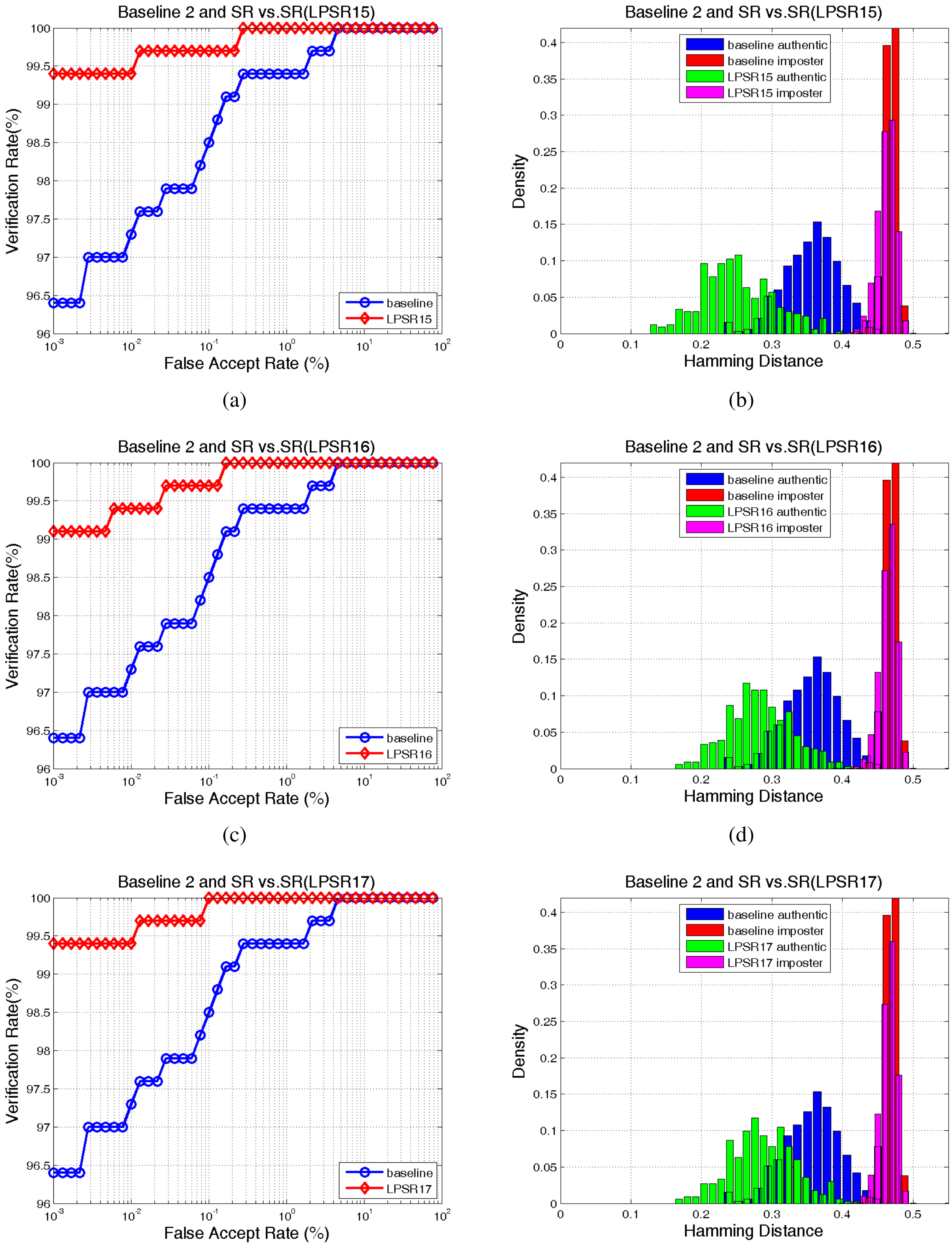


(a) (b)

(c) (d)

(e) (f)

Figure D.7: Performance comparison: baseline scenario 2 (IOM vs. LG) compared to configuration of "SR vs. SR" (LPSR8, 12 and 14). (a) ROC curves of LPSR8; (b) HD histogram distribution of LPSR8; (c) ROC curves of LPSR12; (d) HD histogram distribution of LPSR12; (e) ROC curves of LPSR14; (f) HD histogram distribution of LPSR14



(a)

(b)

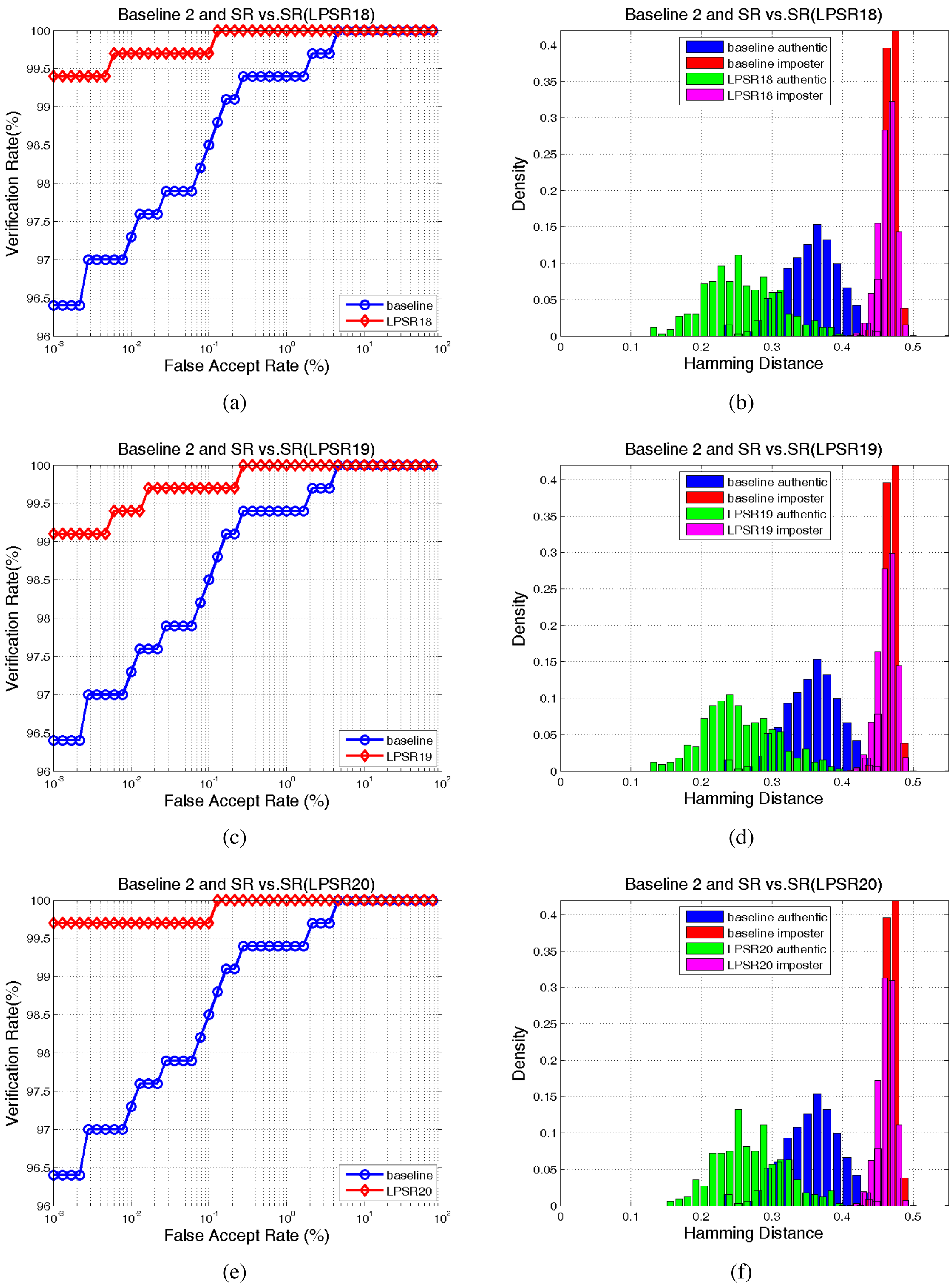
(c)

(d)

(e)

(f)

Figure D.8: Performance comparison: baseline scenario 2 (IOM vs. LG) compared to configuration of "SR vs. SR" (LPSR15, 16 and 17). (a) ROC curves of LPSR15; (b) HD histogram distribution of LPSR15; (c) ROC curves of LPSR16; (d) HD histogram distribution of LPSR16; (e) ROC curves of LPSR17; (f) HD histogram distribution of LPSR17



(a)

(b)

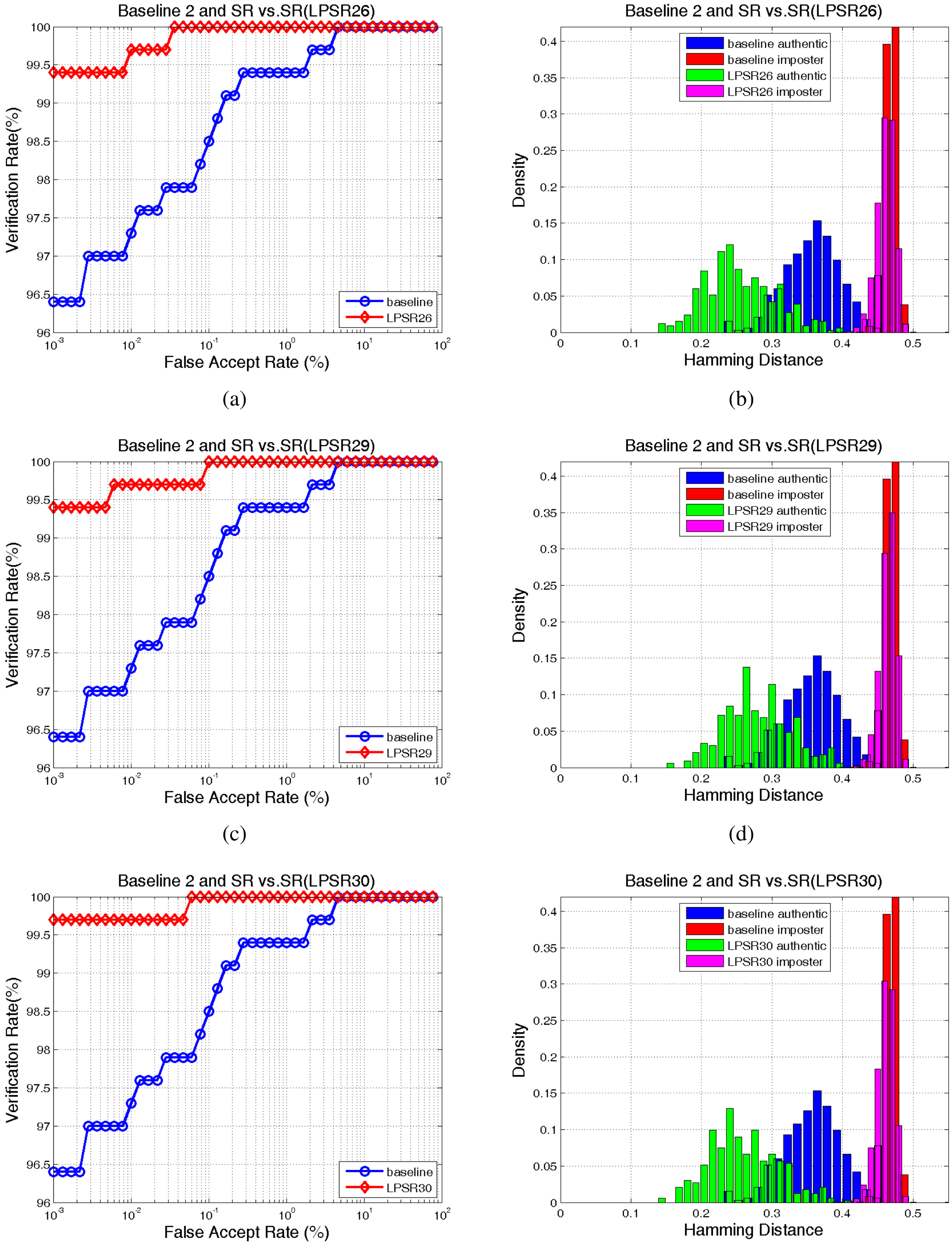
(c)

(d)

(e)

(f)

Figure D.9: Performance comparison: baseline scenario 2 (IOM vs. LG) compared to configuration of "SR vs. SR" (LPSR18, 19 and 20). (a) ROC curves of LPSR18; (b) HD histogram distribution of LPSR18; (c) ROC curves of LPSR19; (d) HD histogram distribution of LPSR19; (e) ROC curves of LPSR20; (f) HD histogram distribution of LPSR20



(a)

(b)

(c)

(d)

(e)

(f)

Figure D.10: Performance comparison: baseline scenario 2 (IOM vs. LG) compared to configuration of “SR vs. SR” (LPSR26, 29 and 30). (a) ROC curves of LPSR26; (b) HD histogram distribution of LPSR26; (c) ROC curves of LPSR29; (d) HD histogram distribution of LPSR29; (e) ROC curves of LPSR30; (f) HD histogram distribution of LPSR30

# Bibliography

[1] A. Jain, A. Ross, and S. Prabhakar, "An introduction to biometric recognition," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 14, no. 1, pp. 4–20, Jan. 2004. 1

[2] J. Daugman, "High confidence visual recognition of persons by a test of statistical independence," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 15, no. 11, pp. 1148–1161, Nov 1993. 1.1, 2.2, 3.1, 3.2, 3.3, 5.7.3

[3] P. Kronfeld, "Gross anatomy and embryology of the eye," *The Eye, H. Davson, Ed.*, 1962. 1.1

[4] "Lg irisaccess 4000," *LG Electronics U.S.A., Inc., http://www.lgiris.com/ps/products/irisaccess4000.htm*. 1.3.2, 5.5.1

[5] E. Hecht, Ed., *Optics*. 1301 Sansome St., San Francisco, CA 94111: Addison-Wesley, 2002, 1301 Sansome St., San Francisco, CA 94111. 1.4.2, 1.4.3, 1.4.3

[6] J. Matey, O. Naroditsky, K. Hanna, R. Kolczynski, D. LoIacono, S. Mangru, M. Tinker, T. Zappia, and W. Zhao, "Iris on the move: Acquisition of images for iris recognition in less constrained environments," *Proceedings of the IEEE*, vol. 94, no. 11, pp. 1936–1947, Nov. 2006. 1.4.9, 2.2, 5.5.1

[7] R. Wildes, "Iris recognition: an emerging biometric technology," *Proceedings of the IEEE*, vol. 85, no. 9, pp. 1348–1363, Sep 1997. 2.1, 2.2, 2.4.1

[8] J. Daugman, "How iris recognition works," *Image Processing. 2002. Proceedings. 2002 International Conference on*, vol. 1, pp. I–33–I–36 vol.1, 2002. 2.1

[9] Y.-P. Huang, S.-W. Luo, and E.-Y. Chen, "An efficient iris recognition system," *Machine Learning and Cybernetics, 2002. Proceedings. 2002 International Conference on*, vol. 1, pp. 450–454 vol.1, 2002. 2.2

[10] Y. Liu, S. Yuan, X. Zhu, and Q. Cui, "A practical iris acquisition system and a fast edges locating algorithm in iris recognition," *Instrumentation and Measurement Technology Conference, 2003. IMTC '03. Proceedings of the 20th IEEE*, vol. 1, pp. 166–168, May 2003. 2.2

[11] H. Sung, J. Lim, J. hyun Park, and Y. Lee, "Iris recognition using collarette boundary localization," *Pattern Recognition, International Conference on*, vol. 4, pp. 857–860, 2004. 2.2

[12] J. Thornton, "Matching deformed and occluded iris patterns: a probabilistic model based on discriminative cues," *PhD thesis, Dept of ECE, Carnegie Mellon University*, 2007. 2.2,

2.4.1, 3.2, 3.4.3, 4.3.2, 5.4

[13] J. S. Lim, “Two-dimensional signal and image processing,” *Prentice Hall signal processing series*, 1990. 2.3.1

[14] R. C. Gonzalez and R. E. Woods, “Digital image processing,” *Prentice Hall, Inc.*, 2008. [Online]. Available: http://www.prenhall.com/gonzalezwoods 2.3.1, 5.4.6.1

[15] “Nist multiple biometric grand challenge,” *National Institute of Standards and Technology, http://face.nist.gov/mbgc/*, 2007. 2.4.1

[16] 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. 2.4.1, 4.3.3

[17] “Iris challenge evaluation,” *National Institute of Standards and Technology, http://iris.nist.gov/ICE/*, 2006. 2.4.1, 3.4.1

[18] L. Ma, T. Tan, Y. Wang, and D. Zhang, “Personal identification based on iris texture analysis,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 12, pp. 1519–1533, 2003. 3.2, 3.3

[19] C. Tisse, L. Martin, L. Torres, and M. Robert, “Person identification technique using human iris recognition,” 2002. [Online]. Available: citeseer.ist.psu.edu/tisse02person.html 3.2, 3.3

[20] J. Daugman, “How iris recognition works,” *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 14, no. 1, pp. 21–30, Jan. 2004. 3.2, 5.7.3

[21] ——, “New methods in iris recognition,” *Systems, Man, and Cybernetics, Part B, IEEE Transactions on*, vol. 37, no. 5, pp. 1167–1175, Oct. 2007. 3.2, 4.3.2

[22] W. Kong and D. Zhang, “Accurate iris segmentation based on novel reflection and eyelash detection model,” *Intelligent Multimedia, Video and Speech Processing, 2001. Proceedings of 2001 International Symposium on*, pp. 263–266, 2001. 3.2, 3.3, 3.4.3, 4.3.2

[23] J. Zuo, N. Kalka, and N. Schmid, “A robust iris segmentation procedure for unconstrained subject presentation,” *Biometric Consortium Conference, 2006 Biometrics Symposium: Special Session on Research at the*, pp. 1–6, 19 2006-Aug. 21 2006. 3.2

[24] E. Krichen, S. Garcia-Salicetti, and B. Dorizzi, “A new probabilistic iris quality measure for comprehensive noise detection,” *Biometrics: Theory, Applications, and Systems, 2007. BTAS 2007. First IEEE International Conference on*, pp. 1–6, Sept. 2007. 3.2

[25] L. Rabiner and B.-H. Juang, *Fundamentals of speech recognition.* Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1993. 3.3

[26] M.-H. Yang and N. Ahuja, “Gaussian mixture model for human skin color and its applications in image and video databases,” *Storage and Retrieval for Image and Video Databases VII*, vol. 3656, no. 1, pp. 458–466, 1998. [Online]. Available: http://link.aip.org/link/?PSI/3656/458/1 3.3

[27] C. Stauffer and W. Grimson, “Adaptive background mixture models for real-time tracking,” *cvpr*, vol. 02, p. 2246, 1999. 3.3

[28] J. Ilonen, J.-K. Kamarainen, H. Kalviainen, and O. Anttalainen, "Automatic detection and recognition of hazardous chemical agents," *Digital Signal Processing, 2002. DSP 2002. 2002 14th International Conference on*, vol. 2, pp. 1345–1348 vol.2, 2002. 3.3

[29] T. Lindh, J. Ahola, J. Kamarainen, V. Kyrki, and J. Partanen, "Bearing damage detection based on statistical discrimination of stator current," *Diagnostics for Electric Machines, Power Electronics and Drives, 2003. SDEMPED 2003. 4th IEEE International Symposium on*, pp. 177–181, Aug. 2003. 3.3

[30] M. Figueiredo and A. Jain, "Unsupervised learning of finite mixture models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 3, pp. 381–396, 2002. 3.3.4

[31] S. Ma, Ed., *Statistical Mechanics*. World Scientific, 1985. 3.4.4.2

[32] S. Gelfand, "Analysis of simulated annealing type algorithms," *PhD thesis, MIT*, 1987. 3.4.4.2

[33] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983. [Online]. Available: http://www.jstor.org/stable/1690046 3.4.4.2

[34] V. Cerny, "Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm," *Journal of Optimization Theory and Applications*, vol. 45, no. 1, pp. 41–51, Jan 1985. 3.4.4.2

[35] V. Granville, M. Krivanek, and J. P. Rasson, "Simulated annealing: A proof of convergence," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 16, no. 6, pp. 652–656, 1994. 3.4.4.2

[36] C. Xu and J. Prince, "Snakes, shapes, and gradient vector flow," *Image Processing, IEEE Transactions on*, vol. 7, no. 3, pp. 359–369, Mar 1998. 4.3.2

[37] "Securimetrics pier device," *SecuriMetrics Inc., http://www.securimetrics.com/solutions/pier.html*. 5.1, 5.5.1

[38] R. Schultz and R. Stevenson, "A bayesian approach to image expansion for improved definition," *Image Processing, IEEE Transactions on*, vol. 3, no. 3, pp. 233–242, May 1994. 1

[39] S. Baker and T. Kanade, "Hallucinating faces," *Automatic Face and Gesture Recognition, 2000. Proceedings. Fourth IEEE International Conference on*, pp. 83–88, 2000. 1

[40] ——, "Limits on super-resolution and how to break them," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 9, pp. 1167–1183, 2002. 1

[41] W. T. Freeman, E. C. Pasztor, and O. T. Carmichael, "Learning low-level vision," *Int. J. Comput. Vision*, vol. 40, no. 1, pp. 25–47, 2000. 1

[42] W. T. Freeman, T. R. Jones, and E. C. Pasztor, "Example-based super-resolution," *IEEE Computer Graphics and Applications*, vol. 22, no. 2, pp. 56–65, 2002. 1

[43] R. Y. Tsai and T. S. Huang, "Multiframe image restoration and registration," *Advances in Computer Vision and Image Processing*, vol. 1, pp. 317–339, 1984. 2

[44] M. Elad and A. Feuer, "Restoration of a single superresolution image from several blurred,

noisy, and undersampled measured images," *Image Processing, IEEE Transactions on*, vol. 6, no. 12, pp. 1646–1658, Dec 1997. 2

[45] P. Vandewalle, S. Süsstrunk, and M. Vetterli, "A frequency domain approach to registration of aliased images with application to super-resolution," *EURASIP J. Appl. Signal Process.*, vol. 2006, no. 1, pp. 233–233, uary. 2, 5.3.1

[46] A. Zomet, A. Rav-Acha, and S. Peleg, "Robust super-resolution," *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, vol. 1, pp. I–645–I–650 vol.1, 2001. 2, 5.3.2

[47] L. Lucchese and G. Cortelazzo, "A noise-robust frequency domain technique for estimating planar roto-translations," *Signal Processing, IEEE Transactions on*, vol. 48, no. 6, pp. 1769–1786, Jun 2000. 2

[48] D. Keren, S. Peleg, and R. Brada, "Image sequence enhancement using sub-pixel displacements," *Computer Vision and Pattern Recognition, 1988. Proceedings CVPR '88., Computer Society Conference on*, pp. 742–746, Jun 1988. 2, 5.3.1, 5.3.2

[49] M. Irani and S. Peleg, "Improving resolution by image registration," *CVGIP: Graph. Models Image Process.*, vol. 53, no. 3, pp. 231–239, 1991. 2, 5.3.2

[50] Y. Altunbasak, A. Patti, and R. Mersereau, "Super-resolution still and video reconstruction from mpeg-coded video," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 12, no. 4, pp. 217–226, Apr 2002. 3

[51] S. Borman and R. Stevenson, "Simultaneous multi-frame map super-resolution video enhancement using spatio-temporal priors," *Image Processing, 1999. ICIP 99. Proceedings. 1999 International Conference on*, vol. 3, pp. 469–473 vol.3, 1999. 3

[52] S. Farsiu, M. Robinson, M. Elad, and P. Milanfar, "Fast and robust multiframe super resolution," *Image Processing, IEEE Transactions on*, vol. 13, no. 10, pp. 1327–1344, Oct. 2004. 5.2

[53] A. Rosenfeld, Ed., *Multiresolution Image Processing and Analysis.* Berlin: Springer, 1984, leesberg, VA, July 1982. 5.3.1

[54] B. Reddy and B. Chatterji, "An fft-based technique for translation, rotation, and scale-invariant image registration," *Image Processing, IEEE Transactions on*, vol. 5, no. 8, pp. 1266–1271, Aug 1996. 5.3.1

[55] R. Kerekes, B. Narayanaswamy, J. Thornton, M. Savvides, and B. V. K. V. Kumar, "Graphical model approach to iris matching under deformation and occlusion," *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, vol. 0, pp. 1–6, 2007. 5.4

[56] J. Thornton, M. Savvides, and B. V. Kumar, "A bayesian approach to deformed pattern matching of iris images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 4, pp. 596–606, 2007. 5.4

[57] M. Bajracharya, A. Diaz-Calderon, M. Robinson, and M. Powell, "Target tracking, approach, and camera handoff for automated instrument placement," March 2005, pp. 52–59. 5.4.5

[58] Y. Z. R. D. D. Christopher J. Hardy, Manojkumar Saranathan, "Coronary angiography by

real-time mri with adaptive averaging," vol. 44, no. 6. Wiley-Liss, Inc., 2000, pp. 940–946. 5.4.5

[59] R. Cutler and L. Davis, "Robust periodic motion and motion symmetry detection," *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, vol. 2, pp. 615–622 vol.2, 2000. 5.4.5

[60] ——, "Robust real-time periodic motion detection, analysis, and applications," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 22, no. 8, pp. 781–796, Aug 2000. 5.4.5

[61] D. D. Diel, P. DeBitetto, and S. Teller, "Epipolar constraints for vision-aided inertial navigation," vol. 2, Jan. 2005, pp. 221–228. 5.4.5

[62] D. Grest, J. michael Frahm, and R. Koch, "A color similarity measure for robust shadow removal in real time," pp. 253–260, 2003. 5.4.5

[63] A. Haasch, N. Hofemann, J. Fritsch, and G. Sagerer, "A multi-modal object attention system for a mobile robot," Aug. 2005, pp. 2712–2717. 5.4.5

[64] R. Manzke, T. Köhler, T. Nielsen, D. Hawkes, and M. Grass, "Automatic phase determination for retrospectively gated cardiac ct," *Medical Physics*, vol. 31, no. 12, pp. 3345–3362, 2004. [Online]. Available: http://link.aip.org/link/?MPH/31/3345/1 5.4.5

[65] H. Najafi, N. Navab, and G. Klinker, "Automated initialization for marker-less tracking: a sensor fusion approach," Nov. 2004, pp. 79–88. 5.4.5

[66] I. Nesnas, M. Bajracharya, R. Madison, E. Bandari, C. Kunz, M. Deans, and M. Bualat, "Visual target tracking for rover-based planetary exploration," vol. 2, March 2004, pp. 747–761 Vol.2. 5.4.5

[67] O. Williams, A. Blake, and R. Cipolla, "Sparse bayesian learning for efficient visual tracking," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 27, no. 8, pp. 1292–1304, Aug. 2005. 5.4.5

[68] J. P. Lewis, "Fast normalized cross-correlation," in *Vision Interface*. Canadian Image Processing and Pattern Recognition Society, 1995, pp. 120–123. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.21.6062 5.4.5

[69] ——, "Fast template matching," *Vision Interface*, vol. 95, pp. 120–123, 1995. 5.4.5

[70] T. Hofmann, "Probabilistic latent semantic indexing," 1999, pp. 50–57. 6.2