ANALYSIS OF METHODS FOR FINGER VEIN RECOGNITION

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF INFORMATICS INSTITUTE
OF
MIDDLE EAST TECHNICAL UNIVERSITY

FARIBA YOUSEFI

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
MEDICAL INFORMATICS

NOVEMBER 2013

Approval of the thesis:

ANALYSIS OF METHODS FOR FINGER VEIN RECOGNITION

submitted by **FARIBA YOUSEFI** in partial fulfillment of the requirements for the degree of **Master of Science in Medical Informatics Department, Middle East Technical University** by,

Prof. Dr. Nazife Baykal
Dean, Graduate School of **Informatics Institute** ________________

Assist. Prof. Dr. Yeşim Aydın Son
Head of Department, **Medical Informatics** ________________

Assist. Prof. Dr. Sinan Kalkan
Supervisor, **Computer Engineering Dept., METU** ________________

Assoc. Prof. Dr. Alptekin Temizel
Co-supervisor, **Work Based Learning Studies Dept., METU** ________________

Examining Committee Members:

Prof. Dr. Ünal Erkan Mumcuoğlu
Medical Informatics Dept., METU ________________

Assist. Prof. Dr. Sinan Kalkan
Computer Engineering Dept., METU ________________

Assoc. Prof. Dr. Alptekin Temizel
Work Based Learning Studies Dept., METU ________________

Assist. Prof. Dr. Aybar Can Acar
Medical Informatics Dept., METU ________________

Assist. Prof. Dr. Banu Günel
Information Systems Dept., METU ________________

Date: **13.11.2013**

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: FARIBA YOUSEFI

Signature :

ABSTRACT

ANALYSIS OF METHODS FOR FINGER VEIN RECOGNITION

Yousefi, Fariba
M.S., Department of Medical Informatics
Supervisor : Assist. Prof. Dr. Sinan Kalkan
Co-Supervisor : Assoc. Prof. Dr. Alptekin Temizel

November 2013, 103 pages

A decade ago, it was observed that every person has unique finger-vein patterns and that this could be used for biometric identification. This observation has led to successful identification systems, which are currently used in banks, hospitals, state organizations, etc. For the feature extraction step of the finger-vein recognition, which is the most important step, popular methods such as Line Tracking (LT), Maximum Curvature (MC) and Wide Line Detector (WL) are used in the literature. Among these, the LT method is very slow in the feature extraction phase. Moreover, LT, MC and WL methods are susceptible to rotation, translation and noise. To overcome these drawbacks, this thesis proposes using some popular feature descriptors widely-used in Computer Vision or Pattern Recognition (CVPR) methods. The CVPR descriptors tested include, Fourier descriptors (FD), Zernike moments (ZM), Local Binary Patterns (LBP), Global Binary Patterns (GBP) and Histogram of Oriented Gradients (HOG) which have not been applied to the finger-vein recognition problem before. The thesis compares these descriptors against LT, MC and WL and analyze their running time, performance and resilience against rotation, translation and noise.

Keywords: Finger-vein recognition, Fourier Descriptors, Zernike Moments, Local Binary Patterns, Histogram of Oriented Gradients

ÖZ

PARMAK DAMAR YAPISI TANIMA METODLARININ ANALİZİ

Yousefi, Fariba

Yüksek Lisans, Tıp Bilişimi Bölümü

Tez Yöneticisi : Yrd. Doç. Dr. Sinan Kalkan

Ortak Tez Yöneticisi : Doç. Dr. Alptekin Temizel

Kasım 2013 , 103 sayfa

On yıl önce, her insanın benzersiz parmak damar yapısı olduğu ve bunun biyometrik kimlik için kullanılabileceği gözlemlenmiştir. Bu gözlem, şu anda bankalarda, hastanelerde ve devlet kurumlarında kullanılan başarılı tanımlama sistemlerine öncülük etmiştir. Daha önceki çalışmalarda, önemli bir adım olan parmak damar algılama probleminin öznitelik çıkarma adımı için Çizgi Takibi (LT), Maksimum Eğrilik (MC), Geniş Çizgi Bulucu (WL) gibi popüler yollar kullanılmaktadır. LT metodu bunlar arasında öznitelik çıkarma aşamasında en yavaş olanıdır. Buna ek olarak, LT, MC ve WL metodları ise döndürülme, taşıma ve gürültü gibi etkenlerden kolay etkilenirler. Bu tez, bahsedilen eksikliklerin üstesinden gelmek için bilgisayarlı görü veya örüntü tanıma (CVPR) metodlarında çokça kullanılan bazı popüler öznitelik betimleyicileri kullanmayı önermektedir. Bu CVPR betimliyicilerinden Fourier Tanımlayıcıları (FD), Zernike Momentleri (ZM), Yerel İkili Örüntüler (LBP), Küresel İkili Örüntüler (GBP) ve daha önce parmak damar algılama problemine uygulanmamış olan Yönelimli Bayır Histogramı (HOG) test edilmiştir. Bu tez, tüm bu bahsedilen betimleyicileri LT, MC, ve WL ile kıyaslamıştır ve çalışma zamanlarını, performanslarını ve döndürülme, taşıma ve gürültüye olan dayanıklılıklarını analiz etmiştir.

Anahtar Kelimeler: Parmak-damar tanıma, Fourier Tanımlayıcıları, Zernike Momentleri, Yerel ikili Örüntüler, Yönelimli Bayır Histogramı

To my life

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to my supervisor Assist. Prof. Dr. Sinan Kalkan for the support, patience, motivation and assistance through the learning process of this master's thesis.

I would also like to thank my co-supervisor Assoc. Prof. Dr. Alptekin Temizel for his help and advice through my study.

I am very thankful to KOVAN lab members especially Güner, Sertaç and Hande for their helps. Güner without you my Turkish would be perfect! Once again thank for all your help.

I would like to express my gratitude towards my family for their best wishes and support especially my mother.

My sincere thanks go to my fiance for his patience, being with me in good and bad times, and being supportive.

My special thanks go to my friends whom were always there for me during my hard times especially Yousef, Negin, Mona and Pardis.

Finally, I would like to thank my loved ones, who have cared about me throughout entire process, and were patient enough to listen to me. I will be grateful forever for your love.

TABLE OF CONTENTS

CHAPTERS

APPENDICES

LIST OF TABLES

LIST OF FIGURES

List of Algorithms

LIST OF ABBREVIATIONS

χ^2	Chi-Square distance
LT	Line Tracking
MC	Maximum Curvature
WL	Wide Line Detector
FD	Fourier Descriptors
ZM	Zernike Moments
FV	Finger Vein
FP	Finger Print
FG	Finger Geometry
Id	Identifier
HP	Hewlett-Packard
Fvr	Favorite
img	Image
LBP	Local Binary Patterns
GBP	Global Binary Patterns
HOG	Histogram of Oriented Gradients
LDP	Local Derivative Patterns
CGF	Circular Gabor Filter
FAR	False Acceptance Rate
FRR	False Rejection Rate
EER	Equal Error Rate
ROC	Receiver Operating Characteristic
PCA	Principle Component Analysis
PIN	Personal Identification Number
EMD	Earth Mover's Distance
CVPR	Computer Vision and Pattern Recognition

CHAPTER 1

INTRODUCTION

1.1 Identification and Authentication Process

Identification and authentication are processes that are used to find out the true identity of an individual [1]. In a virtual world users have unique identifiers (Id's) like email addresses or employee numbers, etc. However unique identifiers are not secure enough and without any permission, they can be used by others. Even though these identifiers can have passwords to be more secure, they can still be robbed easily and furthermore, complicated passwords are difficult to memorize. For these reasons, alternative authentication methods, especially based on biometry have emerged and are widely in use.

Most authentication systems suffer from two types of errors, false acceptance of impostors and false rejection of authorized users and the challenge is to lower the error as much as possible [1]. In brief, authentication can utilize;

- What the user knows: like information that belongs to the user and should be kept as a secret, such as passwords and Personal Identification Numbers (PIN).
- What the user has: refers to physical properties like keys, smart cards, or cards with magnetic strip.
- What the user is: refers to biometrics or behavioral characteristics of people such as signature, fingerprint, etc.

Among these, the first one is the most widely used and the easiest one. In this mode users are asked to make up passwords that are hard to guess and answer some secret questions. This method's security depends on the password and how difficult it is to guess, however, users generally choose weak passwords because they are easier to remember. In other words, this method is not very suitable for high security applications.

The second authentication method is believed to be more trustworthy than the first method. In this form, users have their keys and/or smart cards, however individuals may share their smart cards, lose them or their cards can be stolen. The advantage of this method over the first method is that when your card is stolen, you will notice that you no longer have the physical card and you can easily ask the system administrator to cancel the old card and issue a new one.

The last method is expected to be the best one which is based on individuals' physical or behavioral characteristics and in contrary to the other two methods, users will not forget or lose their credentials which will be discussed in detail in the next chapter.

There are many biometric technology types such as fingerprint[1], finger vein, facial scan, iris scan and voice scan, each of which has their own advantages and disadvantages. However there is no best method among them, depending on the requirements of the application and people that would use these systems, the preferences would be different.

1.2 Motivation, Scope and Contributions of the Thesis

Finger vein recognition is one of the forefront methods in biometric technology in recent years. Many successful methods such as Line Tracking (LT) [2], Maximum Curvature (MC) [3] and Wide Line Detector (WL) [4] have been proposed for finger vein recognition. Among these methods, LT has a very slow matching and feature extraction phase. Moreover, LT, MC and WL are rotation dependent, and they are affected by image noise. To overcome these drawbacks, using some popular feature descriptors widely used for several Computer Vision or Pattern Recognition (CVPR) are proposed. These descriptors include Fourier Descriptors (FD) [5], Zernike Moments (ZM) [6], Histogram of Oriented Gradients (HOG) [7], Local Binary Patterns (LBP) [8] and Global Binary Patterns (GBP) [9]. Among these, FD, ZM, HOG, LBP and GBP have not been applied to the finger vein recognition before. These descriptors are compared against LT, MC and WL. The novelty of the thesis is in (i) applying new feature extraction methods that have not been used for finger vein recognition before and (ii) evaluating the performance of all these methods under translation, rotation and noise. The focus is on the "feature extraction" step, and the preprocessing step is kept as simple as possible. As for matching, the matching method specific to LT, MC and WL which is called mismatch ratio is used and for all other descriptors, three different distance metrics called Euclidean distance, χ^2 (Chi-Square distance) and Earth Mover's Distance (EMD) have been applied and compared to each other. For performance evaluation, the SDUMLA-HMT finger vein database that is publicly available[2] is used. The contributions of the thesis have been accepted for "the 9th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications, VISAPP"[3].

1.3 Outline of the Thesis

The rest of the thesis is organized as follows. Chapter 2 describes background information about biometrics and different biometric systems' advantages and disadvantages. Chapter 3 describes feature extraction methods that has been applied to the finger vein in this thesis like LT, MC, WL, FD, ZM, HOG, LBP and GBP and different matching methods such as Euclidean distance, χ^2 and EMD are analyzed. Moreover, some state of the art studies are discussed in this chapter. Chapter 4 compares the performance of the mentioned methods under rotation, translation and noise, then their running times and accuracies are evaluated and compared against the results in the literature. Finally chapter 5 concludes the thesis with a summary of the contributions and an outlook for how the work can be improved further as future work.

[1] The word "finger vein" is written separately unlike "fingerprint". See: http://www.merriam-webster.com/dictionary/fingerprint

[2] http://mla.sdu.edu.cn/sdumla-hmt.html

[3] http://www.visapp.visigrapp.org/

CHAPTER 2

BIOMETRICS

Biometrics is identifying humans by their physiological, behavioral and biological characteristics [1]. Biometrics can be divided into two categories: physiological biometrics and behavioral biometrics [1]. Physiological biometrics are those which recognize individuals from physiological or biological attributes like face, iris, fingerprint, finger vein, hand geometry, etc. Behavioral biometrics on the other hand, are those which recognize individuals from human attitudes such as hand writing, signature or voice recognition. Figures 2.1 and 2.2 illustrate enrollment to and authentication with the biometric system.

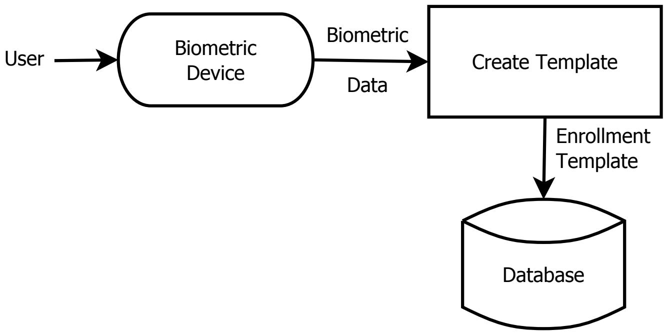


Figure 2.1: Enrollement to a biometric system: First, the biometric data is captured. Then, extracted information is stored in an enrollment template and the template is stored in the database (adapted from [10]).

2.1 Some Advantages and Disadvantages of Biometrics

Tremendous amount of biometric identification systems are in use today. Many of such systems are based on fingerprint, face, iris, voice or finger vein. Biometric technology has many advantages in comparison to the other techniques (e.g., password, PIN, keys and cards). Reasons for using biometric systems include:

- Ease of use: Biometric systems are mostly famous for their convenience to the end users. Individuals do not need to remember different passwords or carry any identity cards; in both cases they can forget the password or forget to carry the card.

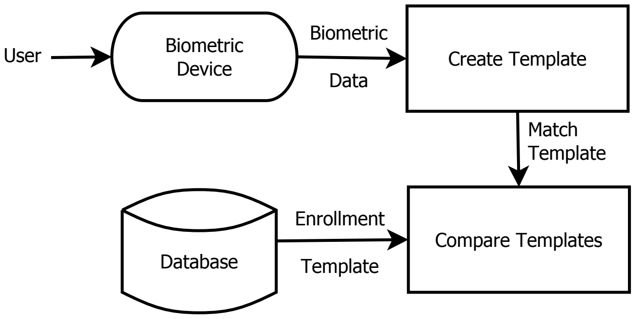


Figure 2.2: Authentication with a biometric system: First, the biometric data is captured. Then, extracted information is stored in an enrollment template and the template is compared to the one in the database (adapted from [10]).

- Enhanced security: Sharing passwords or losing cards can lead to trouble and insecurity, however, using unique biometric identifier reduces the stated risks. Moreover, duplicating biometric identifiers is too difficult or impossible.
- Low cost: Hardware and software improvements have led to decrease in the cost of biometric systems, making them affordable to many organizations and companies. For example, HP and Lenovo laptops are mostly equipped with fingerprint scanners nowadays [1].

In addition to all the advantages of biometric systems, there are some disadvantages that should be mentioned:

- Privacy: Biometric systems gather information of people which is used for identification purposes, however, secure storage of such valuable information against third parties is very crucial, just the same as passwords and PINs, etc. [1].
- Physical harm: Most people believe that touching the same device that strangers touch can lead to diseases. Furthermore, it is believed that iris recognition may damage vision. However, in reality non of the biometric systems caused any damages to human beings ever.
- Fake biometrics: One of the drawbacks of biometric systems is that artificial or simulated samples can be made. There are documents that prove the existence of artificial fingerprints. To overcome these kinds of problems, liveliness detection can be performed which controls if the presented sample is from a live human being or not.
- Stolen credentials: Even though stealing biometric attributes of people is very difficult, there are cases where body parts have been stolen to trick biometric systems and in such cases, biometric data cannot be reset or replaced like passwords.

2.2 Biometric Characteristics

There are different types of biometric systems based on iris, fingerprint, facial recognition and finger vein, etc [11]. Each biometric system calculates different characteristics and then compares them with its database to check if the individual should be given access to the system or not. There are various types of biometric characteristics, however, some important types are:

- Uniqueness: Each person has unique biological characteristics that makes him/her different from others. These particular biometrics can be divided into three groups, called genetics, phenotypic and behavioural [11].
  - Genetics: Features like hair and eye color that the user has inherited from his/her parent.
  - Phenotypic: Certain biometric features like iris patterns, fingerprint patterns and vein network.
  - Behavioural: Features such as signature or voice can be categorized into this group.
- Performance: Measures how much time and equipment is needed for calculating the matching process. Some techniques are cheaper and faster, like fingerprint or finger vein. On the other hand, some techniques are costly and slow, like DNA based biometric systems.
- Accuracy: Measures how accurate the system is working, e.g. how many people are by mistake rejected while others are falsely accepted.
- Acceptability: Acceptability shows if the users are comfortable using the biometric systems or not, for example not many people are comfortable using retina recognition.

2.3 Biometric System Architecture

Four main components of each biometric system are obtaining data, processing, making decision and storage [11] (see Figure 2.3):

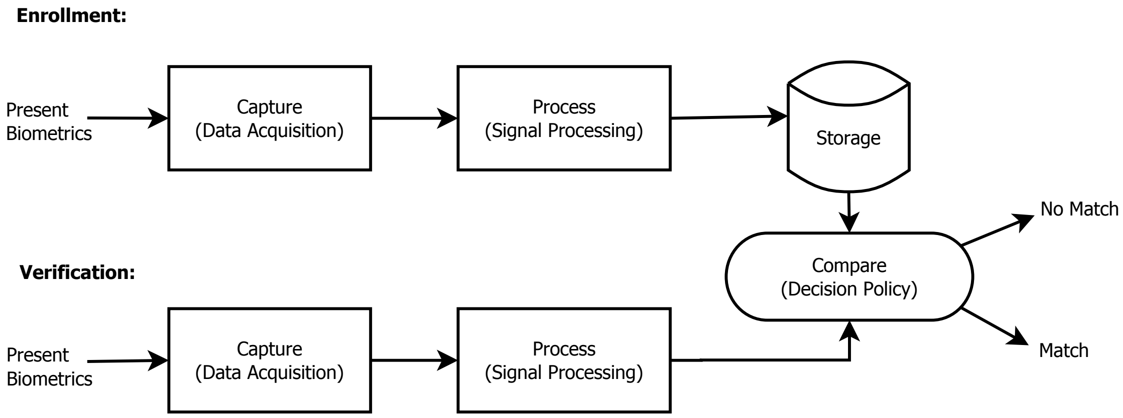

Enrollment:
Present Biometrics
Capture (Data Acquisition)
Process (Signal Processing)
Storage
No Match
Compare (Decision Policy)
Match
Verification:
Present Biometrics
Capture (Data Acquisition)
Process (Signal Processing)

Figure 2.3: Biometric System (adapted from [1]).

- Data acquisition: In this step the biometric data is captured digitally and then sent to the signal processing step.
- Signal processing: Raw data is received in this step and the preprocessing and/or feature extraction methods are performed.

- Decesion policy: Final decision would be made in this step to see if the enrollment should be accepted or rejected.
- Storage: In this step, biometric template is stored for future matching.

2.4 Biometric Types

As mentioned before, there are different types of biometrics, and humans can be recognized by their fingers, hands, face, eyes, and voice, some of which are more widely used than others. In this section each technique will be discussed briefly and its advantages and disadvantages will be mentioned.

2.4.1 Facial Scan

Facial scan technology is mostly used for identification of individuals instead of verification[1]. This technology uses some of the important features of the face, like, eyes, nose, lips, and so on [12]. One of the advantages of facial scan technology is identifying individuals from distance, therefore, it will not annoy people by touching any device. Moreover, images can be captured by different devices like video cameras. On the other hand, this technology has some disadvantages. For example, the quality of the captured image is dependent on lightening, background and angle of the camera. Moreover, users' appearance can change over time, and having glasses, beard, mustache, make-up or different hair styles, would have an impact on accuracy. Furthermore, having higher accuracy requires saving different images of users which in turn requires more memory space.

2.4.2 Iris Scan

Iris scan technology is used for both identification and verification based on unique features of irises. This technology uses patterns that form the visual part of the iris to differentiate between humans [12]. Some advantages are that (i) this system has smallest False Acceptance Rate (FAR) among the other biometric systems, which is critical for high secure applications and (ii) iris does not change in time like some other biometric types (like voice that can change). However, positioning of the head and eye are very important to get accurate results and people should not move their heads during data acquisition. Iris is a very small area, therefore, captured image should be very high resolution which makes the device expensive. Individuals feel discomfortable using this device even though they do not know infrared scan is used during the process [13].

2.4.3 Voice Scan

Various vocal qualities such as frequency, short-time spectrum of dialogue and spectograms (time-frequency-energy patterns) are used for voice scan [12]. The devices e.g. microphones used for this technology are quite cheap. This system lets the users select an expression and repeat it during identification and verification which reduces the risk of impostors guessing the same phrase and entering into the system. However noise and echoes can reduce the system accuracy. Moreover voice can change

[1] The word "identification" has different meaning than "verification". See http://www.merriam-webster.com/dictionary/identification and http://www.merriam-webster.com/dictionary/verification

during illness or in different moods, making authentication problematic. Enrollment to the voice-based system is longer than other biometric systems because users have to repeat the phrase for many times.

2.4.4 Fingerprint Recognition

Fingerprints are the inherent part of humans and they are unique for each person. Even twins have different fingerprints [14]. Using the patterns found on the finger tips, fingerprint biometric data is captured. As with every technology, using fingerprint for biometrics has its own advantages and disadvantages. Age of the user, dryness and wetness of the finger, temporary or permanent cuts on the finger, or dirt on the fingerprint scanner all cause the fingerprints to have weak patterns of ridge and valleys that can degrade the performance of a fingerprint biometric system. The mentioned issues lead to poor recognition rates in fingerprint identification system and users would experience difficulties with the system's authentication process. Furthermore, as discussed before there can be artificial fingers which threaten system's security. For creating artificial fingers a copy of the real fingerprint is required which can be obtained when the victim is touching objects like glasses and can be replicated in gelatin. For making the systems secure against artificial fingers, liveliness detection can be used. Liveliness detection searches for certain attributes which can only be found on the live fingers such as the pulse in the fingertip, however, individuals have different heart rate which can be different from time to time and for detecting the fingertip pulse, users need to keep their fingers on the sensor without moving for few seconds. However, there are artificial fingers called wafer thin artificial fingers, that can fool such a system by producing pulse.

2.4.5 Finger-Vein Recognition

Disadvantages of fingerprint technology made scientists to think about using what is underneath the skin. Under the skin there are blood vessels which are unique to individuals (even in twins) [1] and this uniqueness made a new biometric system based on finger veins. Biometrics based on veins, i.e., vascular biometrics are not limited to the fingers. Retina, face and hands can be identified using vascular properties too, however, the hardware devices used for finger vein identification are more preferred than the others because people are used to using their fingers for identification already. For capturing a vascular network, hemoglobin plays an important role by absorbing infrared light, and after absorbing infrared light vein patterns are captured (see Figure 2.4).

Distance is very important in absorbing infrared light between skin and vessels: bigger distance leads to more noise in the captured image. Palms, back of the hands and fingers can be used as biometric data, however, people mostly prefer to use their fingers.

2.4.5.1 Devices for Finger-vein Image Acquisition

Finger-vein biometric systems use infrared (IR) light to capture blood vessels, however, the position of infrared light source affects the quality of the images. Moreover, the image acquisition device should be small and cheap, and it should provide high resolution images. In captured images, the veins appear as gray patterns. As can be seen in Figure 2.5 finger is placed between the Infrared Light Emitting Diodes (IR-LEDs) and imaging device.

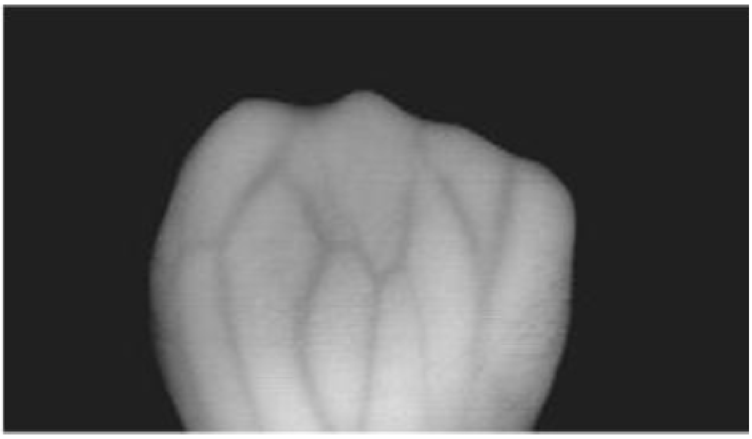
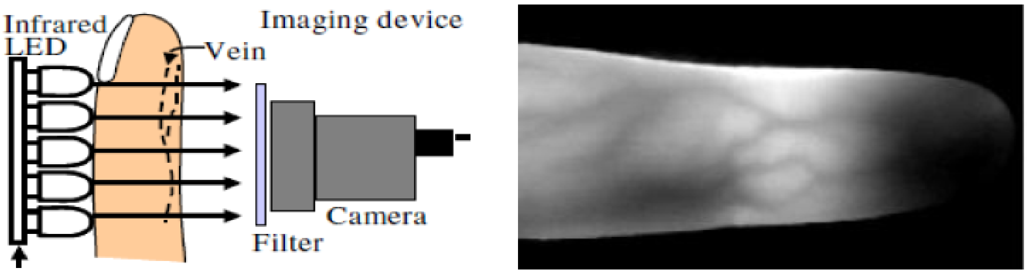


Figure 2.4: Vascular network captured using infrared light (Source: [1]).



(a)

(b)

Figure 2.5: (a) A finger-vein image capturing device, (b) a sample captured image (Source: [2]).

2.4.5.2 Some advantages and disadvantages of Finger-Vein Systems

Some of the advantages of the finger vein systems include:

- Internal nature: Vein patterns are inside the skin and can not be seen by naked eye, therefore, damaged skin will not reduce the chance of finding veins behind the skin. Furthermore, dry, wet or dirty hands would not affect the system.
- Duplicate protection: Vein patterns are difficult to copy because blood needs to flow during image capturing. Scientists in Hitachi proved that it is impossible to cut the finger and register it to the system because blood will seep out [1].
- Hygienic readers: In contrast to fingerprint and hand geometry systems, readers are believed to be free of germs because users do not touch the sensor.
- Usability: These systems are very easy to use.
- No cultural resistance.
- Uniqueness: Finger veins are unique even between twins and do not change by aging.

- Fast processing: There are algorithms that can be processed very fast by the system.
- Harmless: No one reported any side effect of using finger veins' systems on human health.
- Reliability: This system provides trustworthy results that can be used in high secure applications.

Some of the disadvantages of the finger-vein systems include:

- Light controlling: Lights can affect the system, however, for external uses there are covers to solve this problem.
- Alignment: There are finger guidance for putting the finger correctly inside the reader because misalignment would capture wrong patterns.

2.5 Evaluation of Biometric Authentication Systems

Important metrics and methods used in the evaluation of biometric systems are discussed in this section. These are False Acceptance Rate (FAR), False Rejection Rate (FRR), Equal Error Rate (EER) and Receiver Operating Characteristic (ROC) curve [10].

2.5.1 False Acceptance Rate and False Rejection Rate

Biometric systems evaluate a matching score to recognize if the person that tries to enter into the system is a genuine user or an impostor. Selecting the proper threshold (twenty threshold values are used in this study and different threshold values are calculated by dividing the difference between the maximum and minimum of the distances by twenty) is important for matching, though, if the threshold is too low, genuine users will be rejected and if the threshold is too high then impostors would enter the system. Therefore, depending on the security requirements, the threshold should be set properly between them. FAR and FRR metrics are defined as follows:

$$FAR = \frac{\text{number of succesful authentications by impostors}}{\text{number of attemps at authentication by unauthorized users}}. \tag{2.1}$$

$$FRR = \frac{\text{number of failed attemps at authentication by authorized users}}{\text{number of attemps at authentication by genuine users}}. \tag{2.2}$$

Both FAR and FRR are dependent on the threshold; by increasing the threshold, one will be decreased and the other will be increased and vice versa.

2.5.2 Equal Error Rate

As mentioned before, by decreasing the threshold more authorized users will be rejected, and at the same time less unauthorized users will be accepted. At some point, FAR and FRR become equal. The value of FAR and FRR at that point is called the Equal Error Rate. EER does not give any information about the values of FAR and FRR at different thresholds, however, can be used to compare different biometric systems.

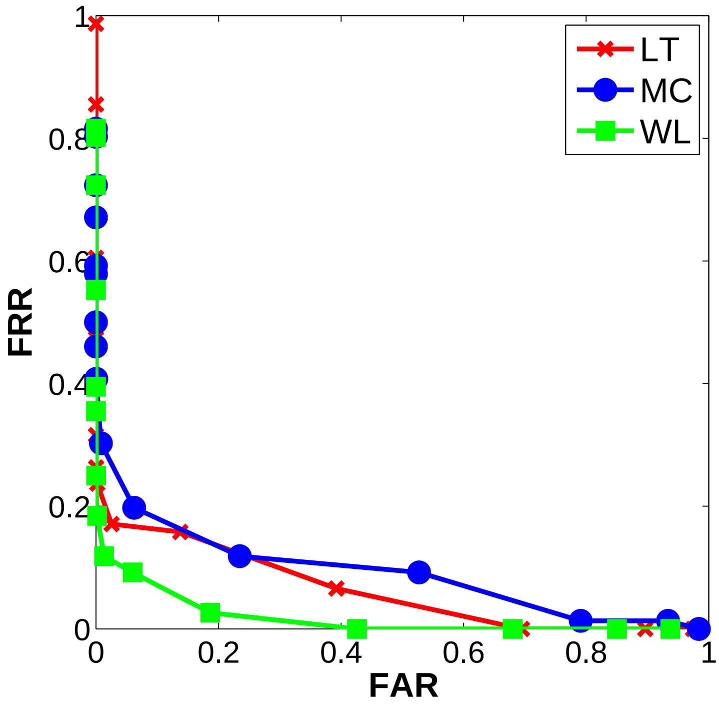


Figure 2.6: Sample ROC Curves.

2.5.3 Receiver Operating Characteristic Curve

A Receiver operating characteristic (ROC) curve displays FAR versus FRR. False accepted impostors rate is shown on the x axis of the curve and false rejected genuine rate is displayed on the y axis of the curve. A ROC curve is useful for summing up the performance of biometric systems. Figure 2.6 shows sample ROC curves.

FAR and FRR are functions of a threshold T, and can be shown as follows:

$$ROC(T) = (FAR(T),\ FRR(T)) \rightarrow \begin{cases} (0,1) & as\ T \rightarrow 0 \\ (1,0) & as\ T \rightarrow \infty. \end{cases} \tag{2.3}$$

As can be seen in Equation 2.3, FAR and FRR are opposite of each other, as one of them increases, the other one decreases.

CHAPTER 3

FINGER-VEIN RECOGNITION METHODS

As discussed in Chapter 2, biometric human identification has become very important due to pitfalls even in extreme security systems, and alternative methods that can be used in place of or together with fingerprint have been being sought during the last decade. It has been shown that finger vein pattern is distinctive enough for human biometric identification [15] and with the advent of technology, and especially since finger vein is more difficult to manipulate compared to, e.g., fingerprints, many successful applications have been seen.
Existing studies regarding finger vein feature extraction can be divided into three main categories (without aiming to be comprehensive):

i) Those that use filtering or transforming methods: Such methods mainly use Gabor filters to extract filter responses using the Gabor function at different scales and/or orientations [16], [17], [18] and [19]. Steerable filters [20] and wavelets [21] are used the same as Gabor filter for image enhancement or feature extraction. After filtering, from the filter responses, using histograms, minimum-maximum or average values, features are extracted, and these features are usually compared using Euclidean distance or Cosine distance [17]. Moreover, Fourier transform [22] can be applied to the finger vein images and the phase components of the images can be taken as feature vectors.

ii) Those that track finger veins, segment the pixels corresponding to the veins, and represent or directly compare these pixels [2], [23].

iii) Those that use descriptive feature extraction methods such as Local Binary Patterns [24], [25] or Local Derivative Patterns [24]. In the literature Kang et al. [26] used LDP for finger vein recognition and Mirmohamadsadeghi et al. [27] used LDP for palm vein recognition.

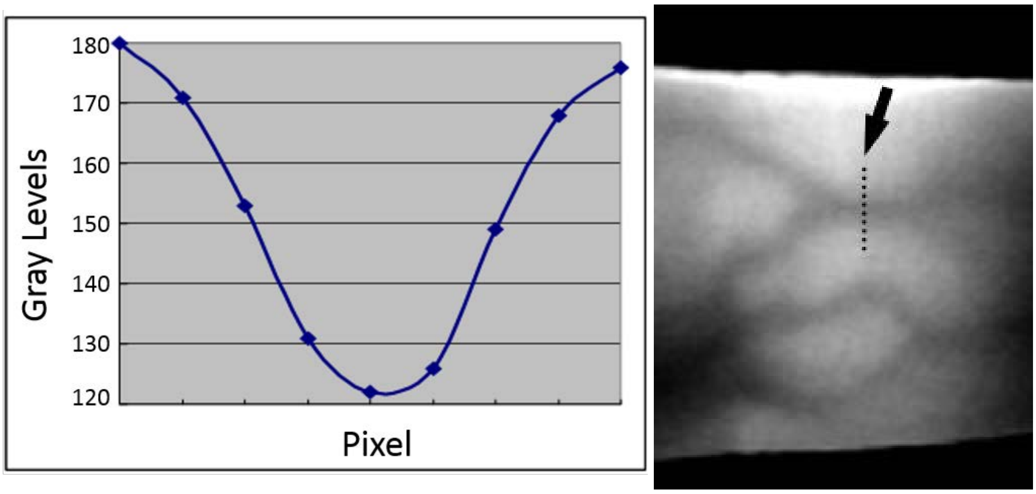
Moreover, there are machine learning methods that can be applied to the finger vein for feature extraction such as neural networks [28] and Principle Component Analysis (PCA) [29].

3.1 Finger Vein Recognition Algorithms Analyzed in this Study

For finger vein extraction, the literature mostly uses methods such as Line Tracking (LT), Maximum Curvature (MC) and Wide Line Detector (WL) and these methods are dependent to translation and rotation and sensitive to noise. To overcome these drawbacks, some popular feature descriptors which are widely used for several computer vision and pattern recognition (CVPR) problems in the literature are used in this thesis. Moreover, various matching methods have been used for matching the features extracted from each method. In this study, for LT, MC and WL, the so-called mismatch ratio is used in which a query map is compared against the stored map pixel by pixel (like in template matching). For FD, ZM, HOG, LBP and GBP, distance metrics like Euclidean distance, χ^2 and Earth Mover's Distance are used.

3.1.1 Line Tracking

The Line Tracking method of Miura et al. [2] [30] is one of the leading approaches used in finger-vein extraction. The method exploits the fact that, due to the light-absorbing nature of the finger veins, they appear darker in the image, making them look like valleys in the infrared image (see Figure 3.1). The LT method is based on randomly finding a pixel in a valley and tracking the pixels along the valley as much as possible. To determine whether a pixel is on a valley (i.e., a part of the finger vein), the LT method checks whether the cross-section $s - p - t$ which is orthogonal to the center pixel, forms a valley in intensity values (see Figure 3.2). The method tracks "valley pixels" and if the pixel is not on a valley, the method restarts randomly in another position in the image for tracking finger-veins. The output is a locus table that lists how many times a pixel has been tracked, and this captures the finger-vein in the infrared image (see Figure 3.3).



(a)

(b)

Figure 3.1: Cross-sectional brightness profile of a vein. (a) Cross-sectional profile, (b) Position of cross-sectional profile (Source: [2]).

Steps required to compute LT method is given in Algorithm 1 [2] [30]. The details about the algorithm of line tracking are as follows;

Step 1: Selecting the starting point and moving direction
The starting point is selected randomly from the finger vein region and the moving direction attribute (which prevents the tracking point from having excessive curvature) is calculated as

follows:

$$D_{lr} = \begin{cases} (1,0) & if\ (R_{nd}(2) < 1) \\ (-1,0) & otherwise. \end{cases} \tag{3.1}$$

$$D_{ud} = \begin{cases} (0,1) & if\ (R_{nd}(2) < 1) \\ (0,-1) & otherwise, \end{cases} \tag{3.2}$$

where $R_{nd}(n)$ is a random number between 0 and n.

Step 2: Dark-line direction determination and the tracking point movement

In this step, the locus position table will be initialized. This table stores how many times a pixel has been tracked. To move the tracking point, some limitations should be considered. These limitations are described verbally below and defined formally in Equations 3.3 and 3.4:

- The next tracking point should be inside the finger region (R_f).
- The next tracking point should not be the previous tracking point in the current tracking round ($\overline{T_c}$).
- Next tracking point should be one of the neighboring pixels as in Equation 3.4.

$$N_c = \overline{T_c} \cap R_f \cap N_r(x_c, y_c), \tag{3.3}$$

$$N_r(x_c, y_c) = \begin{cases} N_3(D_{lr}(x_c, y_c)) & if\ (R_{nd}(100) < p_{lr}) \\ N_3(D_{ud}(x_c, y_c)) & if\ (p_{lr} + 1 \leq R_{nd}(100) < p_{lr} + p_{ud}) \\ N_8(x_c, y_c) & if\ (p_{lr} + p_{ud} + 1 \leq R_{nd}(100)), \end{cases} \tag{3.4}$$

where (x_c, y_c) refer to the current tracking point, $N_8(x, y)$ refer to the eight neighboring pixels of (x_c, y_c) and $N_3(D)(x, y)$ (which is defined as (D_x, D_y)) refer to the three neighboring pixels of (x_c, y_c) that is in the direction of moving direction attribute D. Furthermore, D refers to the direction attribute (see Equations 3.1 and 3.2). $N_3(D(x_c, y_c))$ is defined as follows:

$$\begin{aligned} N_3(D)(x, y) = \Big\{ &(D_x + x, D_y + y), \\ &(D_x - D_y + x, D_y - D_x + y), \\ &(D_x + D_y + x, D_y + D_x + y)\Big\}. \end{aligned} \tag{3.5}$$

p_{lr} and p_{ud} in Equation 3.4 are the probabilities in which three neighboring pixels can be selected in the horizontal or vertical direction. Horizontal neighboring pixels will have larger probability than the vertical ones because veins tend to be horizontal. Using Equation 3.6, the depth of the valley is calculated and the maximum depth in the cross-sectional profile around the current tracking point is selected (see Figure 3.2):

$$\begin{aligned} V_l = \max\Big\{ &F\Big(x_c + r\cos\theta_i - \frac{W}{2}\sin\theta_i, y_c + r\sin\theta_i + \frac{W}{2}\cos\theta_i\Big) \\ &+ F\Big(x_c + r\cos\theta_i + \frac{W}{2}\sin\theta_i, y_c + r\sin\theta_i - \frac{W}{2}\cos\theta_i\Big) \\ &- 2F\left(x_c + r\cos\theta_i, y_c + r\sin\theta_i\right)\Big\}, \end{aligned} \tag{3.6}$$

where W is the width of the cross-section, r is the distance between the current tracking point (x_c, y_c) and the cross-section, and θ is the angle between lines $(x_c, y_c) - (x_c + 1, y_c)$ and $(x_c, y_c) - (x_i, y_i)$.

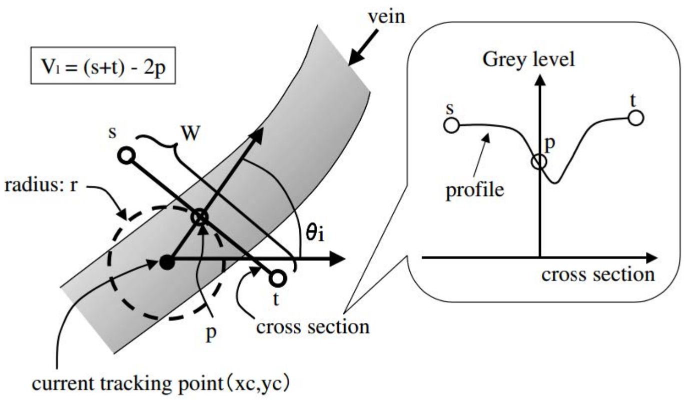


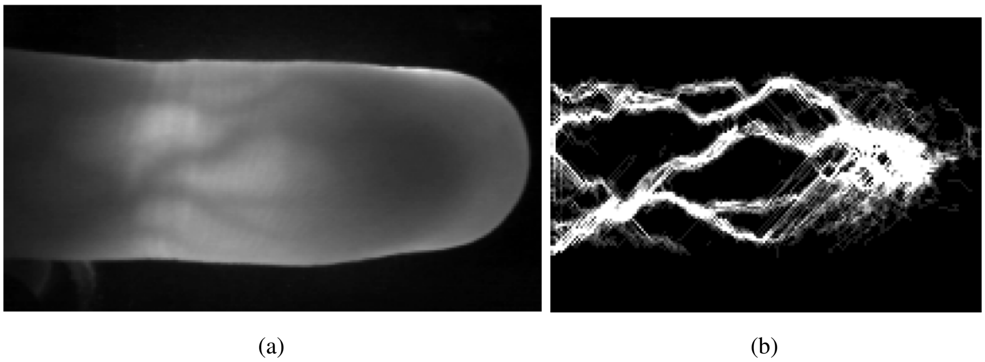
Figure 3.2: Dark line detection; this example shows spatial relationship between the current tracking point (x_c, y_c) and the cross-sectional profile. Pixel p is the neighboring pixel of the current tracking point and the cross-sectional $s - p - t$ looks like a valley. As a result the current tracking point is on a dark line (Source: [2]).

Step 3: Incrementing the values of the elements in the locus table
In this step number of times that points in the locus table have been tracked are updated.

Step 4: Repetition: Steps 1 to 3 are repeated for N many iterations, where N is a parameter to the method.

Step 5: Extracting the finger vein pattern
The last step is extracting the finger vein pattern that will be obtained from the locus space table as shown in Figure 3.3.

In the literature, there are many studies based on the Repeated Line Tracking algorithm such as Yang et al. [31] and Qin et al. [23].



(a)

(b)

Figure 3.3: (a) An example infrared image, (b) Locus space table extracted using the Line Tracking method (Source: [2]).

Algorithm 1 Line Tracking Computation

```
Require: I: Input image, FVR: Favorite region, R: Radius, W: Width of the profiles
  Output: V: Extracted veins

  - Initialize Locus Table (Tr) to Zeros
  - StartPoint: Select random points $(x_s, y_s)$ within FAV region
  - Randomly select $x_s \leftarrow x_s + 1$ OR $x_s \leftarrow x_s - 1$
  - Randomly select $y_s \leftarrow y_s + 1$ OR $y_s \leftarrow y_s - 1$
  while $\mathcal{V} > 0$ do
      - Determine the moving candidate point set $N_c$
          $N_c \leftarrow \overline{T_c} \cap R_f \cap N_r(x_c, y_c)$
      if $size(N_c) = 0$ then
          $\mathcal{V} = -1$
      end if
      for $i = 1$ to $size(N_c)$ do
          - Finding the valley ($\mathcal{V}_l$) using Equation 3.6
      end for
      index $\leftarrow \arg\max_x \mathcal{V}_l(x)$
      $Tr(yc, xc) \leftarrow Tr(yc, xc) + 1$
      $xc, yc \leftarrow Nc(index)$
  end while
```

3.1.2 Maximum Curvature

Miura et al. [3] proposed a method based on calculating the local maximum curvatures in the cross-sectional profiles of a vein image (see Figure 3.4). In this method, the center positions of the veins found using the maximum curvature are extracted and connected to each other to make the final image (Algorithm 2) [3]. The curvature of a profile is calculated using the Equation 3.7.

$$K(z) = \frac{\frac{d^2P_f(z)}{dz^2}}{\left\{1 + \left(\frac{dP_f(z)}{dz}\right)^2\right\}^{\frac{3}{2}}}, \tag{3.7}$$

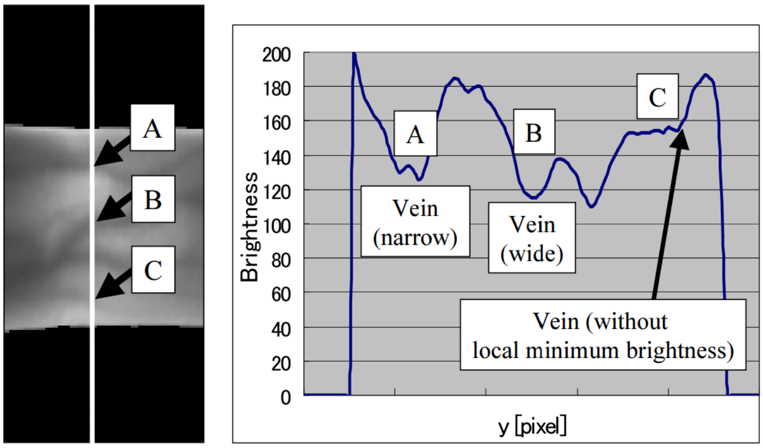
where F is a finger image, $F(x, y)$ is the intensity of the pixel (x, y), z is the position in the profile and $P_f(z)$ is the cross-section profile.

If $K(z)$ is positive, the cross-section profile would be concave, and if $K(z)$ is negative, the cross-section profile would be convex. In each concave area, the local maximums are calculated and these maxima points show the center positions of the veins. The concave width is considered as well, therefore, using both the center position and the width of a concave area, by taking into account scores that are assigned to these center positions as in Equation 3.8:

$$S_{cr}(z_i) = K(z_i) \times W_r(i), \tag{3.8}$$

where $i = 0, 1, ..., N - 1$ and N is the number of local maximas. $W_r(i)$ is the width of the concave regions. The output is the plane V, which is created with the assigned scores as in Equation 3.9 (see Figure 3.5):

$$V(x_i, y_i) = V(x_i, y_i) + S_{cr}(z_i). \tag{3.9}$$



(a) (b)

Figure 3.4: The cross-sectional profile of the veins in MC method. (a) An example of cross-sectional profile, points A,B and C show the veins, (b) Cross-sectional profile of a vein looks like a valley (Source: [3]).

Veins' centers are connected to each other using a filtering operation as follows:

$$C_d(x,y) = min\Big\{max\Big(V(x+1),y), V(x+2,y)\Big), \\ max\Big(V(x-1,y), V(x-2,y)\Big)\Big\}. \tag{3.10}$$

Repeating this process for all of the profiles in the image would give the final vein patterns of the image. Qian et al. [32] used curvature formula in Equation 3.7 to extract finger veins and Song et al. [33] used the mean curvature by calculating the minimum and the maximum of the curvature. Miura et al. [3], Qian et al. [32] and Hong et al. [34] used the maximum curvature extracted as in Equation 3.7.

Algorithm 2 Maximum Curvature Computation

```
Require: I: Input image, FVR: Favorite region, $\sigma$: Sigma

  Output: V: Extracted veins
  $K(z_i)$: Calculating the curvature of the profiles using Equation 3.7.
  if $K(z_i) > 0$ then
      $Local\ Maxima \leftarrow \max K(z_i)$
      $S_{cr} = Local\ Maxima \times W_r(i)$
      $V(x_i, y_i) \leftarrow V(x_i, y_i) + S_{cr}(z_i)$
  end if
  - Connecting the vein centers
```

3.1.3 Wide Line Detector

Wide Line Detector [4] uses a circular sliding window to detect the pixels belonging to a finger vein (see Figure 3.6). At each location, the intensity distribution of the pixels in the neighborhood is eval-

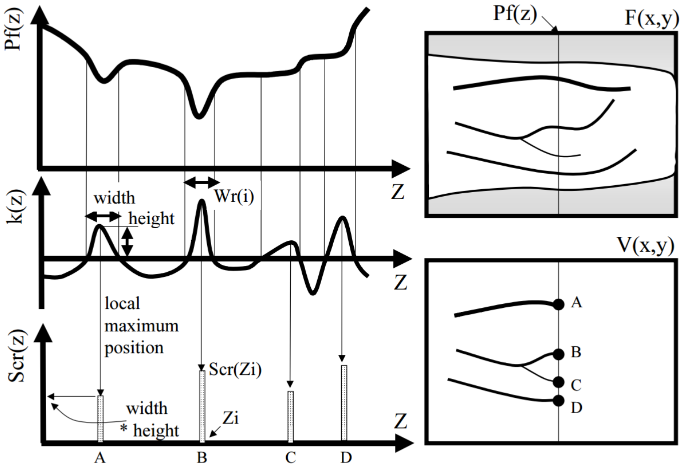


Figure 3.5: Relationship among the profile, the curvature and the probability score of the veins (Source: [3]).

uated with respect to the intensity of the center pixel (see Equation 3.14) [35]. If a small proportion of the pixels have different intensities (see Equation 3.13), then the center of the window is taken as belonging to a finger vein. Let F be the finger vein image, for each (x_0, y_0) in F, the circular

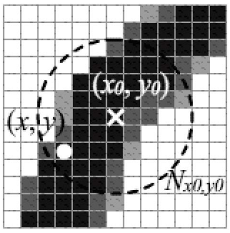


Figure 3.6: The circular neighborhood region (Source: [4]).

neighborhood region with radius r would be:

$$N(x_0, y_0) = \left\{(x, y) \,\middle|\, \sqrt{(x - x_0)^2 + (y - y_0)^2} \leq r\right\}. \tag{3.11}$$

Feature image V, is calculated as:

$$V(x_0, y_0) = \begin{cases} 0 & m(x_0, y_0) > g \\ 255 & otherwise, \end{cases} \tag{3.12}$$

$$m(x_0, y_0) = \sum_{x,y \in N_(x_0,y_0)} s(x, y, x_0, y_0, t), \tag{3.13}$$

$$s(x, y, x_0, y_0, t) = \begin{cases} 0 & F(x, y) - F(x_0, y_0) > t \\ 1 & otherwise, \end{cases} \tag{3.14}$$

where $V(x_0, y_0)$ is the matrix of the finger vein that is calculated using the Equation 3.12, g and t are the threshold values. Huang et al. assigned $g = 41$ and $t = 1$ for the intensity ranges of $[0, 255]$. m calculates sum of the pixels that have different intensities and s groups the pixels that have similar intensities with the one in the center of the mask.

Algorithm 3 Wide Line Detector Computation

```
Require: I: Input image, FVR: Favorite Region, r: Radius, t: Threshold, g: Geometric threshold

  Output: V: Extracted veins
  - Creating circular mask:
          $N \leftarrow X^2 + Y^2 \leq r^2$
  - Calculating s:
          $F(x, y) - F(x_0, y_0) \leq t$
  - Calculating m:
          $\sum_{x,y \in N(x_0,y_0)} s(x, y, x_0, y_0, t)$
  - Apply Equation 3.12
  - Mask the vein image with the finger region
```

3.1.4 Shape Descriptors

There are two ways of representing a shape's region:

- Representing a region by its external characteristics e.g., the boundary.
- Representing a region by its internal characteristics i.e. the pixels that cover the region).

When focus is on a shape's characteristics, boundary representation is mostly used [5]. On the other hand, when focus is on the regional properties, internal representation is used [5]. In both cases, the features that are extracted as descriptors should be insensitive to the changes in rotation, translation and scale [5]. Rotation invariance is achieved if the representation of the shape does not change significantly by rotation. Translation invariance is achieved if representation of the shape does not change notably by translation. Fourier descriptors and Zernike moments are discussed in detail in the following sections because in the literature they are good examples of shape descriptors.

3.1.5 Fourier Descriptors

Fourier descriptors are one of the widely used methods for describing shapes [5]. Fourier descriptors is a contour based descriptor which means that, the pixels on the boundary of the images are used to describe the shape. The first step to compute Fourier descriptors is to extract the image's boundary pixels; $(x_0, y_0), (x_1, y_1), ..., (x_{N-1}, y_{N-1})$. These coordinates can be shown in the form of $x(k) = x_k$ and $y(k) = y_k$. Therefore, the boundary can be represented as $s(k) = [x(k), y(k)]$, for $k = 0, 1, 2, ..., K - 1$. Furthermore, these coordinates can be represented as complex numbers $s(k) = x(k) + jy(k)$, where $j = \sqrt{-1}$. The last step is converting $s(k)$ from the spatial domain to the frequency domain using Discrete Fourier Transform [5] (see Equation3.15).

$$a(u) = \frac{1}{K} \sum_{k=0}^{K-1} S(k) \mathrm{e}^{\frac{-j2\pi uk}{K}}, \tag{3.15}$$

for $u = 0, 1, 2, ..., K - 1$. $a(u)$, which are complex, are the Fourier descriptors of the shape's boundary. FD is normalized using Equation 3.16. Translating the image affects only the first Fourier coefficient, therefore, by removing the first coefficient, FD becomes translation invariant. Rotation invariance can be achieved by using just the magnitude of the Fourier transform. Moreover, scale invariance is obtained by dividing each coefficient by the first coefficient [36], [37]:

$$FD = \left|\frac{FD_1}{FD_0}\right|, \left|\frac{FD_2}{FD_0}\right|, ..., \left|\frac{FD_k}{FD_0}\right|, \tag{3.16}$$

where k is the number of the coefficients. In short, the steps for computing the Fourier descriptors can be found in Algorithm 4.

Algorithm 4 Fourier descriptors Computation

```
Require: I: Input image, k: Number of coefficients

  Output: FD: Fourier coefficients
  $I = I > 0$
  - Converting coordinates to complex numbers:
        $I \leftarrow I(:, 1) + j * I(:, 2)$
  $z \leftarrow FFT(I)$
  - FD Normalization using Equation 3.16
  - Take the $k$ first coefficients
```

3.1.6 Image Moments

Image moments have become one of the popular methods for feature extraction in recent years. Various types of moment descriptors have been used for pattern recognition, registration and compression. When global properties of the image (compared to local properties) are the key points, moments are mostly used. In this section, the geometric moments and the orthogonal moments are explained briefly. In 1962, Hu [38] proposed the first moment invariants for image analysis and pattern recognition.

- Geometric Moments: Two dimensional geometric moment of order $(n + m)$ of a function f(x,y), e.g., the boundary of a shape, is as follows:

$$M_{nm} = \int_x \int_y x^n y^m f(x, y) dx dy. \tag{3.17}$$

  Some properties of the geometric moments are:

  - Mass and Area: Zero order moment of the function $f(x, y)$ is:

  $$M_{00} = \int_x \int_y f(x, y) dx dy, \tag{3.18}$$

    which shows the total mass of a function or an image and in the case of binary image, M_{00} represents the total area of the image.

  - Center of Mass: The two first order moments, (M_{10}, M_{01}), express the center of the mass of the image. Center of the mass is a place where the entire mass of the image can be gathered. Coordinates of the center of the mass are:

  $$\overline{x} = \frac{M_{10}}{M_{00}}, \overline{y} = \frac{M_{01}}{M_{00}}. \tag{3.19}$$

    Equation 3.19 can be used to describe the centroid position of the shape.

The basis set, $x^n y^m$, is not orthogonal, though lack of orthogonality in geometric moments, causes information redundancy which means image would not be reconstructed properly using its geometric moments. On the other hand, these moments are translation and scale invariant, therefore, before using Zernike moments, image preprocessing can be used to achieve translation and scale invariant images.

- Orthogonal Moments: Teague [6] proposed a set of orthogonal moments to overcome geometric moments' problems on information loss. There are different types of orthogonal moments. Since Zernike moments is a good representation of orthogonal moments in the literature, it is described in detail in the following section.

3.1.6.1 Zernike Moments

Zernike moments are shape descriptors which are subcategory of the region-based descriptors [39]. As mentioned in Section 3.1.4, region based descriptors use all the pixels inside an area to build the descriptors. Zernike moments project an image onto a set of complex Zernike polynomials and these polynomials are orthogonal to each other. For computing Zernike moments two steps should be considered. The first step is calculating the Zernike basis functions and the second step is computing the Zernike moments by projecting an image to the basis functions [39]. The set of these complex polynomials called Zernike basis functions are denoted by $V_{nm}(x, y)$, and defined as follows:

$$V_{nm}(x, y) = V_{nm}(\rho, \theta) = R_{nm}(\rho) \exp(jm\theta), \tag{3.20}$$

where $n \geq 0$, $m > 0$ or $m < 0$ with the conditions of $n - |m|$ being even and $|m| \leq n$, ρ is the length of the vector from the origin to (x, y), and θ is the angle between the vector and x axis in counter clockwise direction. The radial polynomial $R_{nm}(\rho)$ is defined as follows:

$$R_{nm}(\rho) = \sum_{s=0}^{\frac{n-|m|}{2}} (-1)^s . \frac{(n-s)!}{s!(\frac{n+|m|}{2} - s)!(\frac{n-|m|}{2} - s)!} \rho^{n-2s}. \tag{3.21}$$

Radial polynomials (see Figure 3.8) are orthogonal and satisfy:

$$\iint\limits_{x^2+y^2 \leq 1} [V_{nm}(x, y)]^* V_{pq}(x, y) dxdy = \frac{\pi}{n+1} \delta_{np} \delta mq, \tag{3.22}$$

with

$$\delta = \begin{cases} 1 & a = b \\ 0 & otherwise, \end{cases} \tag{3.23}$$

where V^* is the complex conjugate of V and δ is the Kronecker delta function. Zernike moment of order n with the repetition m for the digital image $f(x, y)$ is:

$$A_{nm} = \frac{n+1}{\pi} \sum_x \sum_y f(x, y) V^*(\rho, \theta), x^2 + y^2 \leq 1. \tag{3.24}$$

As can be seen in Figure 3.7, the image center is taken as the origin and the pixels' coordinates are projected to the unit circle.

Since ZM is not affected by rotation due to its polar formulation, magnitude of the Zernike moments are used to achieve rotation invariance [40], [41]. Translating the origin of the image $f(x, y)$ to the centroid of the finger vein image, $f(x + \bar{x}, y + \bar{y})$, makes ZM translation invariant [40], [42] and the centroid is calculated using Equation 3.19 (see the Algorithm 5.

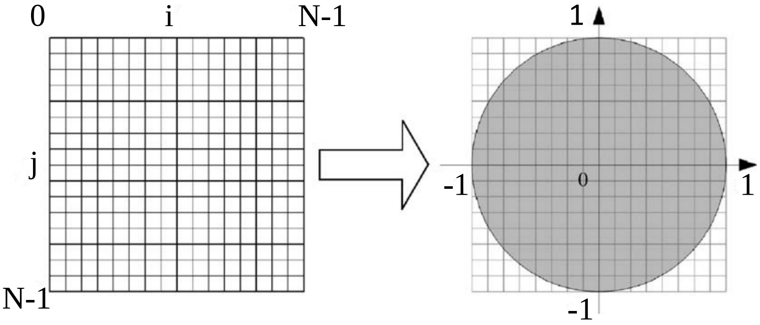


Figure 3.7: Mapping transform (Source: [39]).

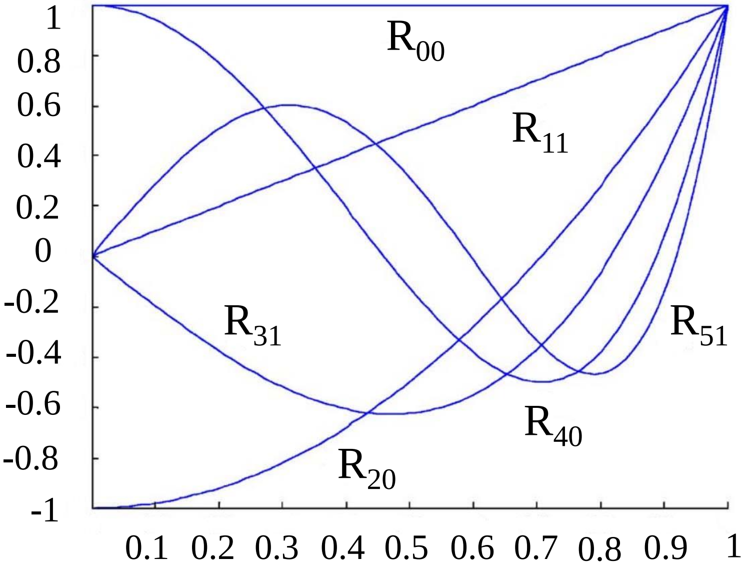


Figure 3.8: Zernike radial polynomials of order 0-5 (Source: [39]).

Algorithm 5 Zernike Moments Computation

```
Require: SZ: Size of the image, n: Order

  Output: ZM: Zernike moments
  $radial \leftarrow 0$
  $szh \leftarrow \frac{SZ}{2}$
  for $x = 1\ to\ SZ$ do
      $\rho \leftarrow \sqrt{(szh - x)^2 + (szh - y)^2}$
      $\theta \leftarrow tan^{-1}(\frac{szh-y}{szh-x})$
      if $\rho > szh$ then
          continue
      end if
      $\rho \leftarrow \rho / szh$
      for $s = 0\ to\ \frac{n-size(m)}{2}$ do
          $radial \leftarrow radial + (-1)^s . \frac{(n-s)!}{s!(\frac{n+size(m)}{2}-s)!(\frac{n-size(m)}{2}-s)!}\rho^{n-2s}$
      end for
      - Calculate Zernike Basis Function (ZBF):
            $ZBF \leftarrow radial \times \exp(n \times \theta \times ij)$
  end for

  Calculate ZM using Equation 3.24
```

3.1.7 Local Binary Patterns

Local Binary Patterns (LBP) [8] is one of the simple and widely used methods for feature extraction. LBP method used in this study is the simple form which is not rotation invariant, however, there are other types of LBP that are rotationally invariant which are not used in this study. In LBP, the intensity of the center pixel is compared against the intensities of the neighboring pixels. During this comparison, bigger or equal intensity values (compare to the intensity of the center pixel) are taken as 1 and smaller values as 0. Binary string which consists of 1s and 0s, can be converted to a decimal number, representing the local intensity distribution around the center pixel with a single number (see Figure 3.9). To describe formally, LBP for a single pixel $p_c = (x_c, y_c)$ in a gray scale image I is defined as:

$$LBP_K(p_c) = \sum_{p=0}^{K-1} s(n_p(p_c) - I(p_c))\, .\, 2^p, \tag{3.25}$$

where K is the number of the neighboring pixels in a 3×3 window, n_p is the p^{th} neighbor of p_c and s is as follows:

$$s(x) = \begin{cases} 1 & if\ x \geq 0 \\ 0 & otherwise. \end{cases} \tag{3.26}$$

Figure 3.9 shows the steps for computing the local binary patterns for the center pixel in 3×3 window. Pseudocode for LBP calculation is presented in Algorithm 6.

Ojala et al. [44] proposed a modified version of the Local Binary Patterns with circular neighborhood which can have neighbors that varies in size. Circular neighborhood has two parameters, P is defined as the number of sampling points and R as the radius and is defined as:

$$LBP_{P,R} = \sum_{P=0}^{P-1} s(g_p - g_c)\, .\, 2^P, \tag{3.27}$$

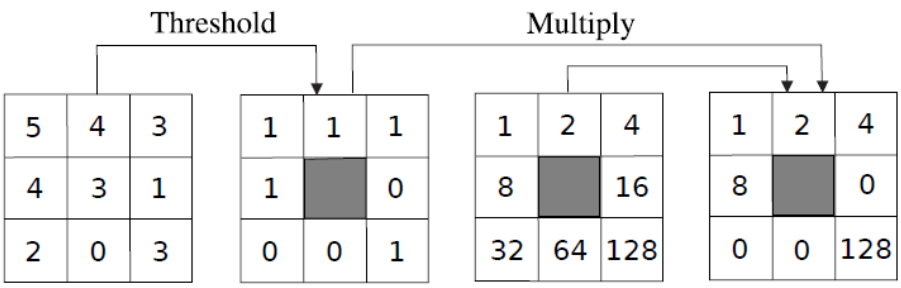


Figure 3.9: Local Binary Patterns computation. The extracted value from the center pixel is: 1+2+4+8+128=143 (Source: [43])

.

Algorithm 6 Local Binary Patterns Computation (adapted from [9])

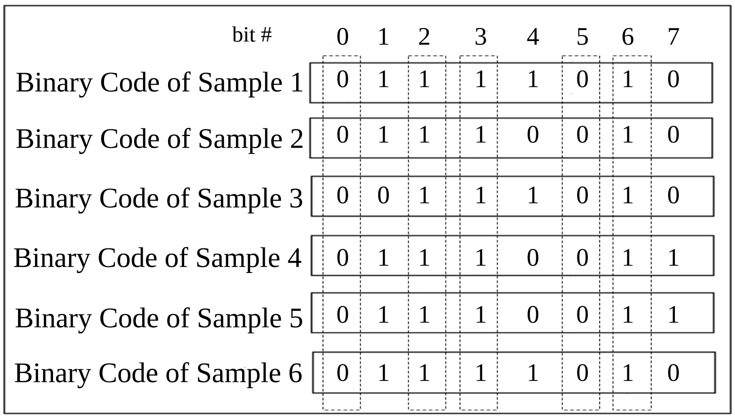
```
Require: I: A gray-scale image

  Output: LBP vector
  - Initialize a $L$ matrix $H$, where $L = N \cdot M$

  for all $p \in I$ do
      - Let $B$ a vector of size 8
      $N \leftarrow 8neight(I, p)$
      for all $n_i \in N$ do
          if $I(p) > I(n_i)$ then
              $B_i \leftarrow B_i + 1$
          end if
      end for
      - Convert the binary vector $B$ to a decimal number to get $D$
      $H(k) \leftarrow \#\{D \in bin(k)\}$          - Increment the kth bin of histogram
  end for
```

where $g_c = (x_c, y_c)$ is the gray value of the center pixel, and g_p is the gray value of the P^{th} sampled point. The coordinates of g_c are $(0, 0)$ and the coordinates of g_p are given by:

$$C_{xp} = x_c + R\cos\frac{2\pi p}{P},\ C_{yp} = y_c - R\sin\frac{2\pi p}{P}. \tag{3.28}$$

In this study, original LBP, defined in Equation 3.26 is used and the results of the original LBP are presented in Chapter 4.

Lee et al. [24] used LBP and LDP for feature extraction and compared them and presented that, even though LBP is faster than LDP, LDP results are better than LBP. Yang et al. [25] proposed a new method based on LBP; they used different images of a finger and named all identical bits of the same finger of various images as Best Bit Map (BBM). Using BBM, noisy pixels are removed and it has better performance in comparison to LBP (see Figure 3.10). Rosdi et al. [45] used Local Line Binary Patterns (LLBP) for the finger vein which proposed by [46] for face recognition. In contrast to LBP which has square shape neighborhood, LLBP neighborhood is a line and consists of horizontal and vertical components and their concatenated binary codes that can be seen in detail in Figure 3.11.



bit #	0	1	2	3	4	5	6	7
Binary Code of Sample 1	0	1	1	1	1	0	1	0
Binary Code of Sample 2	0	1	1	1	0	0	1	0
Binary Code of Sample 3	0	0	1	1	1	0	1	0
Binary Code of Sample 4	0	1	1	1	0	0	1	1
Binary Code of Sample 5	0	1	1	1	0	0	1	1
Binary Code of Sample 6	0	1	1	1	1	0	1	0

Figure 3.10: Best Bit Map computation (adapted from [25]).

3.1.8 Histogram of Oriented Gradients

Dalal et al. [7] proposed a method called Histogram of Oriented Gradients (HOG) for detection the objects which led to successful applications in human detection and tracking. HOG method used in this study is the simple form which is not rotation invariant, however, there are other types of HOG that are rotationally invariant which are not used in this study. This method is based on dividing the image window into the smaller regions called cells and gathering 1-D local histograms of the gradients inside the cells. For normalizing histograms, the cells are combined in larger groups called blocks. Gathering all histograms forms the feature vector. Figure 3.12 and Algorithm 7 represent the steps required for the computation of HOG. In this thesis, HOG descriptors are analyzed and compared with other popular descriptors. The formal definition of HOG is as follows:

$$HOG(k) = \sum_{p \in P} \delta\left(\left\lfloor \frac{\theta^p}{L} \right\rfloor\right), \tag{3.29}$$

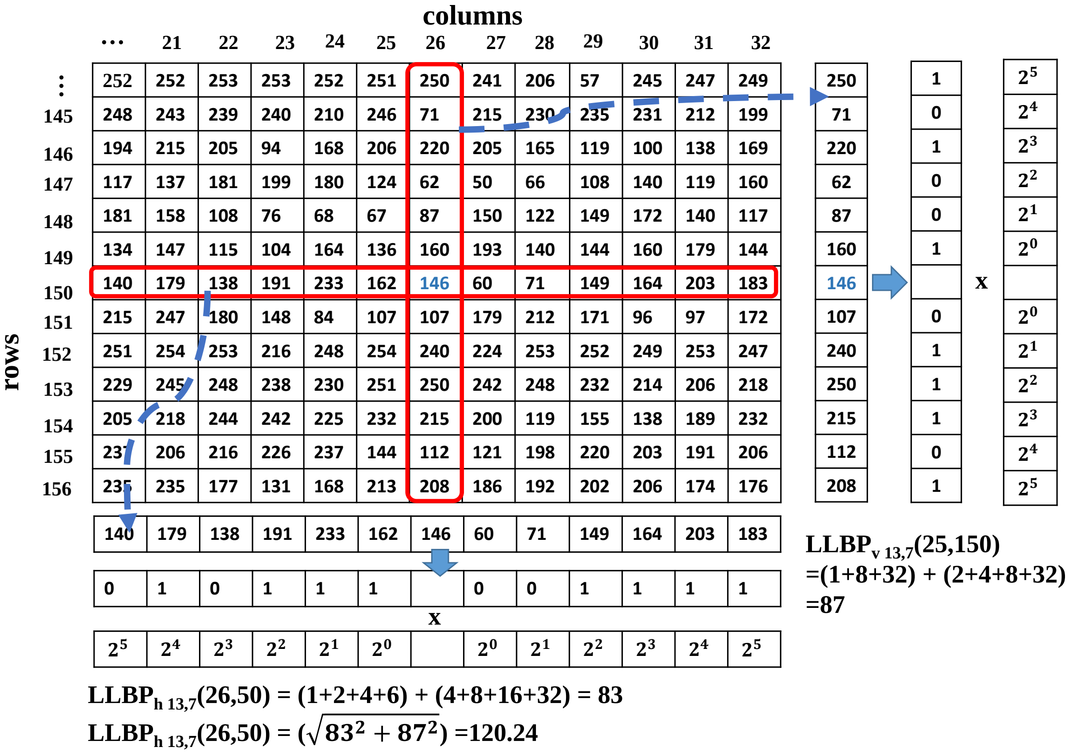
where P is an image, δ is the Kronecker delta given in Equation 3.23, L is the normalizing constant and θ^p is the orientation at point p, that is the same as the image gradient at that point. $HOG(k)$ corresponds to the value of the kth bin in a K-bin histogram. The value of K used in the experiments is set to 9, and the value of the normalizing constant, L, is equal to $180/K = 20$ [7].

3.1.9 Global Binary Patterns

Global Binary Patterns proposed by Sivri [9] is a fast and simple shape descriptor method similar to the Local Binary Patterns [8]. This method creates bit strings in the desired directions for a thresholded binary image and converts these bit strings to the binary numbers to build a global descriptor (see Figure 3.13).

Global Binary Pattern of a row, r, of the binary image I, is as follows:

$$GBP_h(r) = \sum_{j=1}^{R} I(r, j) \,.\, 2^{j-1}, \tag{3.30}$$



	…	21	22	23	24	25	26	27	28	29	30	31	32
⋮	252	252	253	253	252	251	250	241	206	57	245	247	249
145	248	243	239	240	210	246	71	215	230	235	231	212	199
146	194	215	205	94	168	206	220	205	165	119	100	138	169
147	117	137	181	199	180	124	62	50	66	108	140	119	160
148	181	158	108	76	68	67	87	150	122	149	172	140	117
149	134	147	115	104	164	136	160	193	140	144	160	179	144
150	140	179	138	191	233	162	146	60	71	149	164	203	183
151	215	247	180	148	84	107	107	179	212	171	96	97	172
152	251	254	253	216	248	254	240	224	253	252	249	253	247
153	229	245	248	238	230	251	250	242	248	232	214	206	218
154	205	218	244	242	225	232	215	200	119	155	138	189	232
155	231	206	216	226	237	144	112	121	198	220	203	191	206
156	235	235	177	131	168	213	208	186	192	202	206	174	176

250	1		2^5
71	0		2^4
220	1		2^3
62	0		2^2
87	0		2^1
160	1		2^0
146		x	
107	0		2^0
240	1		2^1
250	1		2^2
215	1		2^3
112	0		2^4
208	1		2^5

LLBP$_{v\ 13,7}$(25,150)
=(1+8+32) + (2+4+8+32)
=87

140	179	138	191	233	162	146	60	71	149	164	203	183
0	1	0	1	1	1		0	0	1	1	1	1
						x						
2^5	2^4	2^3	2^2	2^1	2^0		2^0	2^1	2^2	2^3	2^4	2^5

LLBP$_{h\ 13,7}$(26,50) = (1+2+4+6) + (4+8+16+32) = 83

LLBP$_{h\ 13,7}$(26,50) = ($\sqrt{83^2 + 87^2}$) =120.24

Figure 3.11: Local Line Binary Patterns computation (Source: [45]).

```
Algorithm 7 Histograms of Oriented Gradients Computation (adapted from [9])
Require: I: A gray-scale image
    T × V: dimensions of the grid

    Output: HOG vector
    - Let $I$ be decomposed to sub-matrices as: $\begin{pmatrix} P_1^1 & P_1^2 & \cdots & P_1^V \\ P_1^2 & P_2^2 & \cdots & P_2^V \\ \vdots & \vdots & \ddots & \vdots \\ P_T^1 & P_T^2 & \cdots & P_T^V \end{pmatrix}$
    - Let $H$ be a $T \times V \times K$ matrix
    for all $t \in 1 \cdots T$ do
        for all $v \in 1 \cdots V$ do
            $H(t, v) \leftarrow HOG(P_{t,v})$
        end for
    end for
    - Reshape $H$ to get a vector of size $T \cdot V \cdot K$
```

where R is the number of rows in the image I. Likewise, GBP along the vertical direction is as follows:

$$GBP_v(c) = \sum_{i=1}^{C} I(i, c) \, . \, 2^{i-1}, \tag{3.31}$$

where C is the number of columns in the image I. Figure 3.13 illustrates GBP along horizontal and

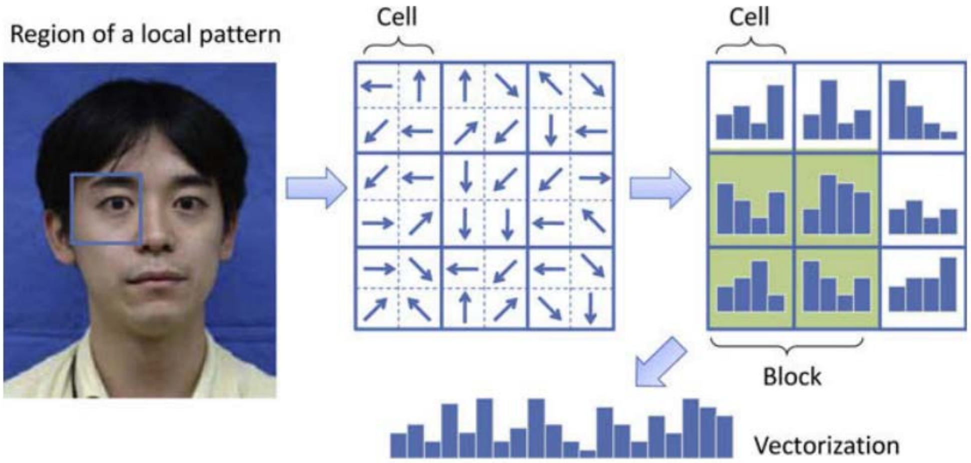


Figure 3.12: Histogram of Oriented Gradients computation (Source: [47]).

vertical axis defined as GBP_{hv} (see Algorithm 8). In order to calculate GBP along any direction with orientation θ (see Figure 3.14), the following equation can be used:

$$GBP_\theta(k) = \sum_{p \in I} \delta\left(L \frac{x^p - \min(x^{p_{ij}})}{\max(x^{p_{ij}}) - \min(x^{p_{ij}})} - k\right) . I(P) . 2^{d^p}, \tag{3.32}$$

where δ is the Kronecker delta function, d^p is the point-to-line distance between the line $l_\theta^{p_0}$ and the pixel p, projection of the pixel p onto the line $l_\theta^{p_0}$ is x^p and L is the length of the GBP_θ descriptor. As can be seen in Figure 3.14, $l_\theta^{p_0}$ is the line with the orientation θ, which passes through p_0. Horizontal and vertical GBP are not invariant to rotation and translation.

Algorithm 8 Global Binary Patterns (GBP_h) Computation (adapted from [9]).

```
Require: I: $R \times C$ thresholded image

Output: GBP vector
- Let $GBP_h$ be a vector of size $N$

for all $i \in 1 \cdots R$ do
    for all $j \in 1 \cdots C$ do
        $GBP_h(i) \leftarrow GBP_h(i) + I(i, j) \cdot 2^{j-1}$
    end for
end for
```

3.2 Finger Vein Matching

Different matching methods have been used for matching the features extracted from each method. For LT, MC and WL, the literature uses the so-called mismatch ratio in which a query map is compared against a stored map pixel by pixel like (in template matching). For FD, ZM, HOG, LBP and GBP, many distance metrics are used in the literature, like Euclidean distance, χ^2 distance and Earth Mover's distance. In this study, the mismatch ratio is used for LT, MC and WL, and Euclidean distance, χ^2 distance and Earth Mover's distance for FD, ZM, HOG, LBP and GBP.

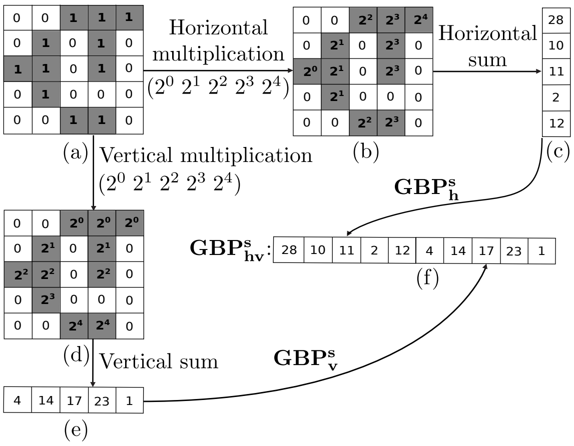


Figure 3.13: GBP computation. (a) The binary image; (b) Rows are multiplied by powers of two; (c) Each row is summed horizontally; (d) Columns are multiplied by powers of two; (e) Columns are summed vertically; (f) Resulting GBP descriptor (Source: [9]).

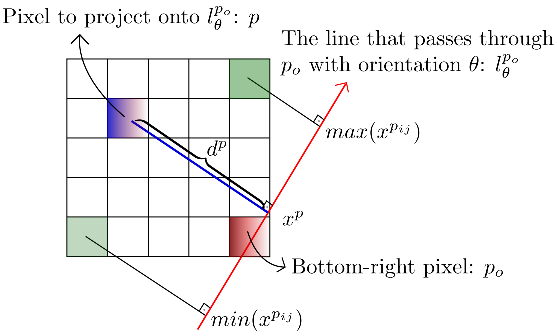

Pixel to project onto $l_\theta^{p_o}$: p
The line that passes through p_o with orientation θ: $l_\theta^{p_o}$
$max(x^{p_{ij}})$
d^p
x^p
Bottom-right pixel: p_o
$min(x^{p_{ij}})$

Figure 3.14: Ilustration of GBP along an arbitrary direction with orientation θ (Source: [9]).

3.2.1 Template Matching

Template matching is one of the oldest and simplest methods in pattern recognition. In template matching, the pattern to be identified is matched against the stored template [48]. $N_m(s,t)$ which is the value of the mismatch is calculated as follows:

$$N_m(s,t) = \sum_{y=0}^{h-2ch-1} \sum_{x=0}^{w-2c_w-1} \phi\Big(I(s+x, t+y),\ R(c_w+x, c_h+y)\Big), \tag{3.33}$$

where c_w and c_h are the distances for the variations in the horizontal and vertical directions, respectively and are used to adjust the displacement, w and h are the width and height of the images and $\phi(P_1, P_2)$ is defined as follows:

$$\phi(P_1, P_2) = \begin{cases} 1 & if\ |P_1 - P_2| = 255 \\ 0 & otherwise. \end{cases} \tag{3.34}$$

P_1 is the pixel value of the image and P_2 is the pixel value of the registered template.

$$N_m = \min_{0 \le s \le 2c_w, 0 \le t < 2c_h} (N_m(s,t)). \tag{3.35}$$

The mismatch ratio, R_m, is calculated as follows:

$$R_m = \frac{N_m}{\sum_{j=t_0}^{t_0+h-2c_h-1} \sum_{i=s_0}^{s_0+w-2c_w-1} \phi(I(i,j),0) + \sum_{j=c_h}^{h-c_h-1} \sum_{i=c_w}^{w-c_w-1} \phi(0, R(i,j))}, \tag{3.36}$$

where s_0 and t_0 are s and t that are minimized in Equation 3.35.

3.2.2 Euclidean Distance

The Euclidean distance between vectors $X = \Big\{x_i \Big| i = 1,2,...,N\Big\}$ and $Y = \Big\{y_i \Big| i = 1,2,...,N\Big\}$ are defined as:

$$Euc(X,Y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}. \tag{3.37}$$

3.2.3 Chi-Square Distance (χ^2)

X^2 distance is a sub group of the Bin-By-Bin dissimilarity measures. The Bin-By-Bin dissimilarity measures pairs of the bins in two histograms that have the same index. This metric is used in wide areas such as shape classification and boundary detection [49].

$$\chi^2(X,Y) = \frac{1}{2} \sum_i \frac{(x_i - y_i)^2}{(x_i + y_i)}, \tag{3.38}$$

where $X = \Big\{x_i \Big| i = 1,2,...,N\Big\}$ and $Y = \Big\{y_i \Big| i = 1,2,...,N\Big\}$.

3.2.4 Earth Mover's Distance (EMD)

Unlike the Bin-to-Bin dissimilarity in which corresponding histogram bins are compared, in Cross-Bin dissimilarity, non-identical indexes of the bins are compared as well. Earth Mover's Distance (EMD) is a solution to the transportation problem which defines the distance between two histograms and the goal is to find the minimum cost when transforming one histogram to the other one [49]. The EMD between the two histograms, $P = (p_i, u_i)_{i=1}^{m}$ and $Q = (q_j, v_j)_{j=1}^{n}$, where m and n are the size and the flow between P and Q, which is shown with $F = (f_{ij})$, is calculated as follows:

$$EMD(P, Q) = \frac{\sum_{i,j} f_{ij}\, d_{ij}}{\sum_{i,j} f_{ij}}, \tag{3.39}$$

where d_{ij} is the ground distance matrix between p_i and q_j, u_i and v_i refer to the weights of the histograms and following constraints should be considered:

$$f_{ij} \geq 0,$$

where $1 \leq i \leq m, 1 \leq j \leq n$,

$$\sum_{j=1}^{n} f_{ij} \leq u_i,$$

where $1 \leq i \leq m$,

$$\sum_{i=1}^{m} f_{ij} \leq v_j,$$

where $1 \leq j \leq n$,

$$\sum_{i,j} f_{ij} = \min\Big(\sum u_i, \sum v_j\Big).$$

CHAPTER 4

EXPERIMENTS

In the previous chapter the methods analyzed in the thesis were explained in detail and in this chapter these methods are compared to each other and tested on their resilience against translation, rotation and noise. This is crucial since in finger vein image capturing systems, fingers tend to be not aligned the same way each time they are scanned.

In this chapter the database contents used in the experiments are explained and their performances and running times are analyzed in depth. Used methods are Fourier descriptors (FD), Zernike moments (ZM), Local Binary Patterns (LBP), Global Binary Patterns (GBP) and Histogram of Oriented Gradients (HOG). Moreover, arbitrary N images are selected from the database which are translated horizontally for different shift amounts $(0, 5, 10, 15$ and 20 pixels) and rotated clockwise using nearest neighbor interpolation for different angles $(0^o, 5^o, 10^o, 15^o$ and $20^o)$ to analyze the methods against their resistance to translation and rotation. Furthermore, Gaussian noise is added to the images. In the case of Gaussian noise the variance has been investigated $(0.0002, 0.0004, 0.0006, 0.0008$ and $0.001)$ and the mean is kept zero (see Figure 4.1).

The aim is to improve the running times and the resilience of LT, MC and WL using FD, ZM, HOG, LBP and GBP. It should be mentioned that these methods (FD, ZM, HOG, LBP and GBP) are applied to the binarized locus table (from LT), the binarized Maximum Curvature map and the Wide Line Detector map.

4.1 The Database

The database in this study is carried out from SDUMLA-HMT finger-vein database that is publicly available[1]. This database contains $3,816$ images from both hands and provided images are index finger, middle finger and ring finger and for each finger six different images are captured. Figure 4.2 displays small sample of the database. The original images are of size 320×240, however, for the sake of faster analysis, the size is reduced to 160×120 using nearest neighbor interpolation. The images are gray scale images in the intensity range of $[0, 255]$.

Using Prewitt edge detector, strong edges are extracted, the boundaries of the finger are found and the mask image is produced (see Figure 4.3). Masking the image is a very important step since it eliminates the irrelevant areas.

[1] http://mla.sdu.edu.cn/sdumla-hmt.html

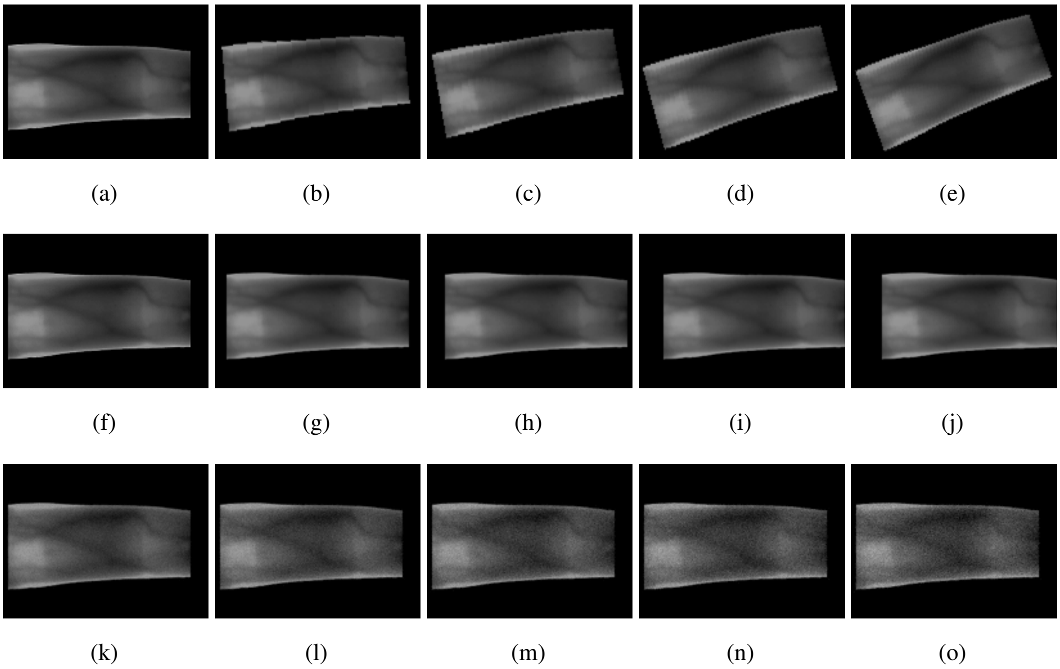


Figure 4.1: Example finger-vein images by (a-e) Rotation ($0^o, 5^o, 10^o, 15^o, 20^o$ degrees, respectively) , (f-j) Translation ($0, 5, 10, 15, 20$ pixels, respectively), (k-o) Gaussian noise ($0.0002, 0.0004, 0.0006, 0.0008, 0.001$ variances, respectively). As can be seen in noisy images, increasing the Gaussian noise degrades the visibility of the veins.

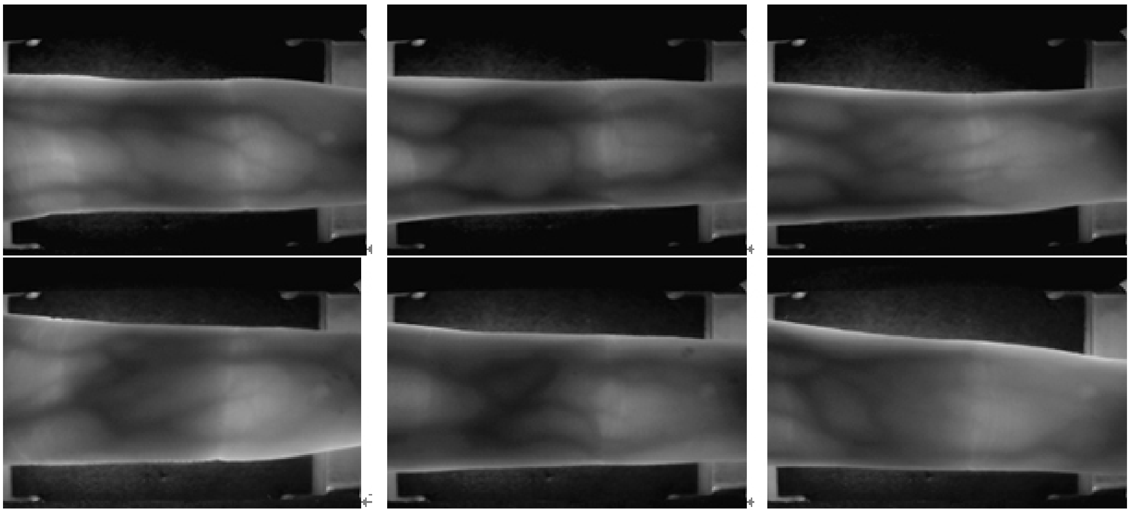


Figure 4.2: Sample of Images from the SDUMLA-HMT Finger-vein Database

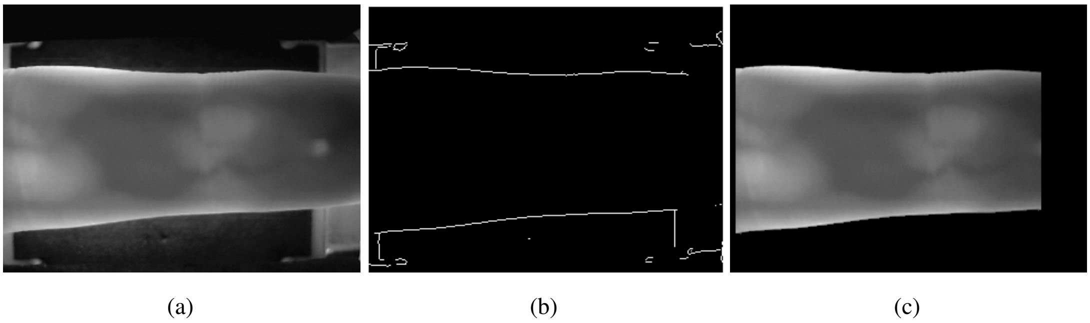


Figure 4.3: Finding the mask for the finger region in the image. (a) Original image, (b) Prewitt edge detection, (c) Masked image.

4.2 Performance Measurement

Performances of algorithms are analyzed using Receiver Operating Characteristic (ROC) curve which depict the False Accepting Rates and the False Rejection Rates of an algorithm at the same time. Drawing a curve required calculating FAR and FRR for different threshold values (twenty threshold values are used in this study, different threshold values are calculated by dividing the subtract of maximum and minimum of the distances by twenty).

4.3 Results

The methods are compared based on the FAR and FRR, as widely used by the finger-vein recognition literature (e.g., Miura et al. [2]).

4.3.1 Line Tracking

Line Tracking method as outlined in Algorithm 1 has two main input parameters. The width of the profiles are selected as $w = 17$ and the *Radius* as $r = 1$ based on the literature [2]. As can be seen in Figure 4.4, using the mask image locus table is produced and then by binarizing the locus table (using the median of the image as a threshold) finger veins are more emphasized. The Line Tracking method is analyzed against rotation, translation and noise. For calculating the mismatch ratio, displacement values are selected c_h=50 and c_w=30, which are the distances for the variations in the vertical and horizontal directions, respectively and are used to adjust the displacement (see Section 3.2.1). For this reason displacements are selected large enough therefore translation will not affect the Line Tracking method and as can be seen in Figure 4.5, Line Tracking method is translation invariant but affected strongly by noise and rotation because of pixel by pixel comparison. The implementation of LT provided online at [50] is used in this thesis.

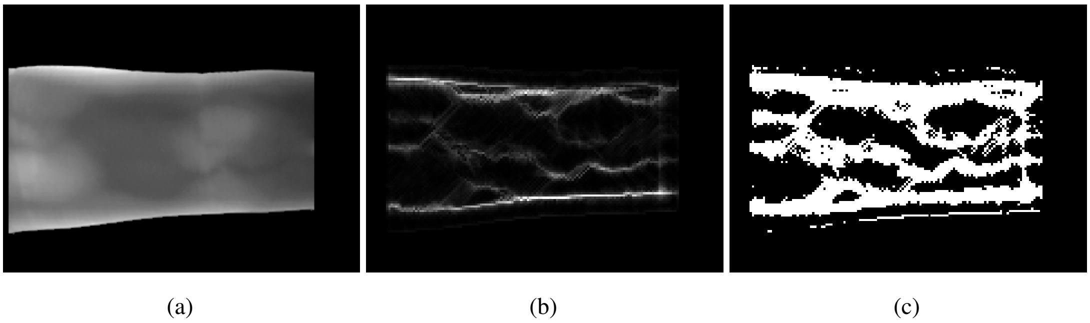

(a) (b) (c)

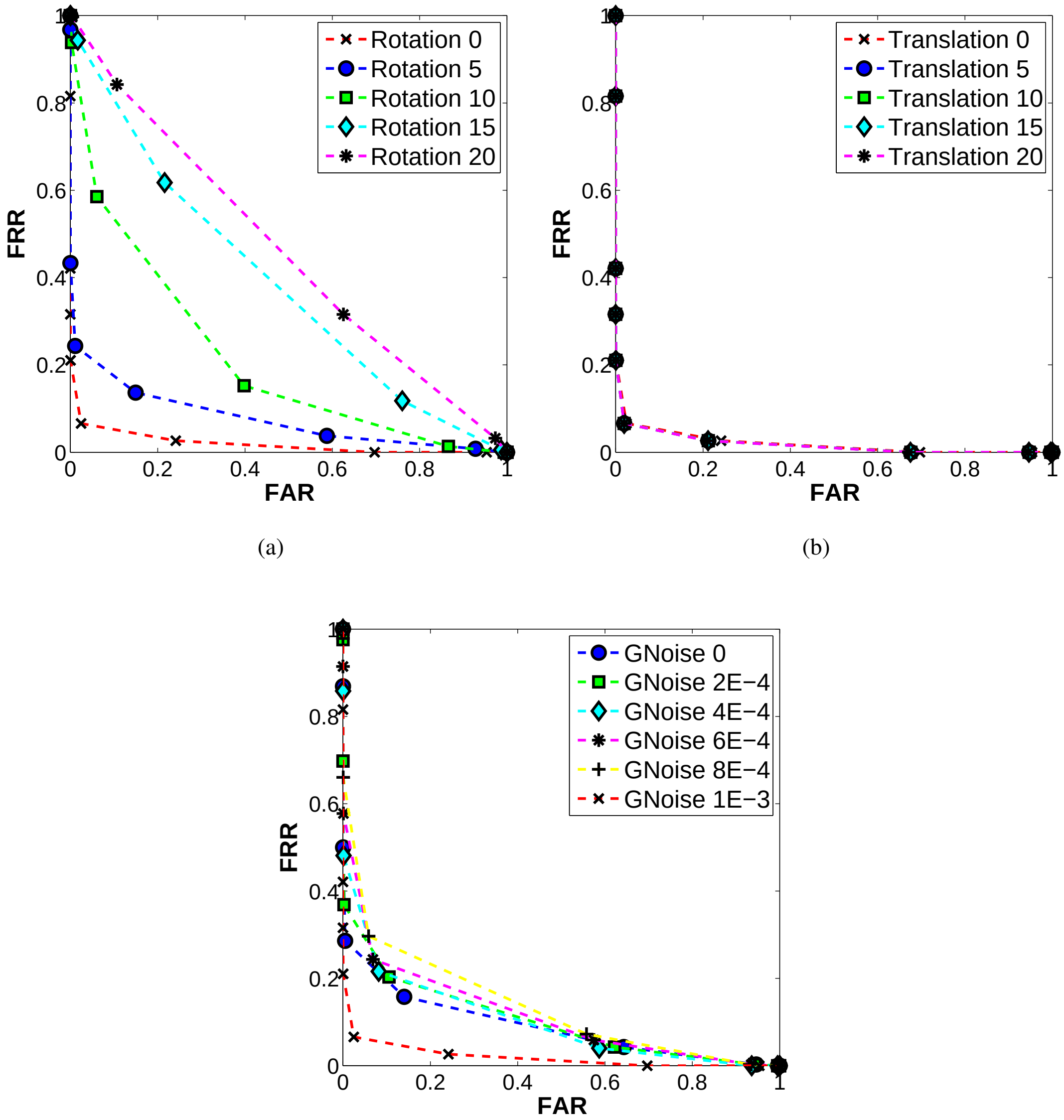
Figure 4.4: Example finger-vein extraction by the LT method. (a) Masked image, (b) Image extracted from locus table, (c) Binarized version of (b) using median thresholding.

4.3.2 Maximum Curvature

The Maximum Curvature method as described in Algorithm 2 has just one parameter called *Sigma* (σ) which is set $\sigma = 3$ based on the literature [3]. As can be seen in Figure 4.6, for extracting veins in the Maximum Curvature method, the maximum curvature map is created and binarized (using the median of the image as a threshold). The MC method analyzed against rotation, translation and noise. For calculating the mismatch ratio, displacement values are selected c_h=50 and c_w=30, which are the distances for the variations in the vertical and horizontal directions, respectively and are used to adjust the displacement (see Section 3.2.1). As can be seen in Figure 4.7, the Maximum Curvature method is translation invariant because of selecting the displacement variables large enough (c_h=50 and c_w=30) and resistant to noise. However, this method is affected greatly by rotation because of changes in the gradient directions. The implementation of MC provided online at [50] is used in this thesis.

4.3.3 Wide Line Detector

Wide Line Detector method as explained in Algorithm 3 has three main parameters, two of which are the threshold values (t=1, g=41) and the third one is *Radius*, which is set to $r = 5$ based on the literature [4]. For extracting veins in the Wide Line Detector method, the WL map is created from the mask image (see Figure 4.8). For calculating the mismatch ratio, displacement values are selected



(a)

(b)

(c)

Figure 4.5: FAR vs. FRR graph for LT based on (a) Rotation, (b) Translation, (c) Gaussian noise.

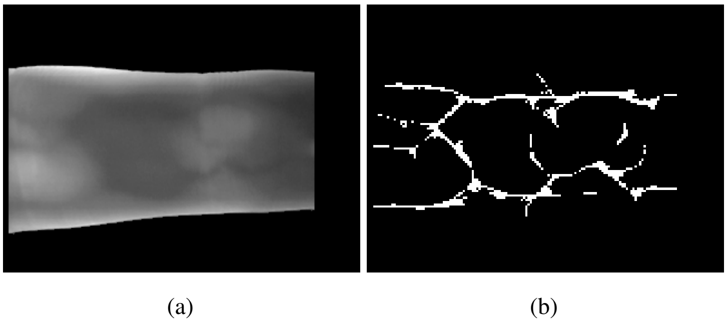


Figure 4.6: Example finger-vein extraction by the MC method. (a) Masked image, (b) Binary image produced by the MC method.

c_h=50 and c_w=30, which are the distances for the variations in the vertical and horizontal directions, respectively and are used to adjust the displacement (see Section 3.2.1). The WL method is analyzed against rotation, translation and noise. It is noticeable that Wide Line method is translation invariant (see Figure 4.9) because of large displacement values, however, is affected strongly by rotation and noise. The implementation of WL provided online at [51] is used in this thesis.

4.3.4 Fourier Descriptors

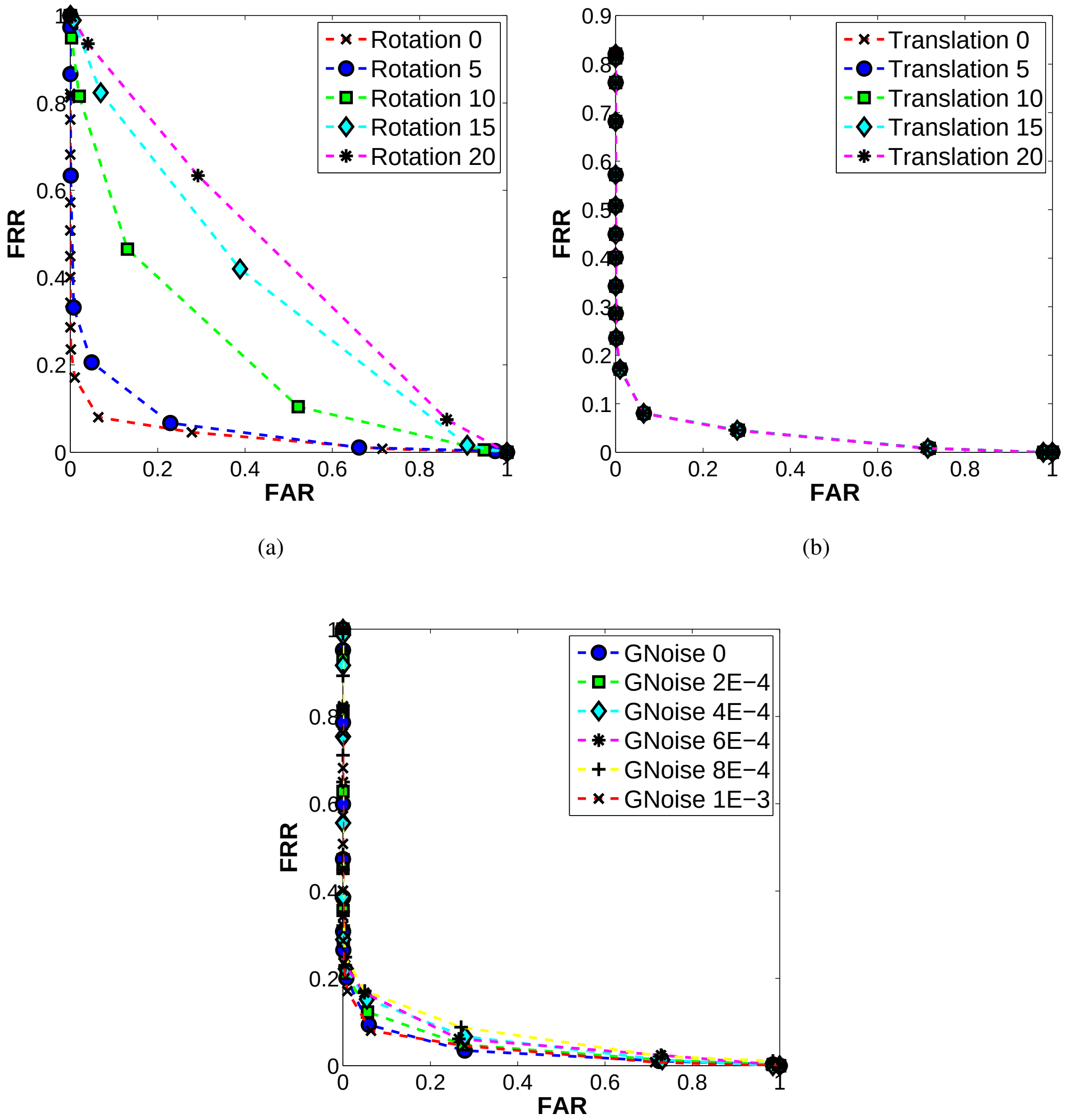
Fourier Descriptors method as mentioned in Algorithm 4 has only one parameter; the number of Fourier terms. Different coefficient numbers are set for different methods based on the experimental results (different coefficient numbers are analyzed and the one with the least error is selected, which have not been written in this thesis). Fourier coefficients value for Line tracking method is set to k=100, Fourier coefficients value for Maximum Curvature method is set to k=40 and for Wide Line Detector method is set to k=30 (See Section 3.1.5). Fourier Descriptors' performance is evaluated using three different distance matching algorithms called Euclidean distance, χ^2 distance and Earth Mover's distance.

4.3.5 Zernike Moments

Zernike moments method as described in Algorithm 5 has single parameter called order. Zernike moments' order for Line Tracking method is set to 44 and for Maximum Curvature and Wide Line methods to 40 (these numbers are set by the experimental results with the least error and are not written in this thesis). Zernike moments' performance is evaluated using three mentioned distances called Euclidean Distance, χ^2 distance and Earth Mover's distance. The implementation of ZM provided online at [52] is used in this thesis.

4.3.6 Local Binary Patterns

Local Binary Patterns as mentioned in Algorithm 6 does not require any parameters. Using three mentioned distances called Euclidean Distance, χ^2 distance and Earth Mover's distance Local Binary Patterns' performance is calculated. The implementation of LBP provided online at [9] is used in this thesis.

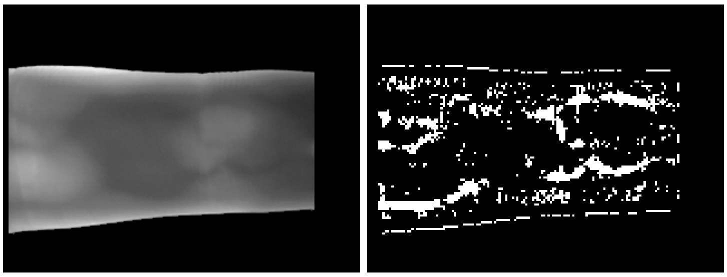


(a)

(b)

(c)

Figure 4.7: FAR vs. FRR graph for MC based on (a) Rotation, (b) Translation, (c) Gaussian noise.



(a) (b)

Figure 4.8: Example finger-vein extraction by the WL method. (a) Masked image, (b) Binary image produced by the WL method.

4.3.7 Histogram of Oriented Gradients

Histogram of Oriented Gradients as described in Algorithm 7 computes histograms of image gradients which are obtained by dividing the image into cells. This method does not require any parameters. The implementation of HOG provided online at [53] is used in this thesis.

4.3.8 GBP

Global Binary Patterns as described in Algorithm 8 has been analyzed over horizontal and vertical projections denoted by GBP_h and GBP_v, respectively. This method does not require any parameters. However, there are other types of GBP that require parameters which are not used in this study and simple form of GBP is analyzed. The implementation of GBP provided online at [9] is used in this thesis.

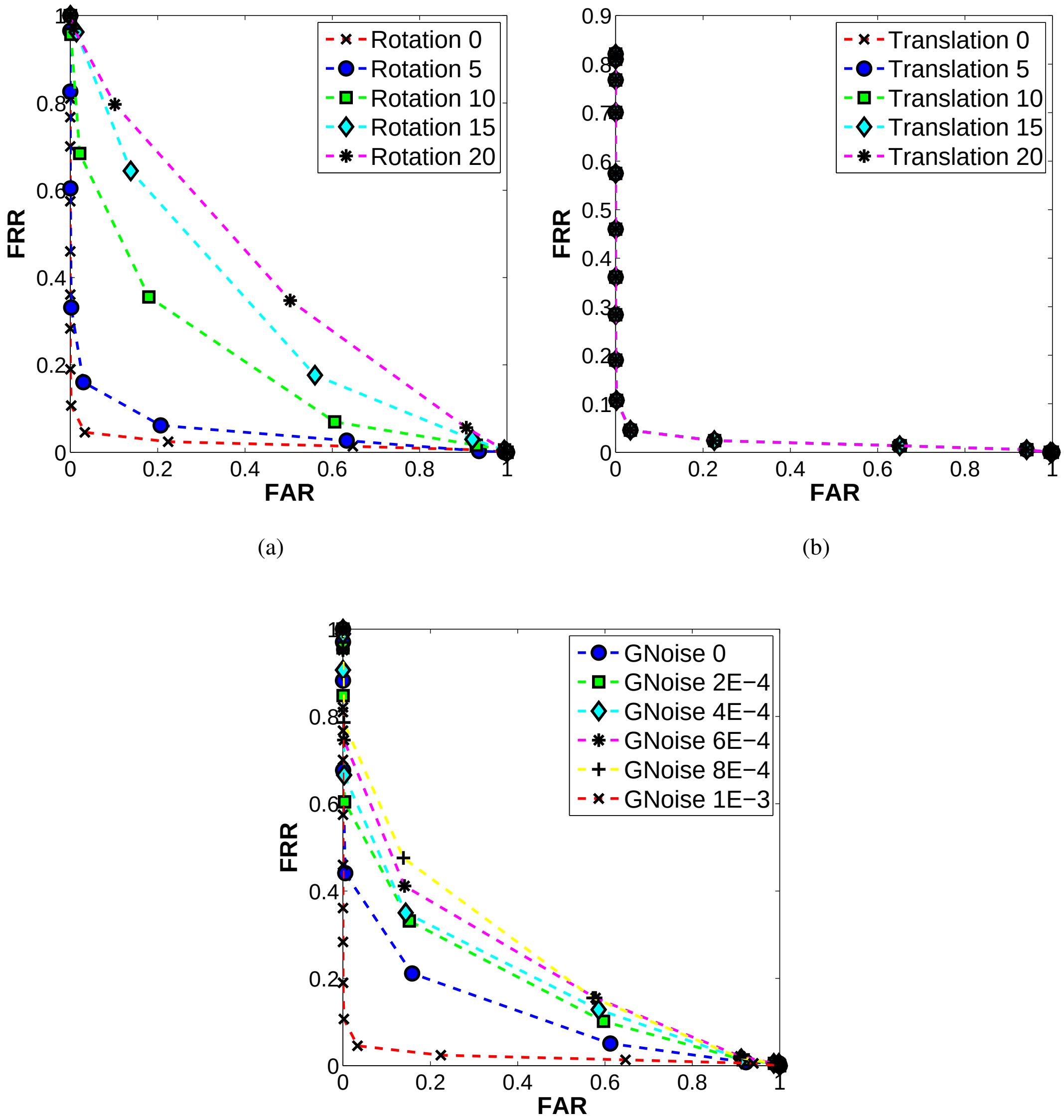
4.4 Comparison of Performances

In this section the methods are compared to each other and their performances are analyzed under translation, rotation and noise. Evaluation is based on the EER graphs and for the sake of space, some of the ROC curves are placed in Appendix A. Original matching method of LT, MC and WL (mismatch ratio) as well as Euclidean distance, χ^2 and Earth Mover's distance are evaluated.

As can be seen in Figure 4.10, for the rotation performance of the methods, all the methods except ZM-variants are severely affected by rotation. ZM is not affected by rotation because calculating the magnitude of Zernike Moments uses distance from the origin which is not affected by rotation.

In terms of translation performance of all methods, it can be seen that HOG-based features are not affected by translation. HOG is not affected by translation because histograms that HOG extracts from each block are not affected by shifts in positions. Moreover, in terms of translation, all methods seem to be good enough to be called translation invariant except GBP (see Figure 4.11).

None of the methods are robust against noise as can be seen in Figure 4.12, except FD which is the worst method (when FAR-FRR is concerned) and GBP that is affected less in comparison to the other methods which are affected severely by Gaussian noise. In FD low frequency components are used, therefore, FD is robust against noise because noise affects high frequency components more than low frequency components.



(a)

(b)

(c)

Figure 4.9: FAR vs. FRR graph for WL based on (a) Rotation, (b) Translation, (c) Gaussian noise.

As smaller FAR and FRR are concerned therefore the least errors in each method are selected. Figure 4.13 displays the FAR-FRR graph for the best variant of each method (e.g., the best of FD among FD+LT, FD+MC and FD+WL). As in Figure 4.13, all methods' performance using Euclidean distance, χ^2 and EMD against WL are better than LT and MC. In the meanwhile, it can be seen that LT, MC and WL provide the best performances. However, HOG+WL provides better FAR-FRR. This is striking because HOG is usually used in human detection and tracking applications and simple histogramming of gradients in grids may provide better results than directly comparing images using mismatch ratios. ZM+WL result is comparable with HOG+WL since ZM is invariant to translation and rotation. Furthermore, GBP+WL has comparable result with HOG+WL and ZM+WL. Surprisingly, FD, which has been successfully applied to similar problems such as hand-written digit recognition, yields the worst results as far as the FAR-FRR values are concerned.

Finally, the best of each distance metric is compared in Figure 4.13. As can be seen in Figure 4.13, using Euclidean and χ^2 matching methods ZM+WL, HOG+WL and GBP+WL are the best and using EMD metric HOG+WL and GBP+WL produce better results. The best results of each distance metric in Figure 4.13 are placed in Figure 4.14. As in Figure 4.14, EMD performs better than χ^2 and Euclidean distance because of Cross-Bin dissimilarity in which non-identical indexes of bins are compared as well and HOG outperforms in comparison to the all other methods.

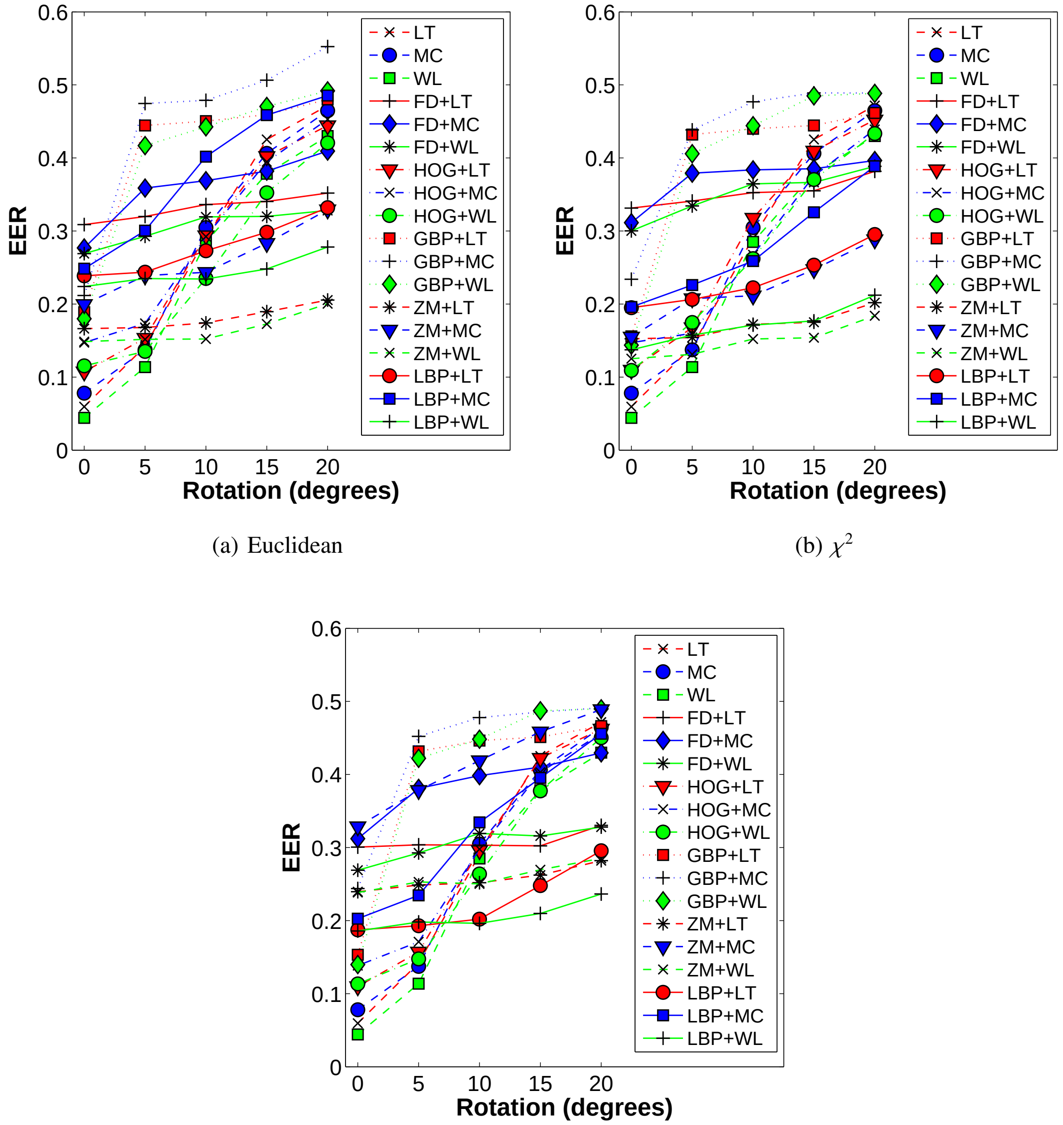
4.5 Comparison of Running Times

In this section, the running time performance of the algorithms is analyzed. All methods are run on the same machine using MATLAB. A single image is compared against a total of 100 images in the database, and the average times for both feature extraction and feature matching over 100 images are calculated.

As shown in Table 4.1, FD, GBP and HOG are the fastest among the methods since their computational complexities are quite low; whereas the WL method, provides the best results when there is no translation, rotation or noise is among the best in terms of speed. On the other hand, the LT method which is the next method that provide the best results after the WL, when there is no translation, rotation or noise, is the slowest. The reason for LT's slowness is due to the fact that the feature extraction step requires tracing veins for several iterations.

4.6 Discussion

In this section, resilience of the methods to rotation, translation and noise are analyzed. For the sake of clarity, ROC curves are placed in Appendix A. From the results provided in Section 4.4 and 4.5, it can be seen that ZM method performs better under rotation. For this reason, for applications requiring invariancy to rotation, ZM method can be preferred. In the case of translation all methods are resistant against translation except GBP. Therefore for applications requiring translation invariancy, all methods except GBP can be preferred. The provided results in Sections 4.4 and 4.5 imply that WL method has the least error rate, therefore in the case of high secure systems WL method can be used. Moreover, HOG+WL, GBP+WL and ZM+WL results are comparable to WL and as mentioned before in the case of rotation invariancy ZM+WL is the best choise, in the case of translation invariancy HOG+WL is the most suitable choice and in the case of resistance to noise GBP is the better choice. The performance of the methods under rotation, translation and noise is summarized in Table 4.2.

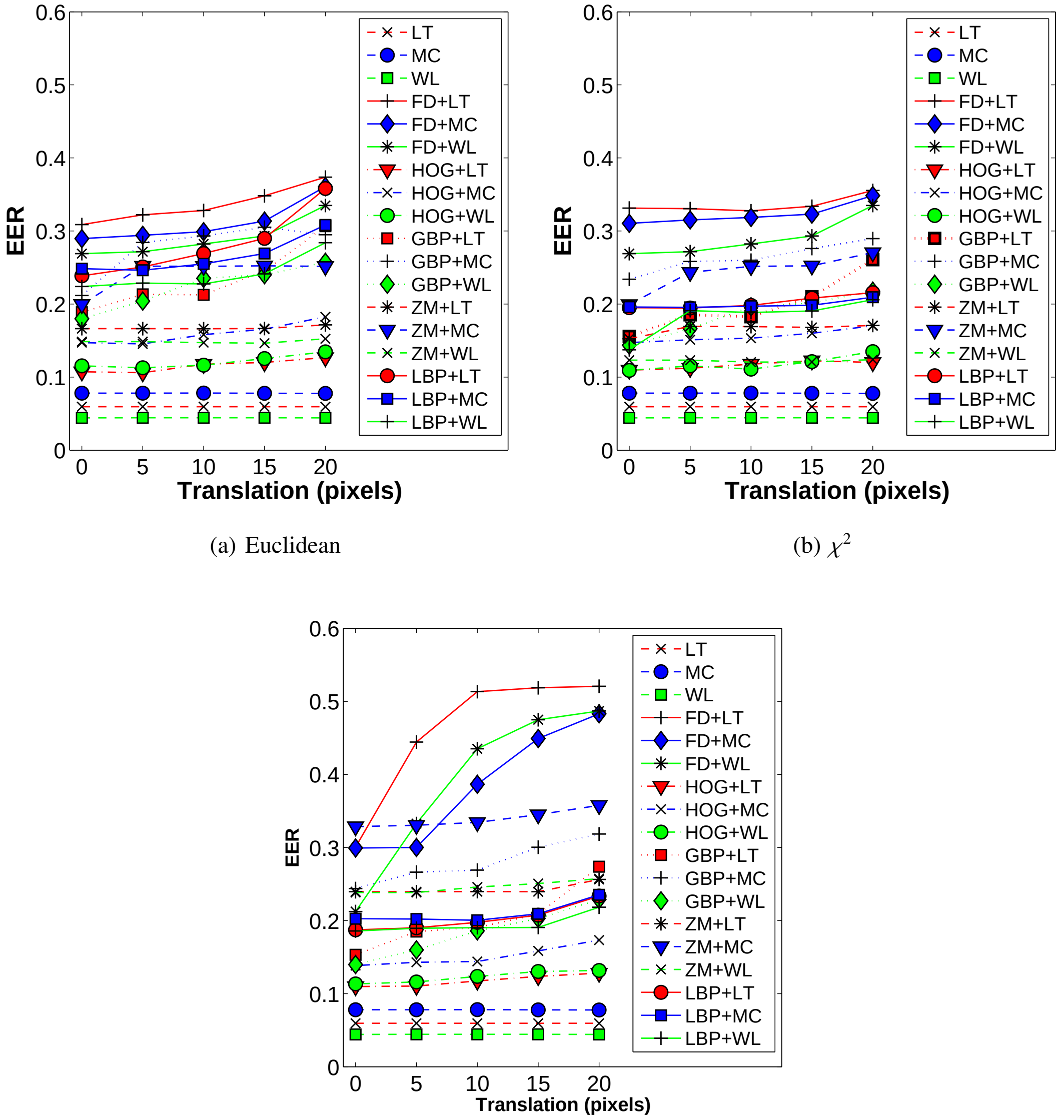


(a) Euclidean

(b) χ^2

(c) EMD

Figure 4.10: Analysis of distance and rotation (a) Euclidean, (b) χ^2, (c) EMD.

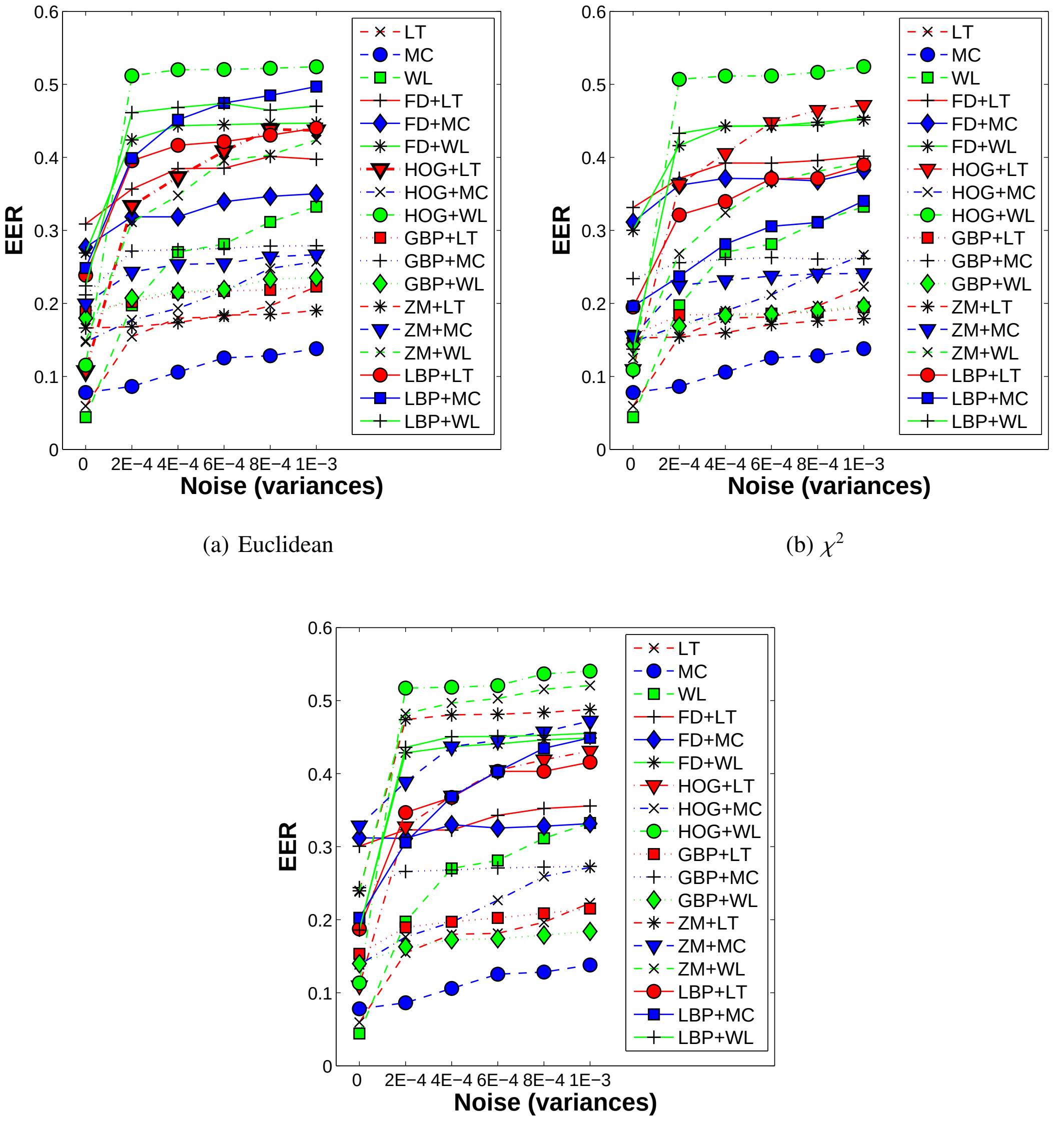


(a) Euclidean

(b) χ^2

(c) EMD

Figure 4.11: Analysis of distance and translation (a) Euclidean, (b) χ^2, (c) EMD.

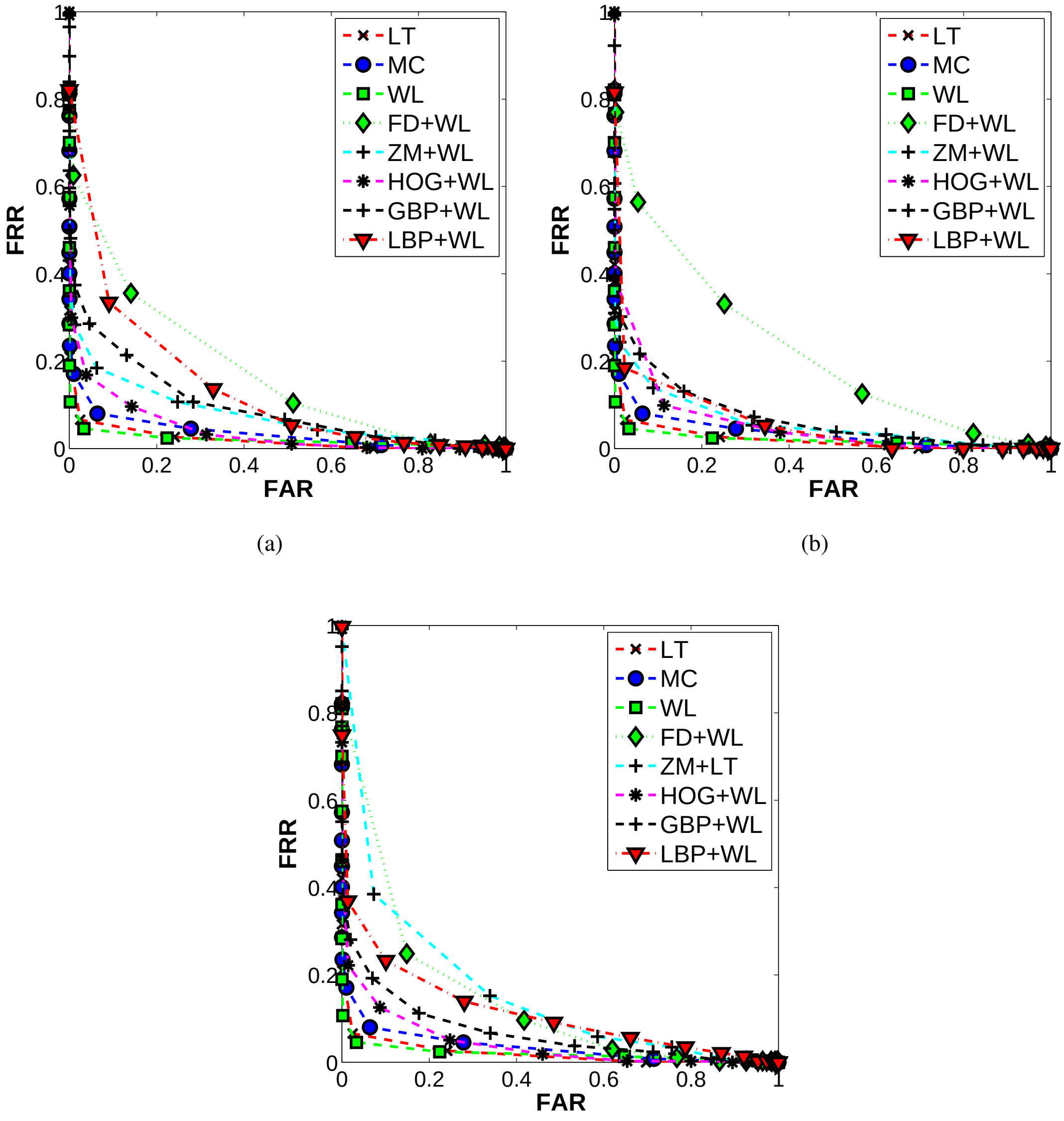


(a) Euclidean

(b) χ^2

(c) EMD

Figure 4.12: Analysis of distance and Gaussian noise (a) Euclidean, (b) χ^2, (c) EMD.



(a)

(b)

(c)

Figure 4.13: FAR vs. FRR graphs for the best of all methods compared with LT+mismatch, MC+mismatch and WL+mismatch (a) Using Euclidean distance, (b) Using χ^2 distance, (c) Using Earth Mover's distance.

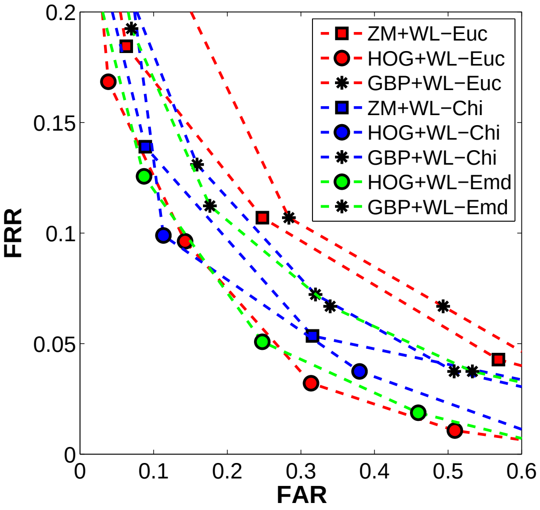


Figure 4.14: Best of Euclidean, χ^2 and Earth Mover's distances

.

4.6.1 Evaluation of the results with respect to the literature

In this section, the obtained EER values are evaluated with respect to the EER values reported by the finger-vein recognition literature. However, note that a direct comparison of the results is not plausible since different datasets and modalities have been used. The selection of the dataset has a great impact on the performance of the methods. This makes the direct comparison of different techniques very hard and in most cases impossible. Furthermore, some methods' EER is very small on a certain dataset like MC method in Miura et al. [3], however, it can have bigger EER in another type of dataset as in Qin et al. [23]. As shown in Table 4.3, it can be seen that the results obtained in the thesis are close to the ones reported in the finger-vein literature. However, the performance of LT in this thesis is better than some of the reported results in the literature, and the reverse is true for MC and WL. Finger vein detection technology is mostly combined with other modalities (mostly fingerprint and finger geometry) to have better performance. For example, as shown by Kang et al. [26], combination of fingerprint with finger vein can decrease the EER value from 1.433% to 0.803% . Note that finger vein and fingerprint can be combined for better performance [26], [54].

Table4.1: Running-time comparison of all methods.

Method	Time (seconds)	Speed Rank
LT (feat. extraction only)	9.2831	15
MC (feat. extraction only)	0.0585	1
WL (feat. extraction only)	0.1286	8
LT	9.4675	19
MC	0.0595	3
WL	0.1335	10
FD + LT + Euc	9.2837	16
FD + MC + Euc	0.0586	2
FD + WL + Euc	0.1289	9
ZM + LT + Euc	9.6507	21
ZM + MC + Euc	0.5183	7
ZM + WL + Euc	0.5920	14
GBP + LT + Euc	9.2850	17
GBP + MC + Euc	0.0601	4
GBP + WL + Euc	0.1454	12
HOG + LT + Euc	9.2925	18
HOG + MC + Euc	0.0664	5
HOG + WL + Euc	0.1368	11
LBP + LT + Euc	9.5831	20
LBP + MC + Euc	0.0890	6
LBP + WL + Euc	0.1686	13

Table4.2: Comparison of resilience to rotation, translation and noise

Methods	Rotation	Translation	Noise
Fourier Descriptors	Yes	Yes	Yes
Zernike Moments	Yes	Yes	No
Local Binary Patterns	No	Yes	No
Global Binary Patterns	No	No	Yes
Histogram of Oriented Gradients	No	Yes	No

Table4.3: Comparison to the reported results in the finger-vein recognition literature.

References	**Method**	**Database** **fingers × samples per each**	**Performance**
[2] (orig.)	Line Tracking	339 × 2 images	EER: 0.145%
[33]	Line Tracking	125 × 1 images	EER: 7.87%
[23]	Line Tracking	200 × 8 images	EER: 0.1561%
This study	Line Tracking	636 × 6 images	EER: 5.94%
[33]	Mean Curvature	125 × 9 images	EER: 0.25%
[3] (orig.)	Maximum Curvature	339 × 2 images	EER: 0.0009%
[23]	Maximum Curvature	200 × 8 images	EER: 0.0726%
[20]	Maximum Curvature	100 × 10 images	EER: 25%
This study	Maximum Curvature	636 × 6 images	EER: 7.80%
[4] (orig.)	Wide Line Detector	10,140 × 5 images	EER: 2.86%
[29]	Wide Line Detector	10,140 × 5 images	EER: 0.87%
This study	Wide Line Detector	636 × 6 images	EER: 4.42%
[32]	Moment invariants	50 × 4 images	EER: 8.93%
[28]	curvelets and local interconnection structure	400 × 8 images	EER: 0.128%
[23]	Region growth- based	320 × 5 images	EER: 0.0369%
[45]	LBP	204 × 10 images	EER: 9.553%
[45]	LDP	204 × 10 images	EER: 12.301%
This study	HOG+WL	636 × 6 images	EER: 10.92%
This study	ZM+WL	636 × 6 images	EER: 12.52%
This study	GBP+WL	636 × 6 images	EER: 14.38%

CHAPTER 5

CONCLUSION

In this thesis, methods widely used for finger-vein recognition, namely, Line Tracking (LT), Maximum Curvature (MC) and Wide Line Detector (WL), are analyzed and extended for robustness against rotation, translation and noise as well as faster running times. For this end, widely used Computer Vision and Pattern Recognition methods such as Fourier descriptors (FD), Zernike moments (ZM), Local Binary Patterns (LBP), Global Binary Patterns (GBP) and Histogram of Oriented Gradients (HOG) are used.

It is observed that WL outperforms in terms of accuracy, however, it is affected by rotation and noise. Methods like ZM, HOG and GBP provide comparable performances and FD, HOG and GBP have better running times and ZM is invariant to rotation and translation, however, its running time is slower than HOG and GBP. Fourier descriptors are robust against noise, however, the error rate is very high. On the other hand, all other methods are affected strongly by noise except GBP.

The results indicate that in the case of no noise, no rotation and no translation, WL is the best method. In a scenario involving rotation, ZM+WL, and in a scenario involving noise GBP+WL can be used. If time is a critical issue, then MC can be chosen since it is the next best method in terms of EER values and the fastest among all methods.

5.1 Future Work

There are different feature description methods in the literature that can be analyzed. For example, there are extensions of LBP, HOG and GBP that provide several invariances, such as rotation invariance, scale invariance. These extensions are worth analyzing as future work. For analyzing resilience to rotation and translation, rotating the finger in different directions and axes will be much realistic, moreover, changing the distance between the finger and camera can be tested instead of artificially rotating, translating and adding noise to the images.

In the thesis, only one database is used for analyzing the performance of the methods. However, for better evaluation and analysis, other databases, especially those with more images and more distortions should be used.

For better analysis of the parameters and the performances of the methods Statistical Package for the Social Sciences (SPSS) software can be used as a future work.

REFERENCES

[1] Hemant Vallabh. *Authentication using finger-vein recognition*. PhD thesis, University of Johannesburg, 2013.

[2] Naoto Miura, Akio Nagasaka, and Takafumi Miyatake. Feature extraction of finger-vein patterns based on repeated line tracking and its application to personal identification. *Machine Vision and Applications*, 15(4):194–203, 2004.

[3] Naoto Miura, Akio Nagasaka, and Takafumi Miyatake. Extraction of finger-vein patterns using maximum curvature points in image profiles. *IEICE Transactions on Information and Systems*, 90(8):1185–1194, 2007.

[4] Beining Huang, Yanggang Dai, Rongfeng Li, Darun Tang, and Wenxin Li. Finger-vein authentication based on wide line detector and pattern normalization. In *20th International Conference on Pattern Recognition (ICPR)*, pages 1269–1272. IEEE, 2010.

[5] Rafael C Gonzalez, Richard E Woods, and Steven L Eddins. *Digital image processing using MATLAB*, volume 2. Gatesmark Publishing Knoxville, 2009.

[6] Michael R Teague. Image analysis via the general theory of moments. *Journal of Optical Society of America (JOSA)*, 70(8):920–930, 1980.

[7] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 886–893. IEEE, 2005.

[8] Timo Ojala, Matti Pietikäinen, and David Harwood. A comparative study of texture measures with classification based on featured distributions. *Pattern recognition*, 29(1):51–59, 1996.

[9] Erdal Sivri. Shape descriptors based on intersection consistency and global binary patterns. Master's thesis, Middle East Technical University, Ankara, Turkey, 2012.

[10] David C Hitchcock. *Evaluation and combination of biometric authentication systems*. PhD thesis, University of Florida, 2003.

[11] John D Woodward, Nicholas M Orlans, and Peter T Higgins. *Biometrics:[identity assurance in the information age]*. McGraw-Hill/Osborne Berkeley, California, USA, 2003.

[12] Anil K Jain, Ruud M Bolle, and Sharath Pankanti. *Biometrics: personal identification in networked society*. Springer, 1999.

[13] Thiems Nanavati. *Biometrics*. John Wiley & Sons, 2002.

[14] Sargur N Srihari, Harish Srinivasan, and Gang Fang. Discriminability of fingerprints of twins. *Journal of Forensic Identification*, 58(1):109, 2008.

[15] Takashi Yanagawa, Satoshi Aoki, and Tetsuji Ohyama. Human finger vein images are diverse and its patterns are useful for personal identification. *2007-12*, 2007.

[16] Jinfeng Yang, Yihua Shi, Jinli Yang, and Lihui Jiang. A novel finger-vein recognition method with feature combination. In *16th IEEE International Conference on Image Processing (ICIP)*, pages 2709–2712. IEEE, 2009.

[17] Jinfeng Yang, Yihua Shi, and Jinli Yang. Personal identification based on finger-vein features. *Computers in Human Behavior*, 27(5):1565–1570, 2011.

[18] Jinfeng Yang and Yihua Shi. Finger–vein roi localization and vein ridge enhancement. *Pattern Recognition Letters*, 33(12):1569–1579, 2012.

[19] Wang Kejun, Liu Jingyu, Popoola P Oluwatoyin, and Feng Weixing. Finger vein identification based on 2-d gabor filter. In *2nd International Conference on Industrial Mechatronics and Automation (ICIMA)*, volume 2, pages 10–13. IEEE, 2010.

[20] Jinfeng Yang and Xu Li. Efficient finger vein localization and recognition. In *20th International Conference on Pattern Recognition (ICPR)*, pages 1148–1151. IEEE, 2010.

[21] Li Xueyan, Guo Shuxu, Gao Fengli, and Li Ye. Vein pattern recognitions by moment invariants. In *The 1st International Conference on Bioinformatics and Biomedical Engineering (ICBBE)*, pages 612–615. IEEE, 2007.

[22] Nurhafizah Mahri, Sas Suandi, and Bakhtiar A Rosdi. Finger vein recognition algorithm using phase only correlation. In *2010 International Workshop on Emerging Techniques and Challenges for Hand-Based Biometrics (ETCHB)*, pages 1–6. IEEE, 2010.

[23] Huafeng Qin, Lan Qin, and Chengbo Yu. Region growth–based feature extraction method for finger-vein recognition. *Optical Engineering*, 50(5):057208–057208, 2011.

[24] Eui C Lee, Hyunwoo Jung, and Daeyeoul Kim. New finger biometric method using near infrared imaging. *Sensors*, 11(3):2319–2333, 2011.

[25] Gongping Yang, Xiaoming Xi, and Yilong Yin. Finger vein recognition based on a personalized best bit map. *Sensors*, 12(2):1738–1757, 2012.

[26] Byung J Kang, Kang R Park, Jang H Yoo, and Jeong N Kim. Multimodal biometric method that combines veins, prints, and shape of a finger. *Optical Engineering*, 50(1):017201–017201, 2011.

[27] Leila Mirmohamadsadeghi and Andrzej Drygajlo. Palm vein recognition with local binary patterns and local derivative patterns. In *International Joint Conference on Biometrics (IJCB)*, pages 1–6. IEEE, 2011.

[28] Zhongbo Zhang, Siliang Ma, and Xiao Han. Multiscale feature extraction of finger-vein patterns based on curvelets and local interconnection structure neural network. In *18th International Conference on Pattern Recognition (ICPR)*, volume 4, pages 145–148. IEEE, 2006.

[29] Gongping Yang, Xiaoming Xi, and Yilong Yin. Finger vein recognition based on $(2d)^2$ pca and metric learning. *Journal of Biomedicine and Biotechnology*, 2012:9, 2012.

[30] Naoto Miura, Akio Nagasaka, and Takafumi Miyatake. Automatic feature extraction from non-uniform finger vein image and its application to personal identification. In *Machine Vision Applications (MVA)*, pages 253–256, 2002.

[31] Wenming Yang, Xiang Yu, and Qingmin Liao. Personal authentication using finger vein pattern and finger-dorsa texture fusion. In *Proceedings of the 17th ACM international conference on Multimedia*, pages 905–908. ACM, 2009.

[32] Xiaohua Qian, Shuxu Guo, Xueyan Li, Fei Zhong, and Xiangxin Shao. Finger-vein recognition based on the score level moment invariants fusion. In *International Conference on Computational Intelligence and Software Engineering (CiSE)*, pages 1–4. IEEE, 2009.

[33] Wonseok Song, Taejeong Kim, Hee C Kim, Joon H Choi, Hyoun J Kong, and Seung R Lee. A finger-vein verification system using mean curvature. *Pattern Recognition Letters*, 32(11):1541–1547, 2011.

[34] Jiang Hong, Guo Shuxu, Li Xueyan, and Qian Xiaohua. Vein pattern extraction based on the position-gray-profile curve. In *2nd International Congress on Image and Signal Processing (CISP)*, pages 1–4. IEEE, 2009.

[35] Laura Liu, David Zhang, and Jane You. Detecting wide lines using isotropic nonlinear filtering. *IEEE Transactions on Image Processing*, 16(6):1584–1595, 2007.

[36] Lionel Evina Ekombo, Noureddine Ennahnahi, Mohammed Oumsis, and Mohammed Meknassi. Application of affine invariant fourier descriptor to shape based image retrieval. *International Journal of Computer Science and Network Security*, 9(7):240–247, 2009.

[37] Berrin Yanikoglu and Alisher Kholmatov. Online signature verification using fourier descriptors. *Journal on Advances in Signal Processing (EURASIP)*, 2009:12, 2009.

[38] Ming K Hu. Visual pattern recognition by moment invariants. *IRE Transactions on Information Theory*, 8(2):179–187, 1962.

[39] Sun K Hwang and Whoi Y Kim. A novel approach to the fast computation of zernike moments. *Pattern Recognition*, 39(11):2065–2076, 2006.

[40] Alireza Khotanzad and Yaw H Hong. Invariant image recognition by zernike moments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(5):489–497, 1990.

[41] Amir Tahmasbi, Fatemeh Saki, and Shahriar B Shokouhi. Classification of benign and malignant masses based on zernike moments. *Computers in Biology and Medicine*, 41(8):726–735, 2011.

[42] Chee W Chong, Paramesran Raveendran, and Ramakrishnan Mukundan. Translation invariants of zernike moments. *Pattern Recognition*, 36(8):1765–1773, 2003.

[43] Topi Mäenpää and Matti Pietikäinen. Texture analysis with local binary patterns. *Handbook of Pattern Recognition and Computer Vision*, 3:197–216, 2005.

[44] Timo Ojala, Matti Pietikainen, and Topi Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):971–987, 2002.

[45] Bakhtiar A Rosdi, Chai W Shing, and Shahrel A Suandi. Finger vein recognition using local line binary pattern. *Sensors*, 11(12):11357–11371, 2011.

[46] Amnart Petpon and Sanun Srisuk. Face recognition with local line binary pattern. In *Fifth International Conference on Image and Graphics (ICIG)*, pages 533–539. IEEE, 2009.

[47] Tatsuo Kozakaya, Tomoyuki Shibata, Mayumi Yuasa, and Osamu Yamaguchi. Facial feature localization using weighted vector concentration approach. *Image and Vision Computing*, 28(5):772–780, 2010.

[48] Anil K Jain, Robert P Duin, and Jianchang Mao. Statistical pattern recognition: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1):4–37, 2000.

[49] Yossi Rubner, Carlo Tomasi, and Leonidas J Guibas. The earth mover's distance as a metric for image retrieval. *International Journal of Computer Vision*, 40(2):99–121, 2000.

[50] Line Tracking and Maximum Curvature methods matlab code. `http://www.mathworks.com/matlabcentral/fileexchange/35716-miura-et-al-vein-extraction-methods`. Accessed: 2012-12-10.

[51] Wide Line method matlab code. `http://www.mathworks.com/matlabcentral/fileexchange/35754-wide-line-detector`. Accessed: 2012-12-20.

[52] Zernike moment method matlab code. `http://liris.cnrs.fr/christian.wolf/software/zernike/index.html`. Accessed: 2013-01-15.

[53] Histogram of Oriented Gradient method matlab code. `http://www.mathworks.com/matlabcentral/fileexchange/33863-histograms-of-oriented-gradients`. Accessed: 2013-07-10.

[54] Beyun J Kang and Kang R Park. Multimodal biometric method based on vein and geometry of a single finger. *IET Computer Vision*, 4(3):209–217, 2010.

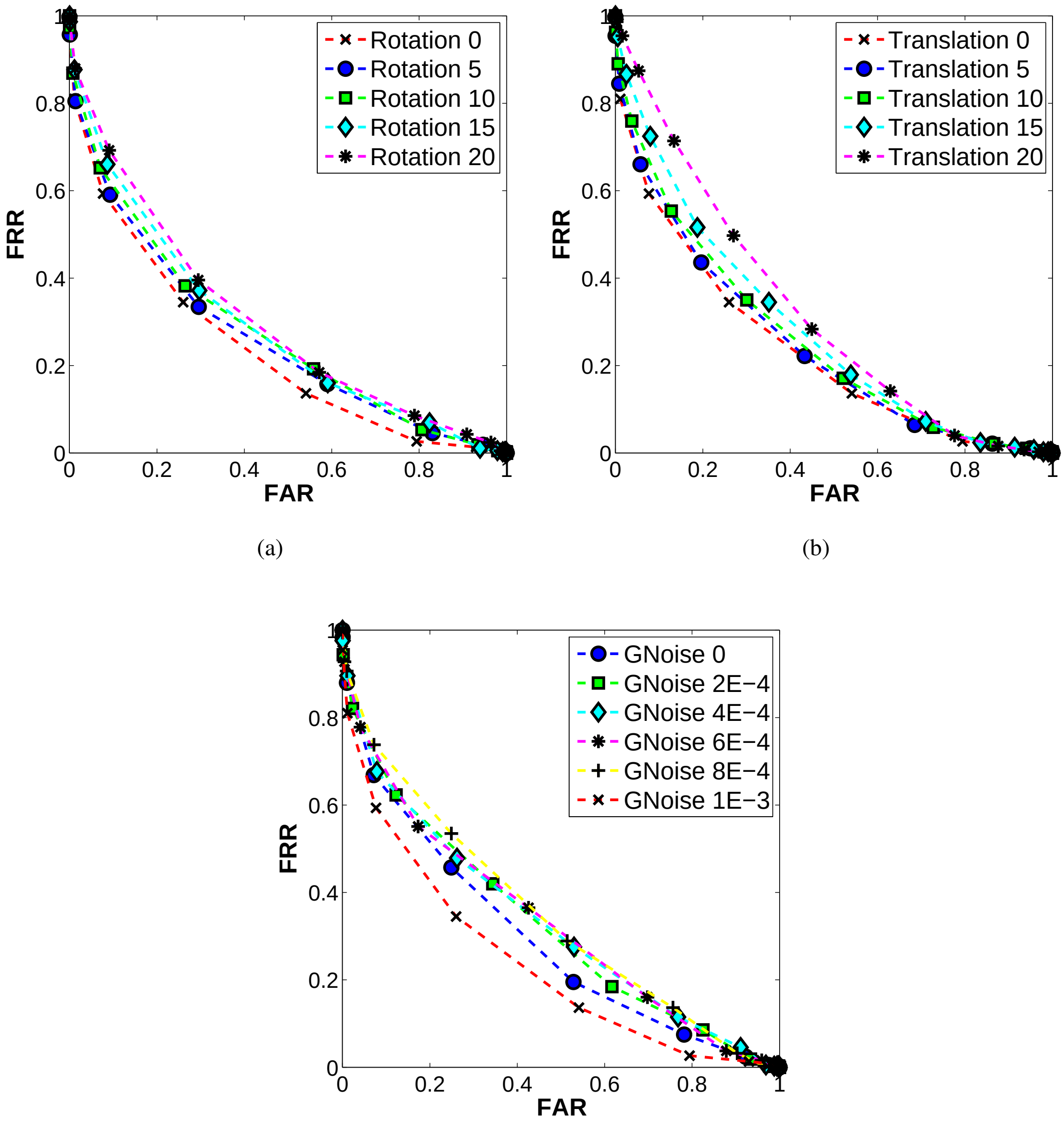
APPENDIX A

DETAILED RESULTS ON THE PERFORMANCE OF THE METHODS

In this section performance of the methods are analyzed in detail and the results are based on the ROC curves. Figures illustrate the affect of rotation, translation and noise on the methods including FD, ZM, HOG, LBP and GBP. As mentioned before in Section 4, arbitrary N images are selected from the database which are translated horizontally for different shift amounts $(0, 5, 10, 15$ and 20 pixels) and rotated clockwise for different angles $(0^o, 5^o, 10^o, 15^o$ and $20^o)$ to analyze the methods against their resistance to translation and rotation. Furthermore, Gaussian noise is added to the images, where the variance of the Gaussian was in the set $(0.0002, 0.0004, 0.0006, 0.0008$ and $0.001)$ and the mean was kept zero.

As can be seen in Figures A.46, A.47 and A.48, FD, ZM, HOG, LBP and GBP are applied to LT, MC and WL and it is noticeable that WL outperforms all other methods.

ZM method performs better under rotation as discussed in Section 4.6 , therefore can be used in the applications requiring invariancy to rotation. In the case of translation all methods are resistant against translation except GBP. However, GBP is resistant against noise and all other methods are affected greatly under noise (figure numbers are not provided here since there are many figures).

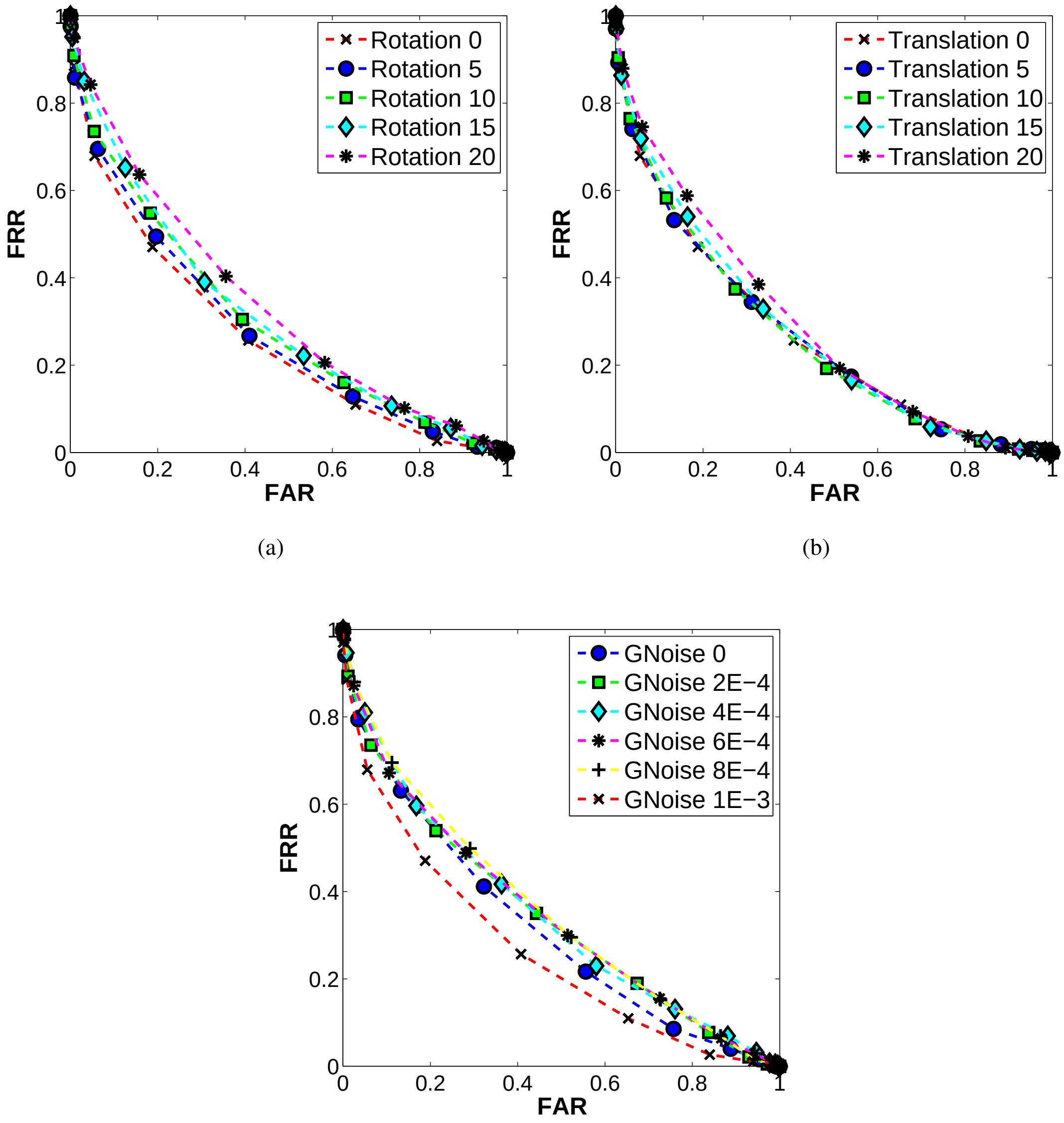


(a)

(b)

(c)

Figure A.1: FAR vs. FRR graph for FD-LT variations using Euclidean distance matching based on (a) Rotation, (b) Translation, (c) Gaussian noise.

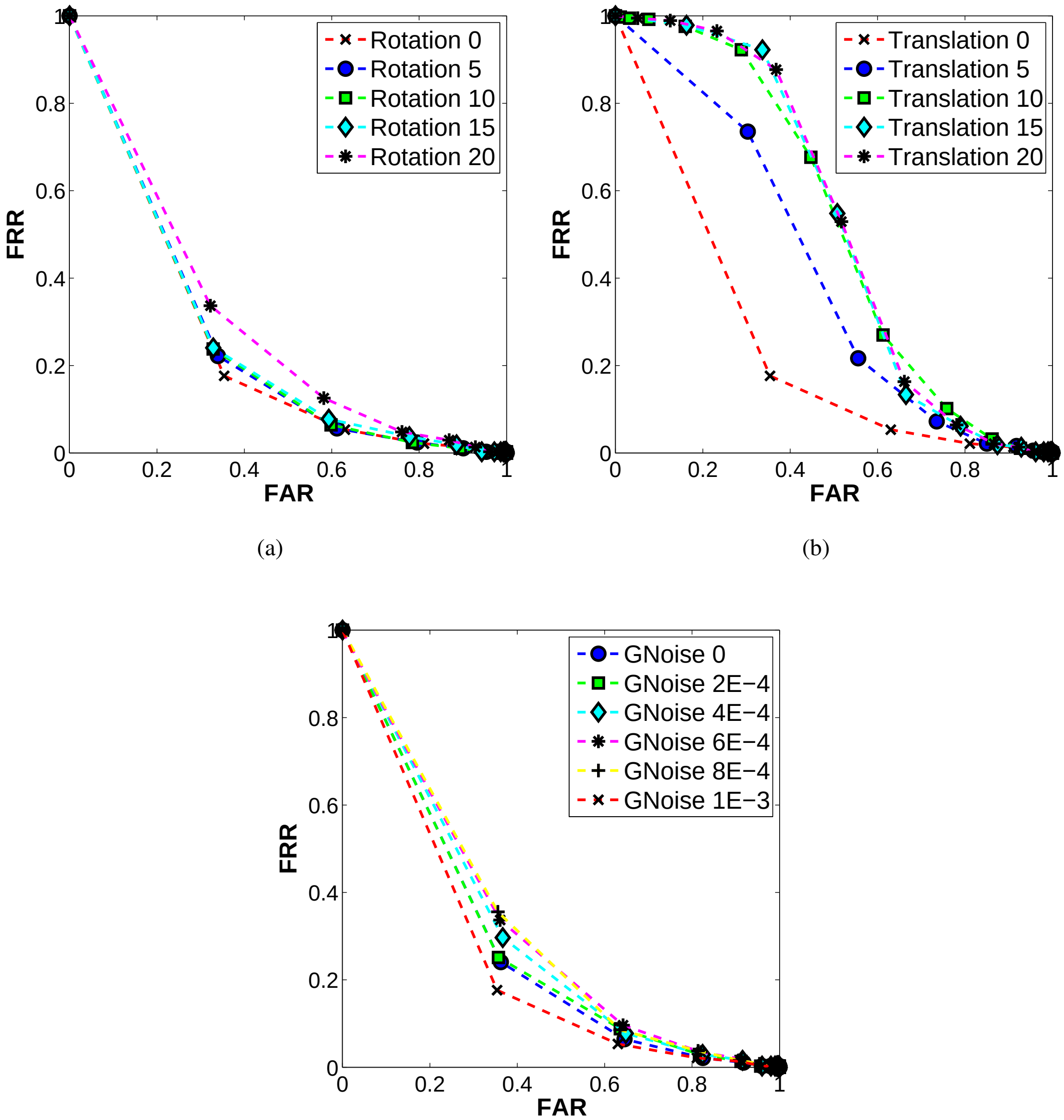


(a)

(b)

(c)

Figure A.2: FAR vs. FRR graph for FD-LT variations using χ^2 distance matching based on (a) Rotation, (b) Translation, (c) Gaussian noise.

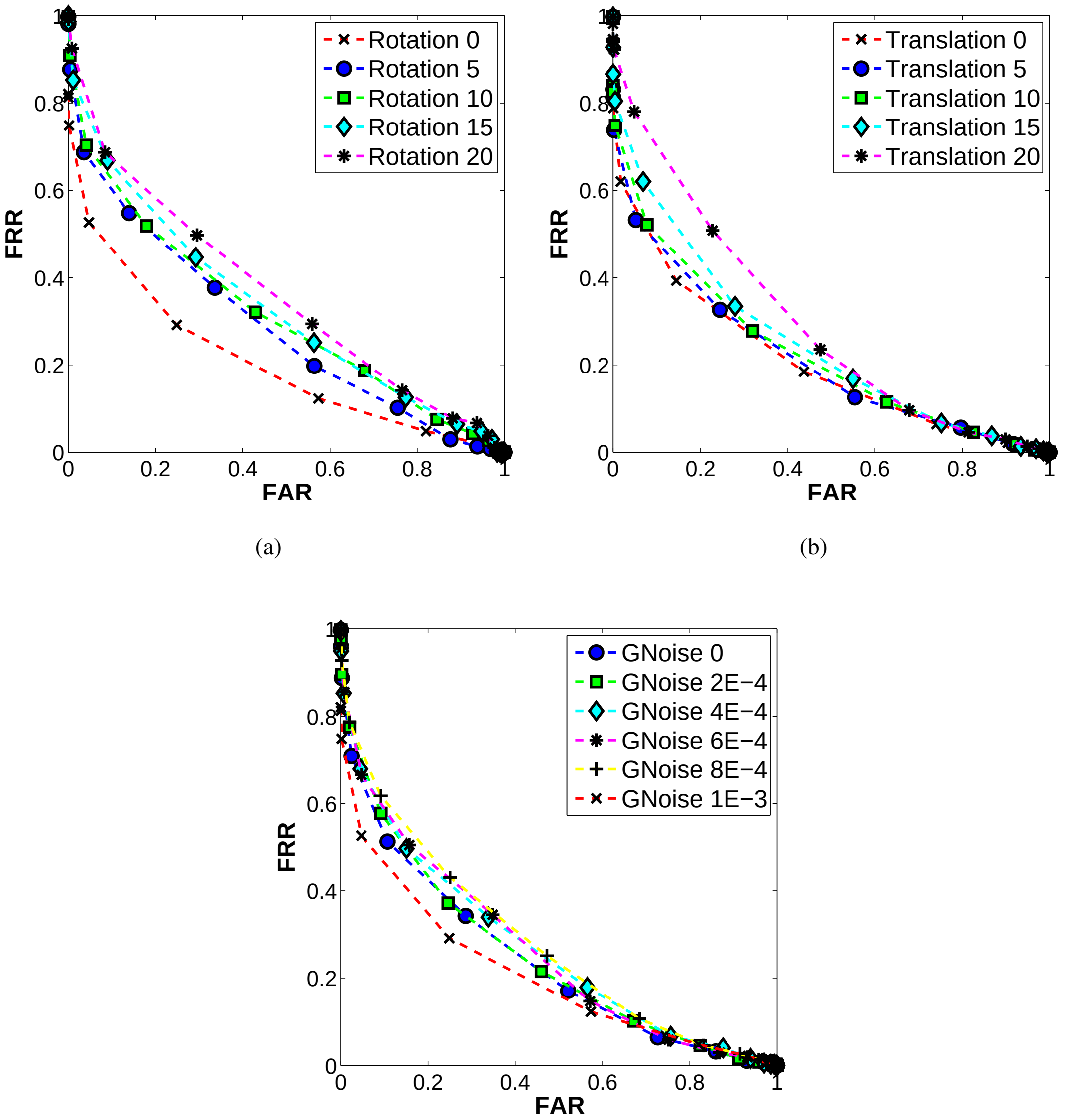


(a)

(b)

(c)

Figure A.3: FAR vs. FRR graph for FD-LT variations using Earth Mover's distance matching based on (a) Rotation, (b) Translation, (c) Gaussian noise.

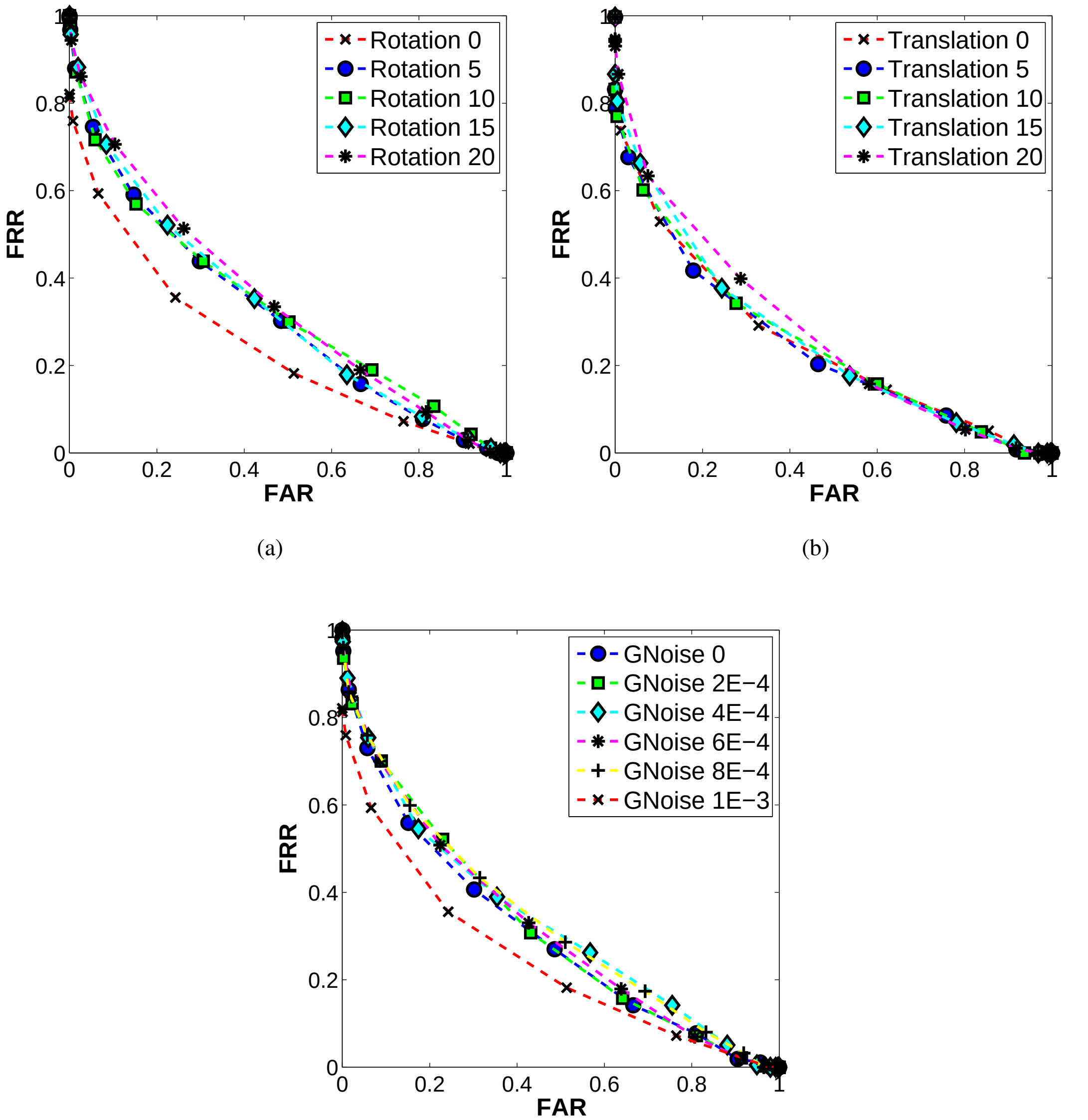


(a)

(b)

(c)

Figure A.4: FAR vs. FRR graph for FD-MC variations using Euclidean distance matching based on (a) Rotation, (b) Translation, (c) Gaussian noise.

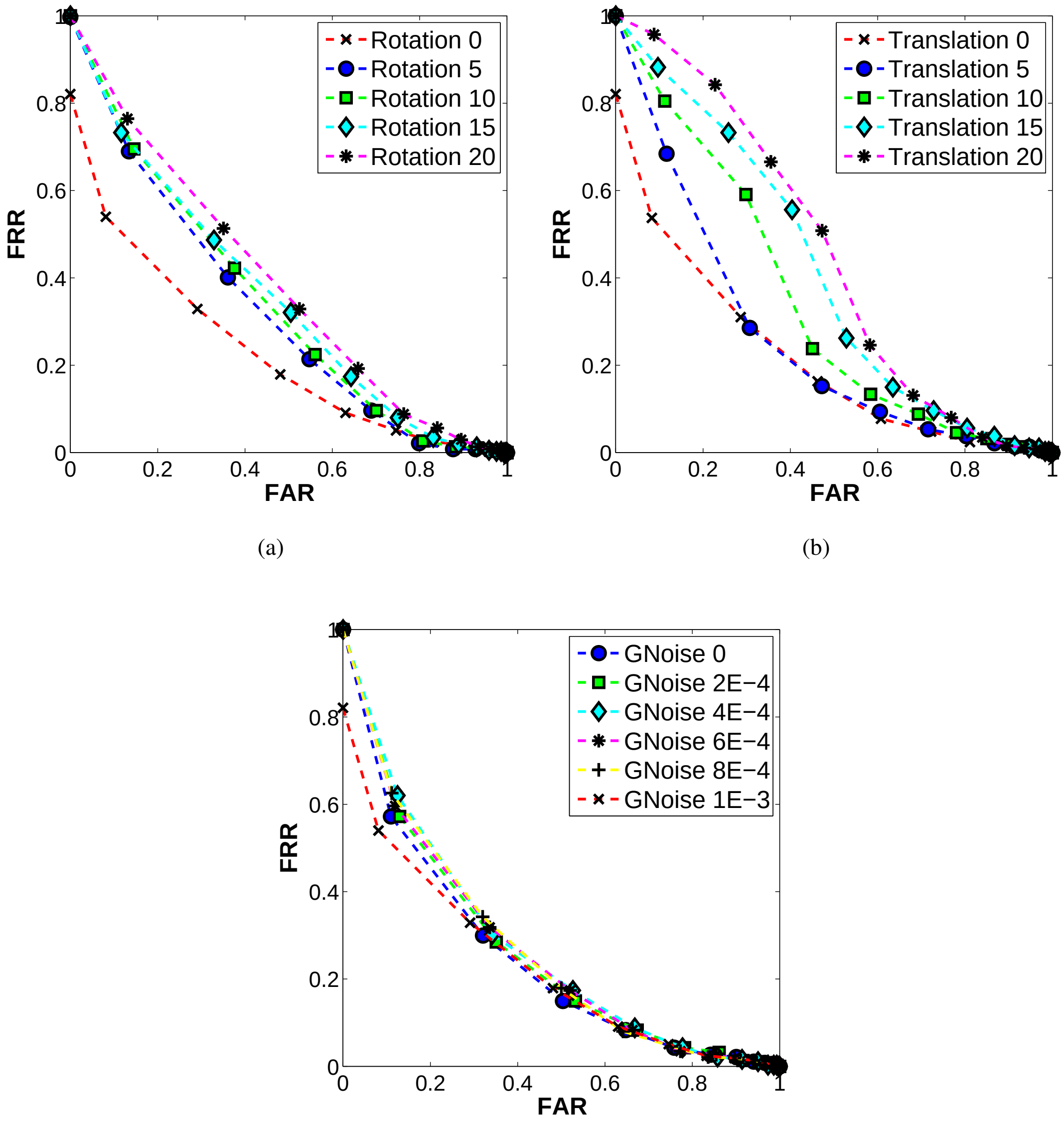


(a)

(b)

(c)

Figure A.5: FAR vs. FRR graph for FD-MC variations using χ^2 distance matching based on (a) Rotation, (b) Translation, (c) Gaussian noise.

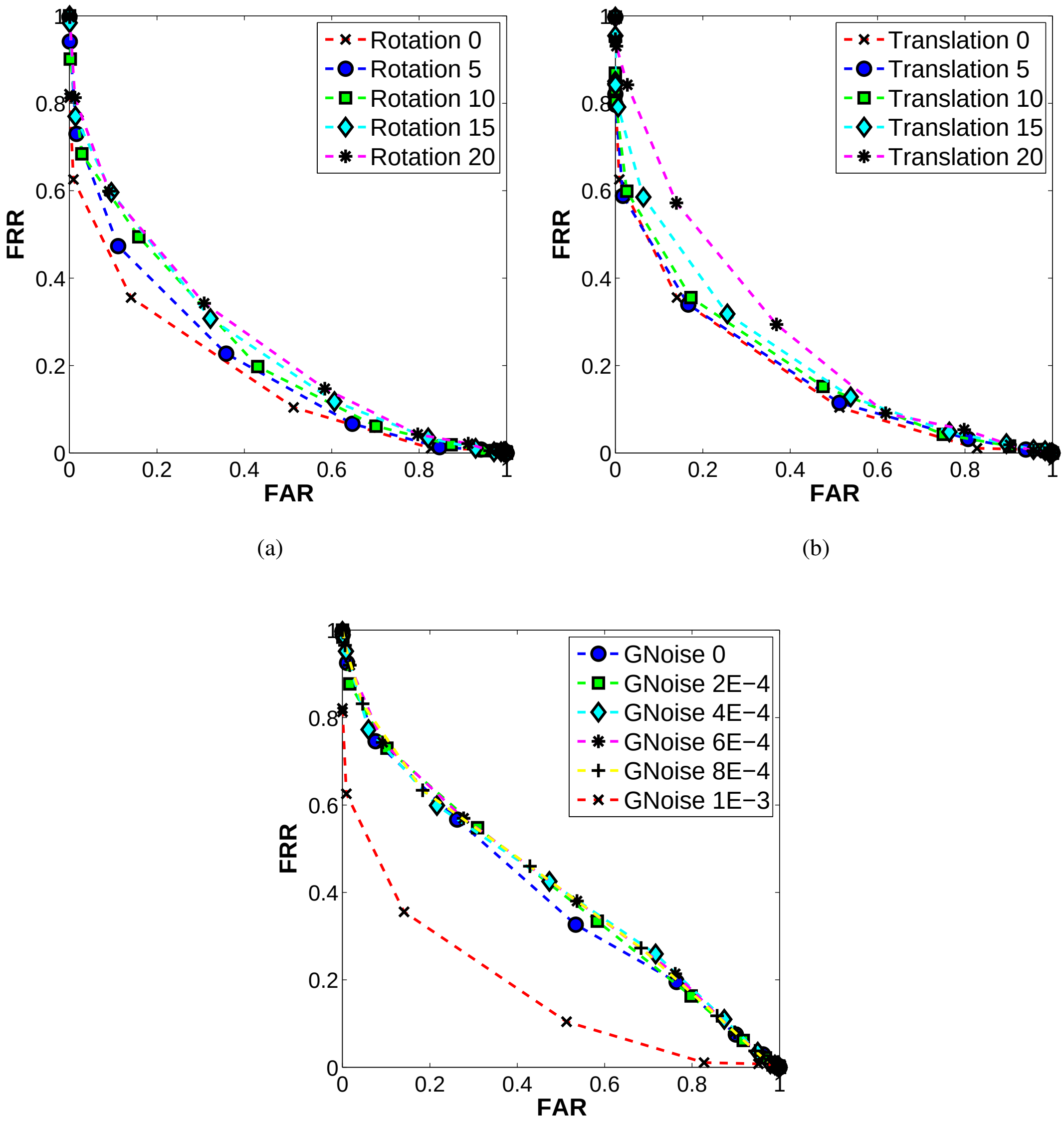


(a)

(b)

(c)

Figure A.6: FAR vs. FRR graph for FD-MC variations using Earth Mover's distance matching based on (a) Rotation, (b) Translation, (c) Gaussian noise.

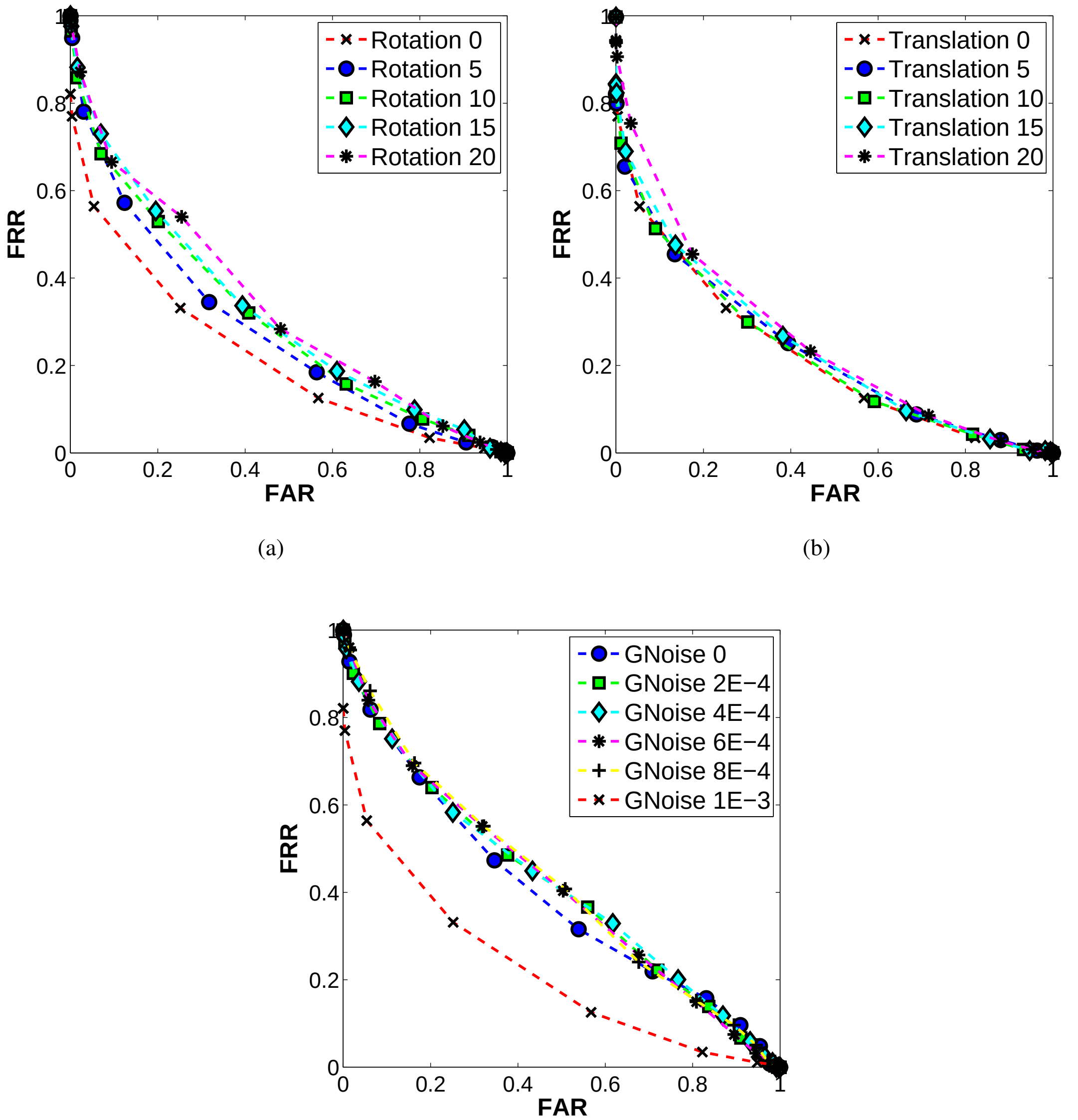


(a)

(b)

(c)

Figure A.7: FAR vs. FRR graph for FD-WL variations using Euclidean distance matching based on (a) Rotation, (b) Translation, (c) Gaussian noise.

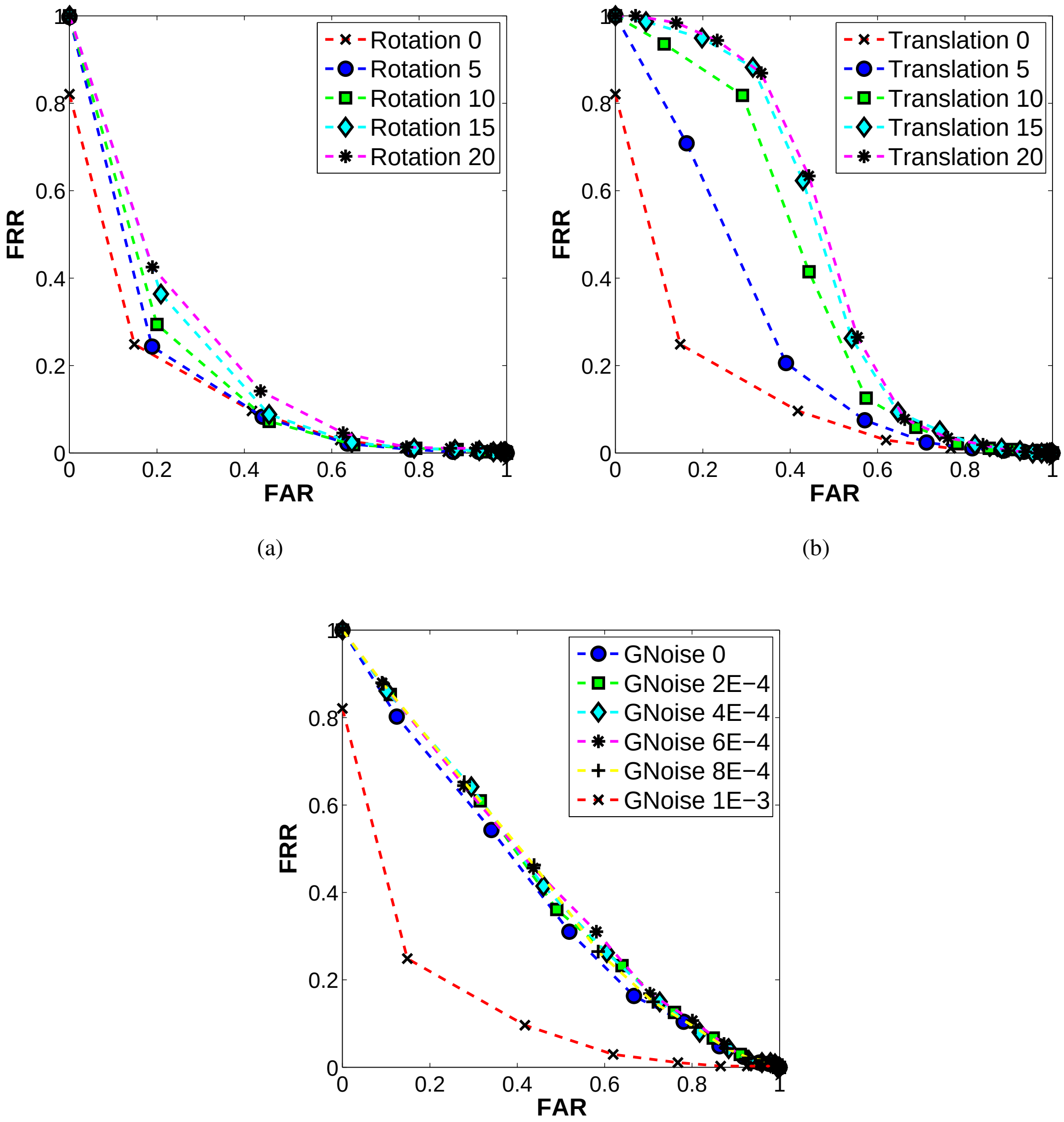


(a)

(b)

(c)

Figure A.8: FAR vs. FRR graph for FD-WL variations using χ^2 distance matching based on (a) Rotation, (b) Translation, (c) Gaussian noise.

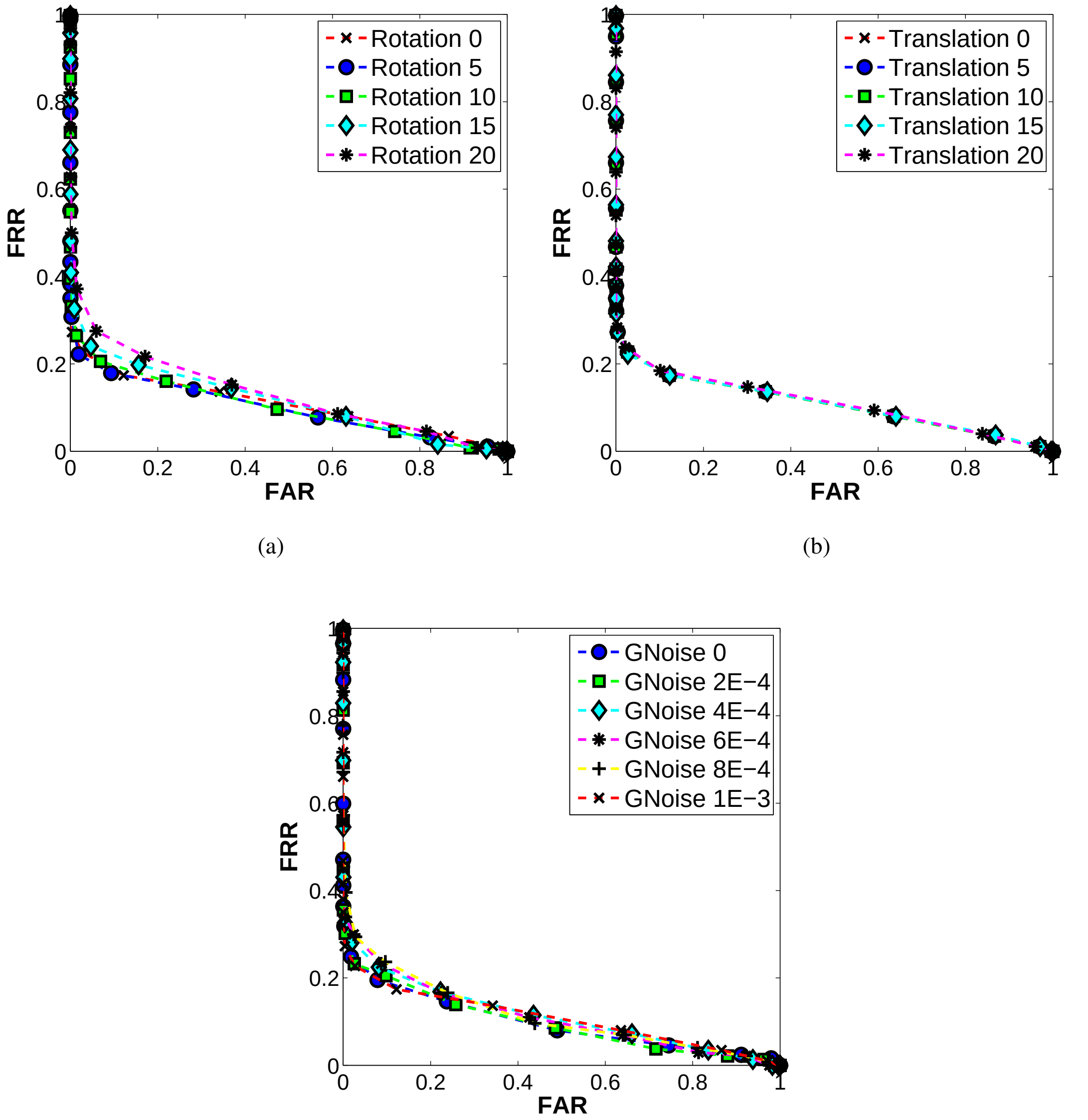


(a)

(b)

(c)

Figure A.9: FAR vs. FRR graph for FD-WL variations using Earth Mover's distance matching based on (a) Rotation, (b) Translation, (c) Gaussian noise.

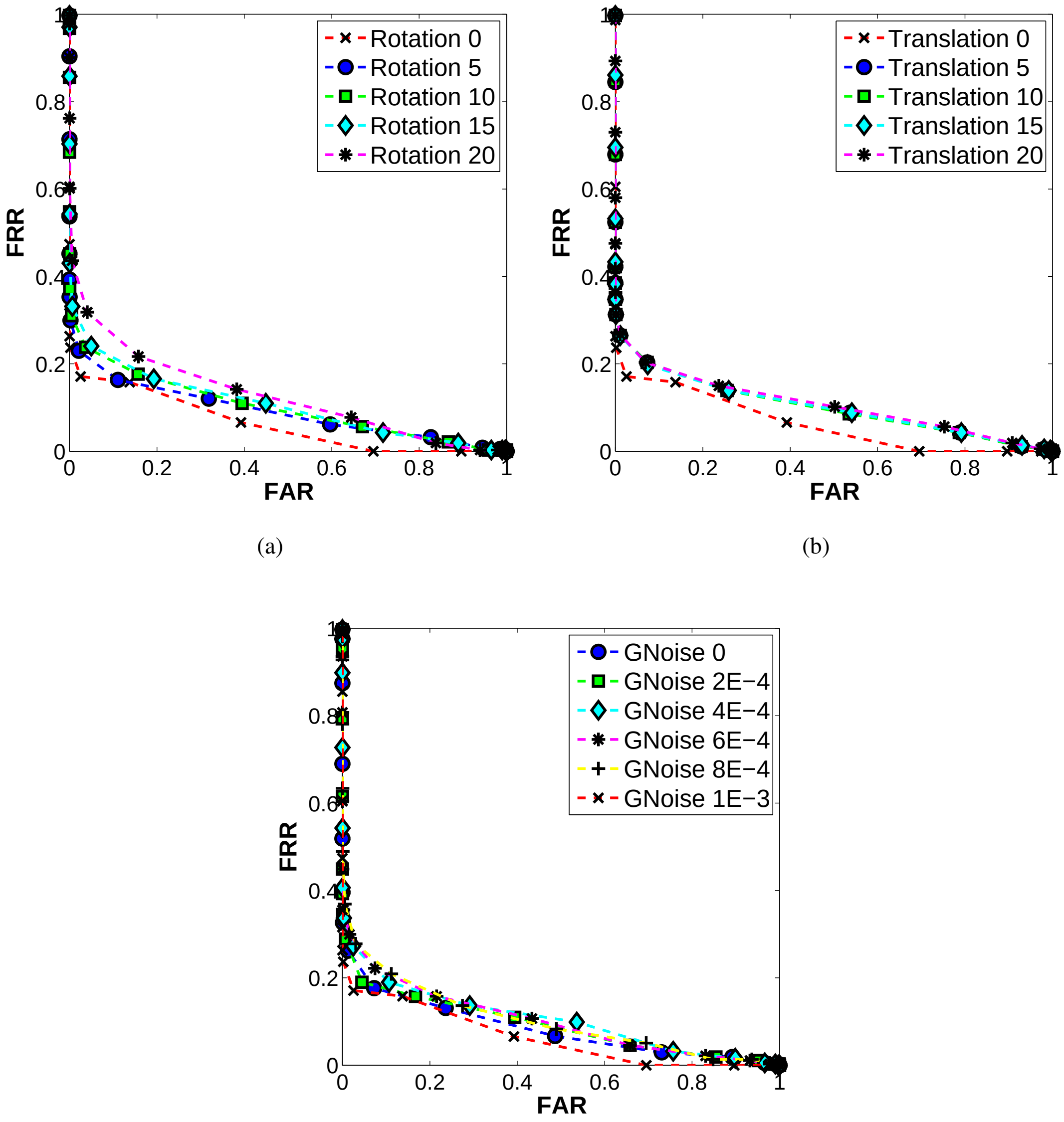


(a)

(b)

(c)

Figure A.10: FAR vs. FRR graph for ZM-LT variations using Euclidean distance matching based on (a) Rotation, (b) Translation, (c) Gaussian noise.

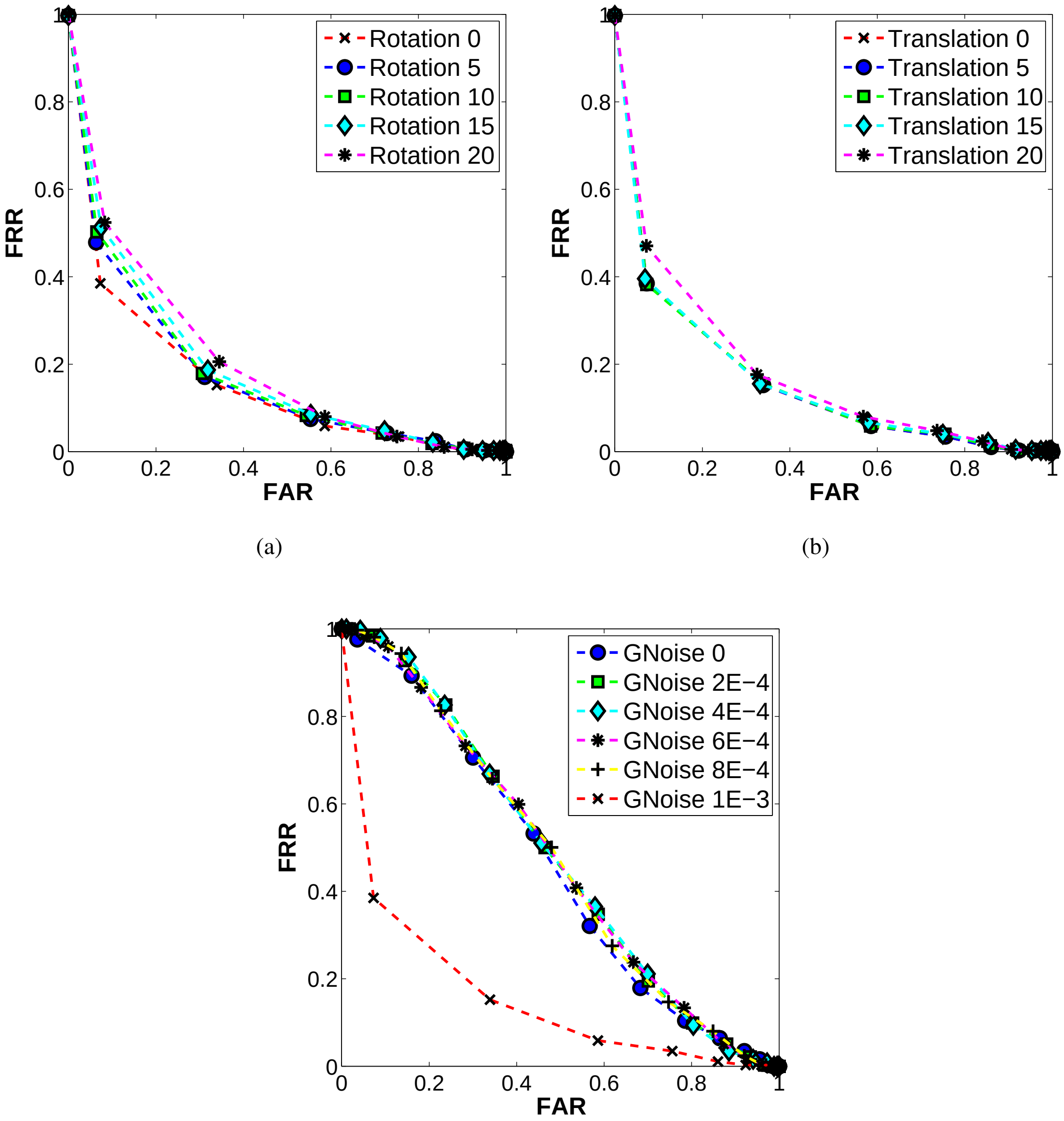


(a)

(b)

(c)

Figure A.11: FAR vs. FRR graph for ZM-LT variations using χ^2 distance matching based on (a) Rotation, (b) Translation, (c) Gaussian noise.

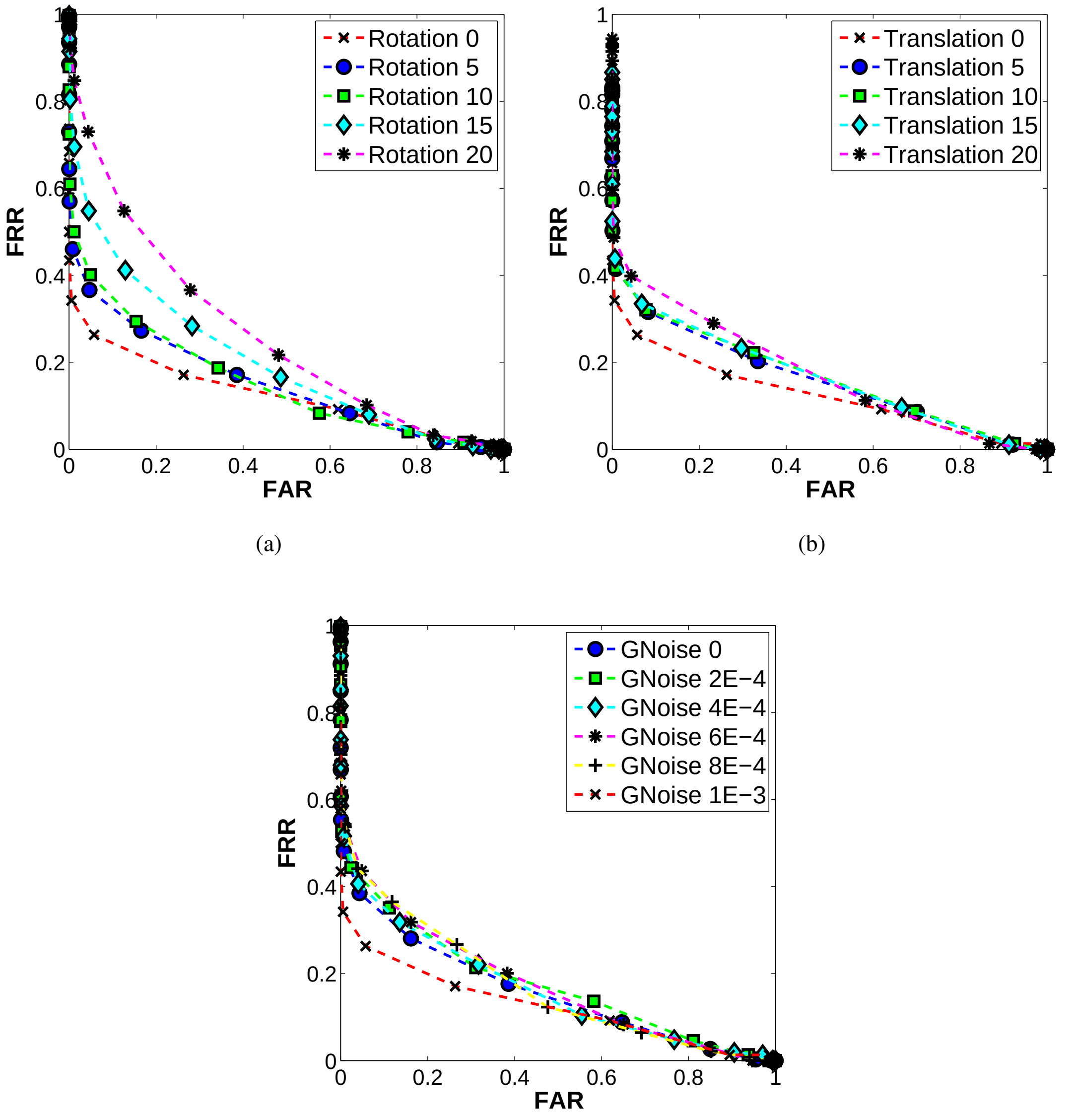


(a)

(b)

(c)

Figure A.12: FAR vs. FRR graph for ZM-LT variations using Earth Mover’s distance matching based on (a) Rotation, (b) Translation, (c) Gaussian noise.

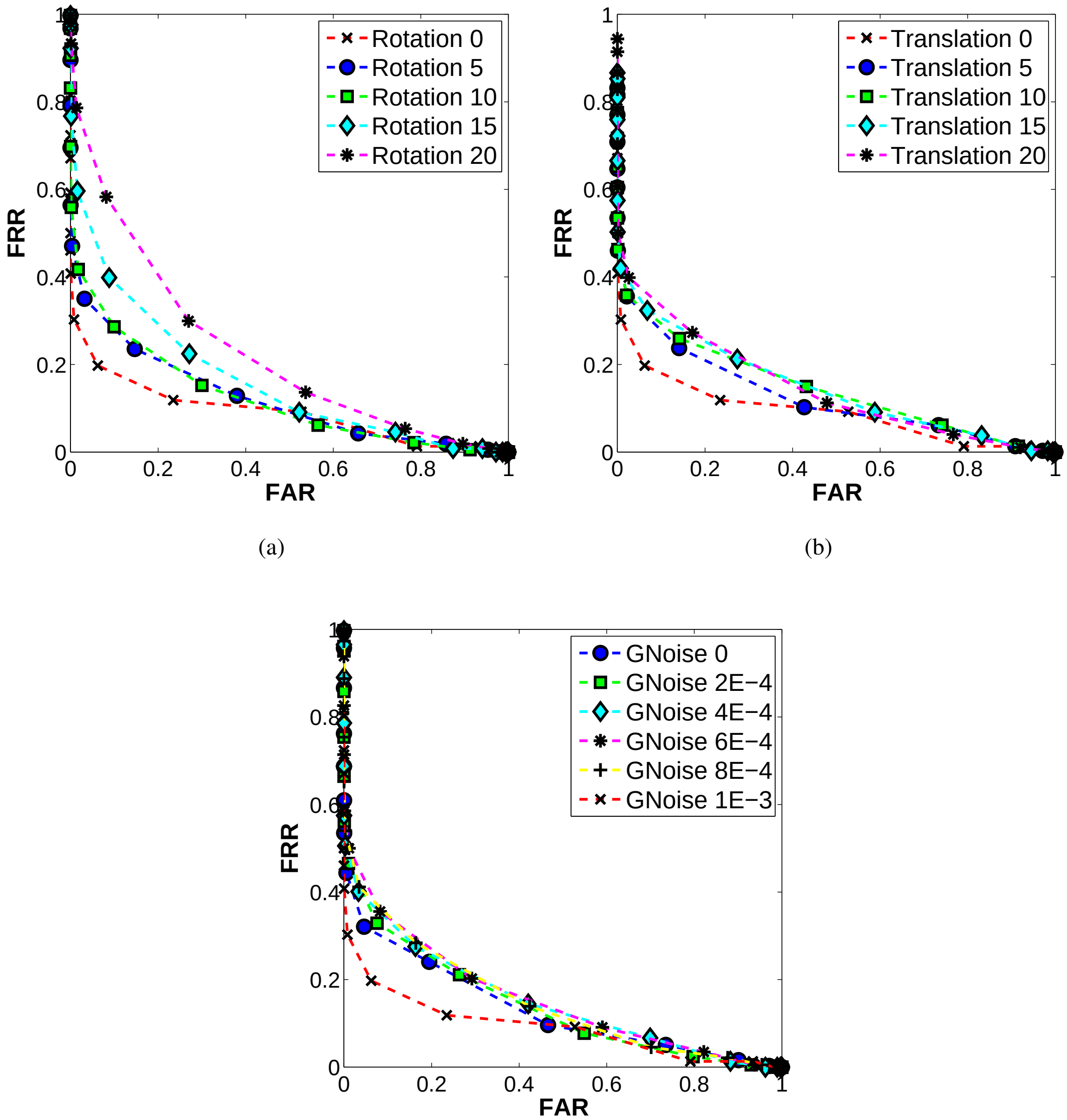


(a)

(b)

(c)

Figure A.13: FAR vs. FRR graph for ZM-MC variations using Euclidean distance matching based on (a) Rotation, (b) Translation, (c) Gaussian noise.

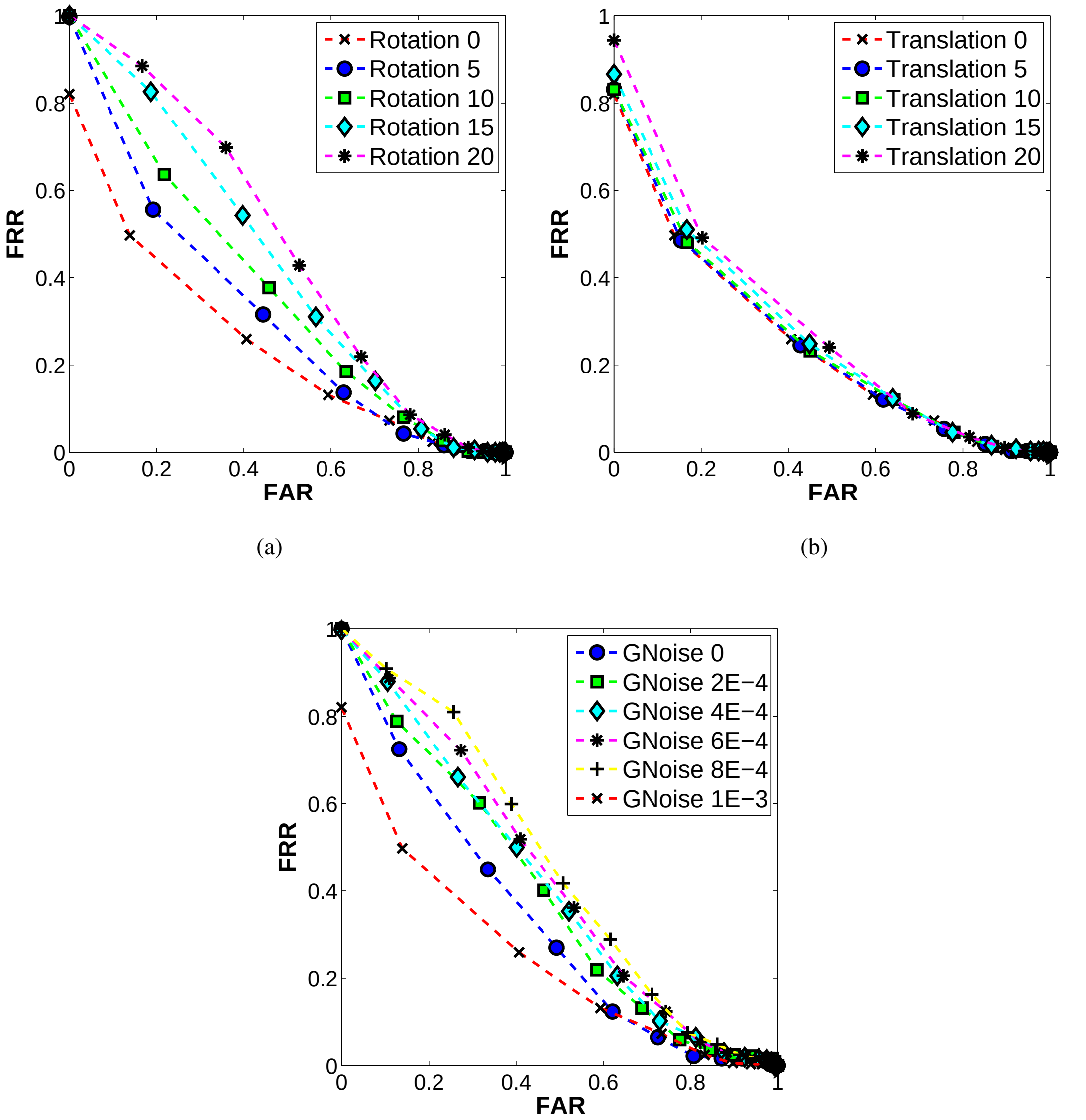


(a)

(b)

(c)

Figure A.14: FAR vs. FRR graph for ZM-MC variations using χ^2 distance matching based on (a) Rotation, (b) Translation, (c) Gaussian noise.

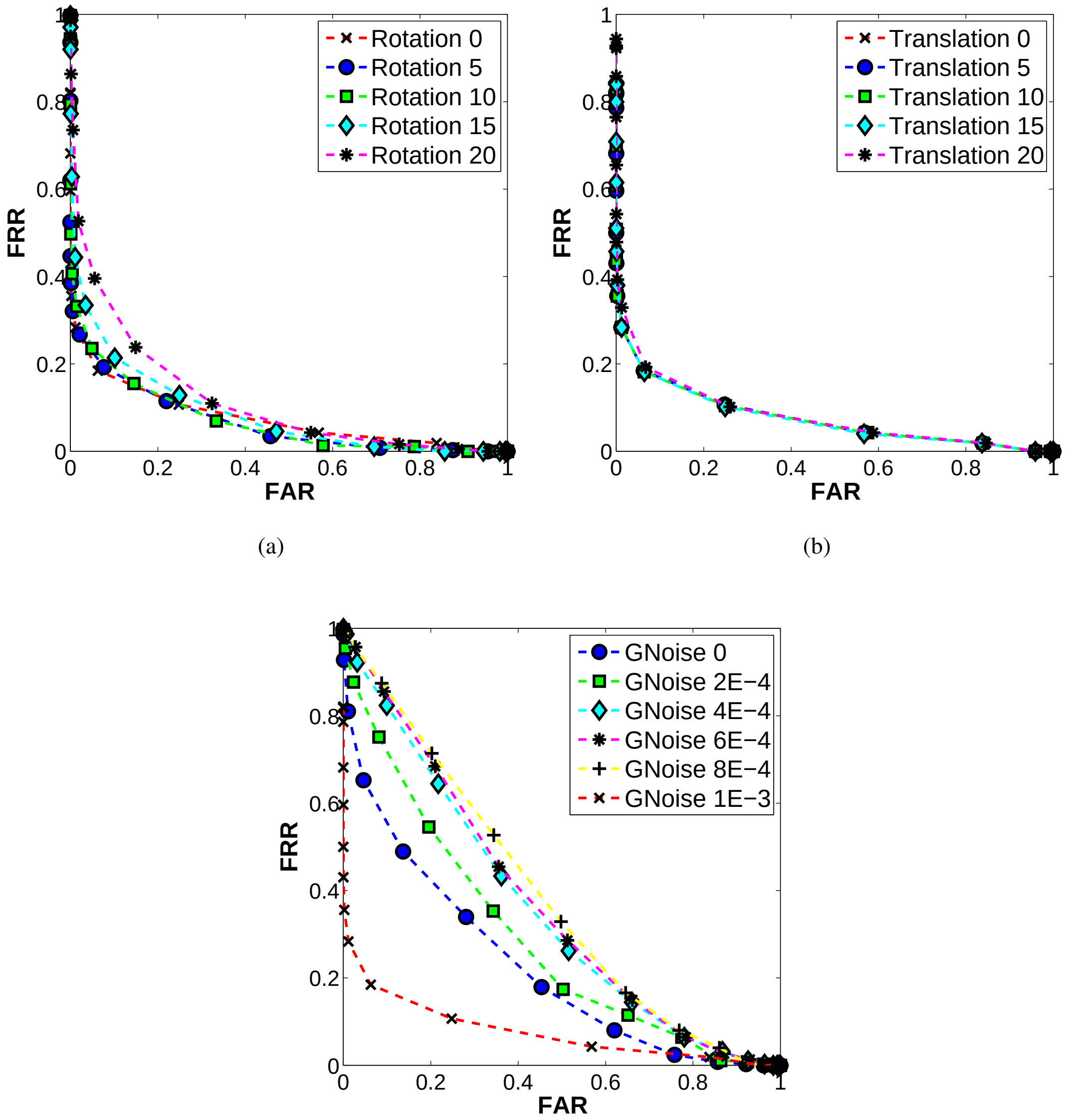


(a)

(b)

(c)

Figure A.15: FAR vs. FRR graph for ZM-MC variations using Earth Mover's distance matching based on (a) Rotation, (b) Translation, (c) Gaussian noise.

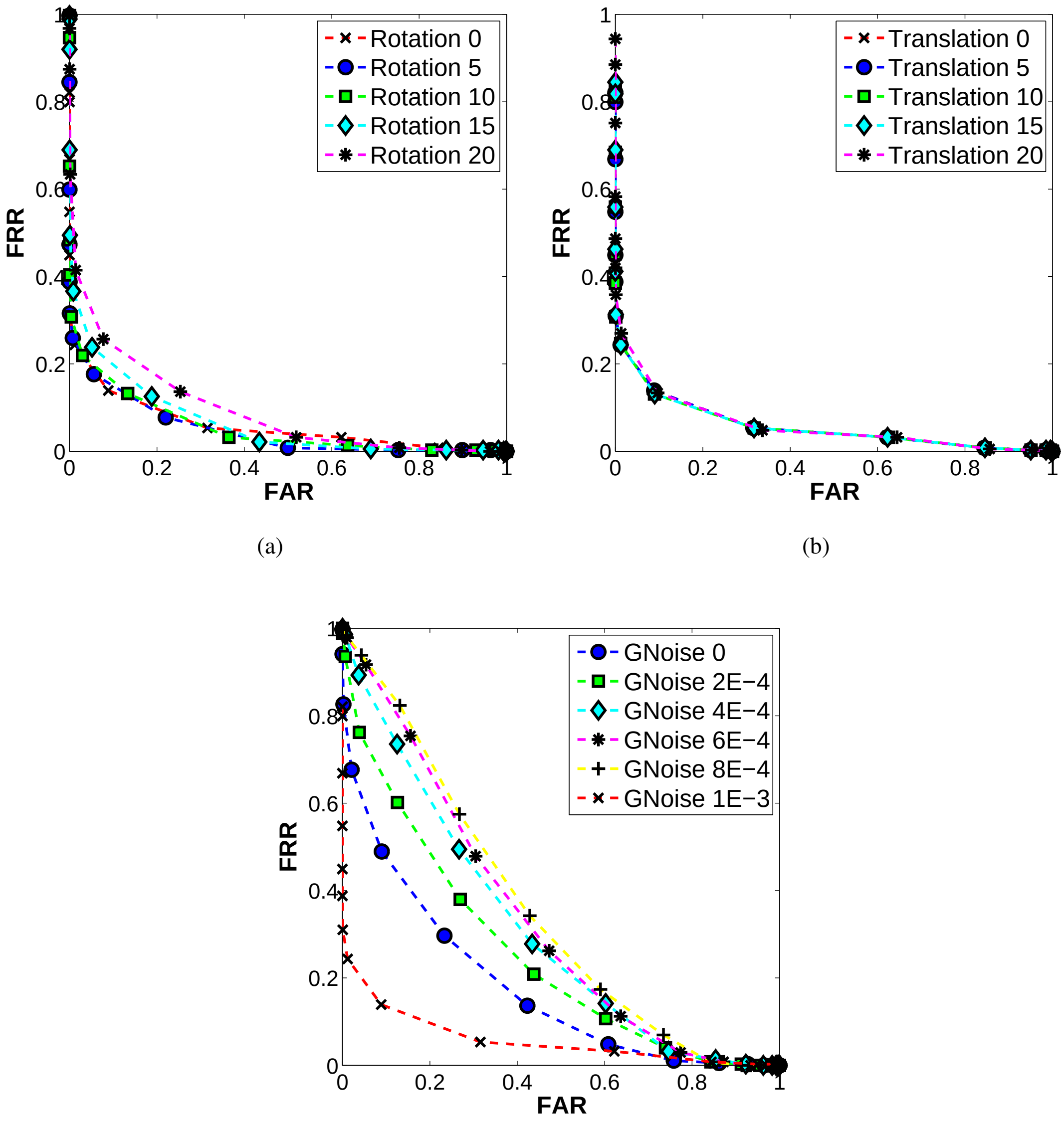


(a)

(b)

(c)

Figure A.16: FAR vs. FRR graph for ZM-WL variations using Euclidean distance matching based on (a) Rotation, (b) Translation, (c) Gaussian noise.

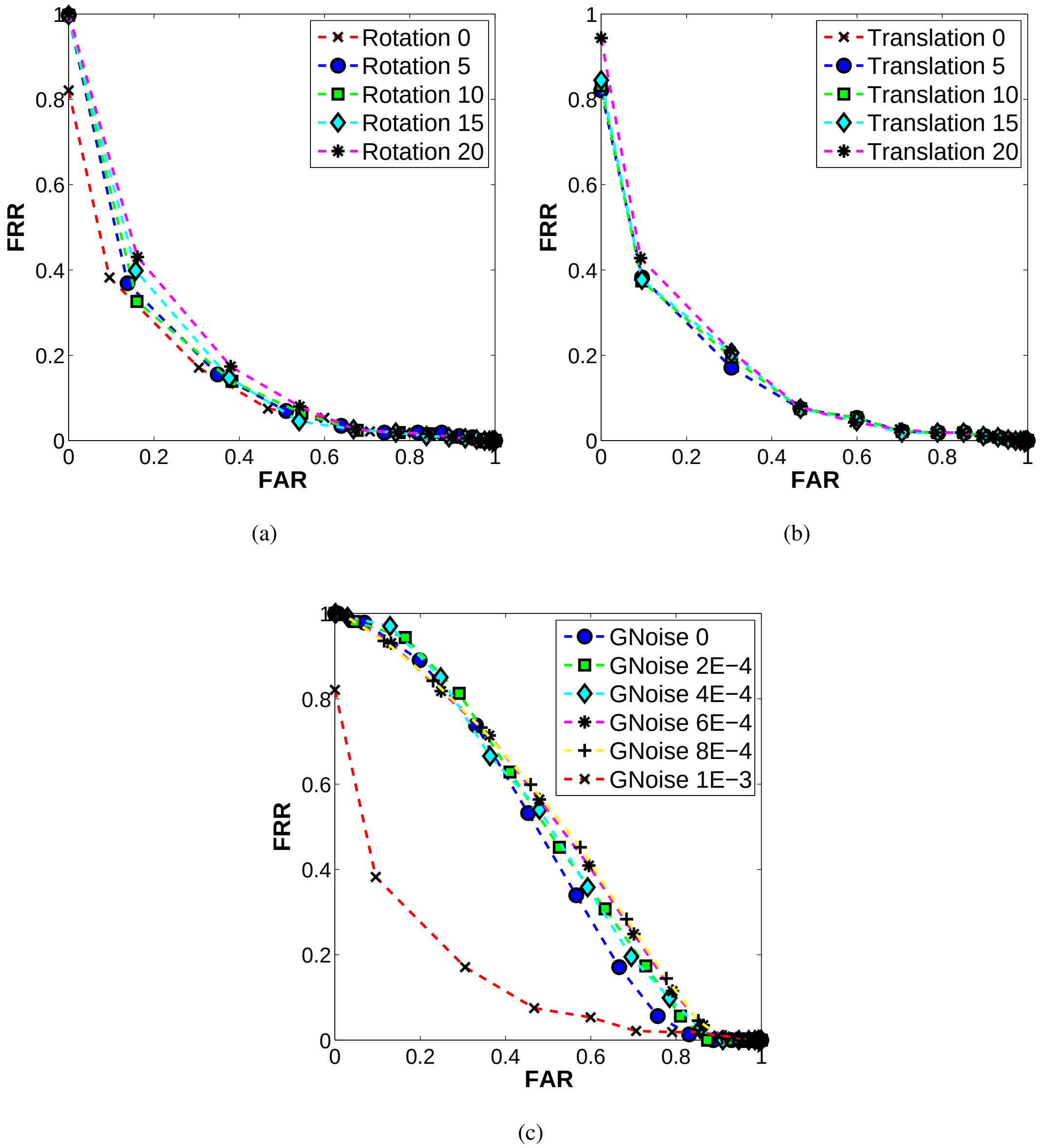


(a)

(b)

(c)

Figure A.17: FAR vs. FRR graph for ZM-WL variations using χ^2 distance matching based on (a) Rotation, (b) Translation, (c) Gaussian noise.

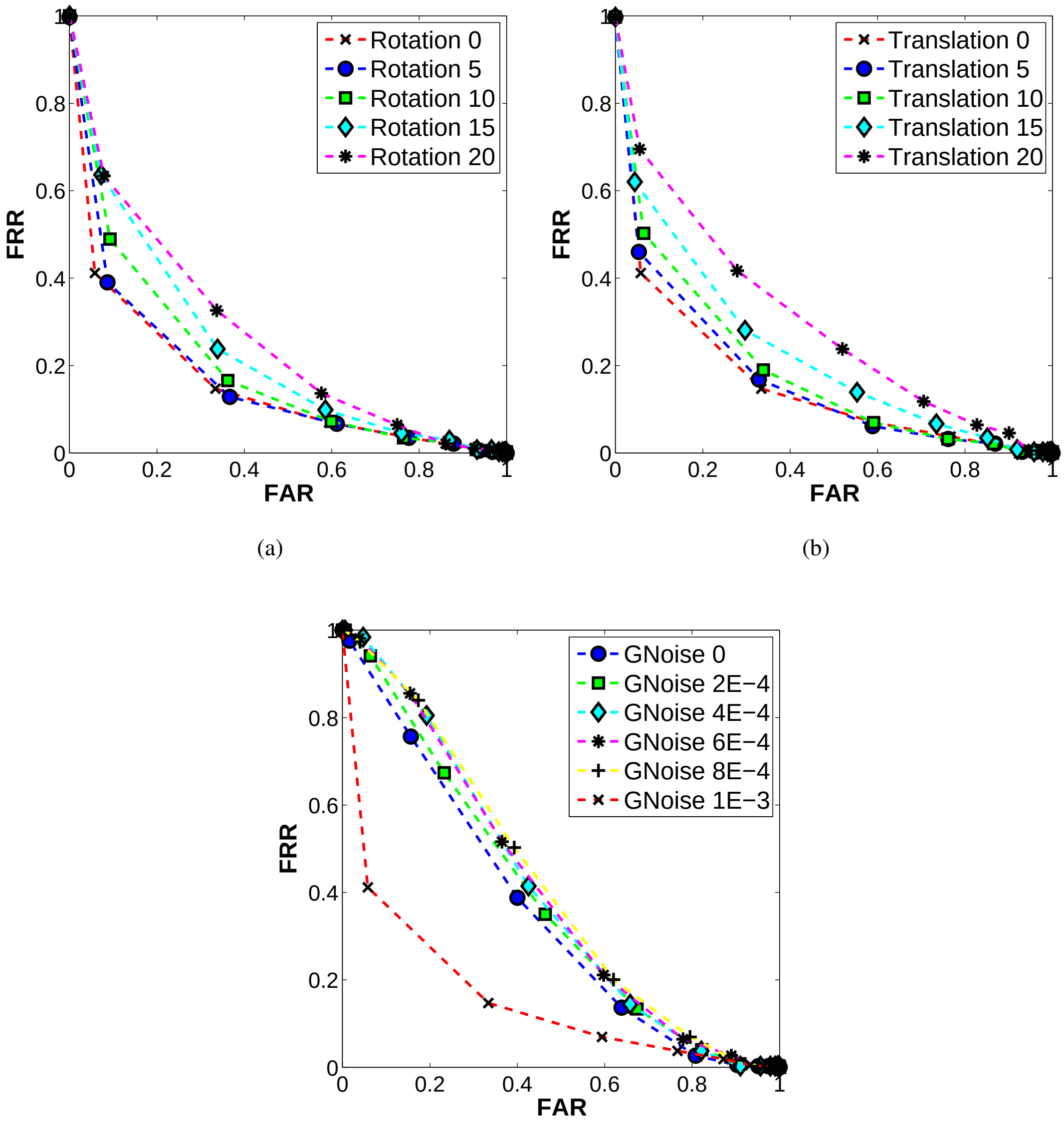


(a)

(b)

(c)

Figure A.18: FAR vs. FRR graph for ZM-WL variations using Earth Mover's distance matching based on (a) Rotation, (b) Translation, (c) Gaussian noise.

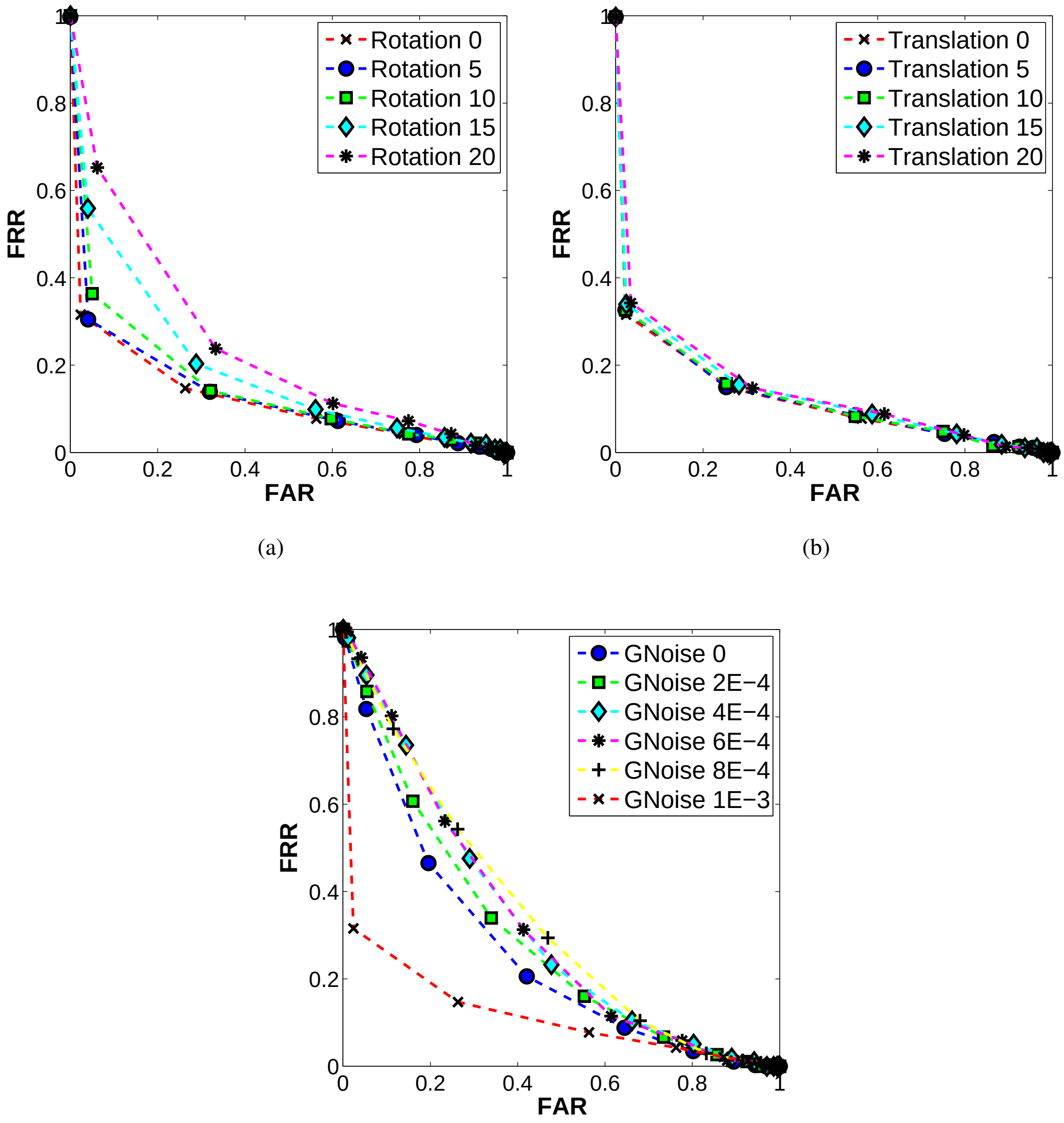


(a)

(b)

(c)

Figure A.19: FAR vs. FRR graph for LBP-LT variations using Euclidean distance matching based on (a) Rotation, (b) Translation, (c) Gaussian noise.

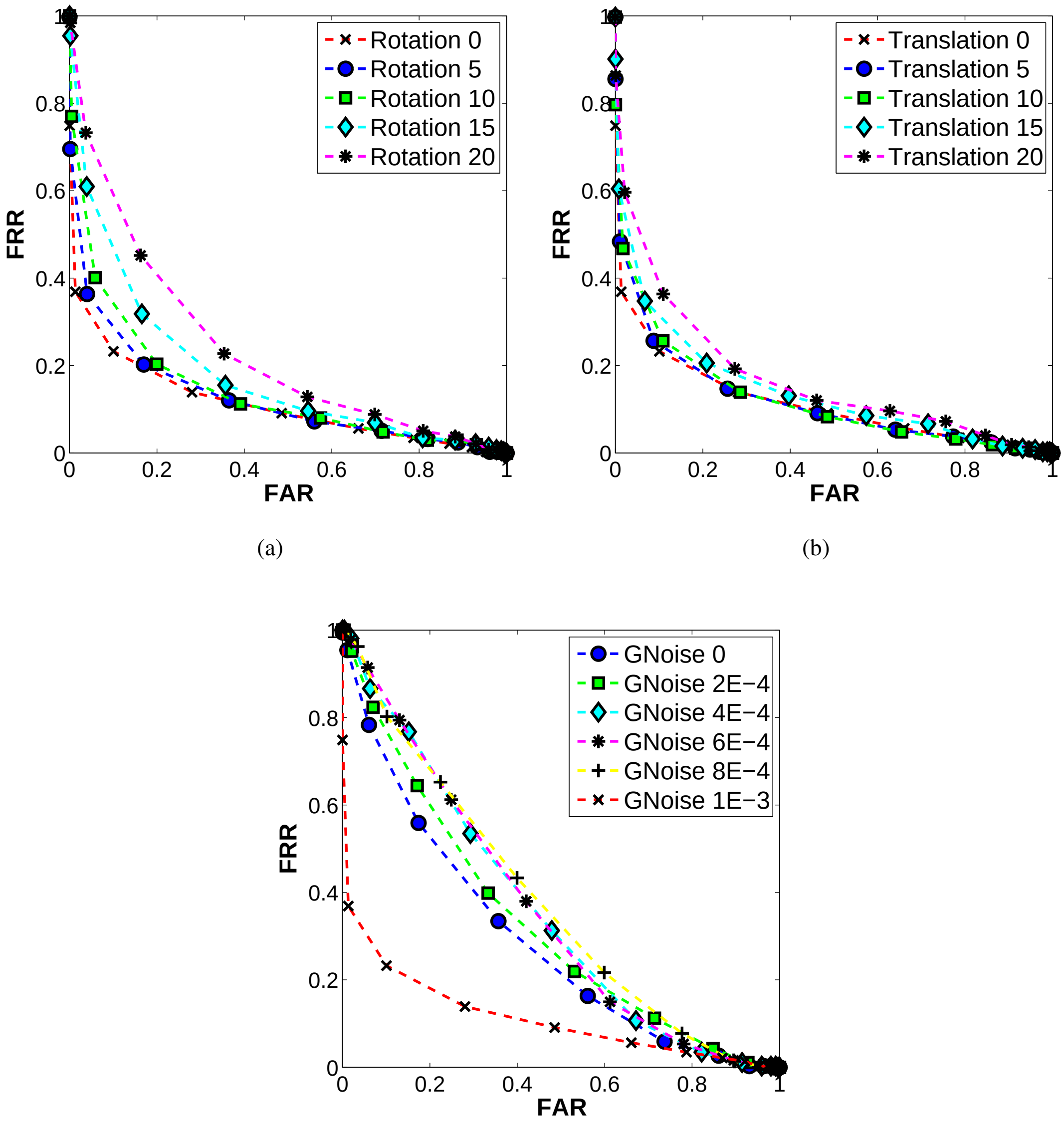


(a)

(b)

(c)

Figure A.20: FAR vs. FRR graph for LBP-LT variations using χ^2 distance matching based on (a) Rotation, (b) Translation, (c) Gaussian noise.

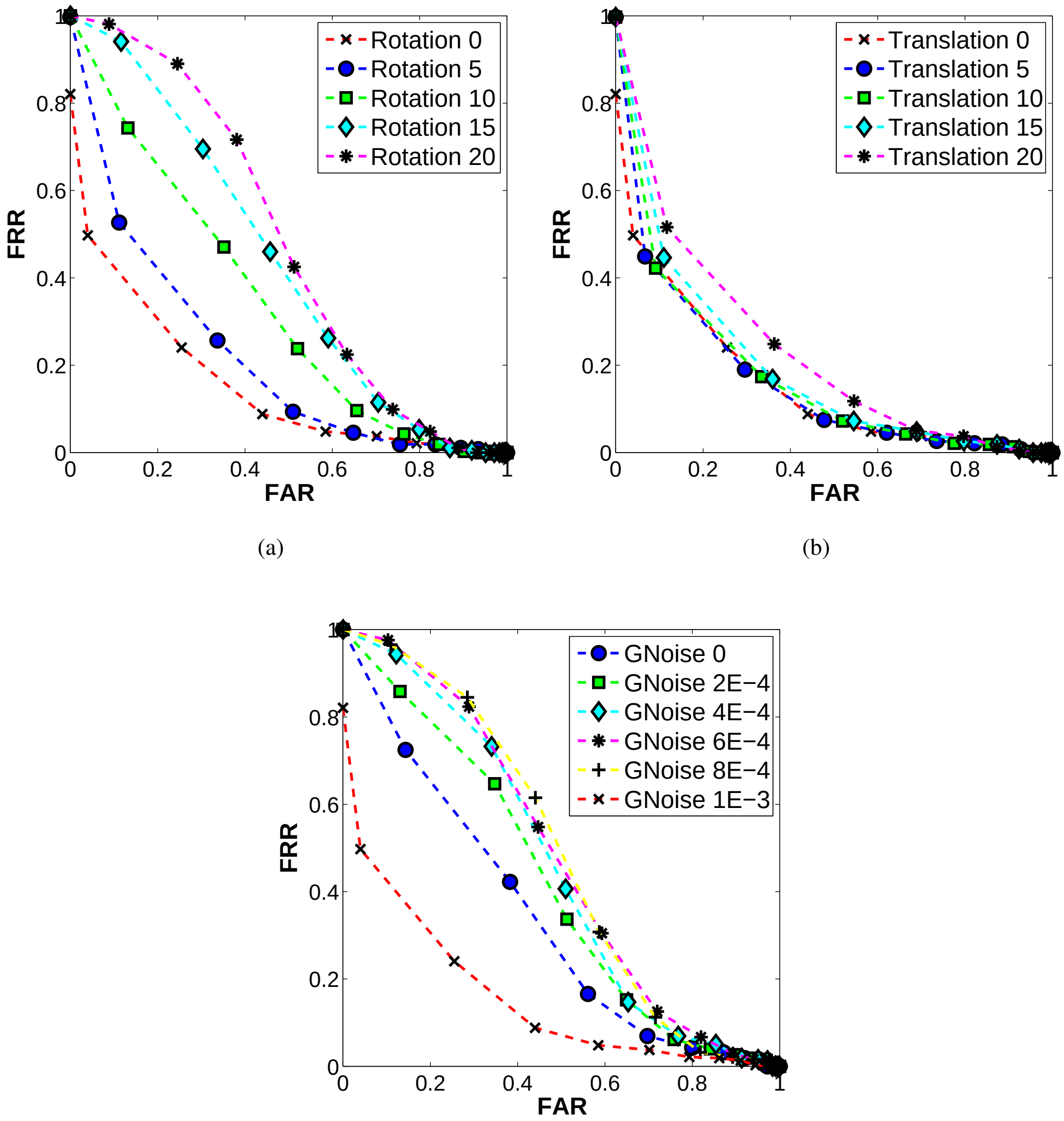


(a)

(b)

(c)

Figure A.21: FAR vs. FRR graph for LBP-LT variations using Earth Mover's distance matching based on (a) Rotation, (b) Translation, (c) Gaussian noise.

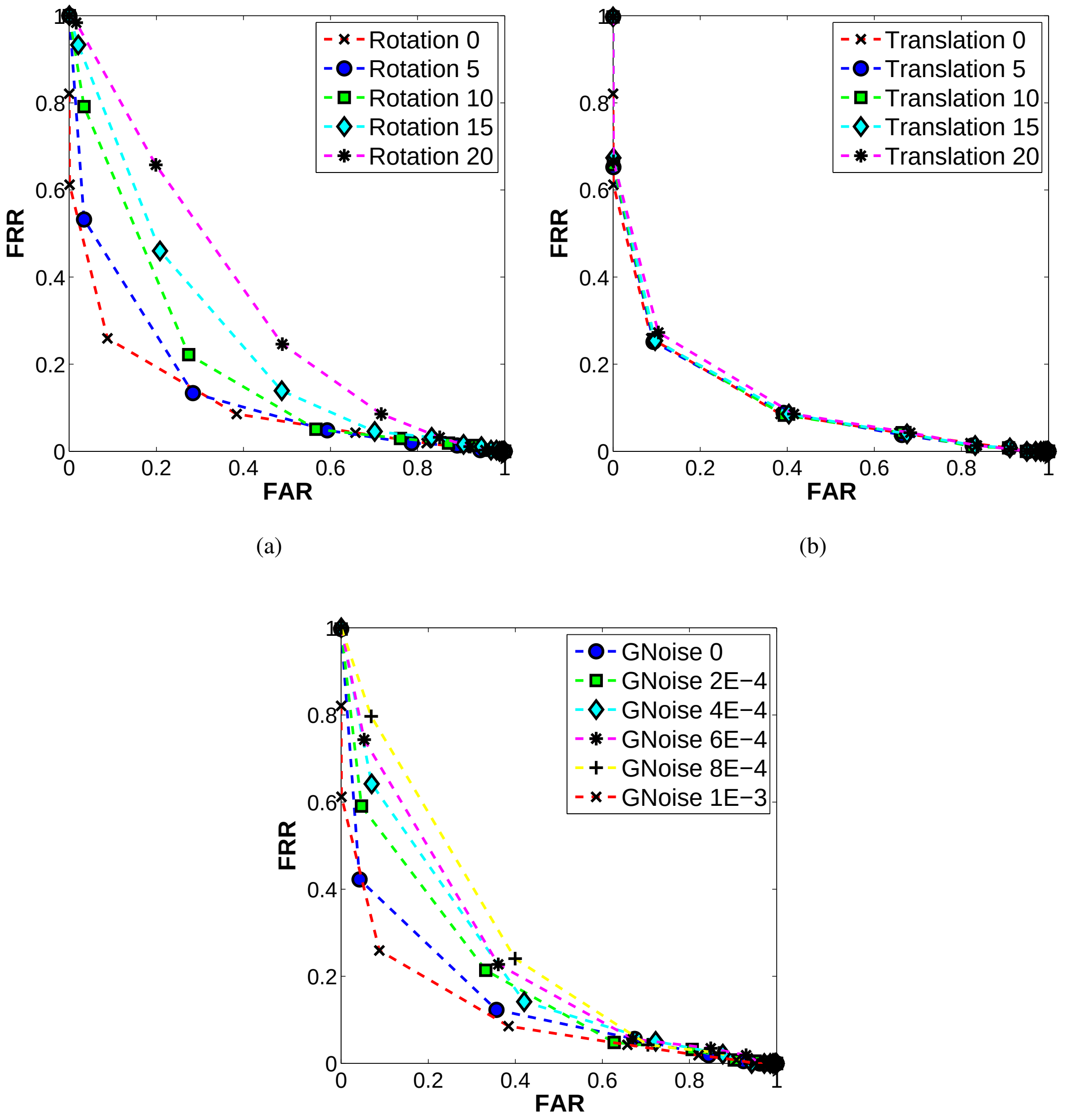


(a)

(b)

(c)

Figure A.22: FAR vs. FRR graph for LBP-MC variations using Euclidean distance matching based on (a) Rotation, (b) Translation, (c) Gaussian noise.

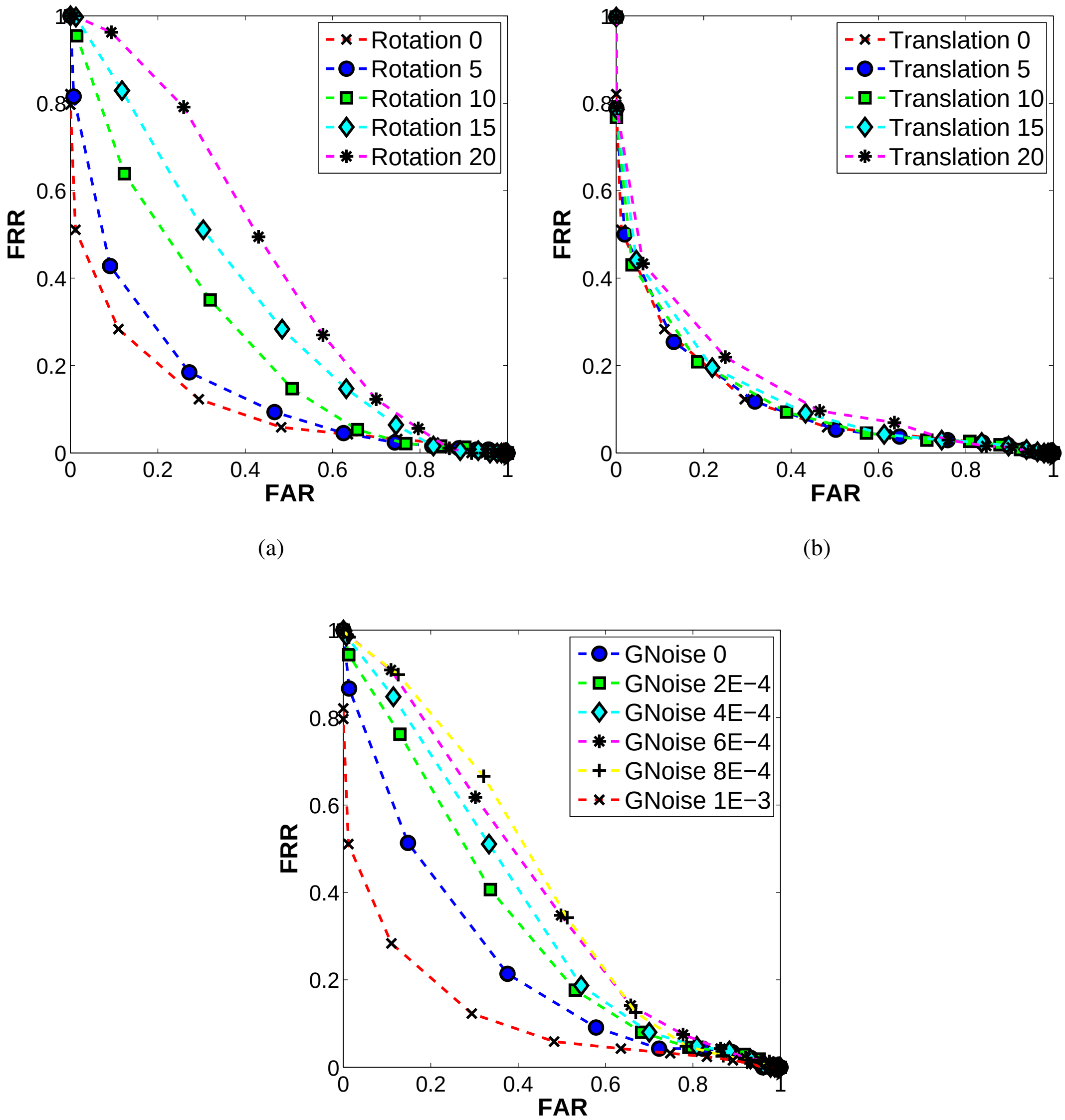


(a)

(b)

(c)

Figure A.23: FAR vs. FRR graph for LBP-MC variations using χ^2 distance matching based on (a) Rotation, (b) Translation, (c) Gaussian noise.

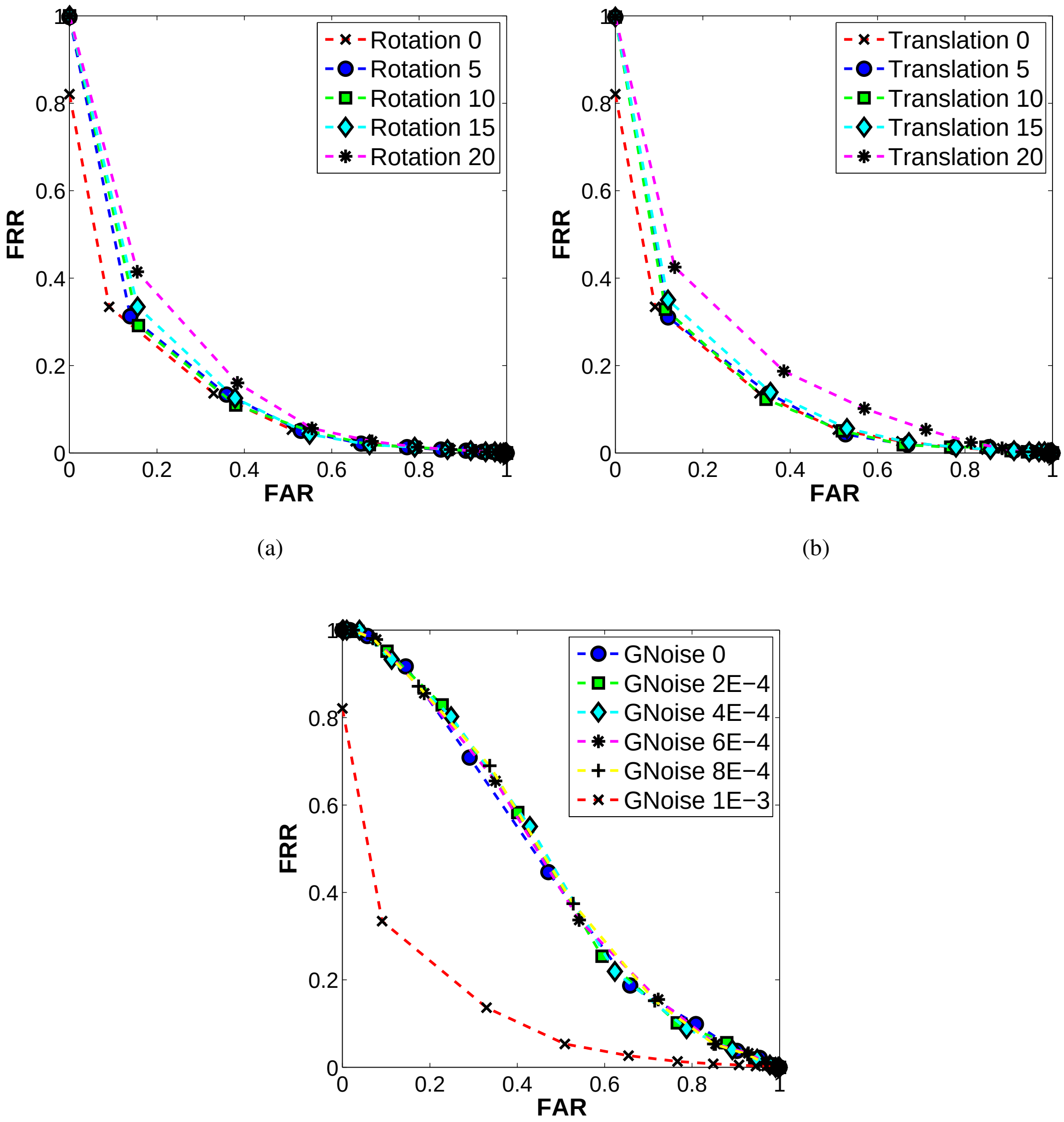


(a)

(b)

(c)

Figure A.24: FAR vs. FRR graph for LBP-MC variations using Earth Mover's distance matching based on (a) Rotation, (b) Translation, (c) Gaussian noise.

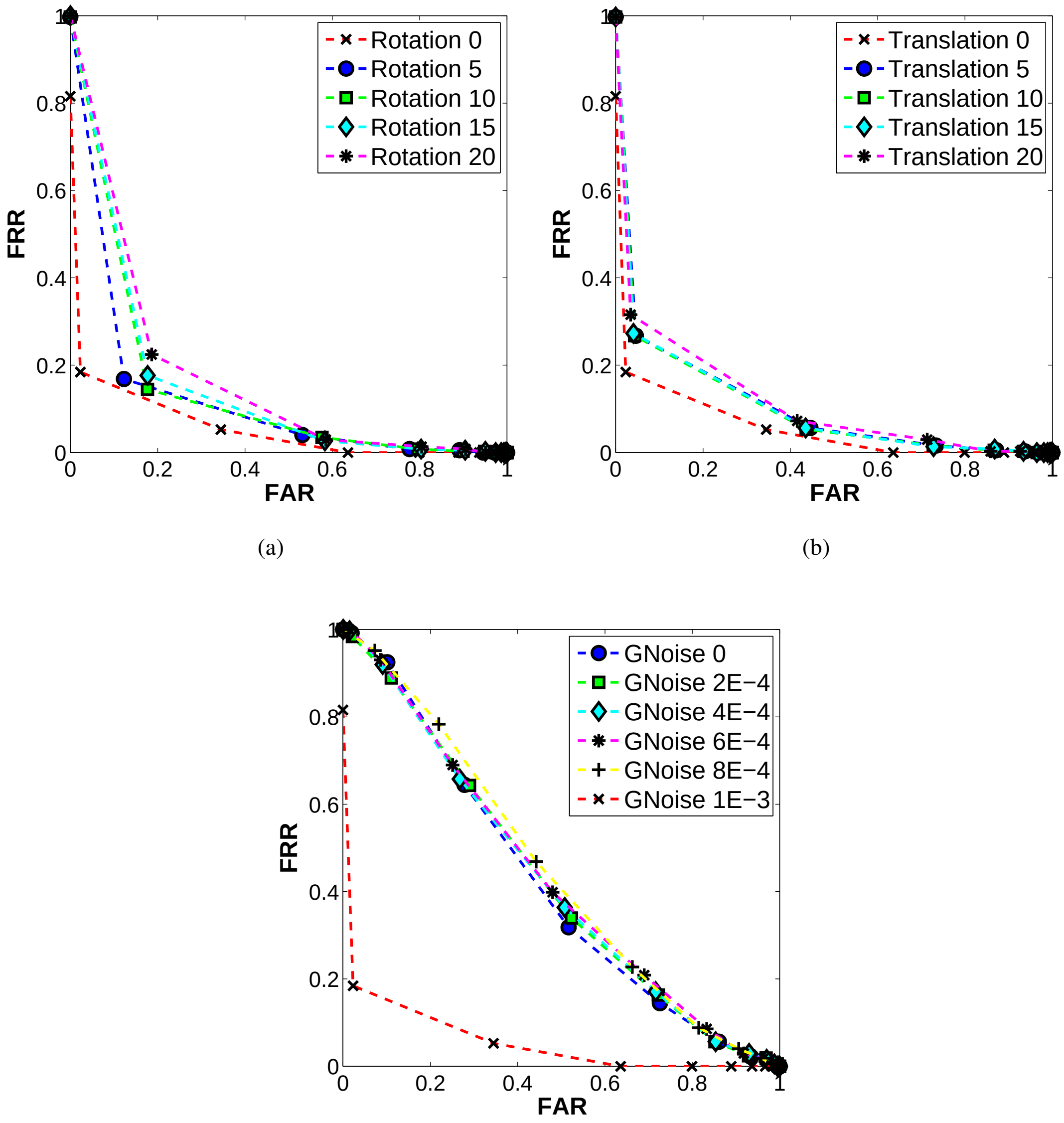


(a)

(b)

(c)

Figure A.25: FAR vs. FRR graph for LBP-WL variations using Euclidean distance matching based on (a) Rotation, (b) Translation, (c) Gaussian noise.

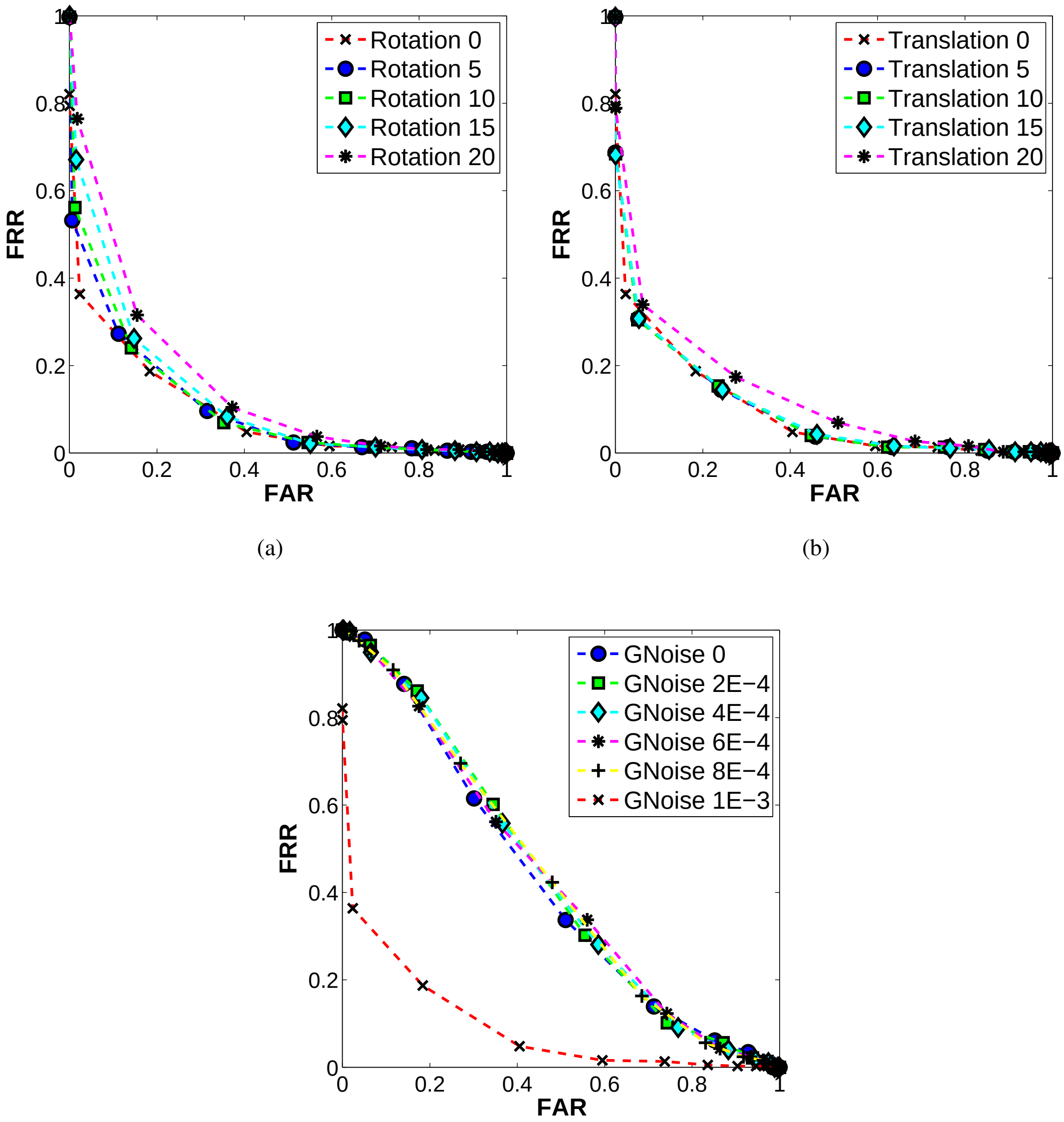


(a)

(b)

(c)

Figure A.26: FAR vs. FRR graph for LBP-WL variations using χ^2 distance matching based on (a) Rotation, (b) Translation, (c) Gaussian noise.

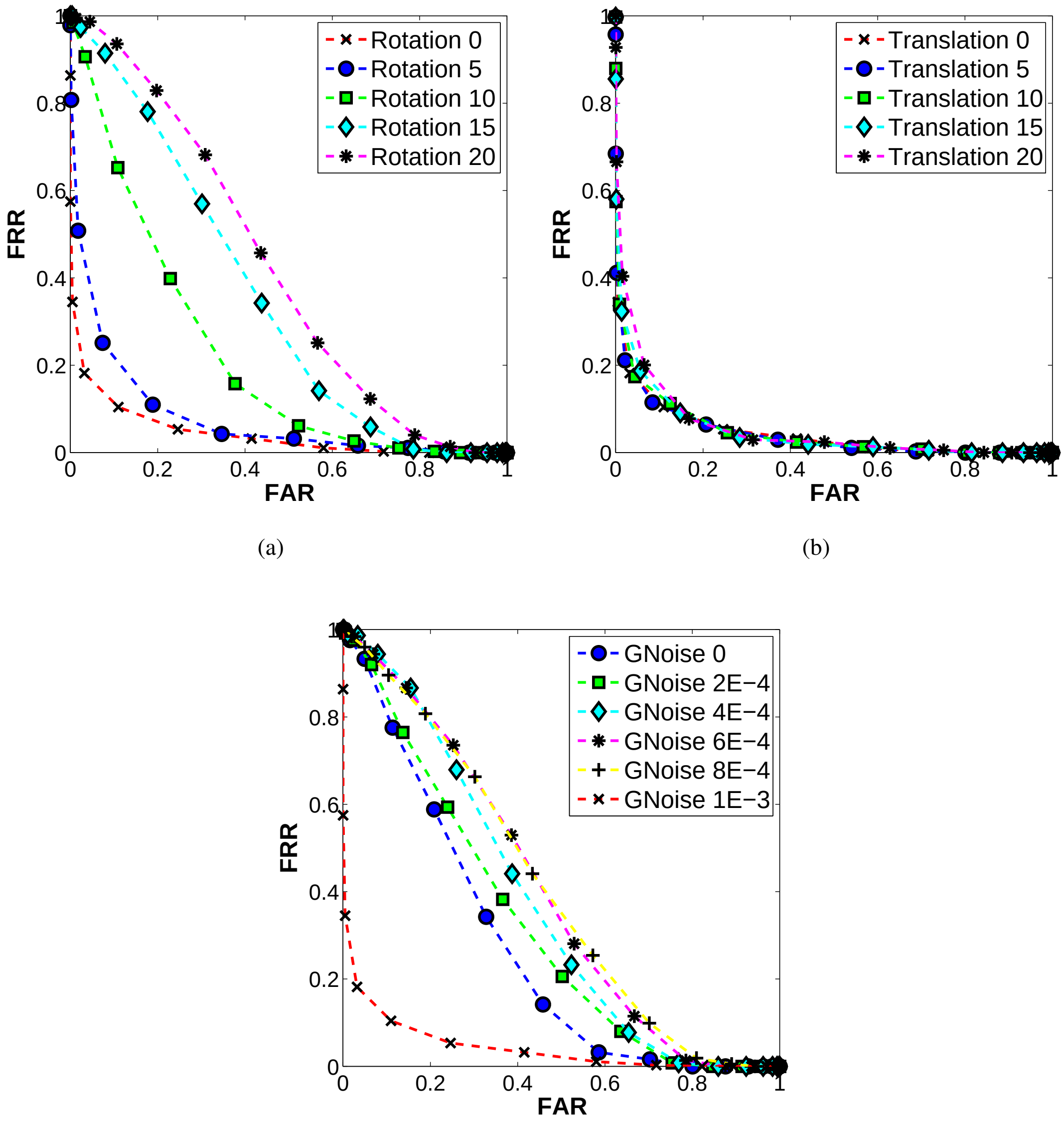


(a)

(b)

(c)

Figure A.27: FAR vs. FRR graph for LBP-WL variations using Earth Mover's distance matching based on (a) Rotation, (b) Translation, (c) Gaussian noise.

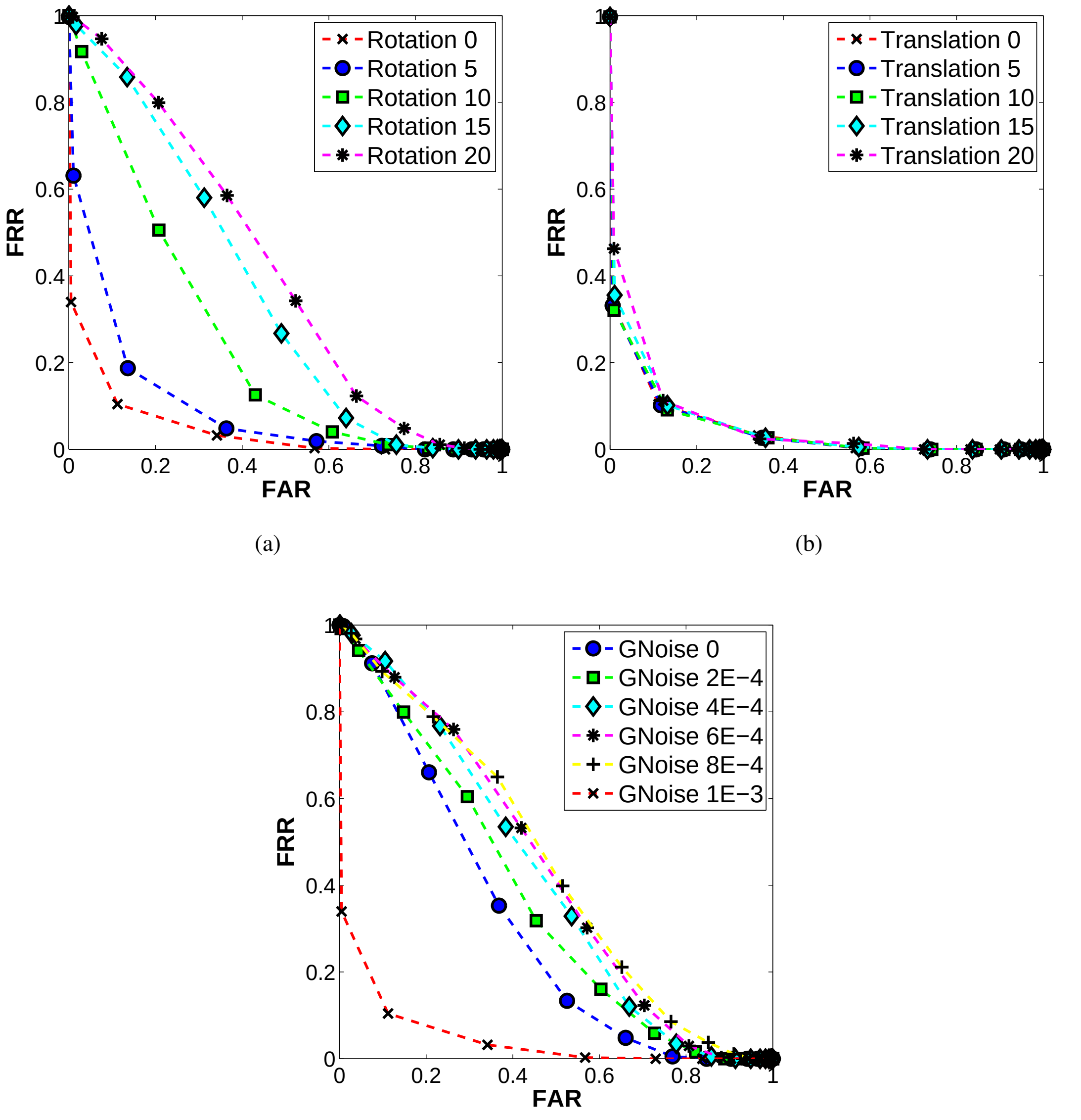


(a)

(b)

(c)

Figure A.28: FAR vs. FRR graph for HOG-LT variations using Euclidean distance matching based on (a) Rotation, (b) Translation, (c) Gaussian noise.

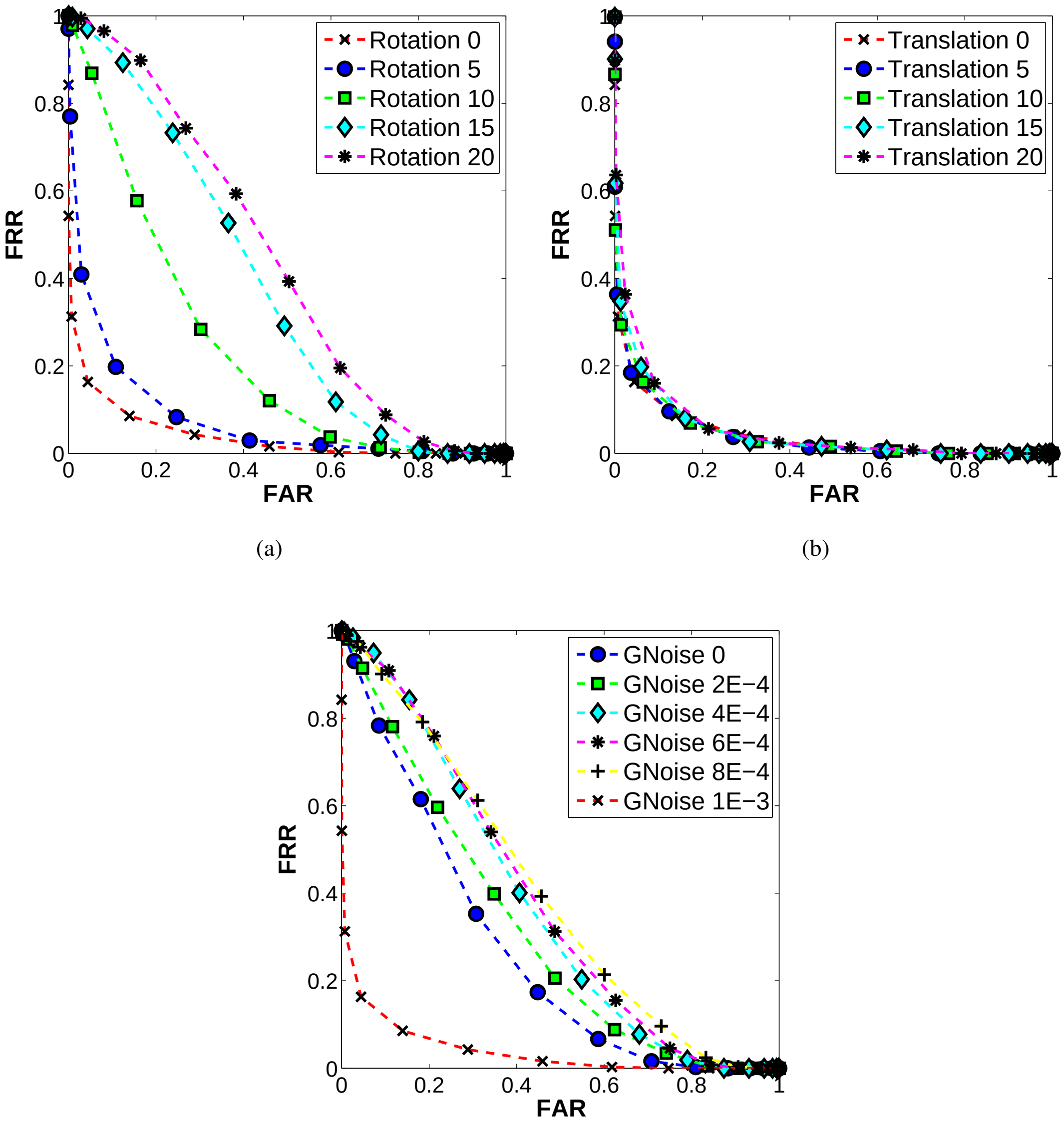


(a)

(b)

(c)

Figure A.29: FAR vs. FRR graph for HOG-LT variations using χ^2 distance matching based on (a) Rotation, (b) Translation, (c) Gaussian noise.

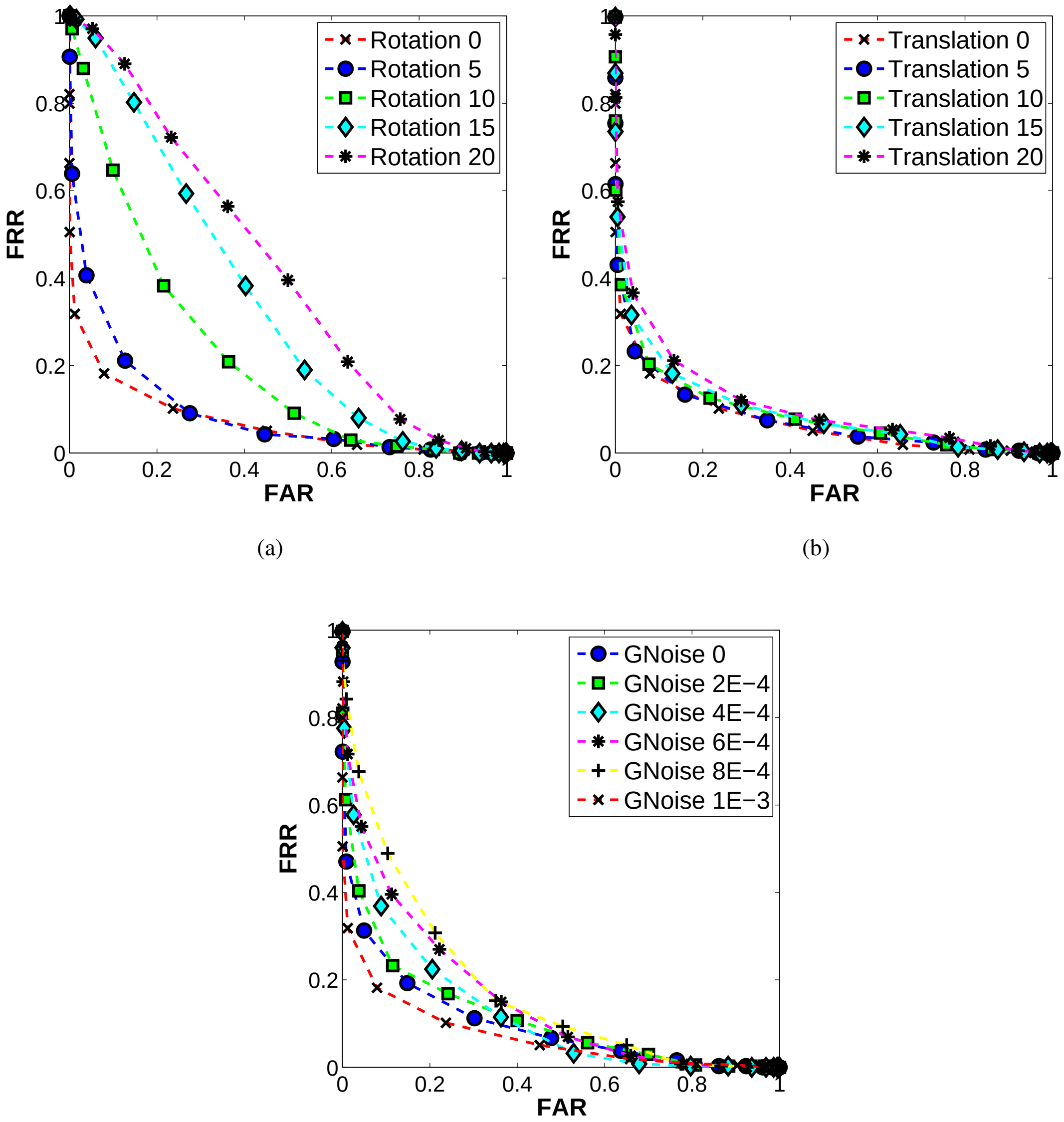


(a)

(b)

(c)

Figure A.30: FAR vs. FRR graph for HOG-LT variations using Earth Mover's distance matching based on (a) Rotation, (b) Translation, (c) Gaussian noise.

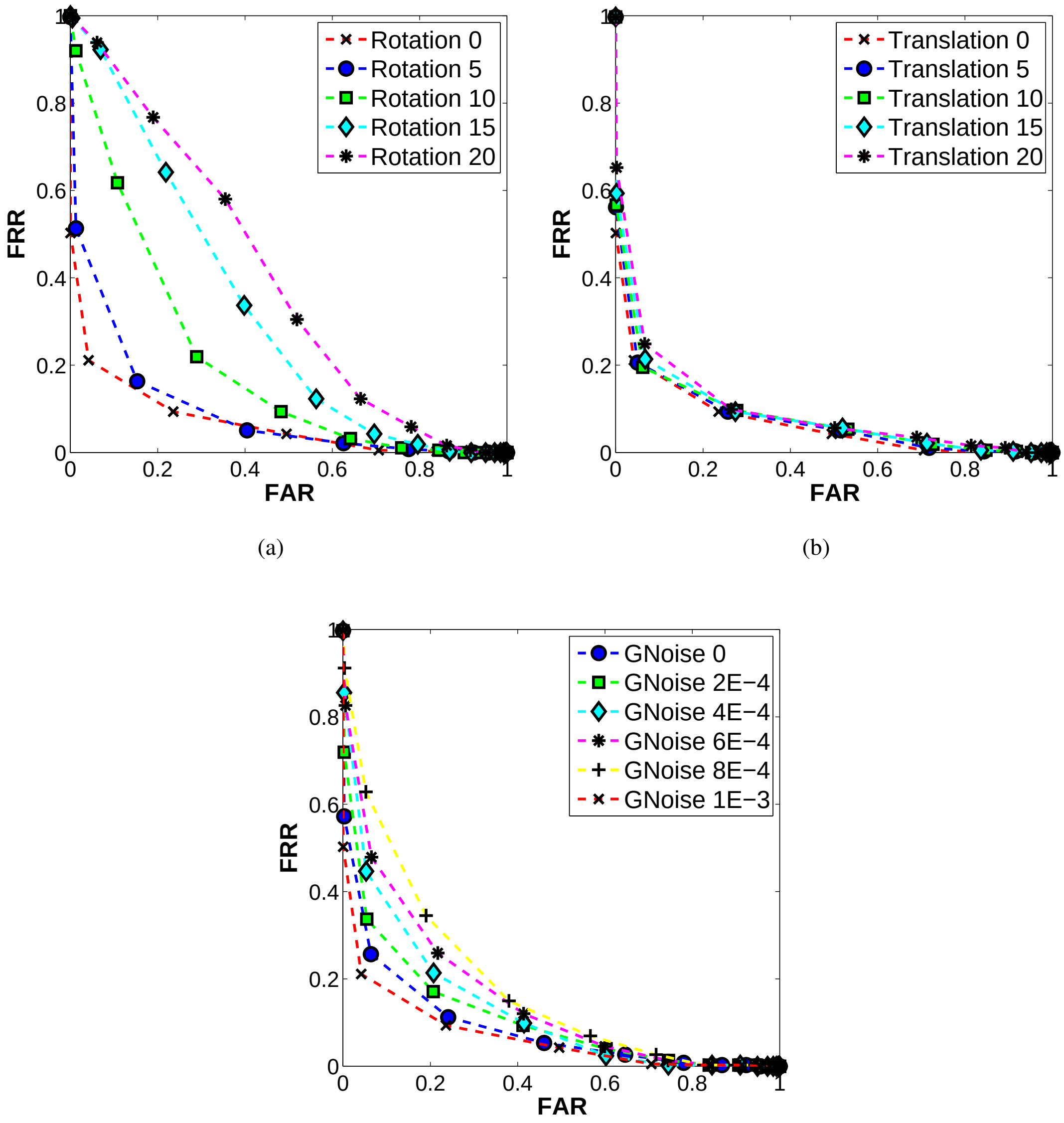


(a)

(b)

(c)

Figure A.31: FAR vs. FRR graph for HOG-MC variations using Euclidean distance matching based on (a) Rotation, (b) Translation, (c) Gaussian noise.

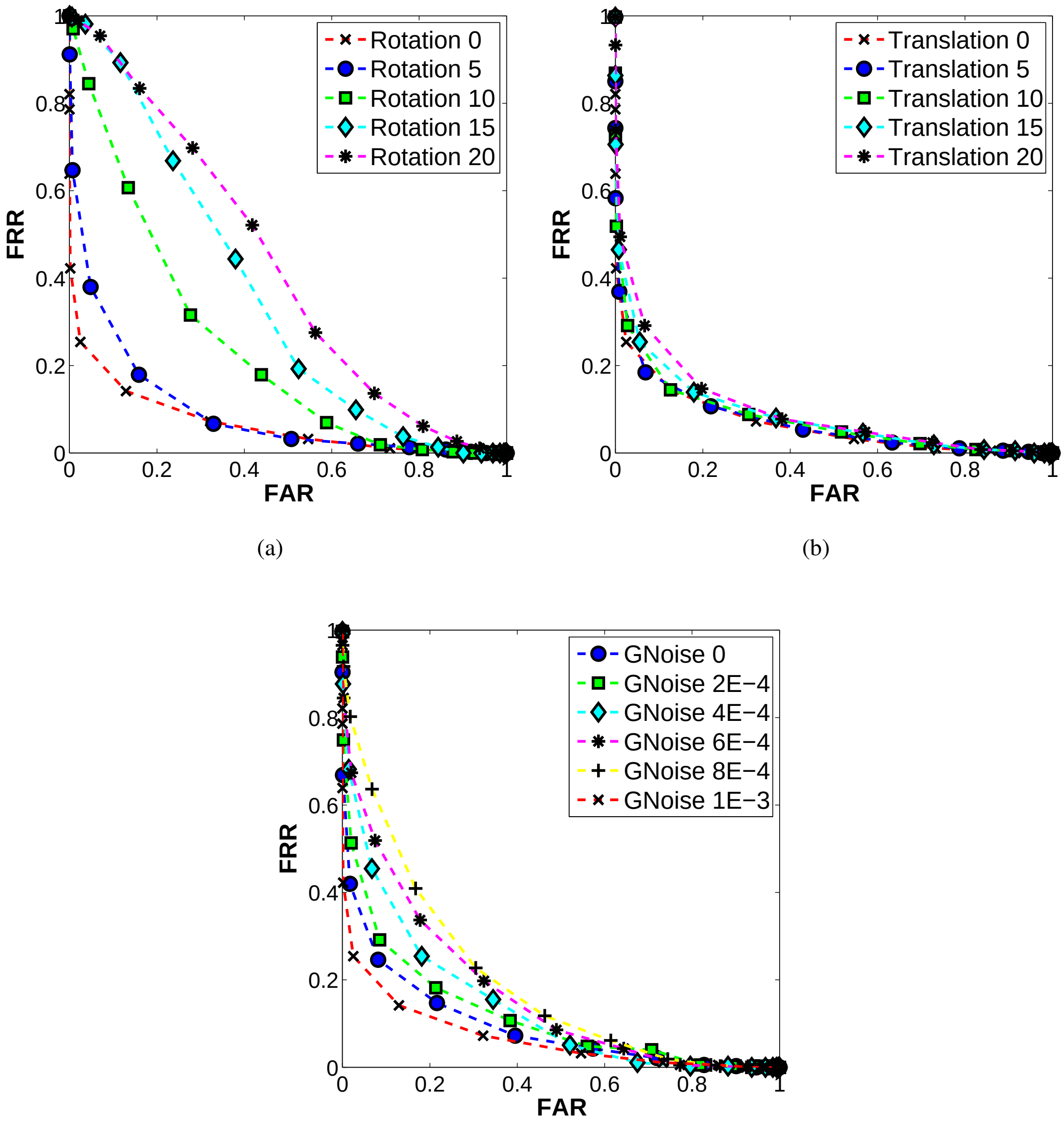


(a)

(b)

(c)

Figure A.32: FAR vs. FRR graph for HOG-MC variations using χ^2 distance matching based on (a) Rotation, (b) Translation, (c) Gaussian noise.

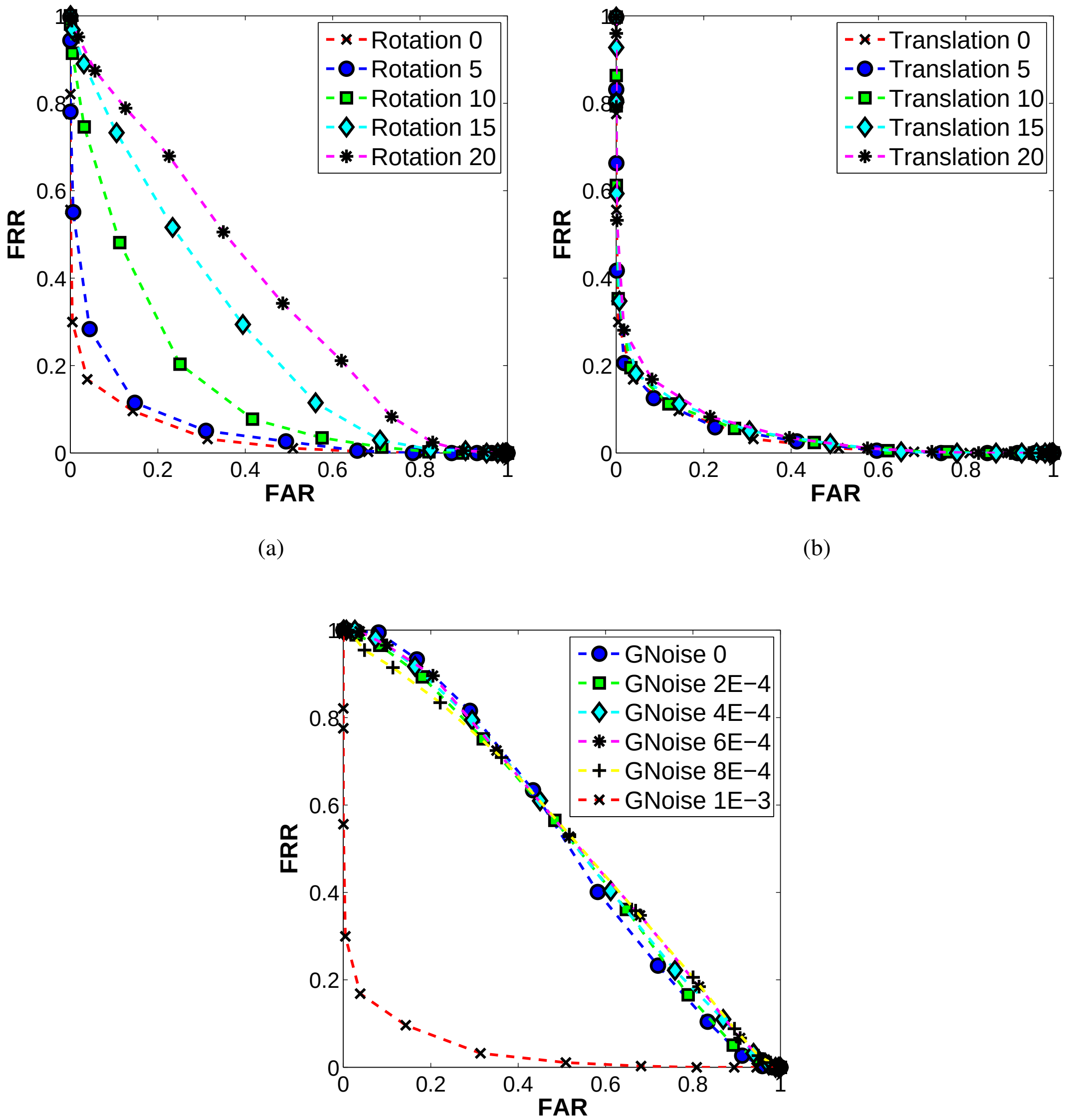


(a)

(b)

(c)

Figure A.33: FAR vs. FRR graph for HOG-MC variations using Earth Mover's distance matching based on (a) Rotation, (b) Translation, (c) Gaussian noise.

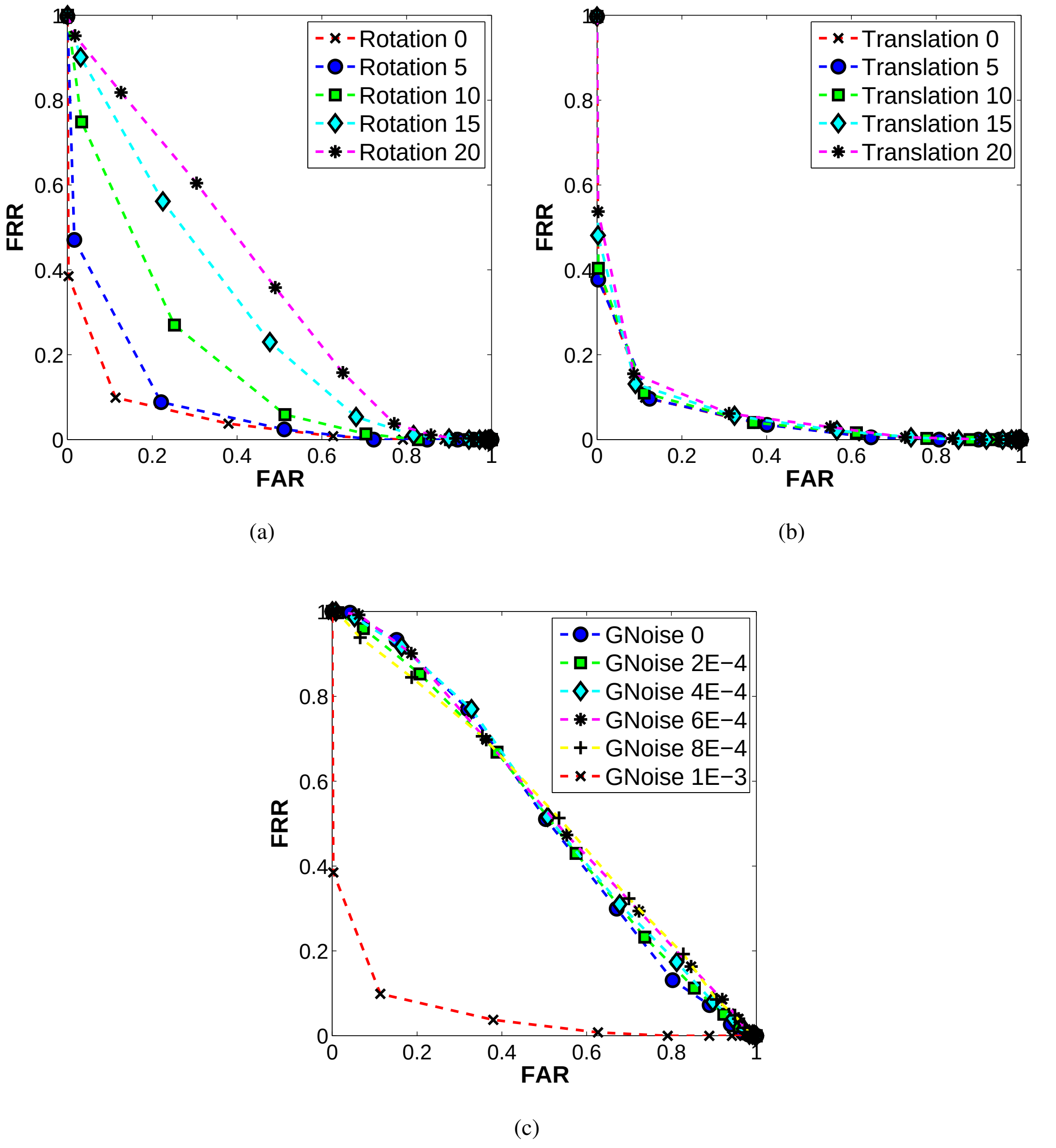


(a)

(b)

(c)

Figure A.34: FAR vs. FRR graph for HOG-WL variations using Euclidean distance matching based on (a) Rotation, (b) Translation, (c) Gaussian noise.

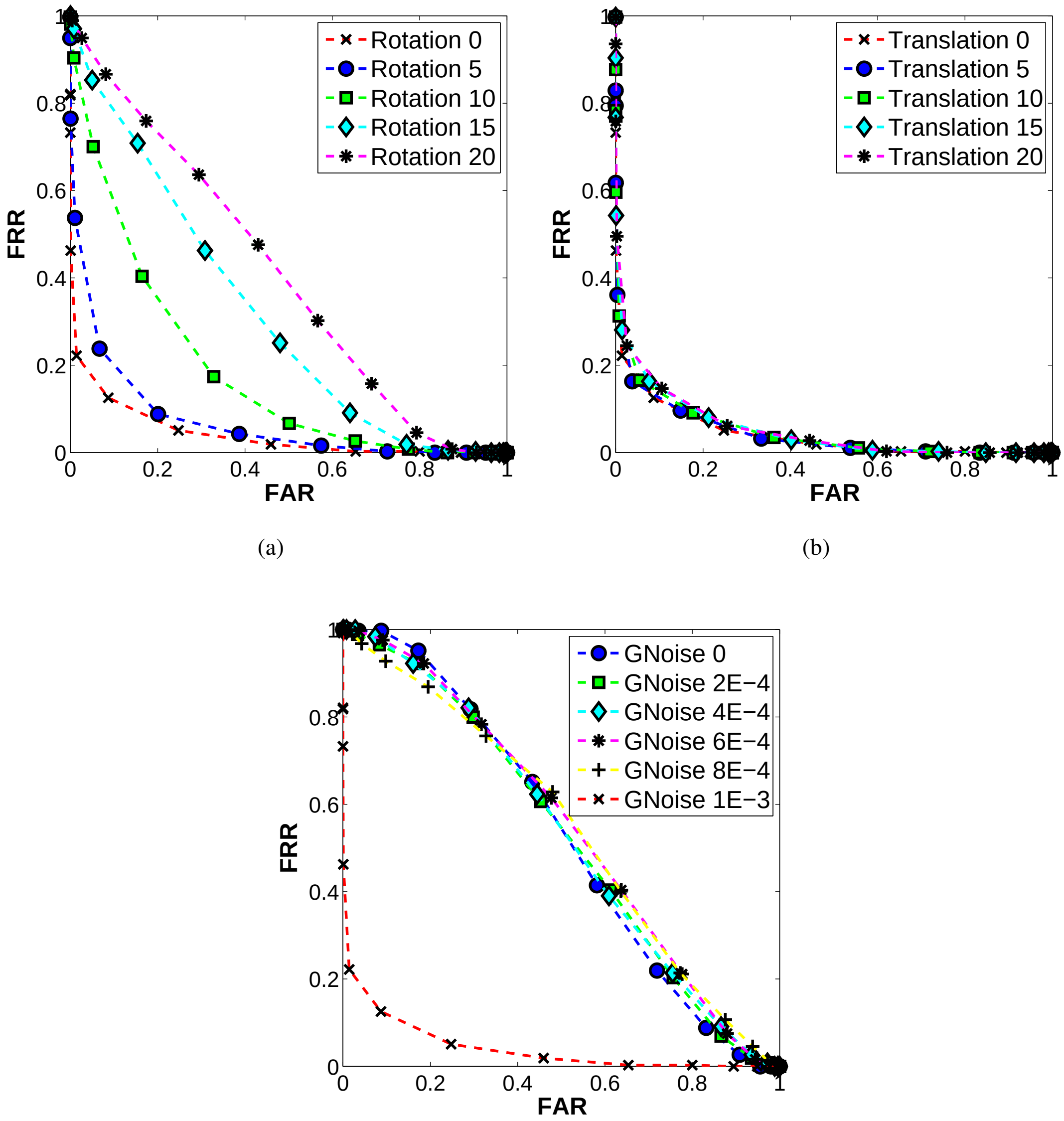


(a)

(b)

(c)

Figure A.35: FAR vs. FRR graph for HOG-WL variations using χ^2 distance matching based on (a) Rotation, (b) Translation, (c) Gaussian noise.

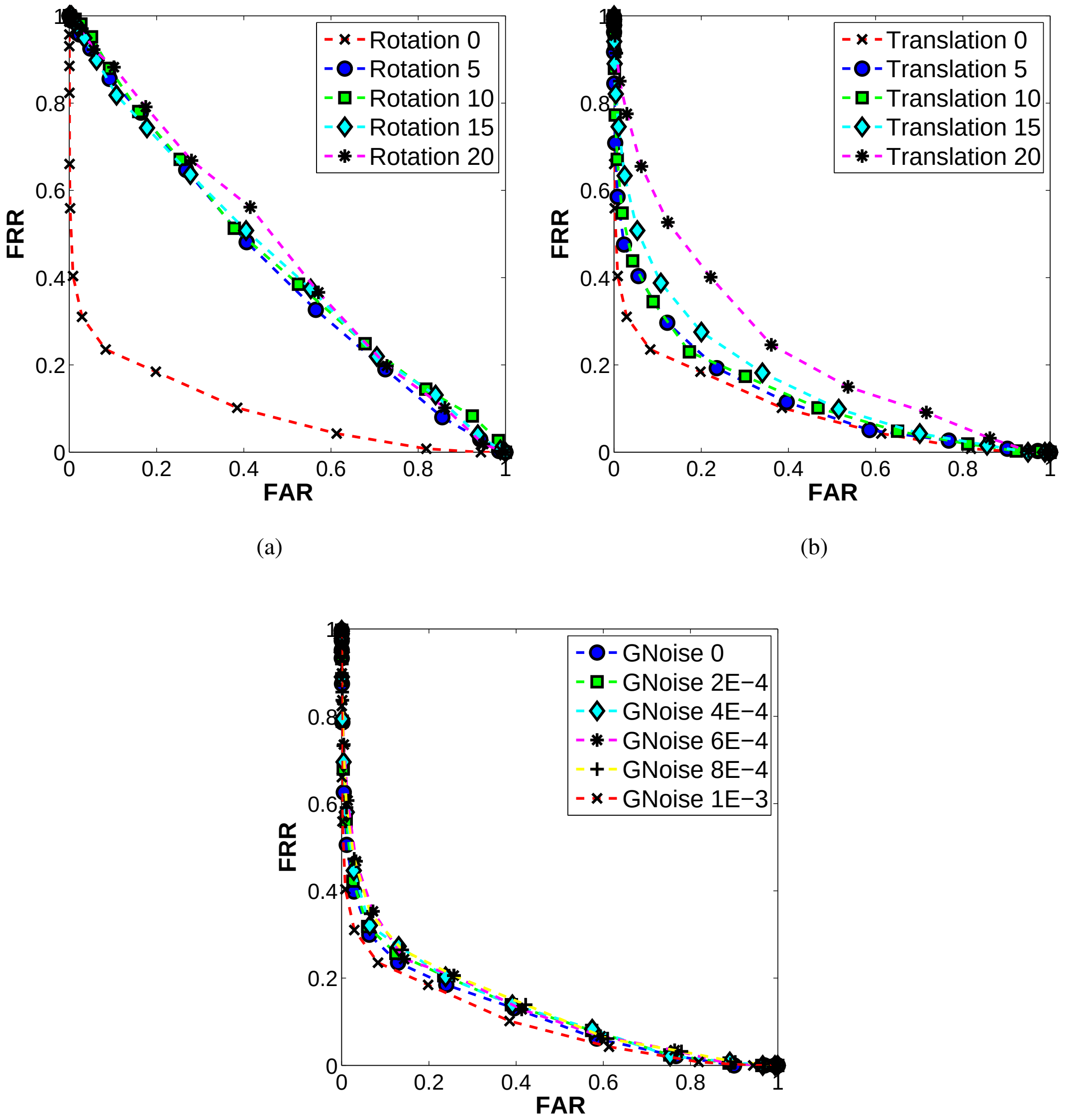


(a)

(b)

(c)

Figure A.36: FAR vs. FRR graph for HOG-WL variations using Earth Mover's distance matching based on (a) Rotation, (b) Translation, (c) Gaussian noise.

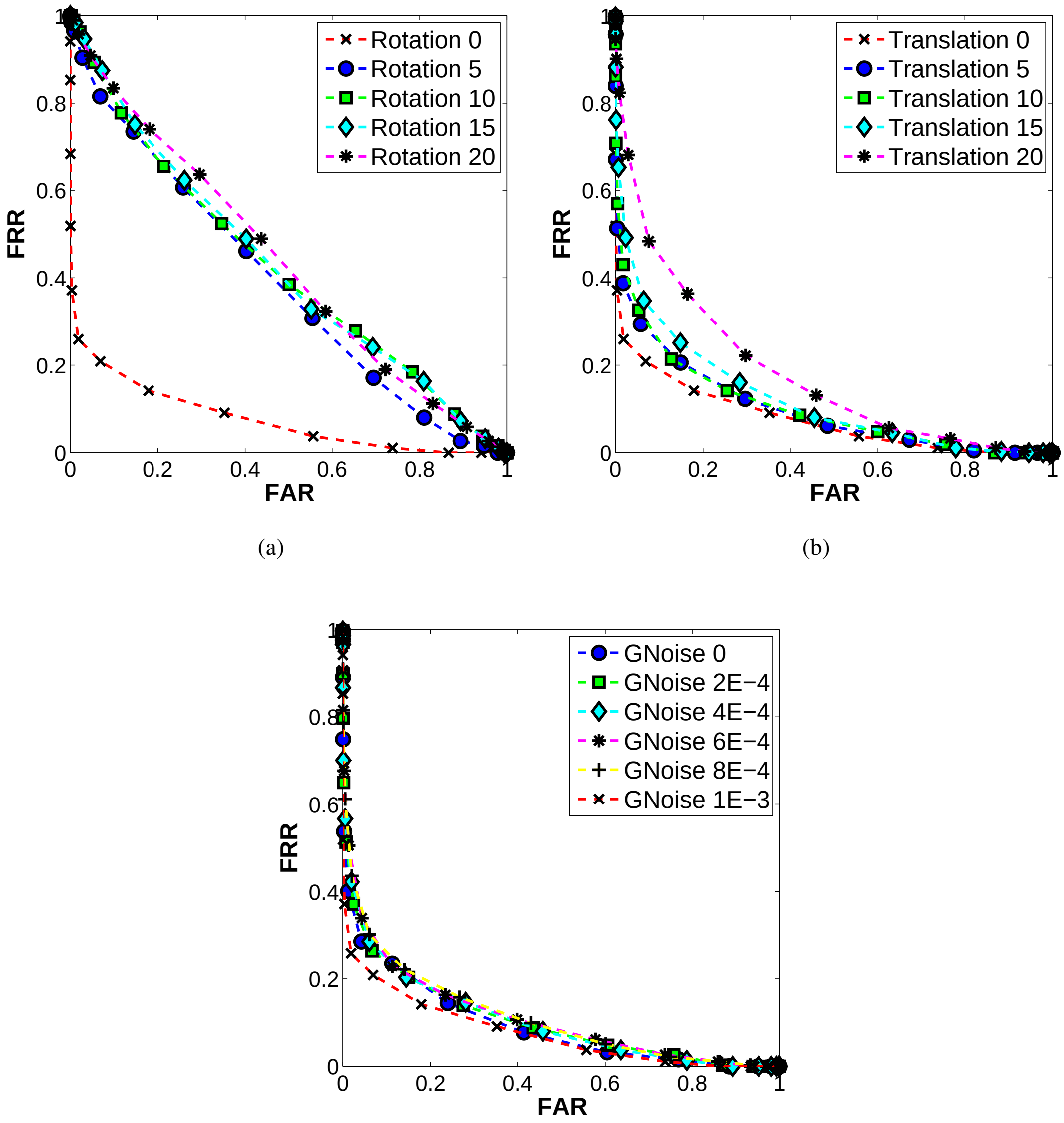


(a)

(b)

(c)

Figure A.37: FAR vs. FRR graph for GBP-LT variations using Euclidean distance matching based on (a) Rotation, (b) Translation, (c) Gaussian noise.

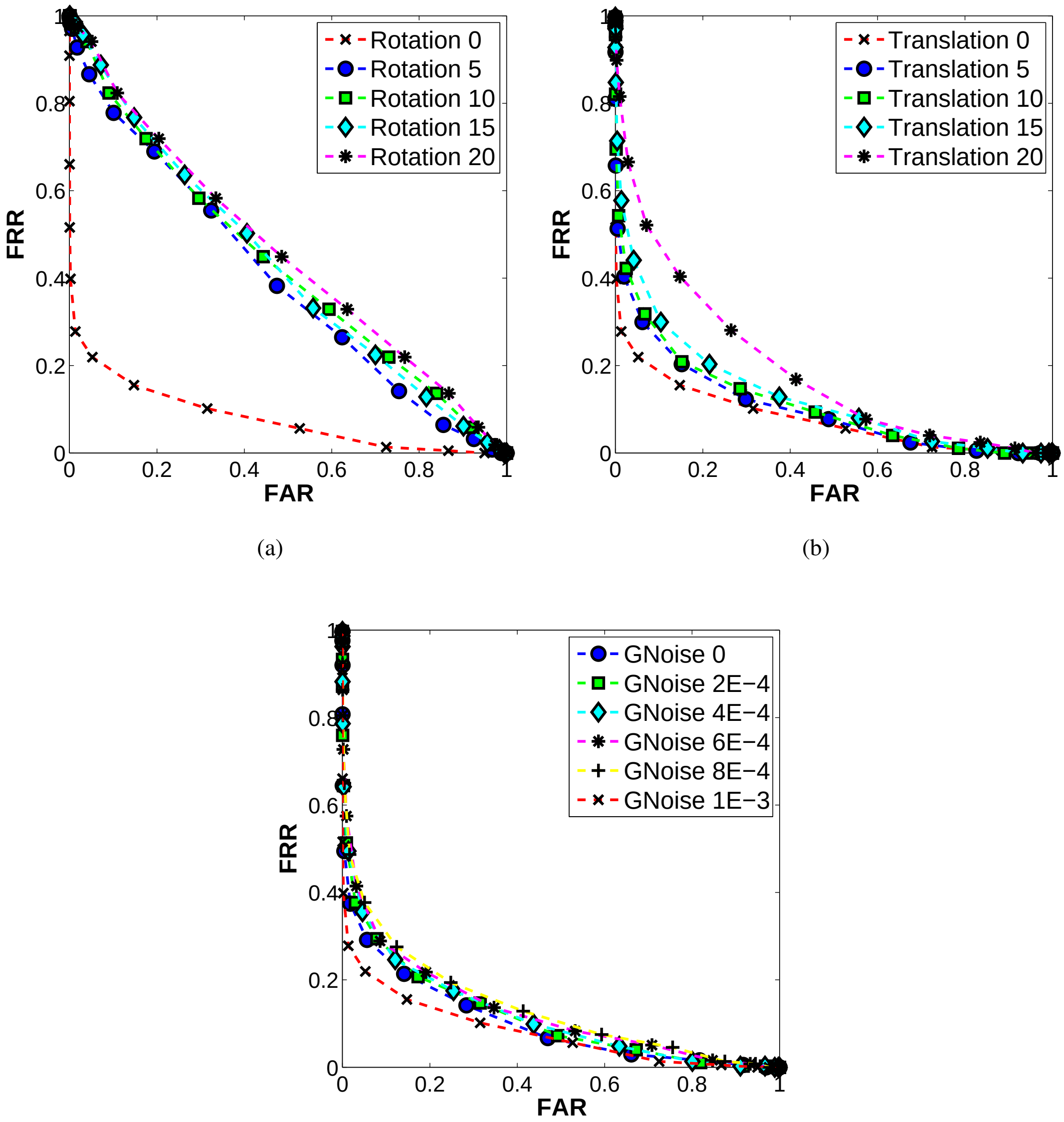


(a)

(b)

(c)

Figure A.38: FAR vs. FRR graph for GBP-LT variations using χ^2 distance matching based on (a) Rotation, (b) Translation, (c) Gaussian noise.

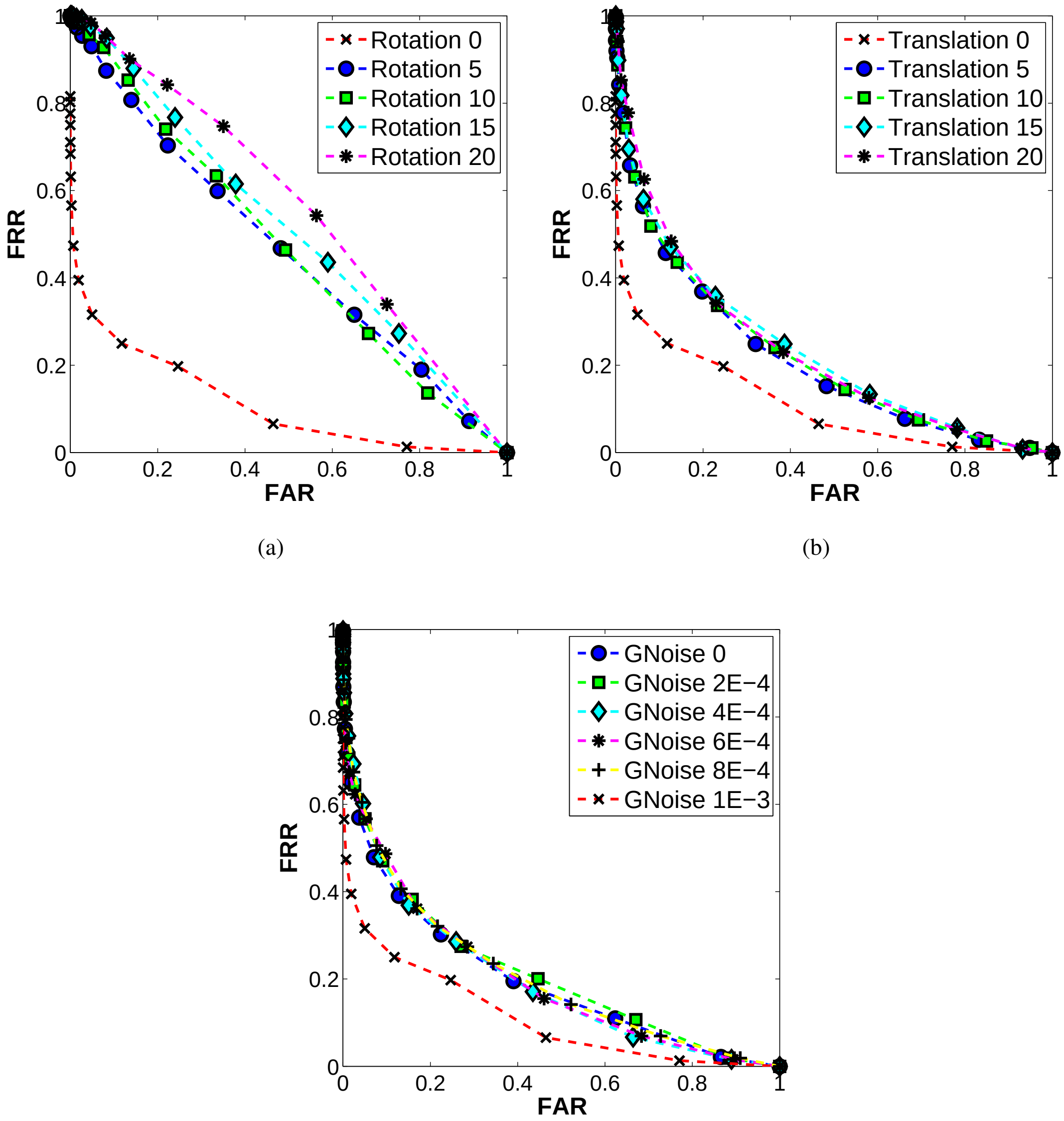


(a)

(b)

(c)

Figure A.39: FAR vs. FRR graph for GBP-LT variations using Earth Mover's distance matching based on (a) Rotation, (b) Translation, (c) Gaussian noise.

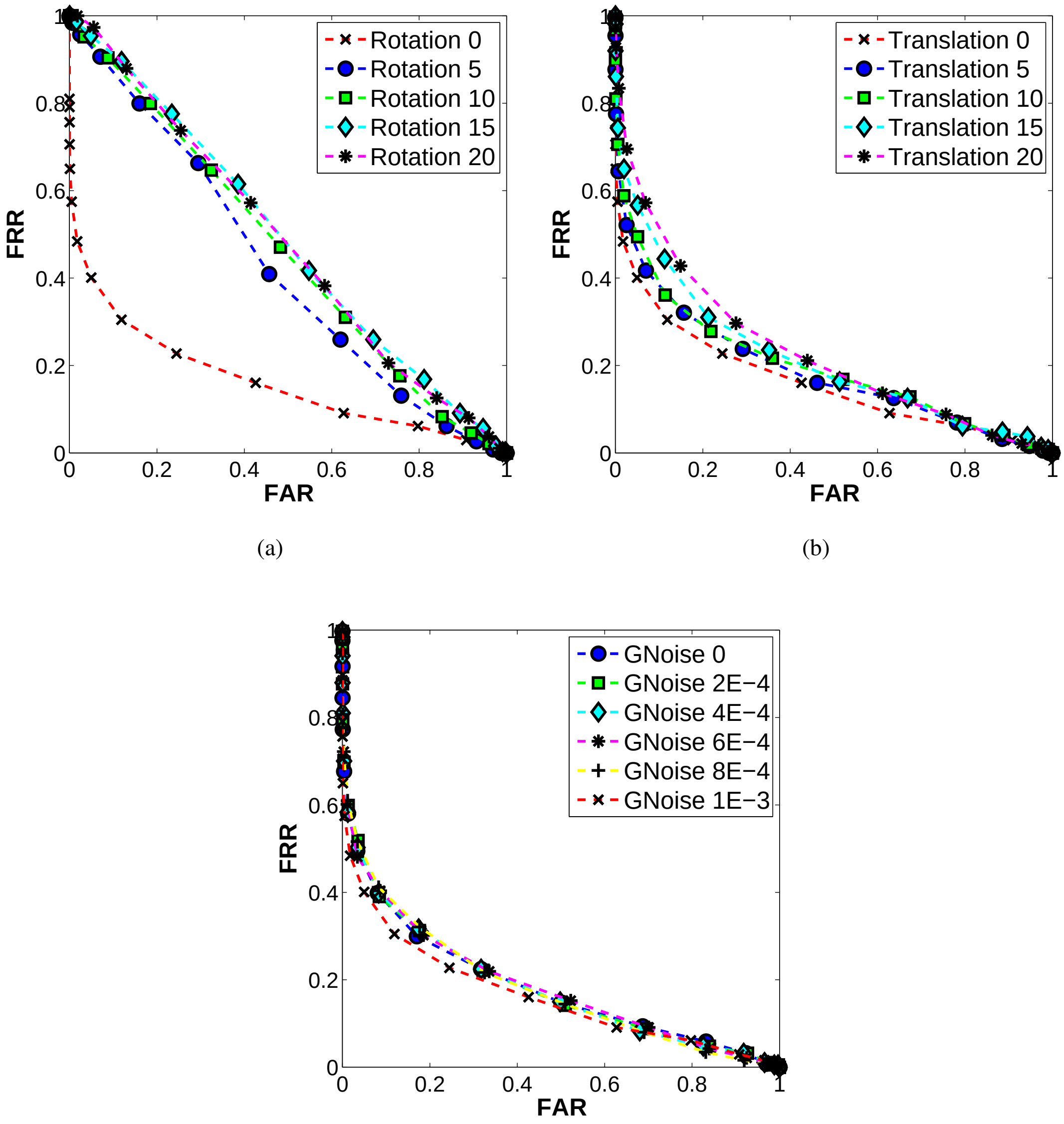


(a)

(b)

(c)

Figure A.40: FAR vs. FRR graph for GBP-MC variations using Euclidean distance matching based on (a) Rotation, (b) Translation, (c) Gaussian noise.

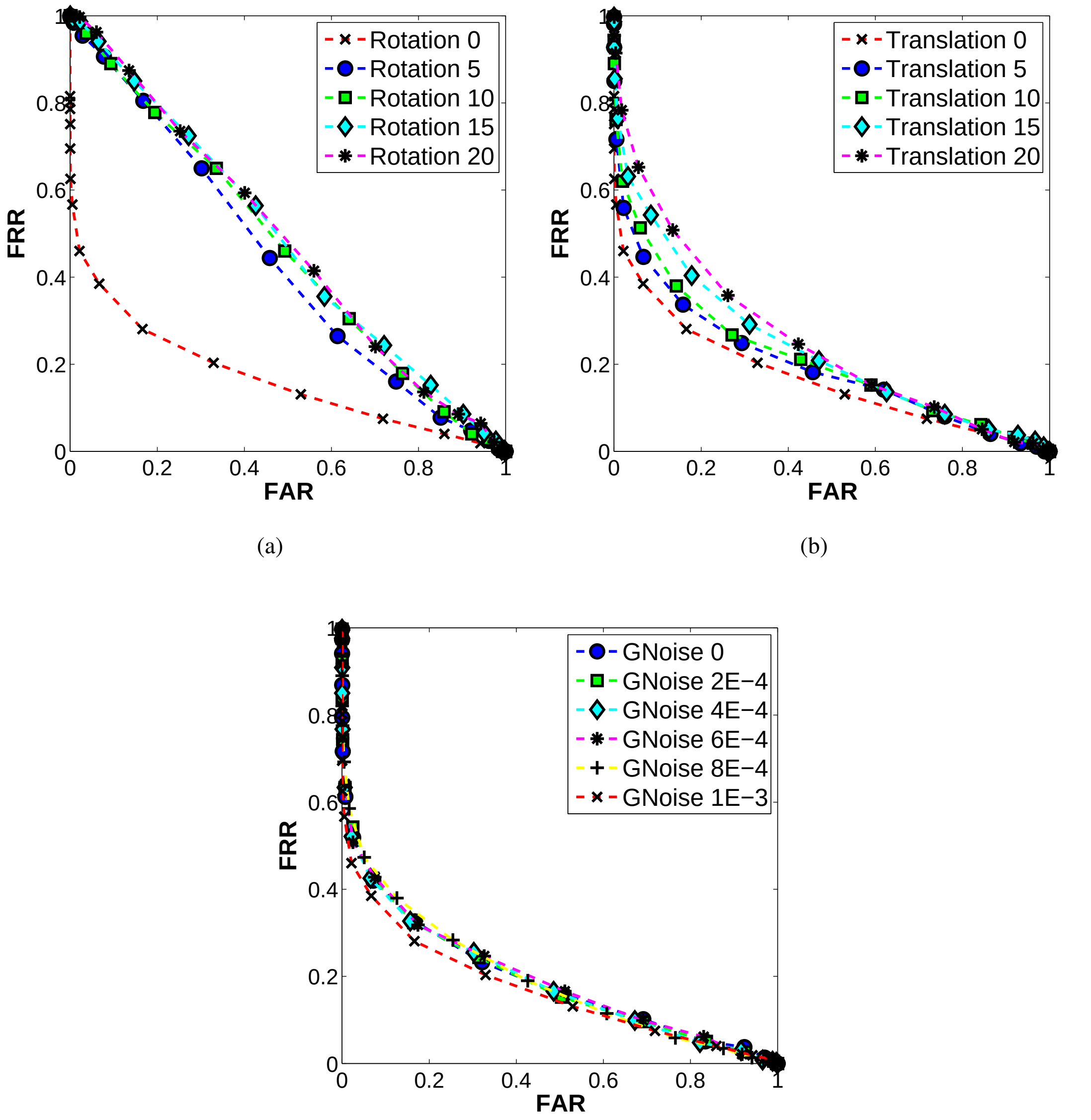


(a)

(b)

(c)

Figure A.41: FAR vs. FRR graph for GBP-MC variations using χ^2 distance matching based on (a) Rotation, (b) Translation, (c) Gaussian noise.

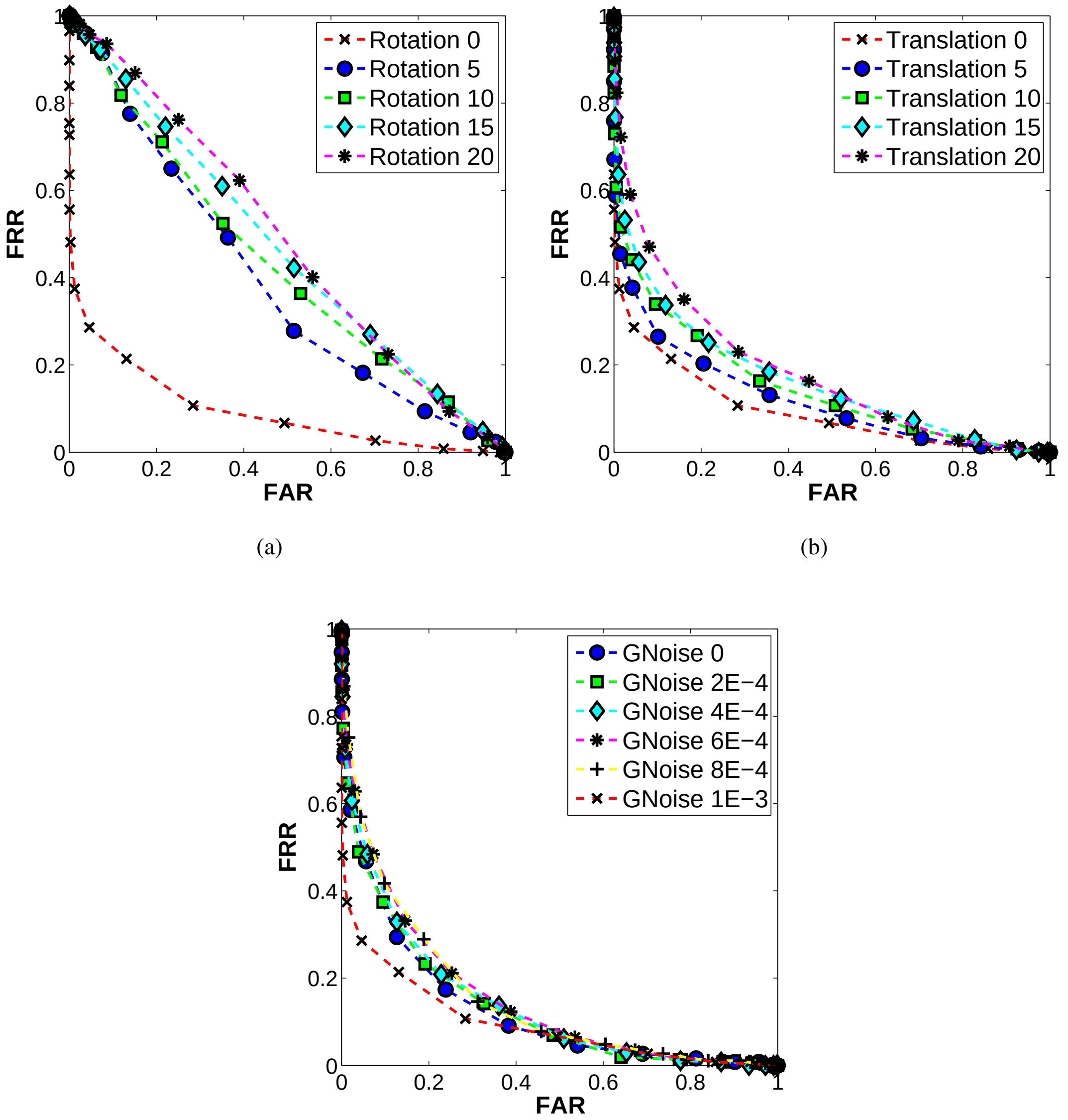


(a)

(b)

(c)

Figure A.42: FAR vs. FRR graph for GBP-MC variations using Earth Mover's distance matching based on (a) Rotation, (b) Translation, (c) Gaussian noise.

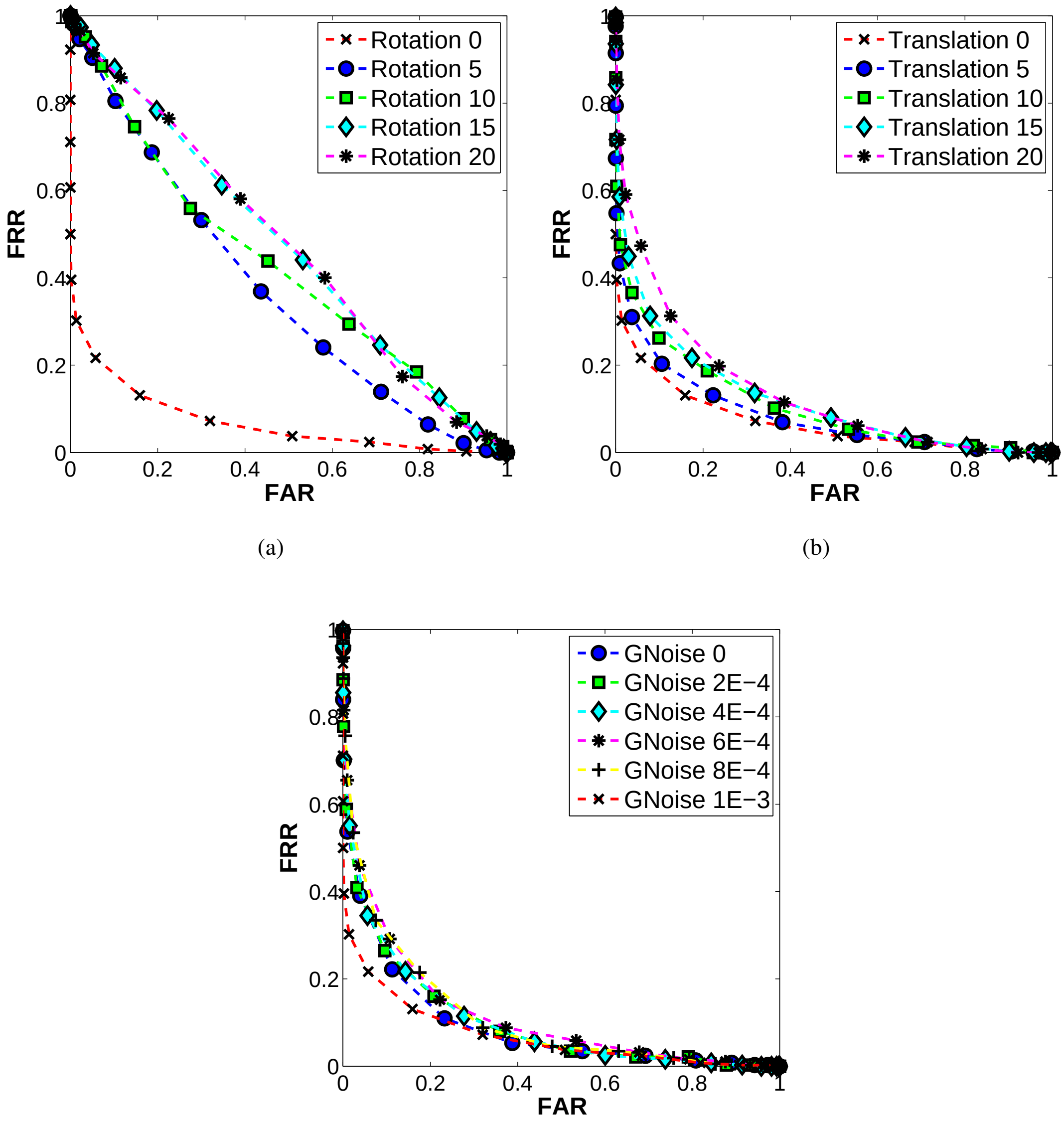


(a)

(b)

(c)

Figure A.43: FAR vs. FRR graph for GBP-WL variations using Euclidean distance matching based on (a) Rotation, (b) Translation, (c) Gaussian noise.

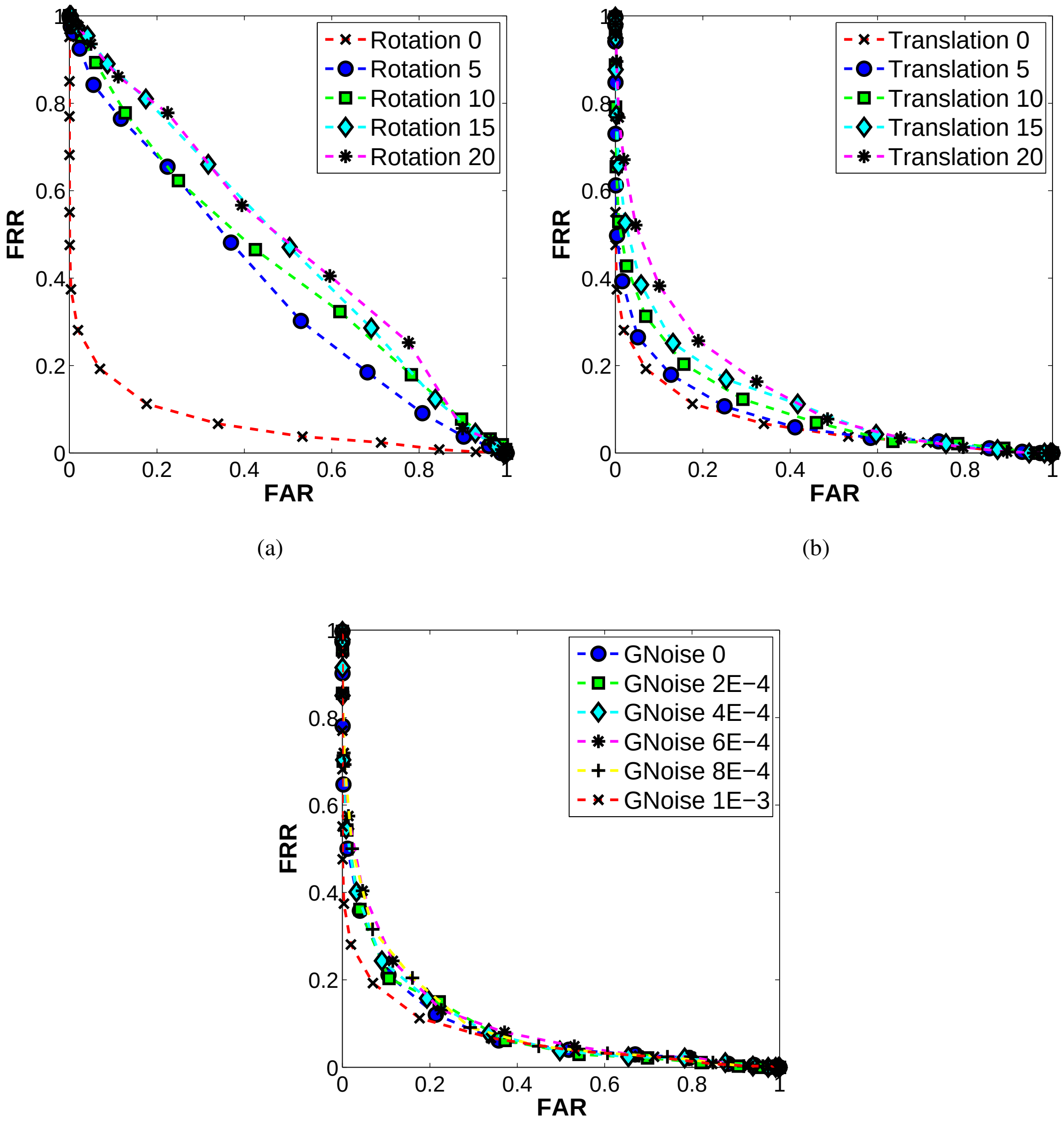


(a)

(b)

(c)

Figure A.44: FAR vs. FRR graph for GBP-WL variations using χ^2 distance matching based on (a) Rotation, (b) Translation, (c) Gaussian noise.

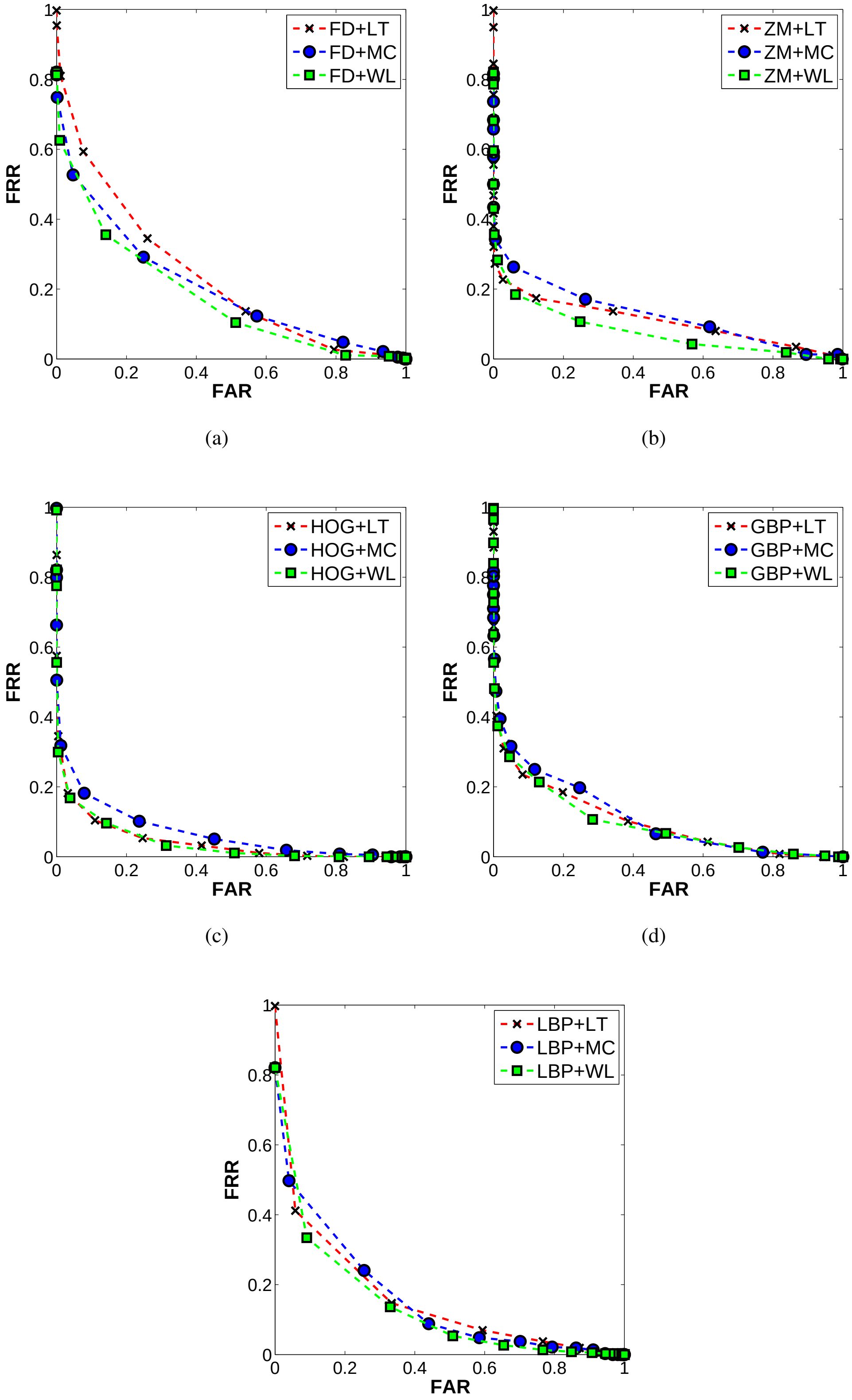


(a)

(b)

(c)

Figure A.45: FAR vs. FRR graph for GBP-WL variations using Earth Mover's distance matching based on (a) Rotation, (b) Translation, (c) Gaussian noise.



(a)

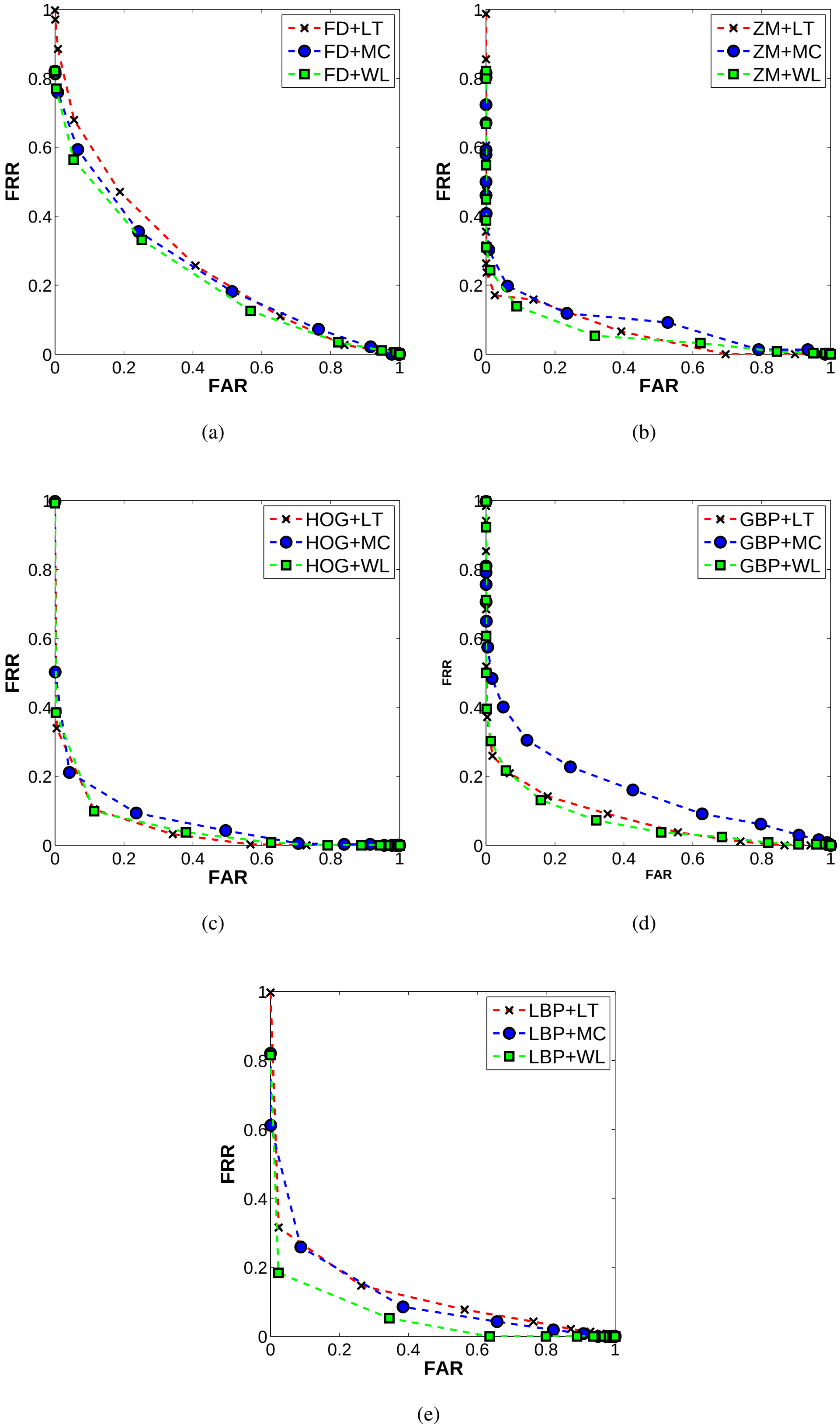
(b)

(c)

(d)

(e)

Figure A.46: FAR vs. FRR graphs for all methods using Euclidean distance (a) FD on LT, MC and WL. (b) ZM on LT, MC, WL. (c) HOG on LT, MC and WL. (d) GBP on LT, MC and WL. (e) LBP on LT, MC and WL.



(a)

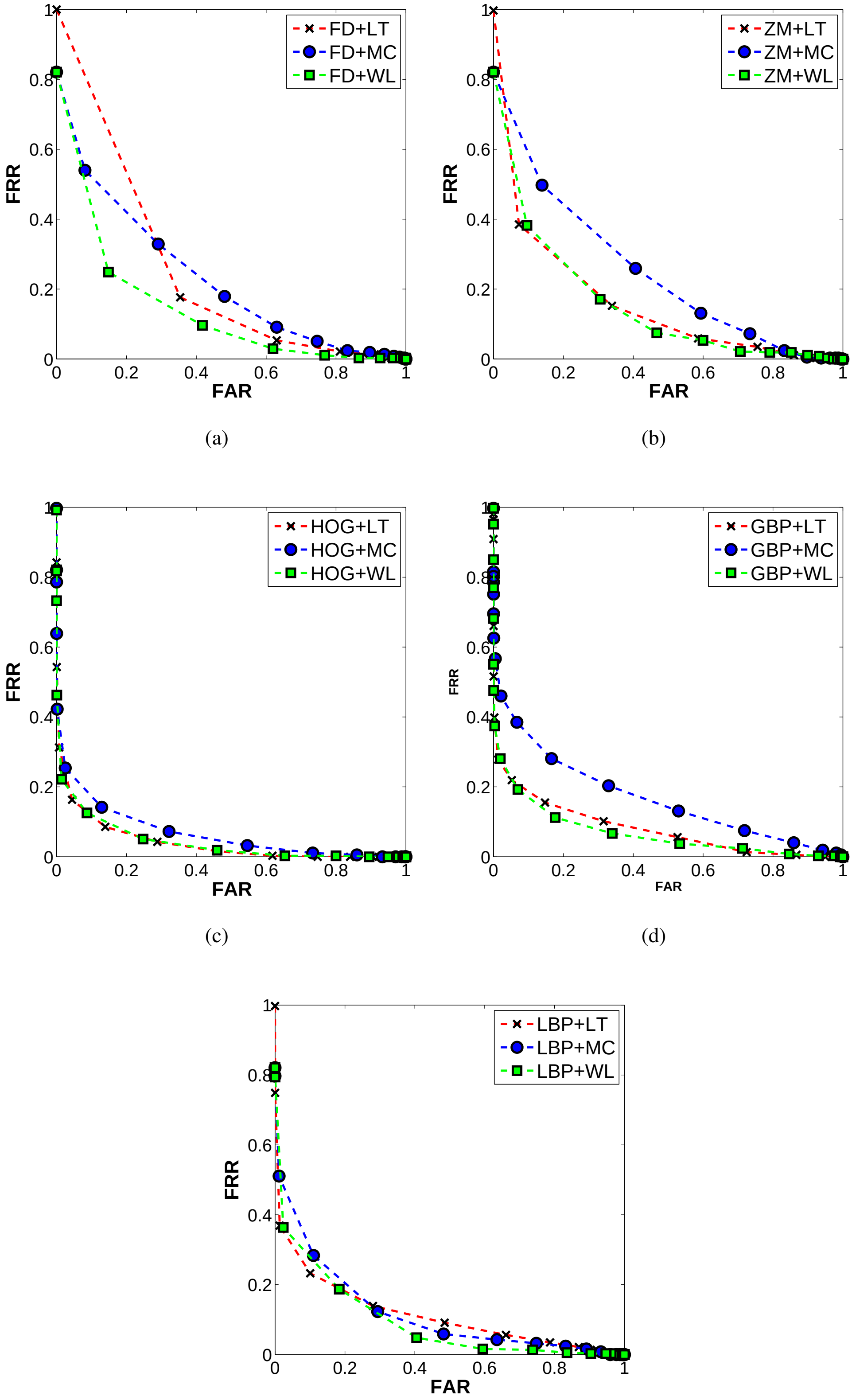
(b)

(c)

(d)

(e)

Figure A.47: FAR vs. FRR graphs for all methods using χ^2 distance (a) FD on LT, MC and WL. (b) ZM on LT, MC, WL. (c) HOG on LT, MC and WL. (d) GBP on LT, MC and WL. (e) LBP on LT, MC and WL.



(a)

(b)

(c)

(d)

(e)

Figure A.48: FAR vs. FRR graphs for all methods using Earth Mover's distance (a) FD on LT, MC and WL. (b) ZM on LT, MC, WL. (c) HOG on LT, MC and WL. (d) GBP on LT, MC and WL. (e) LBP on LT, MC and WL.