Effects of Ageing on Iris Biometric Recognition

Elakkiya Ellavarason

Kongens Lyngby 2013
IMM-M.Sc.-2013-47

Technical University of Denmark
Informatics and Mathematical Modelling
Building 321, DK-2800 Kongens Lyngby, Denmark
Phone +45 45253351, Fax +45 45882673
reception@imm.dtu.dk
www.imm.dtu.dk IMM-M.Sc.-2013-47

Summary (English)

The topic 'ageing' has gained interest in the field of iris biometrics in recent years and is still under investigation. Being fully aware of the fact that iris biometric recognition systems are successfully deployed on large-scale projects such as Indian Unique Identification number (UID), United Arab Emirates (UAE) border immigration service, etc., it is crucial to consider the reliability factor on using iris as a biometric modal for long-term usage. The goal of the thesis is to investigate effects of ageing on iris biometrics. The experimental objectives are three-fold, first, to estimate the presence of iris ageing using several iris processing algorithms. Second, to analyze if the ageing effect is subject-specific. The final one is to analyse the validity of the metrics on which iris ageing is proved.

The investigation of template ageing for iris biometrics is done on the ND-Iris-Template-Aging-2008-2010 database, which contains dataset with 157 subjects having two years of elapsed time between the earliest and most recent iris images. Analysis of ageing effects across six different iris recognition algorithms, revealed performance degradation across all of these algorithms.

The distinguishing factor of this work is that the previous work on iris ageing has always considered large dataset for the overall result, but this thesis deals with the performance analysis of each subject present in the dataset. Subject-specific analysis revealed that variations in pose and illumination greatly contribute to worse comparison score. Further, results obtained using the multi-instance image analysis algorithms across different feature extraction showed promising results which challenges the metrics on which ageing is proven.

Preface

This thesis was prepared at the department of Informatics and Mathematical Modelling at the Technical University of Denmark in cooperation with Center for Advanced Security Research Darmstadt in fulfillment of the requirements for acquiring an M.Sc. in Computer Science and Engineering.

The thesis deals with the evaluation of ageing effects in iris biometrics. Iris ageing effects are explored using six different iris processing algorithms. The main focus is to investigate if the template ageing is subject-specific and the reliability of the metrics on which ageing is proved.

The thesis consists of 8 Chapters and Appendices which contains the detailed description of the background, previous work, experimental evaluations conducted, results obtained, discussions based on the results and future work perspectives.

Lyngby, 30-June-2013

Elakkiya Ellavarason

Acknowledgements

I would like to offer my profoundest gratitude to Prof. Dr. Christoph Busch for providing me with an excellent opportunity to undertake my thesis at Center for Advanced Security Research in Darmstadt whose enthusiasm for "Biometrics" had lasting effect.

My sincere thanks goes to Dr. Christian Rathgeb, Postdoc at CASED, for supervising my project and mentoring me during my time at CASED. His ideas, generous support, and persistent help kept me focused throughout the thesis.

I would like to thank Prof. Rasmus Larsen for his valuable assistance in supervising my project work at Danish Technical University.

I am also indebted to my parents and sisters for their constant encouragement and emotional support. I leaned on many people for advices, both technical and otherwise, during the course of project, and it is here that I thank them: Deepak, Antony, and Vikas. Thank you all for sticking out with me during tough days.

Contents

Chapter 1

Introduction

The necessity for developing sophisticated identity management mechanisms has grown in recent years in order to mitigate security threats as well as to establish reliable identities. One of the ways to do so is using biometrics. The field of biometrics deals with uniquely identifying a person based on set of attributes associated with a person. Biometric technology is used for registering and maintaining personal identities of individuals. This discipline has gained impetus in recent years especially due to its significant advancements, as it offers range of practical implications in commercial, governmental and forensic fields.

While there are various physiological or behavioural traits which are adopted as biometric identifiers, *iris* is believed to be the most distinguishable biometric cue that could be used for personal identification purposes [FS87]. Iris identification technique is considered as an attractive method of biometric recognition due to its unparalleled accuracy, compact template size and remarkably fast comparison speed. Hence, it has inevitably gained traction in challenging application fields such as Department of Defense military operations overseas [KT10], multi-modal capture programs such as largest National ID program (India's UID program) [Zel12], and the U.S. government's Personal Identity Verification (PIV) card program for authentication of federal employees and contractors [All11]. An obvious question that arises in mind based on such large-scale deployments is - what potential challenges could arise on using iris biometrics in such a wide-spread deployments which involves all citizens of a nation?

One such challenge adversely impacting large scale deployment could lie in the ageing impact over the stability of iris texture. For a long time, there existed an optimistic belief in iris biometric field about iris ageing:

"...[iris is] *unique to peron and stable with age"* [Dau93]
"...[iris is] *stable over an individual's lifetime"* [TSK07]

Recent scientific researches [FE11], [BBF09] , [FB12], [BBFP13] suggest that the statements claiming the stability of iris texture over lifetime is no longer true. These findings in all likelihood reflect that the biological ageing of iris would significantly impact the performance profiles of the biometric identification and verification techniques as it raises questions regarding the accuracy and reliability of iris biometric recognition. This thesis addresses the challenge related to iris ageing.

1.1 Problem statement

Several scientific papers in recent times have concluded that iris is not immune to changes with passage of time. Few citations from scientific papers predicting the presence of iris template ageing are listed below.

"*...results demonstrate that iris biometrics is subject to a template aging effect*" [BBFP13]

" *...we present extensive experimental evidence of physical ageing effects on iris recognition and interrelated factors which are important in assessing ageing effects*" [FE11]

"*...iris biometric enrollment templates may undergo aging and that iris biometric enrollment may not be* ***once for lifetime*** " [BBF09]

The conclusions of the papers propose that iris is not immune to changes with passage of time. These claims denote that the long-term use of iris biometric systems in application areas such as security or forensics can be a challenging task. With the deployment of iris recognition technologies at border-crossing system in the UAE [ARAK08], the findings has a huge significance.

In view of the above findings, the basis of this thesis is formed with an objective to investigate effects of ageing in irises. The experimental evaluations concentrate on the findings of Kevin Bowyer *et al.* in the paper [FB12] about template ageing on irises.

The motivation for conducting this thesis is lack of extensive research in the field of iris ageing. Only at the university of Notre Dame, a group of scientists (Kevin Bowyer and his team) have been conducting tests on iris ageing and presenting their findings for world-wide use. This thesis is mainly conducted to verify how far the conclusions given by this group are true about the iris ageing. The conclusions given in the paper are based on metrics with the false non-match rate (FNMR) and false match rate (FMR) comparisons results. Any other possible parameters resulting in degradation of matching scores across years are not taken into consideration.

1.2 Objective

This work distinguishes itself from the existing work in numerous ways which are listed below:

- Results on iris ageing has always considered large dataset for experiments. But, this work closely examines the ageing effects for each subject present in the dataset. That forms the basis for the subject-specific experiment.
- Researches that has taken place so far on iris template ageing are mainly based on the iris recognition software IrisBEE [FB12] which uses single feature extraction technique. This thesis deals with the diagnosis of iris template ageing using a USIT(University of Salzburg Iris Toolkit)[1] software which implements six different iris processing algorithms.
- Results on iris template ageing is always affirmed by a decrease in FNMR over increasing time span. No study has checked how reliable this metric is. This thesis deals with analysing this metric based on a multi-instance image analysis scheme developed during the project phase.

In order to carry out the research, experiments with a dataset of approximately two years of elapsed time between the most recent and the earliest iris image are taken into consideration. Based on the similarity scores generated by six different algorithms used in USIT, match and non-match distributions are obtained for genuine and imposter image comparisons for short and long time span image comparisons. Further, performance analysis of iris processing algorithms

[1] USIT: University of Salzburg Iris Toolkit: http://www.wavelab.at/sources/

is also done. Second part of the thesis deals with determining if the ageing is subject-specific. Followed by experiments to analyse the results using the FNMR metrics on a subset of original dataset in order to check the reliability of this result. Finally, discussion on possible countermeasures to deal with iris template ageing to minimise the deterioration in the long-term performance of biometric authentication systems is done.

1.3 Structure of thesis

The thesis is organised as follows. Chapter 2 gives an overview of general terminology, biometric recognition, and performance standards. Chapter 3 gives detailed description of iris biometrics. Related work on iris ageing is provided in chapter 4. This chapter talks about ageing in other biometric modalities followed by ageing in iris biometrics. Chapter 5 describes the details of the experimental setup and procedure followed to conduct each experimental phase. It is divided into three sections namely, cross-algorithm analysis, subject-specific analysis and multi-instance image analysis. The results of each of the anaysis phases are provided in chapter 6. Discussion based on the results obtained and general discussion on iris template ageing and textural ageing is presented in chapter 7 and conclusion and future work is provided in chapter 8.

Chapter 2

Biometrics: Fundamentals

This chapter gives an overview of the fundamentals of biometrics. It is divided into three sections. The first section describes the basic terminology used in biometric systems, the second section explains biometric recognition. The last section of this chapter explains about the performance testing standards present in biometrics. The terms used in this thesis abides with the standards established by the ISO/IEC [1].

2.1 Terminology

The ISO/IEC JTC1 SC37 [ISO12] defines biometrics as:

"*(automated) recognition of (living) persons based on observation of behavioral and biological (anatomical and physiological) characteristics*".

According to the definition given by ISO/ IEC [ISO12] the biological and behavioral characteristics are physical properties of body parts, physiological and

[1] International Organisation for Standardisation (ISO) and by the International Electrotechnical Commission (IEC)

behavioural processes created by the body and combinations of any of these. Biometric characteristics are categorised as biological/physiological and behavioral.

Biological/Physiological: These characteristics are the biological features of the human body. Example of biological characteristic are fingerprints, iris pattern, face topology.
Behavioral: These characteristics is obtained by observing human behavior pattern. Example of behavioral characteristics are voice patterns, gait, handwritten signature dynamics.

Biometric term '*subject*' is defined as the person whose biometric data is within the biometric system and '*template*' refers to vector of stored biometric features, which is directly comparable to the biometric characteristics of a biometric sample [ISO12]. For a physiological and/or behavioral characteristic of a subject to be considered as a biometric characteristic, the following requirements have to be met:

- ***Uniqueness:*** it is the distinctive characteristic that differentiates one subject from another. It is believed that each subject possesses this characteristic. No two individuals possess the same characteristic.
- ***Performance:*** this property deals with the accuracy, speed and robustness of biometric recognition technology used for biometric identification and verification.
- ***Permanence:*** measure of how well a biometric resists aging effects, how the feature vector remains invariant over time and persistence of the extracted feature.
- ***Collectability:*** it is the measure of ease of acquisition of the biometric trait for measurement.
- ***Acceptability:*** it is the degree of public approval of the biometric technology. The extent of willingness of the data subjects to use it as a biometric identifier.
- ***Circumvention:*** the measure of how easily the system can be fooled using fraud methods. It includes the security of a capturing device measuring the biometric characteristics.

This thesis makes use of the biological feature - iris as a biometric characteristic for the analysis.

2.2 Biometric recognition

The purpose of this section is to give the reader an opportunity to familiarise himself/herself with the biometric recognition concepts. This section gives description of the biometric recognition technology according to ISO SC37 [ISO12]. The biometric recognition system comprises of several subsystems. The workflow of the system is shown in the Figure 2.1.

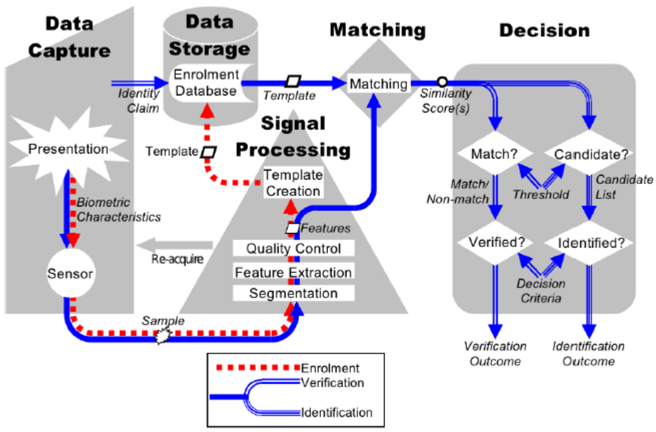


Figure 2.1: Architecture of biometric system taken from [ISO12].

2.2.1 Biometric system module

As depicted in Figure 2.1, the biometric system is largely catagorised into four modules.

- ***Sensor module:*** This module corresponds to the data capture process in Figure 2.1. It is responsible for capturing the biometric data of the subject. An example is a iris sensor that captures the image of the subject's iris.
- ***Feature extraction module:*** This module corresponds to the signal processing phase in Figure 2.1. It processes the biometric data and extracts a set of discriminatory features from the biometric data. For example, edge detection and pupil detection from the iris to extract iris descriptors in a feature extraction module of a iris-based biometric system.
- ***Matcher module:*** This module corresponds to the matching phase in Figure 2.1. It carries out the comparison of the captured biometric data

with the stored template and generates the matching or comparison score. For example, in the matching module of a iris-based biometric system, the fractional Hamming distance is calculated, which measures the fraction of bits for which two iris codes (captured image iris code with template iris code) disagree. Low Hamming distance represents strong similarity of iris codes. Decision making module is also a part of the matcher module, which confirms the identity of the subject using verification or identification depending on the acquired matching score.

- ***System database module:*** This module corresponds to the data storage phase in Figure 2.1. It is a vital module of the biometric system which is responsible for storing the biometric templates of the enrolled users. System database module is used majorly in the enrollment phase. In order to account for variations observed in the biometric traits, multiple templates of the subject is saved in the system database. It is also required that the templates in the database is updated over time for having a reliable and robust biometric system.

2.2.2 Enrollment

The purpose of an enrollment process is to register individuals/subjects into the biometric systems. A biometric template associated with the subject is created and stored in the database which in later stages is used for verification or identification purposes. The enrollment process consists of the following steps:

(1) ***Acquisition:*** Acquiring biometric sample from the subject. In case of iris recognition, we obtain the image of iris.

(2) ***Pre-processing and Feature extraction:*** Extracting the features by processing the biometric sample

(3) Performing quality checks on the acquired features of the biometric sample.

(4) Creation of reference and conversion to a biometric interchange format.

(5) ***Comparison:*** Validating the template usability by performing verification and identification tests

(6) Repeat process from step 1 if the validation in step 5 fails.

2.2.3 System operation modes

Based on the application context, there are two modes in biometrics, namely, verification or identification.

Verification

The system authenticates a subject's identity by comparing the captured biometric data with the biometric template(s) of the same subject stored in the database. The system conducts a (*1:1*) comparison in order to determine if the identity claim is true or not. Verification is usually used for positive recognition for preventing multiple people using the same identity [Way01].

The verification process can be mathematically formulated as follows:

For a input feature vector X_Q(derived from the biometric data), and a claimed identity I, determine if (I, X_Q) belongs to class *(*w_1*,*w_2*)*, where w_1 is true claim (genuine user) and w_2 is false claim (imposter user). Usually, X_Q is compared against , X_1 - the biometric template corresponding to user the same [JRP04]. Thus the formula can be given as:

$$(I, X_Q) \in \begin{cases} w_1, & \text{if } S(X_Q, X_1) >= t \\ w_2, & \text{otherwise} \end{cases} \tag{2.1}$$

where t is the defined threshold and the function S(X_Q,X_1) measures the similarity between feature vectors X_Q against the template X_1 belonging to the same subject. S(X_Q,X_1) is known as the matching or comparison score between the biometric measurements of the subject and the claimed identity [Way01]. If the comparison score is greater than or equal to the threshold t, then w_1 is returned else w_2 is returned.

Identification

The identification of a subject is carried out by searching the template against all the users present in the database for a match. The biological biometric sample of a subject is compared to N other templates stored in the database. In this case, $(1 : N)$ comparisons are carried out. One of the problems with identification is the long processing time. Identification component is a critical element of the biometric system in negative recognition applications where the system tries to find out whether the subject is who he/she (implicitly or explicitly) denies to be.

It can be mathematically formulated as following:

Given an input feature vector X_Q(derived from the biometric data), determine the identity I_k , $k \in 1, 2, 3, ...N, N+1$, where I_1, I_2,..., I_N are the enrolled identities for a subject in the system and I_{N+1} represents the reject case where no match is identified for the subject on which the identity test is conducted. Thus the formula can be given as:

$$X_Q \in \begin{cases} I_k, & \text{if } max(S(X_Q, X_{IK})) >= t, K = 1, 2, .., N \\ I_{N+1}, & \text{otherwise} \end{cases} \tag{2.2}$$

where t is the threshold and X_{IK} is the biometric template corresponding to identity.

2.3 Biometric performance testing

The performance of a biometric system is evaluated using different metrics. It is based on the failure type that occurs in the system modules of the biometric system. The failure types are explained in detail in this section.

The final output of a biometric matching system is the matching or comparison score S(X_Q,X_1) which indicates the similarity between the input (X_Q) and the stored template (X_1) present in the database. Threshold t helps in making this decision. The pairs of biometric samples generating scores higher than or equal to t are known as mate pairs (belonging to the same subject) and pairs of biometric samples generating scores lower than are inferred as nonmate pairs (i.e., belonging to different subjects) [JRP04]. Genuine distribution is the distribution of scores produced by the pairs of sample comparisons from the same subject. Similarly, comparisons from the different subjects is known as imposter distribution.

At the enrollment stage, the biometric system can have the following failures:

- ***Failure-to-Capture (FTC):*** It is failure of a biometric system to form a proper enrollment template for a subject. For example, in the enrollment phase, the capture process not generating a sample of good quality.
- ***Failure-to-eXtract (FTX):*** This occurs when the generation of biometric template fails due to the feature extraction process. For example, the problems occurring at the feature extraction phase like long processing time for feature extraction exceeding systems time limit.
- ***Failure-to-Enroll (FTE):*** This failure is defined as failure of the biometric system to form a proper enrollment reference for a subject. For

example, the sensor capturing incorrect information from the biometric trait, insufficient quality of the captured biometric data to develop as a template.

- ***Failure-to-Acquire (FTA):*** It is the proportion of verification or identification attempts for which the system fails to capture or locate an image or signal of sufficient quality [iso06a].

At the verification stage, two types of errors can occur. The errors are explained in mathematical terms for clear understanding. If for a subject I, X_I represents the stored template of the subject I present in the database and X_Q represent the acquired input for recognition, then there are two possible hypothesis formulation:

- ***Null hypothesis (H_0):*** X_Q input does not come from the same subject as the template X_1
- ***Alternative hypothesis (H_1):*** X_Q input comes from the same subject as the template X_1

Based on the hypotheses formed, the associated decisions are:
D_0 : person is not who he/she claims to be
D_1 : person is who he/she claims to be

The decision is formed based on the check conditions : if comparison score S(X_Q,X_I) is less than threshold t, then D_0 else D_1. The hypothesis testing generates two types of errors:

Type I : D_1 is decided when H_0 is true
Type II : D_0 is decided when H_1 is true

These two kinds of errors are known as false-match rate (FMR) and false non-match rate (FNMR). It is shown in Figure 2.2. The definition of these terms are as follows :

- ***False Match Rate(FMR):*** This error occurs in case of imposter comparisons when the imposter probe falsely claims a match with the compared non-self template. ISO standards [iso06a] defines *False Accept Rate (FAR)* as the proportion of verification transactions with wrongful claims of identity that are incorrectly confirmed. Mathematically, it could be phrased as:

$$FMR = P(D_1 \mid H_0) \tag{2.3}$$

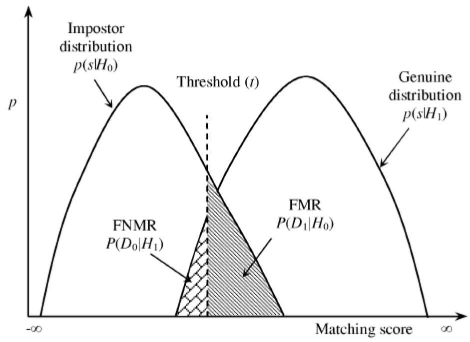


Figure 2.2: Biometric system errors: False non-match rate and False match rate for a given threshold t for imposter and genuine distribution [JRP04].

- ***False Non-match Rate(FNMR):*** This error occurs when the genuine match attempts fails given that the match is between the match of the template of the same characteristic of the same subject supplying the sample. ISO standards [iso06a] defines *False Reject Rate(FRR)* as tproportion of verification transactions with truthful claims of identity that are incorrectly denied. Mathematically, it could be phrased as:

$$FNMR = P(D_0 \mid H_1) \tag{2.4}$$

- ***Equal Error Rate(EER):*** This is defined as the point at which the False acceptence rate (FAR) is equal to False reject rate (FRR). It is visualised through detection error-tradeoff (DET) curve. The accuracy of the biometric system can be evaluated based on the EER. Lower the value of the equal error rate, higher the accuracy of the biometric system.

2.3.1 Graphical representation

The relation between different error rates are represented using ROC and DET curves. In every biometric system, there exists trade off between the FMR and FNMR. In order to make the system more tolerant to noise and variations, threshold t is decreased, then the FMR increases. On the contrary, if t is increased for making the system more secure, FNMR increases. System performance at all operating points or thresholds is shown using ROC curves.

- ***ROC Curve:*** It is used for conducting performance analysis of different systems under same conditions or single system under varied conditions. It has three fold usage:

  - *Matching algorithm performance* plot of measuring the FMR against 1-FNMR for various threshold values t
  - *End-to-End Verification system performance* plot of measuring 1-FRR against FAR
  - *Open-set identification system performance* plot of measuring correct identification rate against false-positive identification rate

- ***Detection Error Tradeoff (DET) Curve:*** In the same way, DET can be used for different purposes.

  - *Matching error rates* plot of measuring the FMR against FNMR for various threshold values t
  - *End-to-End Verification system performance* plot of measuring FRR against FAR
  - *Open-set identification system performance* plot of measuring false negative identification rate against false positive identification rate

CHAPTER 3

Iris Biometrics

John Daugman, pioneer in iris biometric identification work, defines iris biometrics as:

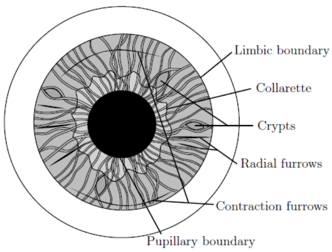
"*Iris biometrics refers to high confidence recognition of a person's identity by mathematical analysis of the random patterns that are visible within the iris of an eye from some distance*" [Dau04b].

It is the uniqueness of the iris which makes it a reliable biometric entity used for identification and recognition of a person. This chapter describes biometric recognition using iris. It is categorised into three sections. The first section gives a brief history of the field of iris biometrics, the next section gives in-depth detail of iris image processing used in iris recognition systems and the final section talks about the practical application areas where iris biometrics has been successfully deployed.

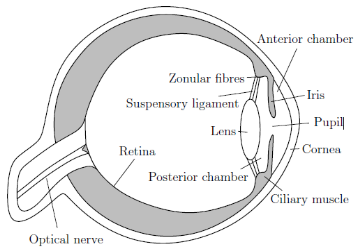
3.1 The iris as a biometric characteristic

The iris is the pigmented circle that rings the dark pupil of the eye. It is the annular region of the eye which controls and directs light to the retina. It is bounded by the sclera (white region of the eye) and pupil (dark region in the center of the eye surrounded by the iris). Tiny muscles attached to the iris dilate (widen) and constricts (narrow) the size of the pupil. Under bright light, constriction of the sphincter muscle (which lies around the edge of the pupil) causes the pupil to constrict.

The color of the iris texture originates from microscopic pigment cells called melanin. The iris consists of rich pattern of ridges, furrows and pigment spots. The iris surface is made up of two regions - the central pupillary zone and the outer ciliary zone. The border between these two regions is called collarette. The anatomy of iris is seen in Figure 3.1.



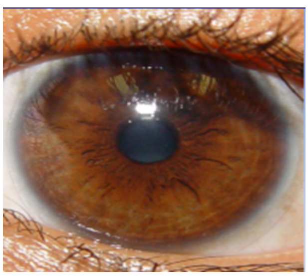
(a) Front view.



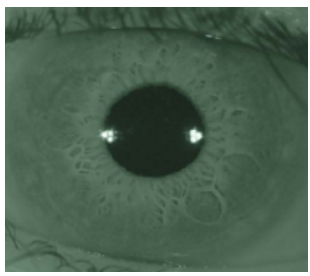
(b) Side view.

Figure 3.1: Anatomy of iris. Taken from [RUW13].

Iris texture patterns are visible differently under different light settings. Figure 3.2 shows the texture pattern under two different light setting. In visible light, the layers of the eye are clearly visible but limited texture information is acquired. On the contrary, under infrared light, melanin muscles reflects most infrared light which gives more texture pattern information.



(a) Iris texture pattern under visible light.



(b) Iris texture pattern under near infrared light.

Figure 3.2: Iris texture pattern in different light settings.

3.2 History of iris biometrics

Iris biometric had its beginning in the 19th century. Since then, it had been an evolving discipline of biometrics. The earliest recorded use of eye for identification was for arrestee in year 1882, by police officer Alphonse Bertillon. In 1886, he suggested human eye (eye color) for biometric recognition[ddi86]. In 1936, Frank Burch proposed using of iris patterns for identification [Dau01]. Iris biometrics papers in Google Scholor from 1990 through 2010 in Computer Science and Mathematics literature is shown in Figure 3.3.

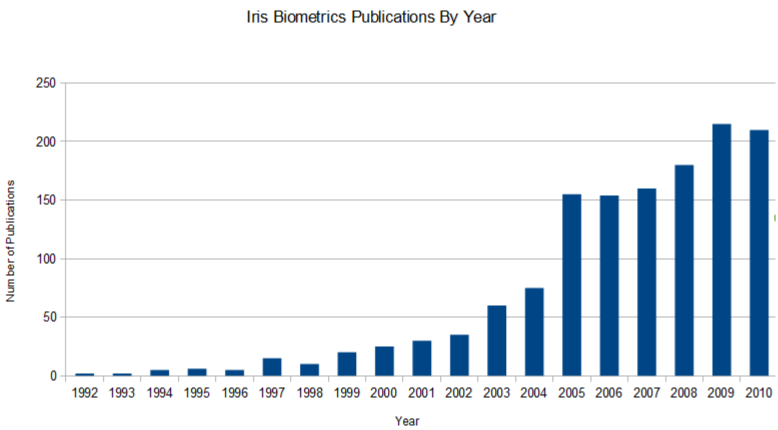


Figure 3.3: Iris biometrics papers in Google Scholar. Taken from [BBFP13].

The origin of automated iris recognition dates back to 1986, when Leonard Flom and Aran Safir filed a patent for the first iris recognition system. In 1987, they recieved the approval for their patent for developing the conceptual design of the automated iris biometric system [FS87]. Following this, in year 1992, Johnston published a report on experiments on the feasibility of iris biometrics conducted at Los Alamos National Laboratory [Joh91]. The purpose of the experimental test conducted by Johnston was to observe iris images of 650 persons aquired each month over 15 month period. The iris pattern seemed unchaged and the specular highlights and reflections were noted for further study. The results of the paper revealed that iris biometrics could potentially be used for biometric identification and verification.

After 10 years from acceptence of Flom and Safir's patent, the most important work in iris biometrics field was contributed by Daugman. He inventing an operational iris recognition system [Dau94]. Since then, Daugman's approach served as a standard reference model in iris biometrics field. It was based on doubly dimensionless coordinates for normalisation, 2D Gabor filters as feature extraction, and Hamming distance (HD) scores as comparator [RUW13]. Following Daugman, Wildes came up with a new approach of iris biometric recognition system which was developed at Sarnoff Labs [Wil97]. In year 1996 and 1998, Wildes *et al.* filed two patents [WAH+96] presenting an acquisition system which allowed a user to self-position his or her eye, automated segmentation method and the normalised spatial correlation for matching.

3.3 Iris recognition system

The iris is a internal protected organ, which is unique. Therefore, it can serve as a living password that one always carries along rather than remembering it. The iris recognition system makes use of the iris texture to identify individuals. The iris recognition system consists of several subsystems. Figure 3.4 shows the overview of each phase of a conventional iris processing chain.

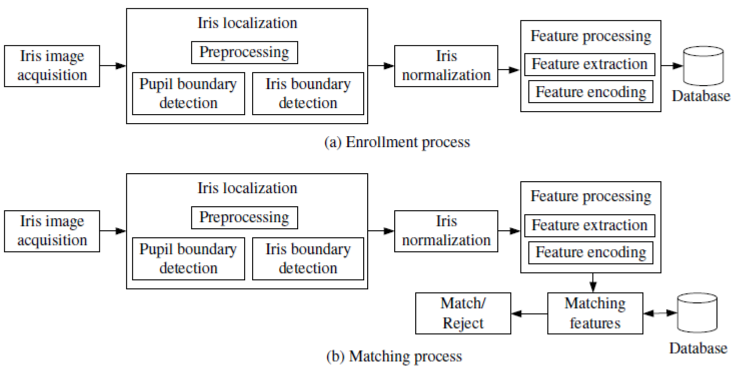


Figure 3.4: Overview of iris processing phases. Taken from [DS08].

It is divided into the following steps:

- ***Iris image acquisition:*** Capturing the image of the eye of the subject to be identified.
- ***Iris localisation:*** Detecting and isolating the iris from the acquired image. This step involves defining the boundary between the iris and pupil portions of the eye and also defining boundary between the iris and sclera.

- ***Iris normalisation:*** Converting the iris region into rectangular area.
- ***Iris feature extraction:*** The iris's distinct features consists of number of characteristics such as freckles, stripe, furrows, crypts etc. Extracting these features from the iris is called feature extraction. The filtered output is mapped to a binary feature vector known as iris code.
- In case of matching process, comparing the code with a previously generated reference iris code to get the similarity measure is done.

3.3.1 Image acquisition

Image acquisition is the phase where the subjects's eye image is captured. There are two types of challenges faced at this stage. First, making the image acquisition phase less intrusive for the subject. For instance, 'Iris on the Move' project [MNH$^+$06] aims at tackling with this problem. Second is to develop quality metrics for acquiring a good quality iris image for verification and identification process.

Iris biometric systems usually have defined and constrained image acquisition conditions. The standard iris image acquisition procedure involves steps where the subject is prompted to position and focus the eye in a particular point where it is possible get sufficient information of the eye and near-infrared illumination around the 700–900 nm range is used to light the face. Daugman suggested that iris should have a diameter of minimum 140 pixels [Dau04b]. In 2005, ISO Iris image standard specification stated to have 200 pixels for the diameter of the iris.

Less intrusive to users

An image capturing system was developed by Sensar Inc. and the David Sarnoff Research Center in year 1996 that would find the eye of the nearest user who is positioned in between 1 and 3 ft from the cameras [HMM$^+$96]. The system consisted of two wide field-of-view cameras and a cross-correlation-based stereo algorithm which would search for the coarse location in the face. The Narrow field-of-view(NFOV) camera affirms the presence of eye and retrieves the image of the eye. Two incandescent lights were placed focusing on the face, which illuminated the face. The eye-finding algorithm locates the eye using the specular reflections from the lights on both sides where the camera is placed. Eventhough Sensar's system gave high performance, but still it required specialised lighting condition for detecting the eye.

Sung *et al.* [SCZY02] came up with the idea of considering the shape and orientation of the eye corner to detect the eye. Fancourt *et al.* [BHG+05] demonstrated that it is possible to acquire images at a distance of up to ten meters that are of sufficient quality to support iris biometrics [BHF08]. But once again, their system imposed constrained conditions.

Abiantun *et al.* [ASK05] tried to increase the vertical range of an acquisition system and on the contrary Narayanswamy and Silveira [NS06] tried to increase the depth-of-field where in camera with a fixed focus could capture an acceptable quality iris image, without having to use the zoom lens. The vertical range increase was acquired by integrating the face detection system on a video stream and a movable camera using the rack-and-pinion system for detecting the largest part of the face.

Most of the recent work focuses on improving and speeding up the entire process. Out of them, the least constrained system is the one described by Matey *et al.* [MNH+06] which acquires the image of the person who walks with a normal speed through access control point, for instance airports. The image acquisition is "based on high-resolution cameras, video synchronised strobed illumination, and specularity based image segmentation" [BHF08]. The system aims to be able to capture useful images in a volume of space 20 cm wide and 10 cm deep, at a distance of approximately 3 m. The height of the capture volume is nominally 20 cm, but can be increased by using additional cameras. The envisioned scenario is that "subjects are moderately cooperative" [BHF08].

The 'Iris on the Move' (IOM) project has developed a system that can capture recognition quality iris image from subject walking with a normal speed through a minimal confining portal [MNH+06]. The schematic view of the system is shown in Figure 3.5.

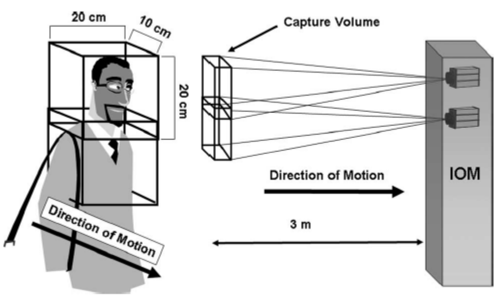


Figure 3.5: Schematic view of the "Iris On The Move" image acquisition system. Taken from [MNH+06].

The two main challenges are the capture volume and the standoff and verification time. The capture volume can be defined as the volume within which the eye must be placed for acquiring a reasonable quality iris image. The standoff distance is the distance between the camera and the subject. Existing systems require reasonably close proximity for capturing a good quality iris image. Verification time refers to the approximate time for capturing sufficient quality iris image from the subject and to perform identification. In some of the cases, the system requires two irises at a time for verification purpose.

Usually, the image acquisition is carried out with close control of a trained operator who helps the the subjects to preposition near the optimum position for the system. Project IOM claims self-positioning often adds 5-10 seconds to verification time for non-habituated subjects. Ongoing projects such as IOM continues to concentrate on methods needed to reduce constraints on the subject so that iris recognition gets easier to use.

Quality metrics of iris image

Iris, having approximately 1 cm in diameter, is relatively small target to capture, which makes acquisition of iris images of sufficient quality a challenging task. ISO/IEC 19794-6, a standard which supports the existing iris recognition algorithms, considers a resolution of more than 200 pixels or more across the iris to be of good quality, of 150–200 pixels across the iris to be of acceptable quality, and of 100–150 pixels to be of marginal quality [iso02].

3.3.2 Iris pre-processing

Iris pre-processing consists of steps to demarcating iris's inner and outer boundaries between the pupil and sclera. Advanced methods include detecting occlusions caused by the upper and lower eyelid boundaries, detecting superimposed eyelashes or reflections from eyeglasses and excluding them.

Iris region segmentation

Iris segmentation process is the process of locating the inner (pupilary) and outer boundary (limbic) of the iris. Daugman and Wildes suggest two different techniques for performing iris localisation. The two approaches are explained below.

- **Daugman's approach**

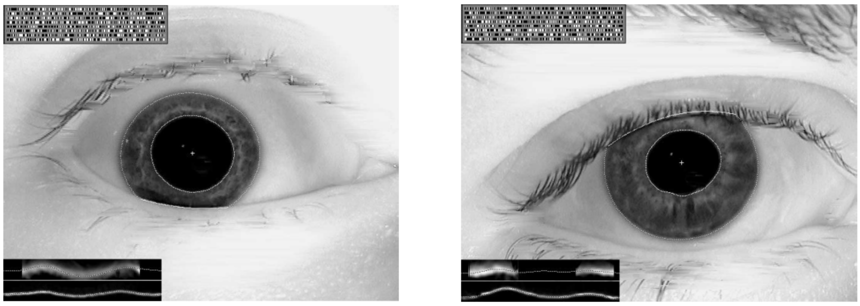
Early work of Daugman assumed the pupillary and limbic boundaries boundaries to be circular. Therefore, the boundary was described with three parameters : radius 'r' and the coordinates of the center of the circle (x_0, y_0). He defined a integro-differential operator for detecting the iris boundary which is,

$$max(r, x_0, y_0) \mid G_\sigma(r) * \frac{\partial}{\partial r} \oint (r, x_0, y_0) \frac{I(x,y)}{2\pi r} \mid \qquad (3.1)$$

where $I(x, y)$ is the image of the eye, * denotes convolution, $G_\sigma(r)$ is known as the smoothing function. The complete operator behaves in effect as a circular edge detector, blurred at a scale set by 0, that searches iteratively for a maximum contour integral derivative with increasing radius at successively finer scales of analysis through the three parameter space of center coordinates and radius (r, x_0, y_0) defining the path of contour integration [Dau93].

However, it has been discovered that often the pupillary and limbic boundaries are not completely circular, which led Daugman to study alternative segmentation techniques for modeling the iris boundaries. His recent contributions to iris biometrics contains more methods to detect the iris inner and outer boundaries with active contours which leads to more embedded coordinate systems and using Fourier-based methods in order to solve iris trignometry and projective geometry for handling off-axis gaze by rotating the eye into orthographic perspective [Dau07]. Daugman came up with a solution for mapping the iris by creating a nonconcentric pseudo-polar coordinate system. He describes the boundaries using 'active contours' which are based on the Fourier series expansion of the contour data. He employed Fourier components whose frequencies are integer multiples of $1/(2\pi)$. The degree of smoothness depends on the number of frequency components chosen. Truncating the discrete Fourier series after a certain number of terms amounts to low-pass filtering the boundary curvature data in the active-contour model [Dau07]. An example is given in Figure 3.6.

The left bottom corner box is the discrete Fourier series approximation of the data. They are known as two "snakes" which represent the curvature map of the inner(lower box) and outer boundary(upper box), with the endpoints joining up at the six o'clock position. If the shape of the pupillary and limbic boundary were circular, these "snakes" would have been just a straight line. The occlusions caused due to eyelids are indicated by the separate splines in the Figure 3.6. The dotted curve which is used to plot the snake is the discrete Fourier series approximation to the corresponding loci of points in the iris image.



(a) Normal iris image. Left bottom corner box containing curvature maps for the inner and outer iris boundaries.

(b) Iris image with occlusion. Left bottom corner box containing curvature maps for the inner and outer iris boundaries.

Figure 3.6: Active contours enhance iris segmentation. Taken from [Dau07].

The estimation method is carried out by computing a Fourier expansion of the N regularly spaced angular samples of radial gradient edge data r_θ for $\theta = 0$ to $\theta =$ N - 1. A set of M discrete Fourier coefficients C_k, for $k =$ 0 to $k = M-1$, is computed from the data sequence r_θ as follows [Dau07]:

$$C_k = \sum_{\theta=0}^{N-1} r_\theta e^{-2\pi ik\theta/N} \tag{3.2}$$

From these M discrete Fourier coefficients, an approximation to the corresponding iris boundary (now without interruptions and at a resolution determined by M) is obtained as the new sequence. R_θ as:

$$R_\theta = \frac{1}{N}\sum_{k=0}^{M-1} C_k e^{2\pi ik\theta/N} \tag{3.3}$$

The stiffness of the active contours is set by M and the number of Fourier components used.

- **Wildes approach**

  Wildes approach varies from Daugman's method in several ways. For image acquisition, Daugman uses standard video camera along with LED-based point light source. On the contrary, Wildes system makes use of low light camera along with a diffuse source and polarisation. For segmentation, Wildes approach computes a binary edge map and then performs Hough transform. In order to produce a template, at multiple scales, Wildes applies a Laplacian of Gaussian filter and then calculates the normalised correlation for similarity measure.

  There are several pros and cons of Daugman and Wildes approach. Daugman's acquisition system is easier and simpler than that of Wildes approach. Wildes makes use of less-intrusive light source for eliminating specular reflections. Wildes approach is considered to be more stable for segmentation process, however, due to the implementation of binary edge abstraction, it makes less use of the data. One of the advantages of Wildes approach is it also contains eyelid detection and localisation. For the matching process, Wildes makes use of more available data.

3.3.3 Iris region normalisation

Once the iris segmentation is done, next step is to describe the features which would make it possible for performing iris comparisons. One of the challenges at this stage is, not all the iris images acquired at the image acquisition stage are of same size. Few of the factors causing this issue are distance from the camera, illumination causing contraction and dilation of iris. This issue is solved by mapping the extracted iris region of interest from the eye into a normalised coordinate system.

The resulting normalised image is a rectangular image with angular and radial coordinates on horizontal and vertical axis respectively. It is shown in Figure 3.7. In such normalised image, the pupillary boundary is considered to be on top of the image and the limbic boundary on the bottom. The left side represents the 0 degrees and the right side of the iris image represents 360 degrees.

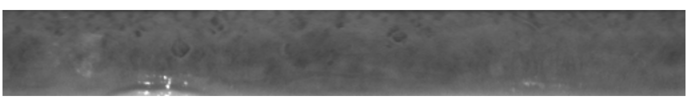


Figure 3.7: Example rectangular texture of the iris region.

3.3.3.1 Daugman's Rubber Sheet Model

This is the traditional model widely used in iris processing at the normalisation phase. This model was devised by Daugman, in which each point within the iris region is remapped to a polar coordinate pair consisting of (r,θ), where 'θ' is the angle between 0 and 360 degrees(or angle [0,2π]) and 'r' a radial coordinate system ranging between 0 and 1. This normalisation is based on the assumption that when pupil dilates and contracts, iris stretches linearly.

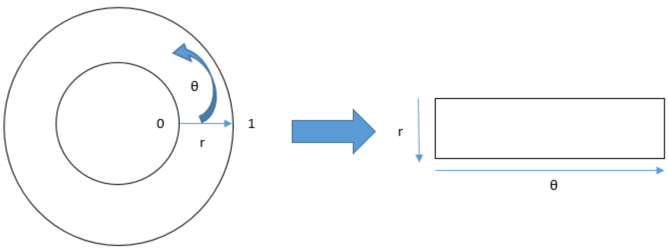


Figure 3.8: Daugman's rubber sheet model.

The modeling of the non-cocentric normalised polar representation from iris region of *(x, y)* Cartesian coordinates is done using the following formula.

$$I(x(r,\theta), y(r,\theta)) \rightarrow I(r,\theta) \tag{3.4}$$

where,

$$x(r,\theta) = (1-r)x_p(\theta) + rx_1(\theta) \tag{3.5}$$

$$y(r,\theta) = (1-r)y_p(\theta) + ry_1(\theta) \tag{3.6}$$

where *(x,y)* are the original Cartesian coordinates, the iris region image is represented as I*(x,y)* and (r,θ) are the corresponding normalised polar coordinates.

3.3.4 Iris feature extraction

For providing accurate recognition, extraction of the most discriminating features from the iris pattern is vital. These significant features of the iris are encoded so that template comparison is made more efficient. Feature extraction deals with extraction the discriminating features and encoding it. Daugman makes use of convolution along with 2-D Gabor filters to extract the texture from the normalised image as the differences in lighting during image acquisition, directly comparing the pixel intensity of two iris images will possibly yield wrong results.

The iris feature extraction is normally divided into three categories: a) phase-based b) zero-crossing representation c) texture analysis based method. Phase based methods are for instance, Gabor wavelet and Log-Gabor wavelet. Zero-crossing representation method is 1-D wavelet, texture analysis based techniques are Laplacian of Gaussian filter and Gaussian-Hermite moments.

3.3.4.1 Feature encoding algorithms

- ***Wavelet Encoding:*** In order to decompose data present in the iris region into components present in different resolutions, wavelets are used. Wavelet filters are applied to the 2-D iris region, one for each resolution. The output from the wavelet application is encoded for providing discriminating iris pattern representation.

- ***Gabor Filters:*** Gabor filters are used for texture representation and discrimination. They provide a conjoint representation of a signal in spacial frequency and space. It is constructed by modulating a sine/cosine wave with a Gaussian. Modulation combination with sine wave gives localisation in space but not of frequency. Signal decomposition is done using a quadrature pair of Gabor filters with a cosine modulated by a Gaussian as the real part and sine modulated with Gaussian as imaginary part. The real filter is also known as even symmetric component and the imaginary filter is known as odd symmetric component. Figure 3.9 shows the odd symmetric and even symmetric 2D-Gabor filters.

  For iris encoding, Daugman makes uses of a 2-D version of Gabor filters [Dau94].

  A 2-D Gabor filter over the an image domain (x, y) is given as:

$$G(x,y) = e^{-\pi[(x-x_0)^2/\alpha^2+(y-y_0)^2/\beta^2}|e^{-2\pi i[u_0(x-x_0)+v_0(y-y_0)]]} \tag{3.7}$$

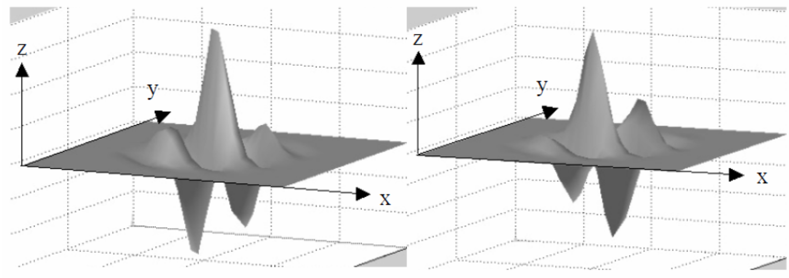


Figure 3.9: A quadrature pair of 2-D Gabor filters left (real component or even symmetric filter characterised by a cosine modulated by a Gaussian right) imaginary component or odd symmetric filter characterised by a sine modulated by a Gaussian. Taken from [M+03].

where the pair (x_0, y_0) represent the position in the image, the (α, β) pair specify the effective width and length, and (u_0, v_0) specify modulation having the spatial frequency.

In order to compress the data, Daugman demodulates the output from the Gabor filters by quantising the phase information into four levels, for each possible quadrant in the complex plane [M+03]. Each of these four levels are represented using two bits of data. Therefore, each pixel in the normalised iris pattern corresponds to two bits of data in the iris template [M+03]. Totally, 2048 bits are calculated for the template. Similarly the masking bits are also calculated for masking out corrupted regions within the iris. Finally, it generates a 256-byte compact template.

- **Log Gabor Filters**

  Problem with the Gabor filter is that whenever the bandwidth is larger than one octave, the bandwidth even symmetric filter is associated with the direct current (DC) component. A zero DC component is possible to obtain for any bandwidth by using a Gabor filter with a Gaussian on a logarithmic scale. This is termed as Log-Gabor filter. The frequency response using Log-Gabor filter is given as:

$$G(f) = exp\frac{-(log(f/f_0))^2}{2(log(\sigma/f_0))^2} \tag{3.8}$$

  where the center frequency is f_0 and σ gives the bandwidth of the filter.

Comparitor

The template generated by the feature encoding process needs a matching metric for calculating the measure of similarity between two iris templates. Since binary iris codes are most commonly used, Hamming distance algorithm is taken into consideration for explanation. A brief description of the algorithm is given below.

- ***Hamming distance***

  Hamming distance is the measure of similar bits between two bit patterns. It is used to find out whether two patterns were generated from same irises or different irises.

  In comparing the bit patterns X and Y, the Hamming distance (HD), is defined as the sum of disagreeing bits (sum of the exclusive-OR between X and Y) over N, the total number of bits in the bit pattern [M+03].

$$HD = 1/N \sum_{j=1}^{N} X_j \oplus Y_j \tag{3.9}$$

  Areas of the iris that are obscured by eyelids, eyelashes, or reflections from eyeglasses, or that have low contrast or a low signal-to-noise ratio, are detected by the image-processing algorithms and prevented from influencing the iris comparisons through bit-wise mask functions [Dau94]. The formula for masking is given as:

$$HD_{raw} = \frac{\| (codeA \oplus codeB) \cap maskA \cap maskB \|}{\| maskA \cap maskB \|} \tag{3.10}$$

3.4 Public deployments

This section covers some significant large public deployments of iris recognition system.

- **India's Unique Identification project - 'Aadhar'**

  This project aims at providing unique identification number (UID) to the citizens which contains the basic demographic and biometric details and stored in the central database known as UIDAI . One of the key challenge for this project is to ensuring the uniqueness of biometrics across a population of 1.2 billion.

The relevant recommendation of the Committee dealing with iris reads as follows [UID] :

"While 10 finger biometric and photographs can ensure de-duplication accuracy higher than 95% depending upon quality of data collection, there may be a need to improve the accuracy and also create higher confidence level in the de-duplication process. Iris biometric technology, as explained above, is an additional emerging technology for which the Committee has defined standards. It is possible to improve de-duplication accuracy by incorporating iris. Accuracy as high as 99% for iris has been achieved using Western data. However, in the absence of empirical Indian data, it is not possible for the Committee to precisely predict the improvement in the accuracy of de-duplication due to the fusion of fingerprint and iris scores. The UIDAI can consider the use of a third biometric in iris, if they feel it is required for the Unique ID project."

- **UK Project IRIS: Iris Recognition Immigration System**

  A "frequent flier" programme that allows enrolled participants to enter the UK from abroad without passport presentation, and without asserting their identity in any other way. Cameras at automated gates operate in identification mode, searching a centralised database exhaustively for any match [UKB].

  IRIS statistics as of 30 July 2007,

  "100,000 frequent travellers have been enrolled, growing by 2,000 per week, and there have been around 500,000 IRIS automated entries since January 2006, with currently around 12,000 IRIS arrivals into the UK per week." [JDS] (– Pat Abrahamsen, UK Home Office).

- **The United Arab Emirates - Iris-based border security system**

  It is deployed at all air, land, and sea-ports in UAE. Total number of 1,190,000 IrisCodes registered in a watch-list. On a normal day atleast 12,000 irises are compared to all on the watch-list which is around 14 billion comparisons/day. Each search takes less than 2 seconds. Statistics reveals that about 9 trillion (9 million-million) comparisons have been done since 2001. Around 150,000 illegal re-entry cases have been found [JDS].

- **U.S. Marines in Iraq: control of points of entry into Fallujah**

  All males of military age entering the city are identified. Overall, some 3,800 iris cameras are in use by U.S. Forces in Iraq [JDS].

- **Takhtabaig Voluntary Repatriation Centre, Pakistan-Afghan border**

  The United Nations High Commission for Refugees (UNHCR) administers cash grants for returnees, using iris identification [JDS].

Chapter 4

Related Work

This chapter discusses ageing in biometrics. It is categorised into two sections. First section gives a brief information about ageing in biometric modalities and the second section describes related work on ageing in iris biometrics.

4.1 Ageing in biometric modalities

For identifying a subject, biometrics-based systems make use of physiological or behavorial characteristics of a subject. However, these characteristics do not stay stable with time, they sometimes undergo suttle or significant changes with the passage of time. Hence, developing biometric applications for long-term use is a challenging task. Ageing effects are usually found in the biometric modalities such as face, signature and fingerprint, etc. These age-related changes in the modalities, raise questions about the reliability of the biometric system. It also affects the accuracy of the computer-automated recognition system.

'Age' is a continuous variable which includes progressive and slow changes. There has been an significant problem limiting the comprehensive age progression study as the age bands used to track ageing issues are not consistent which lead to conflicting results in the literature. Ageing factor is related with three unique characteristics which exerts different challenges in biometrics:

- ***Uncontrollable:*** Ageing is a continuous process. Nothing or no one can advance or delay it. Usually it is slow and irreversible.
- ***Personalised ageing patterns:*** Ageing pattern differs for every individual. It is sometimes caused due to genetic factors as well as many external factors such as lifestyle, health, weather conditions etc.
- ***Temporal data:*** Status of the modal at a particular age will only affect all older modals but not the younger ones. Example, facial ageing at a particular time will have effects on the older faces.

A brief description of ageing in different biometric modalities is given below.

4.1.1 Face

Facial ageing has the most apparent and visible effect. It is usually manifested in the form of complex synergy of textural changes in skin, soft tissues, deep structural components of the face and loss of facial volume. Most of these changes are due to combined effects of decreased tissue elasticity, bone resorption and gravity.

Research has been conducted for simulation of ageing in facial models or images. The different techniques identified are as follows:

- ***Bio-mechanical simulation:*** Few developed methods using this technique are layered facial simulation model for skin ageing with wrinkles [WBM99], flaccidity-deformation approach [BPG06] and analysis approach that applies vectors of ageing to the orbicularis muscle in virtual faces [BJ03].
- ***Anthropometric deformation:*** This method has been developed for both adult ageing [BTN04] and growth and development in young faces [RC06].
- ***Image-based approach:*** This approach makes use of active appearance models (AAM) for estimating growth and development [LTC95], [LTC02] and adult ageing [PRAB06]. 3-D method to the image-based approach is also presented that indicate ageing using 3-D morphable model parameters [OVV+97].

4.1.2 Fingerprint

Fingerprint age determination is highly beneficial in the field of forensics as the morphological, physical, chemical and biochemical transformations in fingerprints provides important information regarding the traces existing at the crime scene. Work on study of evolution of fingerprints ageing process has been conducted based on the factors such as ridge thickness, distance between ridges/valleys, number of pores etc.

Hotz *et al.* did a statistical analyses to determine the fingerprints growth [HGL$^+$]. Scientific paper titled 'Method for fingerprints age determination' [PPP10] considers factors affecting the fingerprint ageing process such as: chemical composition of a fingerprint trace, external influences and background material. On the basis of experience accumulated over a long period of time, standards have been set allowing the determination of the time span during which traces of different chemical compositions stored in various ambient conditions can be effectively used for dactyloscopic purposes [PPP10]. This method also quantifies fingerprint ageing specific to four human blood groups.

4.1.3 Signature

Ageing factor also effects the handwriting pattern. Studies reveal handwriting could be used to categorise 'middle-aged' and 'elderly' individuals [Wal97]. Reports also state that younger adults performed significantly faster handwriting activity than the older adults [DKF93]. As in [Wal97] and [DKF93], it is shown that writer speed decreases (velocity decreases and execution time increases) with increasing age (18 – 30, 31– 55 and 56) in a signing task [EF12]. In particular, a sharp change is noted between the 31 – 55 age groups [EF12].

4.2 Ageing in iris biometrics

'Single enrollment for lifetime' concept was long accepted in the iris biometrics community. But there are few researches conducted who claim that the iris could in fact change over time, which imposes the need of re-enrollment. Iris ageing and changes in the iris texture is a topic of current interest in iris biometric field.

The first experimental results of iris template ageing was published by Baker *et al.* [BBF09]. The experiments conducted by them contained dataset with 26 irises (13 subjects). The images acquired over the time period 2004-2008 using a LG 2200 iris sensor. The authors used the IrisBEE matcher for evaluation. Their experiment involved comparison of the authentic and imposter distributions for two kinds of matches - short term and long term. The short term matches contained comparison of images taken in the same academic semester and long comparison was of comparison with images taken across years. There were no significant changes found in the imposter matches between long term and short term, but the authentic matches for long-term revealed an increase in false non-match rate. They concluded that at the false accept rate of 0.01%, the false reject rate increased by 75% for long-time lapse. But as we can see that the results are presented for small-sized dataset, it is unreliable.

Tome-Gonzalez *et al.* followed experimenting on template ageing by aquiring iris images with one to four week time difference, using an LG 3000 sensor. They used Masek's iris matcher implementation, which revealed a weak overall performance. Their experiment was based on comparison of images of same and different sessions across four weekly sessions. They reported that at a FMR of 0.01%, there was an increase in FNMR of 8.5% to 11.3% for within-session matches and increase in FNMR of 22.4% to 25.8% for across session matches [TGAFOG08].

Fenker and Bowyer conducted experiments on 86 irises (43 subjects), imaged over a two-year period. Iris matchers - IrisBEE and VeriEye were used for analysis. IrisBEE matcher results showed an increase in false reject rate ranging from 157% at a Hamming distance threshold of 0.28 to 305% at 0.34. Whereas VeriEye matcher showed an increase in false reject rate from short to long time-lapse by 195% at a threshold of 0.3 fractional Hamming distance and up to 457% at a Hamming distance threshold of 1 [FB11].

Rankin *et al.* [RSMP12] explored variation in the appearance of iris in three-month intervals for 2 times. Their study involved dataset with high resolution images of 238 irides, captured with a specialised biomicroscope at three and six month intervals, and classified according to texture, measured recognition failure rates resulting from the application of local and non-local feature extraction techniques [RSMP12]. Their results revealed that the recognition failure was detected in 21% of intra-class comparisons cases overall, taken at both three and six month intervals. However, they did not make use of near-infrared illumination rather they used visible-light illumination in the visible band (400–700 nm). Commercial iris biometric systems do not make use of the visisble-light illumination. Therefore these results remain obscure.

The article published by Rankin *et al.* [RSMP12] is under debate. Daugman and Downing pointed out few critics from their paper which is given in [DD13]. In return, Rankin *et al.* address back to the number of assertions pointed out by Daugman and Downing in [RSMP12]. The description of their arguments are given in the table 4.1.

Daugman and Downing	Rankin *et al.*
(a) Performance measure at two point of time (after three months and after six months) revealing 20% failure rate is due to fact that they did not consider at zero interval (initial measure from first time the three months interval was considered) and their multipole algorithm implementation used was terrible at both of these time intervals yielding constantly bad results. This is caused due to **small head tilts or eye cyclotorsion**, giving rise to **unstable segmentation** of the iris boundaries which finally would have caused deformations in coordinates yielding to high dissimilarity scores.	(a) Ensured that **head tilts or eye cyclotorsion were avoided by using a clinical biomicroscope** for image capture. These instruments are used routinely in ophthalmic clinics and are relied upon by surgeons to keep eye and head position still. Additionally **cyclic rotation** (as proposed in Masek's original implementation) was implemented for registration in the matching process which accounts for head-tilt differences between images. The assertion that the study included incorrect segmentation is incorrect. **Accurate segmentation with no coordinate shifts were only taken into account**.
(b) Proof for ageing is provided based on the assumption that a non-zero Hamming distance implies a change in the iris pattern. But as a matter of fact, such scores can commonly arise from **algorithm weaknesses** for instance **unstable coordinate alignments**.	(b) **Unstable coordinate alignments were taken care of by implementing cyclic rotation** (as proposed in Masek's original implementation) technique with shifting of ± 4 rather than ± 5 as used by Masek. Different shifting values were used in experimentation which yielded different comparison scores. But, the minimum of the computed Hamming distances were taken into consideration, ensuring even if there were unstable coordinate alignments, such shifting would correct and obtain the lowest match score. The **cyclic rotation combined with the steps taken to avoid head tilts and cyclotorsion** accounts for avoiding unstable coordinate alignments.

(c) **No photographic evidence** is provided which shows textural changes in iris. The study makes use of illumination in the visible band (400–700 nm), which would detect pigmentation changes.	(c) No claim to have found distinct, visible changes in texture that is possible to be seen in photograph. Reference of **changes is at level of binary code**. Therefore, there is no comparison of image itself, rather comparison is of the bit strings which encode bit strings.
(d) **Freckles, pigment blotches and colour changes can develop over time, but are irrelevant** as the publicly deployed iris recognition system make use of monochrome cameras and infrared illumination in the 700–900 nm band, where melanin is almost completely non-absorbent.	(d) These features are not irrelevant as they are the features associated with the characteristics of pigmentation - clumps of pigment, uneven distribution, variations in density. These features vary between individuals in density, shape and location. Rather **it is important to consider these features with greater details for analysis**.

Table 4.1: Table with Daugman critics and Rankin *et al.*'s reply.

Recent scientific paper on the topic 'The Prediction of Old and Young Subjects from Iris Texture' reveals that it is possible to categorise iris images as representing a young or older person at levels of accuracy statistically significantly greater than random chance [SBF]. This indicates presence of age-related information in the iris texture. But once again, this experiment has been conducted for a small number of dataset with 50 subjects between the ages of 22 and 25 as the "younger" group, and 50 subjects older than the age of 35 as the "older" group [SBF].

The most relevant paper which forms the basis of this thesis work is 'Analysis of template ageing in iris biometrics' [FB12]. The authors of this paper state:

"We find clear and consistent evidence of a template ageing effect that is noticeable at one year and that increases with increasing time lapse"

Major contribution of this thesis deals with finding out how far these claims hold true.

Chapter 5

Experimental Evaluations

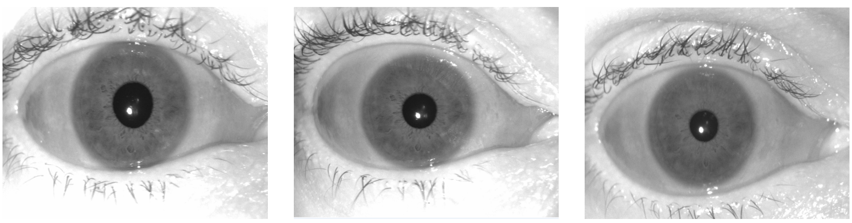
This chapter describes about the experiments undertaken for analysing ageing in iris biometrics. It is divided into four sections. Each section is related to the next part as each phase of the project evolved based on the results obtained in the previous sections. First section of the thesis titled 'Cross Algorithm Analysis' gives detailed information about the experiments and evaluations done on analysing iris ageing effects using several iris processing algorithms, more precisely, feature extraction algorithms. The second section named 'Subject-specific analysis', deals with finding out if the ageing/performance degradation found in the first section is present in specific subjects available in the dataset. The result of the second section led to designing a multi-instance image comparison scheme. Final section deals with the analysis of multi-instance algorithm across multiple feature extraction algorithms.

5.1 Experimental setup

Detailed information of the experimental setup used for conducting each analysis of this research is explained in this section.

5.1.1 Image dataset

The dataset used for the study of iris ageing is from ND-Iris-Template-Aging-2008-2010 [1]. The original dataset which is not available publicly on the internet contains a dataset for a time span of four years which is from 2008 to 2011. The iris images considered for this experiment are of equal subjects throughout two year time span from spring of 2008 through spring of 2010. LG 4000 sensor was used for the image acquisition procedure and same procedure was carried out in the same laboratory for image acquisition process across different years. The iris images were aqcuired from total number of 157 subjects. The subject age ranges from 20 to 64 years old. None of the subjects wore spectacles during data acquisition. The following figures : 5.1(a), 5.1(b), 5.1(c) are examples of irises of the same subject (subject ID - 02463) taken in the year 2008, 2009, and 2010, respectively [ER13].



(a) 2008 (b) 2009 (c) 2010

Figure 5.1: Sample iris images of a single subject of the ND-Iris-Template-Ageing-2008-2010 database acquired in 2008, 2009, and 2010.

Dataset categorisation

The iris image set is divided into two categories, namely, short-time lapse (containing image pairs of two images taken within the same year, with no more than 3 months of time lapse between them) and long-time lapse (image pairs of two images taken in different years for example - 2008 and 2009). The long-time lapse is once again categorised into one-year time lapse (2008-2009 and 2009-2010) and two-year time lapse (2008-2010). The iris image comparison is divided based on genuine (authentic) and imposter comparisons. Genuine comparisons are based on comparison of iris images of same subject at different time lapse, whereas imposter comparison is based on a balanced comparison of iris images of different subjects [ER13]. Detailed information about the number of genuine

[1] ND-Iris-Template-Aging-2008-2010: http://www3.nd.edu/~cvrl/CVRL/

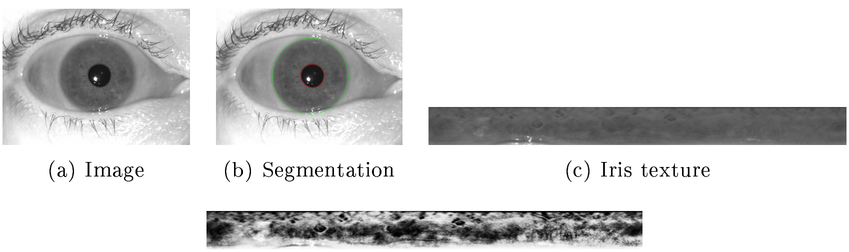
and imposter iris images used for the experiments according to the time lapses is given in table below.

Time-Period	Short/ Long	Subjects	Genuine Comparisons	Imposter Comparisons
2008-2009	short	88	10229	10229
	long	88	26633	26633
2009-2010	short	157	4735	4735
	long	157	10942	10942
2008-2010	short	40	21718	21718
	long	40	45267	45267

Table 5.1: Table summarising the ND-Iris-Template-Ageing-2008-2010 database used for the experimental evaluations.

5.1.2 USIT framework

University of Salzburg Iris Toolkit USIT[2] abbreviated as USIT is used for processing the irises. It is a software package available for Windows and Linux platform for iris recognition which includes algorithms for iris preprocessing, feature extraction and comparison. The iris image undergoes processing which includes iris detection, segmentation, preprocessing and feature extraction. USIT is based on easy-to-use command line tools. The iris recognition tool applies pattern matching techniques to compare two iris images and retrieve a comparison score that reflects their degree of (dis-)similarity. The traditional iris processing chain adopted by this toolkit is depicted in Figure 5.2.



(a) Image (b) Segmentation (c) Iris texture

(d) Enhanced texture

Figure 5.2: Conventional iris-biometric pre-processing chain for a sample iris image.

[2]USIT: University of Salzburg Iris Toolkit: http://www.wavelab.at/sources/

The list of various iris processing algorithms employed by USIT and used for analysis are given below.

Iris Processing Phase	Algorithm
Face/Face-part Detection	Gaussian Face and Face-part Classifier Fusion
Iris Segmentation	Contrast-adjusted Hough Transform (CAHT) Weighted Adaptive Hough and Ellipsopolar Transform (WAHET)
Iris Feature Extraction	1D-LogGabor Feature Extraction Algorithm of Ma *et al.* (re-implementation) Algorithm of Ko *et al.* (re-implementation) Algorithm of Rathgeb and Uhl Context-based Feature Extraction Algorithm of Monro *et al.* (re-implementation)
Iris Biometric Comparators	Hamming Distance-based Comparator Context-based Comparator Comparator of Ko *et al.* (re-implementation) Comparator of Monro *et al.* (re-implementation)

Table 5.2: Table summarising iris processing algorithms present in USIT.

At pre-processing, the segmentation algorithm proposed in [UW12] has been utilised. The two-stage segmentation algorithm employs a weighted adaptive Hough transform iteratively refining a region of interest to find an initial center point, which is utilised to polar transform the image and extract polar and limbic boundary curves one after another from an (ellipso-)polar representation. Since the described segmantation algorithm does not obtain correct segmentation results for the entire dataset another segmentation technique based on Canny edge detection and Hough circle detection is applied. Once the pupil and iris circles are localised, the area between them is transformed to a normalised rectangular texture of 512×64 pixel, according to Daugman's rubbersheet approach [Dau04a]. Finally, lighting across the texture is normalised using block-wise brightness estimation. Preprocessing for a sample image is depicted in Fig. 5.2.

Feature extraction algorithms

As previously mentioned, six different feature extraction techniques are employed in USIT, where the majority of these algorithms are custom re-implementations of existing approaches:

- ***1D-LogGabor Feature Extraction*** **[M$^+$03]:** this algorithm performs a row-wise application of a 1D-LogGabor filter and encodes phase angles of the resulting complex values for each pixel.

- ***Algorithm of Ma et al.*** **[MTWZ04]:** a dyadic wavelet transform is performed on signals obtained from the texture stripes, and two fixed sub-bands are selected from each transform and all local minima and maxima are encoded to generate iris-codes.
- ***Algorithm of Monro et al.*****[MRZ07]:** based on differences of discrete cosine transform (DCT) coefficients of overlapped angular patches from normalised iris images iris-codes are extracted.
- ***Algorithm of Ko et al.*** **[KGYC07]:** this algorithm estimates cumulative sums over grayscale values obtained from neighboring pixel blocks in order to generate a binary feature vector.
- ***Algorithm of Rathgeb and Uhl*** **[RU10]:** intensity variations within iris texture stripes are analysed and encoded in order to generate iris-codes.
- ***Context-based Feature Extraction*** **[RU09]:** a trivial feature extractor which obtains a binary code from quantising pixel blocks while the according comparator aims at detecting clusters of matching bits.

These algorithms extract a great diversity of differnt features of iris textures and, thus, form an adequate basis for the investigation whether reported perfromance drops over time are algorithm-specific.

USIT usage

USIT was used in windows operating system is the command line. The following commands were used for processing of the iris images from the dataset.

Listing 5.1: Iris Segmentation

```
#Weighted Adaptive Hough and Ellipsopolar Transform
> ./wahet -i image -o texture -m mask -s width height
#Contrast-adjusted Hough Transform
> ./caht -i image -o texture -m mask -s width height
```

Listing 5.2: Iris Feature Extraction

```
#1D-LogGabor Feature Extraction
> ./lg -i iris_texture -o iris_code
#Algorithm of Ma \textit{et al.} (re-implementation)
```

```
> ./qsw -i iris_texture -o iris_code
#Algorithm of Ko \textit{et al.} (re-implementation)
> ./ko -i iris_texture -o iris_code
#Algorithm of Rathgeb and Uhl
> ./cr -i iris_texture -o iris_code
#Context-based Feature Extraction
> ./cb -i iris_texture -o iris_code
#Algorithm of Monro \textit{et al.} (re-implementation)
> ./dct -i iris_texture -o iris_code
```

Listing 5.3: Iris Code Comparison

```
#Hamming Distance-based Comparator
> ./hd -i iris_code_1 iris_code_2 -s shift min
shift max -m mask_file_1 mask_file_2 -a algorithm -n
from_bit to_bit -o
#Context-based Comparator
> ./cbc -i iris_code_1 iris_code_2 -o log_file
#Comparator of Ko \textit{et al.}
> ./koc -i iris_code_1 iris_code_2 -o log_file
#Comparator of Monro \textit{et al.}
> ./dctc -i iris_code_1 iris_code_2 -o log_file
```

5.1.3 Statistical calculations

The performance metrics is estimated in terms of FNMR at a targeted FMR. In accordance to the ISO/IEC IS 19795-1 [ISO06b], the FNMR of a biometric system defines the proportion of genuine attempt samples falsely declared not to match the template of the same characteristic from the same user supplying the sample. By analogy, the FMR defines the proportion of zero-effort imposter attempt samples falsely declared to match the compared non-self template.

In order to determine the presence of template ageing effects, various statistical experiments were carried out on the dataset. The first step was to determine the density distributions of the similarity scores for genuine and imposter comparison for each (sub)dataset for each algorithm in USIT. For each dataset, equal number of imposter and genuine image comparisons were considered. Imposter distributions were obtained by randomly comparing pairs of iris-codes extracted form iris images of two different subjects. For every long and short category of the dataset, the FNMR and FMR were computed. Receiver operating characteristic graphs were drawn for each of the dataset using six iris processing algorithms and the value of 1-FNMR at a FMR of 0.01% was calculated. The computation of FNMR and FMR was done by applying the the following formula on the genuine and imposter comparison score obtained for each dataset [ER13].

For computing the verification rates namely FNMR, FMR and genuine match rate the following formulas are used:
Φ_g is the set of all genuine comparison score
Φ_i is the set of all imposter comparison score
$\Phi_g(t)$ is the set of all genuine scores s > t
$\Phi_i(t)$ is the set of all imposter scores s > t

$$GMR(t) = \frac{\parallel \Phi_g(t) \parallel}{\parallel \Phi_g \parallel} \tag{5.1}$$

$$FMR(t) = \frac{\parallel \Phi_i(t) \parallel}{\parallel \Phi_i \parallel} \tag{5.2}$$

$$FNMR(t) = 1 - GMR(t) \tag{5.3}$$

5.2 Cross-algorithm analysis

Purpose

In contrast to majority of existing iris ageing investigations, which restrict to applying a single feature extraction algorithm using the software IrisBEE and VeriEye SDK, the iris ageing effects is explored using six different iris recognition algorithms present in the iris processing software USIT. Based on the similarity scores generated by different algorithms available in USIT, the match and non-match distributions for genuine and imposter image comparisons for short and long time span image comparisons is acquired. Two types of performances are tested using this method. Firstly, the performance of short and long time lapse comparison tests of one and two year time lapse dataset obtaining receiver operating characteristic curves for each algorithm. Second, performance of each of the feature-extraction algorithms against each other is evaluated.

The density distribution for each of the short and long time comparisons for all the other algorithms were acquired in similar manner. These density distribution histogram graphs are provided in Chapter 6. Once the density distribution of the similarity scores were acquired, next step was to find out the FNMR and FMR. The graphs of the FNMR and FMR versus the similarity score were drawn. The

FNMR and FMR are calculated using the following formulas:

$$FNMR(t) = \int_0^t \Phi_g(s)\, ds \tag{5.4}$$

$$FMR(t) = \int_t^1 \Phi_i(s)\, ds \tag{5.5}$$

where $\Phi_g(s)$ is the probability density distribution function of genuine comparison score s and $\Phi_i(s)$ is the probability density distribution function of imposter comparison score s for threshold t.

5.3 Subject-Specific analysis

Purpose

Having identified the performance degradation in FNMR across one-year and two-year time span iris image comparisons, the next logical step was to analyse if the performance degradation that is termed as 'ageing' factor is specific to few subjects present in the database. This forms the basis of the second phase of the thesis.

Evaluation

The resulting files from the previous section (cross-algorithm analysis) are text files containing genuine and imposter comparison image names and scores. These files contain the names of iris images which are being compared in two-column format and associated comparison score. The first five digits of the image name represent the subject number and the digits following the 'd' represent a unique image from that subject. The format of the final file is given in table below.

Image 1	Image 2	Comparison Score
02463d1890	02463d1908	0.24556
02463d1890	02463d1910	0.26433
02463d1890	02463d1912	0.28635

Table 5.3: Example text file containing comparison scores.

The final files are given in table below.

Genuine	Imposter
Matches 2008 2009 short	Imposter matches 2008 2009 short
Matches 2008 2009 long	Imposter matches 2008 2009 long
Matches 2009 2010 short	Imposter matches 2008 2010 short
Matches 2009 2010 long	Imposter matches 2008 2010 long
Matches 2008 2010 short	Imposter matches 2009 2010 short
Matches 2008 2010 long	Imposter matches 2009 2010 long

Table 5.4: Example text file containing the comparison scores.

5.4 Multi-instance image comparisons

Purpose

Due to the results yielded by the subject-specific analysis, multi-instance image comparison analysis scheme was devised for further investigation. As the subject-specific results did not show a linear trend for each subject in the dataset across different feature extraction algorithms, it was difficult to conclude that the so-called 'ageing' or performance deterioration is subject-specific. Therefore, deeper look into the files containing the original comparison scores revealed non-homogeneous scores from the score set of each image of few subject. Every single image of the subject is compared with a number of other images of the same subject present in the dataset yielding a set of comparison scores of the image, which is termed as 'comparison score set'. Each subject had an average of 300 genuine comparisons. The example score set of a particular image of the subject is shown in the table below.

First Image	Second Image	Comparison Score
02463d1891	02463d2783	0.436426
02463d1891	02463d2785	0.252051
02463d1891	02463d2787	0.264160
02463d1891	02463d2845	0.445508
02463d1891	02463d2847	0.441113
02463d1891	02463d2877	0.283691

Table 5.5: Example image comparison score set of subject ID (02463), image ID (02463d1891) of year 2008-2010 Long.

The important point to notice from the table above is that the comparison score set of the image (02463) contains a score of 0.252051, which is considered to be good score as it is closer to genuine score level and it also contains score of 0.445508 which is closer to imposter score. This is a critical problem. In order to handle this, multi-instance image comparison technique was developed. The goal for developing this technique is to design a combination scheme of comparison scores, without excluding the bad scores and use this for further analysis to check if even with this combination scheme the performance degradation is same or not.

For this purpose, standard iris image quality assessment could have been used for filtering out the good images. But the reason for adopting this method is that there is a possibility that the criteria set for classifying good and bad images can be biased which might exclude image scores having 'ageing' effect. Therefore, multi-instance image comparison algorithm had to be developed which would consider each score from the comparison score set in an unbiased manner.

5.4.1 Algorithm

Multi-instance image comparison algorithm was mainly designed to check if the performance level seen in the plots from cross-algorithm analysis, remains the same for a subset of comparison score set taken from the original set or not. For classifying the subset, the most commonly used and standard normalisation technique like the arithmetic mean, median, best score (minimum score) was chosen. The reason why the standard methods are chosen for filtering is because of the fact that for an arbitrary distribution, functions like mean, meadian are reasonable estimates of location and scale. The algorithm of multi-instance image comparison is given on the next page. This algorithm was implemented using MySQL. MySQL was chosen as large dataset had to be processed. The implementation code for this algorithm is given in the appendix A.1.3.

Algorithm 1 Multi-instance image comparison analysis algorithm

Input and Output

Input *Comparison score set of iris image from the comparison score file*
Output *Combination of subset of the comparison score set*

Steps

```
procedure MultiInstanceImageAnalysis(comparison score file)
    while Image ID is present in the comparison score file do
        Read scores of the current Image ID
        Calculate count - number of scores available for the current image ID
        if count = 2 then
            BestOfTwo ← minimum of two scores
            MeanOfTwo ← mean of two scores
        else if count = 3 then
            BestOfThree ← minimum of three scores
            MeanOfThree ← mean of three scores
            MedianOfThree ← median of three scores
            MeanOfBestOfThree ← mean of two best scores
                                                        ▷ Subset processing
            Set i=0
            Set input parameter as 2
            Calculate number of subsets - (length + 1)/inputparameter
            for i<=number of subset do
                Create random subset of scores of length 2
                BestOfTwo ← minimum of two scores
                MeanOfTwo ← mean of two scores
            end for
```

Algorithm 1 Multi-instance image comparison analysis algorithm (continued)

```
    else if count = 4 then
                                                    ▷ Subset processing
        Set i=0
        Set input parameter as 2
        Calculate number of subsets - (length + 1)/inputparameter
        for i<=number of subset do
            Create random subset of scores of length 2
            BestOfTwo ← minimum of two scores
            MeanOfTwo ← mean of two scores
        end for

        Set j=0
        Set input parameter as 3
        Calculate number of subsets - (length + 1)/inputparameter
        for j<=number of subset do
            Create random subset of scores of length 3
            BestOfThree ← minimum of three scores
            MeanOfThree ← mean of three scores
            MedianOfThree ← median of three scores
            MeanOfBestOfThree ← mean of two best scores
        end for
    else if count = 5 then
        BestOfFive ← minimum of five scores
        MeanOfFive ← mean of five scores
        MedianOfFive ← median of five scores
        MeanOfBestOfFive ← mean of three best scores
                                                    ▷ Subset processing
        Set i=0
        Set input parameter as 2
        Calculate number of subsets - (length + 1)/inputparameter
        for i<=number of subset do
            Create random subset of scores of length 2
            BestOfTwo ← minimum of two scores
            MeanOfTwo ← mean of two scores
        end for

        Set j=0
        Set input parameter as 3
        Calculate number of subsets - (length + 1)/inputparameter
        for j<=number of subset do
            Create random subset of scores of length 3
            BestOfThree ← minimum of three scores
            MeanOfThree ← mean of three scores
            MedianOfThree ← median of three scores
            MeanOfBestOfThree ← mean of two best scores
        end for
```

Algorithm 1 Multi-instance image comparison analysis algorithm(continued)

```
        else if count > 5 then
                                                            ▷ Subset processing
            Set i=0
            Set input parameter as 2
            Calculate number of subsets - (length + 1)/inputparameter
            for i<=number of subset do
                Create random subset of scores of length 2
                BestOfTwo ← minimum of two scores
                MeanOfTwo ← mean of two scores
            end for

            Set j=0
            Set input parameter as 3
            Calculate number of subsets - (length + 1)/inputparameter
            for j<=number of subset do
                Create random subset of scores of length 3
                BestOfThree ← minimum of three scores
                MeanOfThree ← mean of three scores
                MedianOfThree ← median of three scores
                MeanOfBestOfThree ← mean of two best scores
            end for

            Set k=0
            Set input parameter as 5
            Calculate number of subsets - (length + 1)/inputparameter
            for k<=number of subset do
                Create random subset of scores of length 5
                BestOfFive ← minimum of five scores
                MeanOfFive ← mean of five scores
                MedianOfFive ← median of five scores
                MeanOfBestOfFive ← mean of three best scores
            end for
        end if
    end while
end procedure
```

5.4.2 Cross-algorithm comparisons

Cross-algorithm comparison analysis is carried out in order to check the results from multi-instance image analysis, across different feature-extraction algorithms. The resulting files from multi-instance image analysis contained these following final files with comparison score set in them.

Final files
Best of Two
Mean of Two
Best of Three
Mean of Three
Median of Three
Mean of Best Two of Three
Best of Five
Mean of Five
Median of Five
Mean of Best Three of Five

Table 5.6: Final files containing the comparison scores.

These files were generated for the three best performing algorithms chosen from the performance analysis of algorithms given in Cross-algorithm analysis. The three best performing algorithms are 1D- LogGabor, Ma *et al.* and Monro *et al.* algorithm.

5.4.3 Evaluation

In order to analyse it across different feature extraction algorithms, the following steps were carried out for each of these algorithms.

- Apply multi-instance image analysis algorithm in the previous section to the genuine and imposter comparison score files obtained from Cross-algorithm analysis for each of the short and long comparison files - 2008-2009 long, 2008-2009 short, 2009-2010 long, 2009-2010 short, 2008-2010 long, 2008-2010 short.

- Generate the density distribution histogram from the data obtained from resulting files of the previous step - for instance best of two, mean of three etc.

- Calculate the FNMR for the genuine and FMR for the imposter comparisons from the resulting files from the first step.
- Plot the 1-FNMR(%) versus FMR(%) graphs for each of these files.
- Compare the obtained curve with the original curve from Cross-algorithm analysis.

Chapter 6

Experimental Results

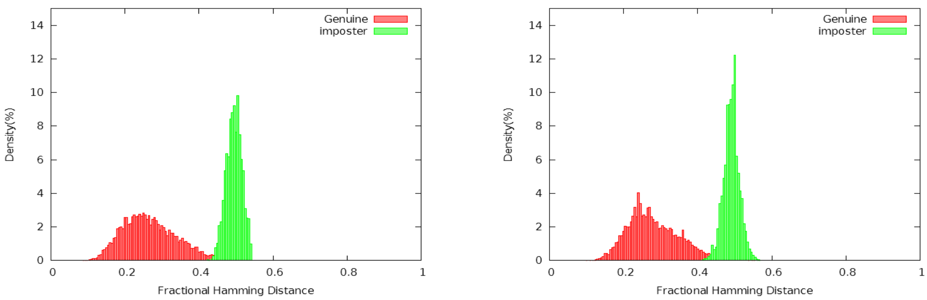
The previous chapter described the experimental evaluations carried out at each phase of this thesis. In this chapter, experimental results of each analysis are presented. It is divided into three sections, namely, cross-algorithm analysis, subject-specific analysis and multi-instance image comparison analysis.

6.1 Cross-algorithm analysis

The cross-algorithm analysis consists of three experimental steps undertaken during the thesis work. The first step was to determine the density distribution of the genuine and imposter comparisons, calculating the FNMR and FMR, plotting ROC curves for each subset of the dataset in order to compare the short and long comparisons. Followed by conducting a cross-algorithm performance analysis using the ROC curves.

6.1.1 Density distribution

This step consists of determining the density distributions of the comparison scores for genuine and imposter comparison for each short and long dataset using each algorithm present in USIT. For each short and long dataset, an equal number of imposter and genuine image comparisons were considered. Imposter distributions were obtained by randomly comparing the iris images of two different subjects. The sample density distribution histogram for 1D-LogGabor algorithm for comparison of 2008-2009 short and long time lapses is shown in Figure 6.1(a) and 6.1(b) [ER13].



(a) Density distribution histogram of similarity scores for 1D-LogGabor 2008-2009 short comparisons.

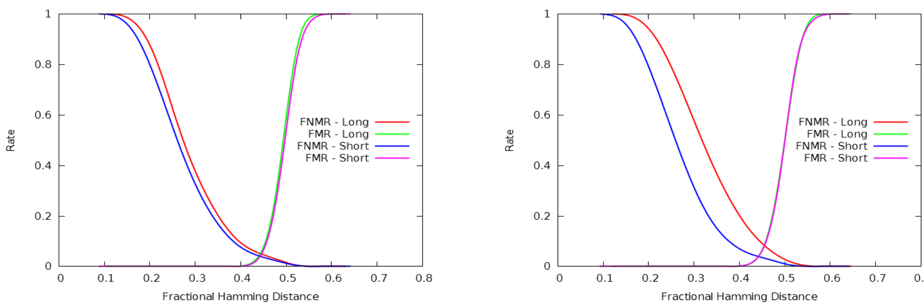
(b) Density distribution histogram of similarity scores for 1D LogGabor 2008-2009 long comparisons.

Figure 6.1: Graph Density Distribution Histogram of 1D LogGabor.

The above density distribution for 1D-LogGabor algorithm shows that the comparison scores of the genuine comparisons are spread across 0.1 to 0.43 fractional Hamming distance and the imposter distribution is from 0.41 to almost upto 0.52 fractional hamming distance. Density distribution of algorithm Ma *et al.* [A.1.1] reveals that the genuine comparison scores lie in the range of 0.15 to 0.45 and the imposter ranges from 0.45 to 0.5. For algorithm of Ko *et al.*, genuine comparison score is from 0.0 to around 0.4 fractional hamming distance and imposter from 0.15 to 0.48 fractional Hamming distance. Algorithm of Rathegb and Uhl genuine comparison is from 0.2 to 0.45 and imposter 0.35-0.45. Context-based algorithm reveals the imposter comparison score spread from 0.18 to 1 fractional Hamming distance. Algorithm of Monro *et al.* shows the genuine comparison score's density distribution to be from 0.22 to 0.45 and imposter scores in the range of 0.42 to 0.45 fractional Hamming distance.
The density distribution for each of the short and long time comparisons for all the other algorithms were acquired in similar manner. These density distribution histogram graphs are provided in A.1.1.

6.1.2 Performance degradation across different algorithms

Once the density distribution of the similarity scores were acquired, next step was to estimate the FNMR of the genuine comparison scores and FMR for imposter comparison scores. The graphs of the FNMR and FMR versus the fractional Hamming distance for 1D-LogGabor are shown in Figure 6.2 [ER13].



(a) 1D LogGabor: FNMR and FMR versus Fractional Hamming Distance Short and Long comparisons 2008-2009.

(b) 1D LogGabor: FNMR and FMR versus Fractional Hamming Distance Short and Long comparisons 2008-2010.

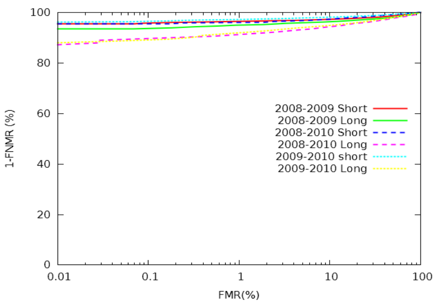
Figure 6.2: Graph FNMR/FMR versus similarity score for 1D LogGabor.

The graphs shown in Figure 6.2 are the example graphs which depict the comparison of the short- time lapse (2008-2009 - one year time lapse) and long-time lapse (2008-2010 - two year time lapse). As we can see from Figure 6.2, the gap between FNMR of short and long comparisons is increased for year 2008-2010 [ER13]. In the similar manner, for the rest of the feature extraction algorithms, this gap is visible. It is given in A.1.2.

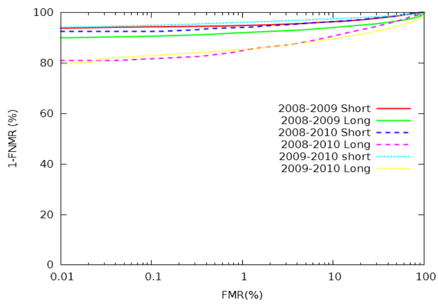
6.1.3 Analysis of short and long comparisons

To evaluate the performance of iris image processing algorithms on the datasets, ROC curve is applied and further used for analyzing individual performance level for short and long term iris image comparisons.

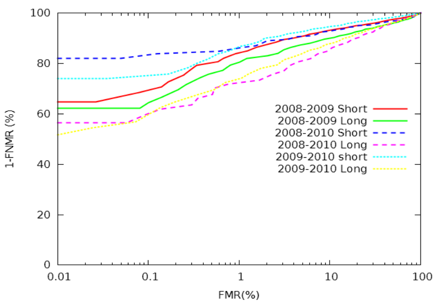
As we can see from the ROC graphs shown in 6.3, the short time lapse comparison outperforms the long time lapse comparisons, the two year time lapse (2008-2010) comparisons have the lowest performance level compared to one-year time lapse comparisons (2008-2009 and 2009-2010). If no ageing factor was involved, the values for short and long comparisons would have been expected



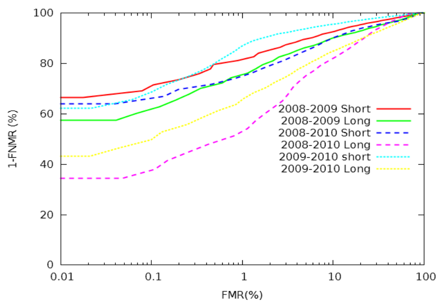
(a) Receiver operating characteristic graph for short and long comparisons for 1D Log-Gabor algorithm.



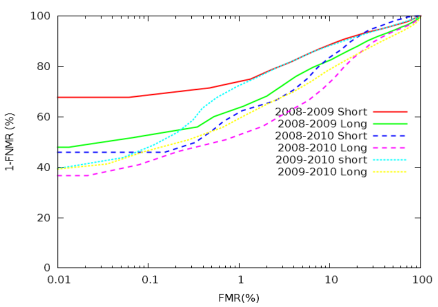
(b) Receiver operating characteristic graph for short and long comparisons for algorithm Ma *et al.*



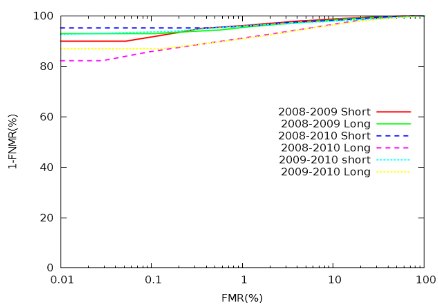
(c) Receiver operating characteristic graph for short and long comparisons for algorithm Ko *et al.*



(d) Receiver operating characteristic graph for short and long comparisons for Context-based algorithm.



(e) Receiver operating characteristic graph for short and long comparisons for algorithm Rathgeb and Uhl.



(f) Receiver operating characteristics graph for short and long comparisons for algorithm Monro *et al.*

Figure 6.3: ROC for short and long time comparison for each algorithm.

to remain equal. It is important to note that the applied algorithms extract a great diversity of biometric features. However, we can clearly see that in all the feature extraction algorithms, the short comparisons outperform the long comparisons by yielding a higher 1-FNMR% value at FMR of 0.01%. This indicates iris ageing is not algorithm-specific, it is visible in all the algorithms [ER13]. But it is rather quick to conclude that the performance degradation seen here is due to ageing factors. Therefore, further investigation such as subject-specific analysis, multi-instance image comparison analysis and visual examination was done on the dataset to examine the ageing effects.

6.1.4 Performance evaluation of feature extraction algorithms

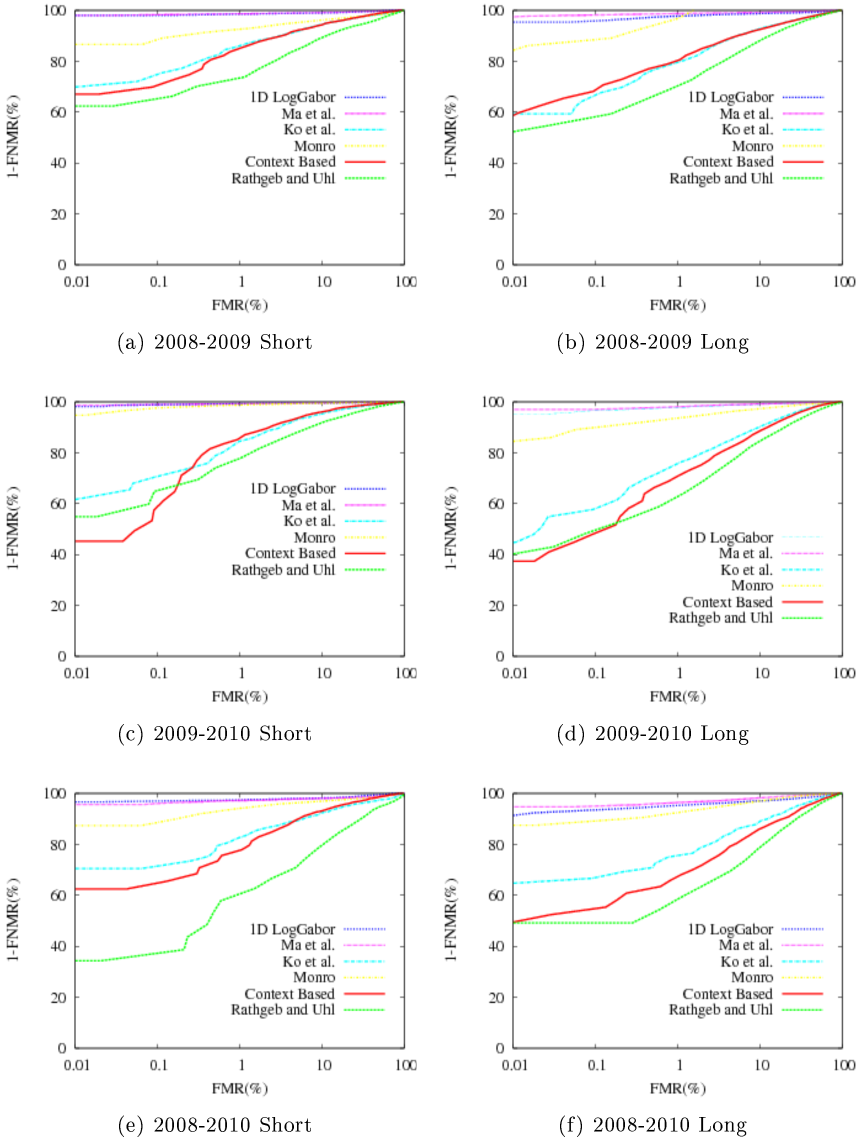
Algorithm Type	**2008-2009**		**2009-2010**		**2008-2010**	
	Short	**Long**	**Short**	**Long**	**Short**	**Long**
1D LogGabor	98.00	95.41	98.21	94.47	96.18	94.45
Monro *et al.*	86.06	86.08	94.79	87.45	84.73	89.10
Ma *et al.*	98.00	97.82	98.67	96.74	94.94	96.33
Rathgeb and Uhl	62.46	54.71	54.90	40.16	34.36	48.22
Ko *et al.*	72.01	59.58	62.83	44.97	68.14	64.14
Context-based	67.01	59.28	45.28	37.38	62.56	48.14

Table 6.1: Verification results in terms of 1-FNMR (in %) at FMRs of 0.01% for 2008-2009, 2009-2010, and 2008-2010 for short and long time lapse comparisons.

The accurate 1-FNMR values at 0.01% FMR are given in table above. Furthermore, it is important to notice that the performance of algorithms which apply coarse quantisation on rather large pixel blocks, e.g. Context-based feature extraction algorithm, varies in 2008-2009, 2009-2010 and 2008-2010 graphs. Overall, algorithms - Ma *et al.*, the 1D-LogGabor feature extraction, Monro *et al.*, and Ko *et al.* have revealed similar behavior in all the short and long comparisons, which enables us to draw a conclusion that algorithm Ma *et al.* performs better than all the other algorithms, followed by the 1D LogGabor feature extraction and the algorithm of Monro *et al.*

6.1.5 Summary

The ROC curve was drawn to analyze the performance of each of the feature extraction algorithms used which is shown in the 6.4. Figure 6.4 shows one and



(a) 2008-2009 Short
(b) 2008-2009 Long
(c) 2009-2010 Short
(d) 2009-2010 Long
(e) 2008-2010 Short
(f) 2008-2010 Long

Figure 6.4: Receiver operating characteristic curves for different iris recognition algorithms for short time lapses 2008-2009, 2009-2010, and 2008-2010.

two year time lapse comparisons for short and long time comparisons for all the six algorithms. It is obvious that the higher the 1-FNMR value at 0.01% FMR, the better the biometric performance level. We can see from Figure 6.4 that the algorithm of Ma *et al.* and the 1D LogGabor feature extraction achieve the highest 1-FNMR value. The curves for 1D LogGabor and Ma *et al.* almost coincide for all the three short time lapse graphs. But looking at the long time lapse graphs, it is clear that the biometric performance of algorithm Ma *et al.* is better than all the other.

Another assertion is, the performance degradation is seen across different feature extraction algorithms. Therefore, this performance degradation which is termed as 'ageing' is not algorithm specific.

6.2 Subject-specific analysis

The subject-specific analysis was done for subjects present in 2008-2010 dataset. The reason for choosing subjects from 2008-2010 dataset is because if ageing was present, this subset would be the most affected as the comparison is for two-year time span, hence the oldest in the dataset. This dataset was analyzed across different feature extraction algorithms.

- 1D-LogGabor algorithm

  As we can see from the 6.5 plot, the difference in mean of short and long comparisons indicates that there is a change in comparison score in two years. Higher mean comparison score for long can possibly suggest presence of 'ageing' in iris texture. But, if this was due to ageing effects, the same change in means for each subject should be seen across all the other feature extraction algorithms. Rather, we see a different trend in the mean change across different algorithms.

  The results obtained from this graph could be categorized into three categories (refer Figure.6.5):

  - *Large difference in short and long mean*: PersonID 1 and 26 reveal a large difference in the mean of the comparison scores.
  - *Mean of short outperforming the long*: PersonID 3, 10, 29, 39 show a higher mean for short comparison than that of long.
  - *Mean of short and long at the same level*: PersonID 18, 37 show that that the mean of short and long almost coincide. Does this mean that there is no ageing effect for these subjects? This might also suggest that iris ageing varies.

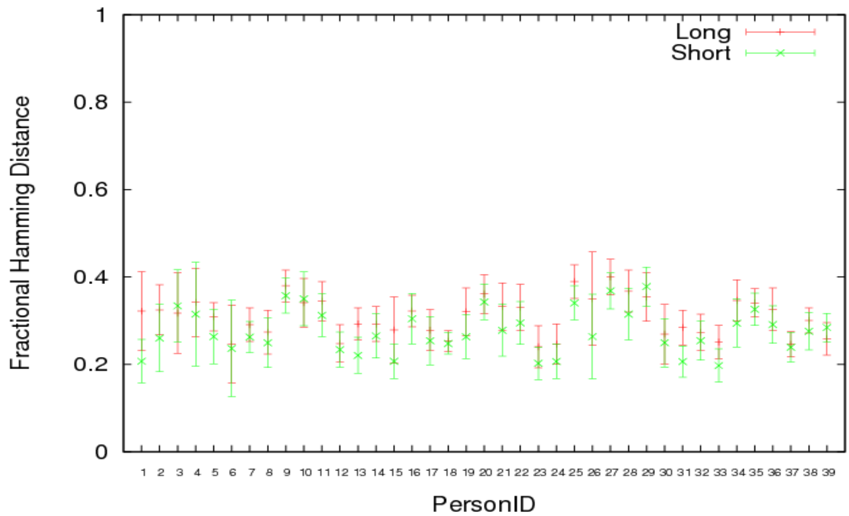


Figure 6.5: Subject-Specific analysis of 2008-2010 dataset using 1D-LogGabor algorithm.

- Algorithm of Ma *et al.*

  Algorithm of Ma *et al.* reveals a different trend than that of 1D-LogGabor. For example, PersonID 3 and 4 are different in Ma *et al.* than that of 1D-LogGabor. For PersonID 37, the mean of short and long do not coincide. PersonID 1 and 26 still have large mean difference for long and short comparisons.

- Algorithm of Monro *et al.*

  Refer Figure.6.7. For PersonID 3, the mean of short and long is equal, while in algorithm of Ma *et al.* and 1D-LogGabor mean of short is greater than long. The trend for 3 and 4 is almost similar to that of 1D-LogGabor. For PersonID 17, the mean is equal for short and long.

- Algorithm of Ko *et al.*

  Refer Figure.6.8. Surprisingly, PersonID 3, 4, 10, 11, 25, 26, 27, 28, 29 show a high mean, in the range of 0.28 to 0.3 fractional Hamming distance. This was only seen in PersonID 1 and 26 in the other algorithms. For subjects 3 and 10, short outperform long as seen in 1D-LogGabor and Ma *et al.*.

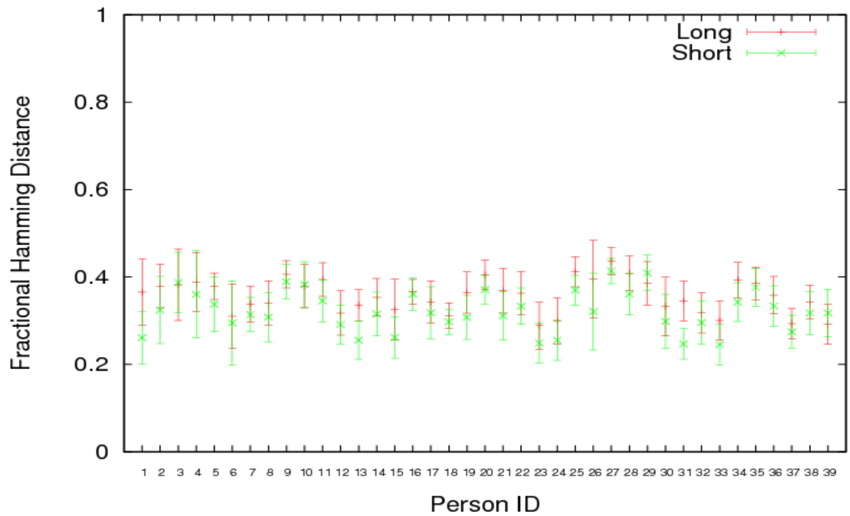


Figure 6.6: Subject-Specific analysis of algorithm of Ma *et al.*

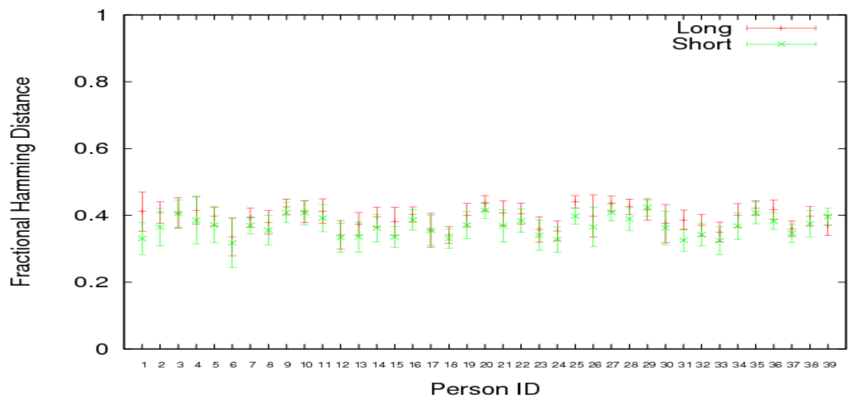


Figure 6.7: Subject-Specific analysis of Algorithm of Monro *et al.*

- Algorithm of Rathgeb and Uhl.

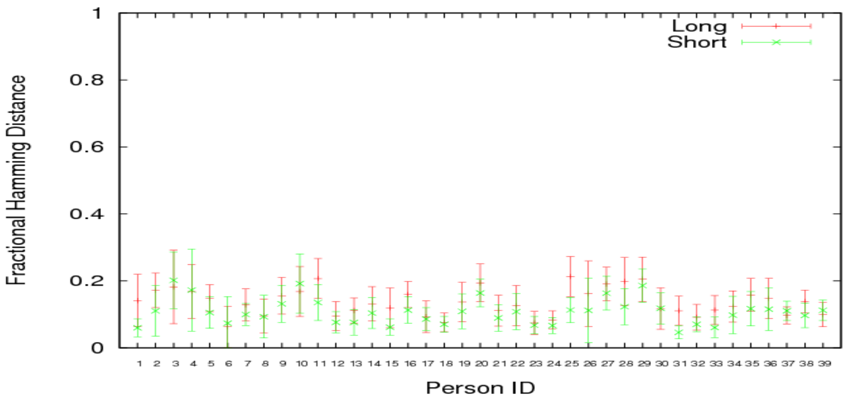


Figure 6.8: Subject-Specific analysis of Algorithm of Ko *et al.*

Refer Figure.6.9. PersonID 10 show a equal mean for short and long which is different from the other algorithms. For PersonID 29 and 37, the mean is not at the same level for short and long as seen in other algorithms. Also for subjects, 3 and 25, the standard deviation is less for small.

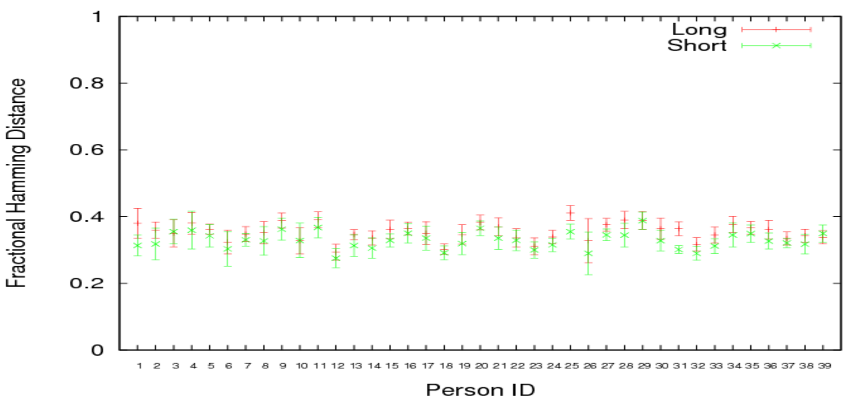


Figure 6.9: Subject-Specific analysis of Algorithm of Rathgeb and Uhl.

- Context-based feature extraction algorithm

Refer Figure.6.10. PersonID 20, 28 and 10 reveal a high mean fractional Hamming distance. For PersonID 18, the mean is at the same level as seen in other algorithms.

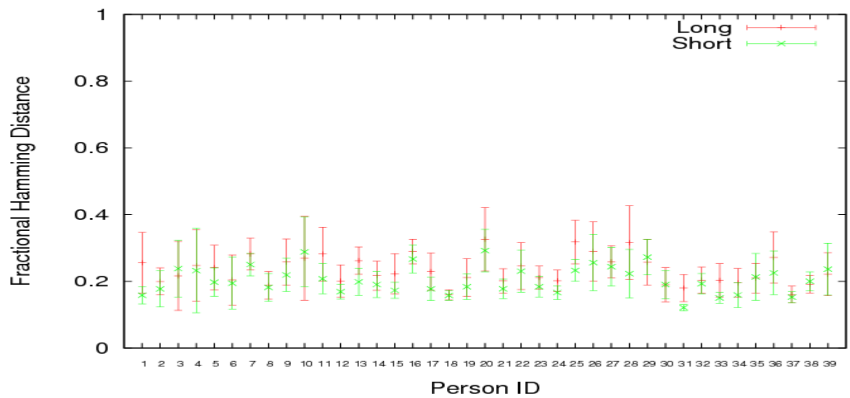


Figure 6.10: Subject-Specific analysis of Context-based feature extraction algorithm.

Overall, the plots do not show a linear change in the mean for long and short comparisons using different feature extraction algorithms.

6.2.1 Summary

As the plots of subject-specific analysis yielded different trend across different feature extraction algorithm, it is hard to say if the ageing is present in few subjects or not. One comment that could be made is, if ageing is present, its not seen in few subjects, as the mean of the comparison scores for short and long remained the same.

6.3 Multi-instance image comparison analysis

The output of multi-instance image comparison analysis is a subset of the comparison scores from the original dataset. These subsets were formed based on arithmetic operations such as mean, median etc. Once these subsets were obtained, the FNMR of the subsets for short and long comparisons were plotted and analyzed.

6.3.1 Density distribution

The first step was to analyze the density distribution of the various subsets obtained from the multi-instance image comparison analysis. The graph below shows the density distribution of genuine comparisons for various subsets. This is an example graph of 2008-2010 long comparisons of 1D-LogGabor. We can see from 6.11 that the 'Best of five' subset yields around average of 0.24 fractional Hamming distance and the original dataset's average is around 0.32.

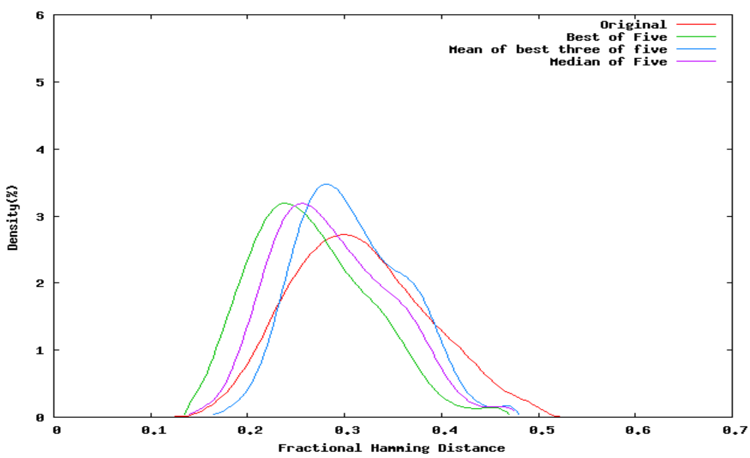


Figure 6.11: Density distribution histogram of the subsets against the original dataset.

6.3.2 Cross-algorithm analysis

Cross-algorithm analysis was performed to analyze the performance variations using multi-instance image analysis. For this purpose, three best performing feature extraction algorithms were taken into consideration. Namely, 1D-LogGabor, Ma *et al.*, Monro *et al.*. Using these algorithms, the FNMR of subsets - best of five, median of five, mean of best three of five, mean of five curves were analyzed. The graph for each of this is given below.

- **1D-LogGabor algorithm:**

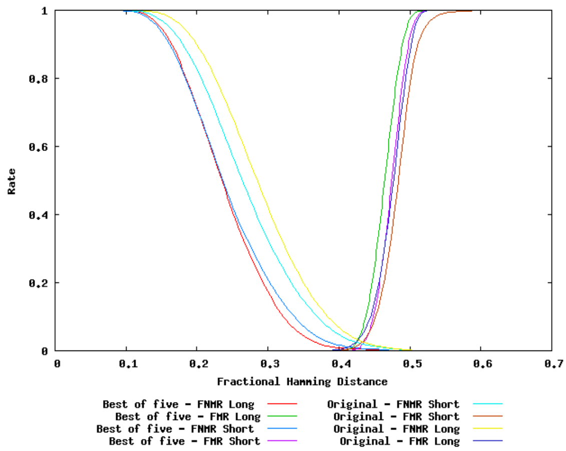


Figure 6.12: Best of five Vs Original FNMR of 2008-2009 using 1D-LogGabor.

As we can see from 6.12, the original FNMR of short and long differs from FNMR of subset best of five for year 2008-2009. The gap between the short and long FNMR's of best of five is less. From 0.81 to 0.48 FNMR, they coincide and from 0.48 FNMR long performs better than short. And around point 0.01 FNMR, they are equal again.

- **Algorithm of Ma *et al.*:**

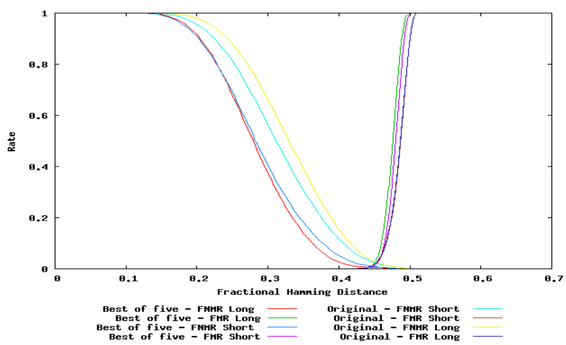


Figure 6.13: Best of five Vs Original FNMR of 2008-2009 using algorithm of Ma *et al.*.

As we can see from 6.13, at 0.8 to 0.68 FNMR, the curves coincide. From 0.6 FNMR, the long outperforms short. At point 0.01 they almost coincide again.

- **Algorithm of Monro *et al.*:**

  6.14 shows that at 0.75 FNMR, long and short coincide. From 0.6 FNMR, the long outperforms the short.

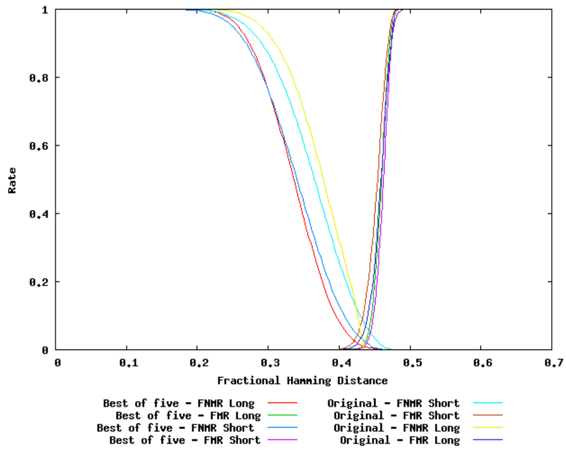


Figure 6.14: Best of five Vs Original FNMR of 2008-2009 using algorithm of Monro *et al.*

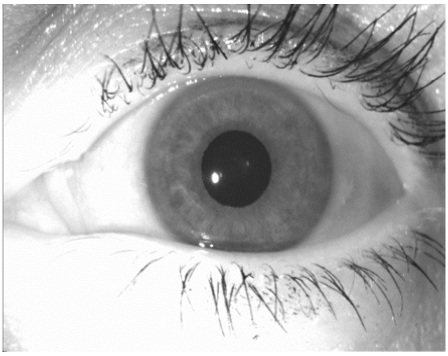
6.3.3 Summary

The experiment using multi-instance analysis was carried out in order to estimate the validity of the metrics using FNMR and FMR with a subsets of comparison scores from the original dataset. The results of the multi-instance image analysis show that the performance of the FNMR of the short and long for the subset is not same as that of the original dataset, rather we obtain a contradicting result. The degradation in FNMR for short and long has to be the same for a subset of the original data.

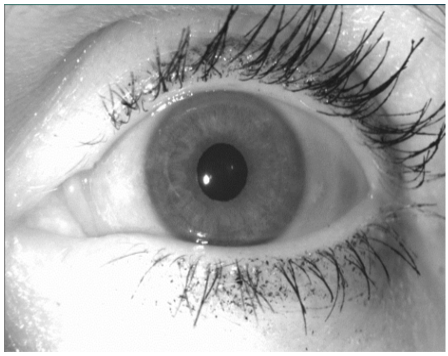
6.4 Visual examination

Few of the genuine comparisons of iris images yielded a comparison score closer to that of imposter score. On conducting visual examination, it was discovered that few images had some visible variations in illumination like reflections etc.

which is responsible for high comparison score. Few examples are given in this section.



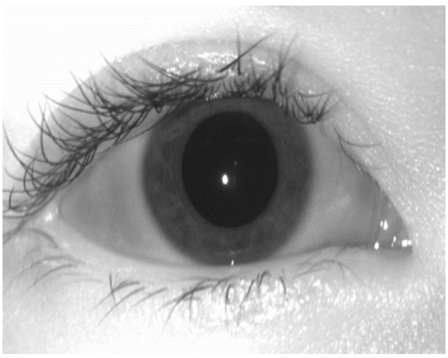
(a) Good image.



(b) Bad image.

Figure 6.15: Iris image of subject ID 04385.

The reflection due to lighting in the lens leaves a white mark on the lens in the bad iris image. Another obvious factor causing bad comparison score is the pupil dilation. This bad iris image yielded a bad comparison score throughout each of the comparisons.



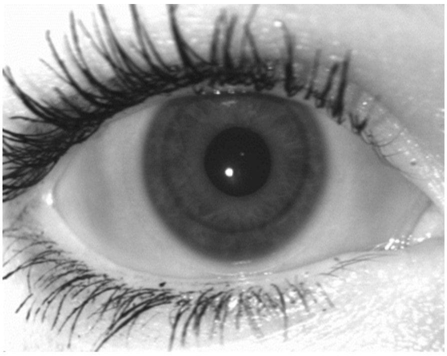
(a) Good image.

(b) Bad image.

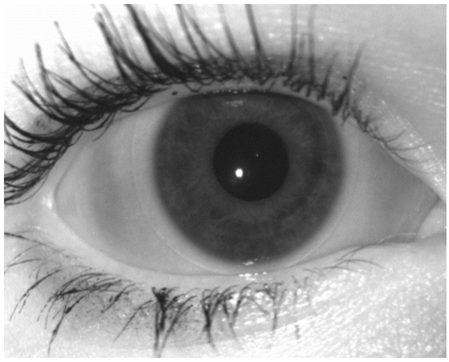
Figure 6.16: of subject ID 05455.

The bad image here yields comparison score greater than 0.49 which is closer to imposter score. The bad score is due to occlusion of eyelids and pupil dilation.

It is quite obvious that this subject was wearing contact lens during the image



(a) Good image.



(b) Bad image.

Figure 6.17: Iris image of subject ID 05303.

acquisition. The contact lens forms a interior circular boundary inside the iris. Each comparison containing this image yielded score close to imposter score.

Chapter 7

Discussion

This chapter is divided into three sections. The first section contains discussions based on the results obtained from the experiments. Followed by discussion based on Kevin Bowyer's paper [FB12] on iris template ageing. The final section contains general discussion based on template and iris ageing.

7.1 Based on interpretation of results

The goal of this thesis was to investigate effects of ageing on irises. For this purpose, the iris dataset used by Bowyer *et al.* [FB12] for his experiments was taken into consideration which contained images of subjects across two years. It was observed that the short comparisons yielded higher 1-FNMR values than that of the long comparisons across six different feature extraction algorithms. It is expected that a subset of the original dataset yields the same performance. But subsets formed based on criteria such as mean, median etc. did not show the same performance as that of the original dataset. The plots for FNMR versus fractional Hamming distance for long and short comparisons using multi-instance image analysis show that at some FNMR points, the long performs better than short and sometimes the short and long are at the same level. Hence, the metrics on which iris ageing is proved becomes debatable.

This thesis work is distinguished from the existing work on iris ageing in several ways. The study on iris template ageing so far do not contain analysis based on each subject. Previous work on this topic presents results for a large dataset but does not explicitly deal on per subject basis. One main objective of this thesis was to perform subject-specific analysis for the ND-Iris-Template-Ageing-2008-2010 database. While conducting the subject-specific experiments, it was observed that for few subjects, the mean of short and long comparison scores for year 2008-2010 were at the same level, which illustrate that there is no ageing effect for these subjects. And for few subjects the short outperforms the long comparisons. Such results make it difficult to draw conclusions. And this change is seen to be variable across different feature extraction algorithms. If ageing effect existed, atleast similar trend should have been visible across different algorithms. These results suggested that the dataset contains iris images with variations in pose and illumination, which might have been the cause for the bad comparison scores which were closer to imposter scores. Therefore it is of absolute importance that the analysis of ageing is done on a dataset which limits these minute variations to a minimum level. Careful measures should be taken during the image acquisition phase to keep this variations at a minimum level.

7.2 Based on Bowyer *et al.'s* work [FB12]

While conducting the experiments on the ND-Iris-Template-Ageing-2008-2010 database, few missing information was discovered which might have been constructive for analysing and performing further experiments. These missing features are listed below.

- ***Dataset limited:*** The dataset available in the internet is limited only to year 2008-2010, whereas the original dataset specified on the paper [FB12] contains dataset of 2008-2011.
- ***No age-related information of the subjects:*** Information regarding the age of the subjects could have been helpful to do further investigation. It would have been beneficial to classify the range of age in which the effects of ageing is most visible. It would have been helpful to analyse the speed of ageing.
- ***No information related to subjects wearing lenses:*** The database used in [FB12] does not contain information regarding the number of subjects wearing lens. It is also important to find out if any of the subject changed their lens within two years span. There is a possibility of subjects changing their lens during these two years, which may contribute for the bad comparison score. Different lens may have different effects on the iris.

- ***No information on the ethnicity and geographical origin of the subjects:*** While conducting subject-specific analysis, few subjects did not seem to have any ageing effect (as the mean of short and long comparison score were equal). It is possible that the ageing effect varies from person to person based on their diet, medications habits, smoking or alchohol habits etc. These habits to some extent correlates to the cultural ethnicity of the subjects. We also cannot neglect the possibility of ageing being divided based on population based on different geographical locations. The information about subjects geographical location could have been helpful for further interesting analysis.

It was found that there are various factors which account for high dis-similarity score in this datset such as pupil dilation, lens, illumination etc., it was logical to exclude the images containing such variations. But, there were large number of images yielding bad scores. It was hard to analyse each image based on the factors contributing a bad comparison score. As the dataset contained many number of comparisons using one single bad image. Therefore, considering the number of images causing bad scores, it was not an effective solution. Therefore, multi-instance image analysis algorithm had to be developed for analysis which would choose the best of comparison scores from the comparison score set of each image.

Further, visual examination of iris images revealed that pupil dilation is not taken into account for comparison. Comparing a iris image with small pupil against a comparatively large pupil gives a high comparison score closer to imposter score. Therefore, it is likely that the overall results provided in [FB12] contains comparison scores which include large variations in pupil dilations. Visual examination show iris images with some variation in illuminaiton causing reflections, occlusions etc., yielding bad comparison scores throughout (for each comparison with any other genuine image). Therefore, it is confirmed that the dataset contains iris images with variations such as occlusions, pupil dilation, reflections due to illumination etc., which are a prevalent contributor for higher comparison score. One way to effectively eliminate the pupil dilation in iris images is to enroll an iris with a set of template containing multiple images acquired at different dilation values. This is possible by exposing the iris to different illumination effects. This makes it possible to perform the verification against multiple iris images representing varying range of dilation values. In order to reduce the pose and illumination variation component, it is required not to have simple and stringent quality checks on iris images. Rather much concise and strict quality checks on iris images would serve the purpose.

7.3 Based on ageing in iris biometrics

7.3.1 Template ageing

It is crucial to understand that certain biological changes in the eye might contribute to template ageing in iris. Few are listed below:

- ***Eyelid drooping:*** Studies state that eyelid drooing increases as age increases [PH]. This suggest increased occlusions in iris images with the passage of time. Less iris area suggests fewer bits available for matching, which might yield large comparison score. Therefore, eyelid droop can potentially contribute to template ageing.

- ***Cornea shape and distance:*** The distance from the corneal surface to the iris also changes with age. "Despite considerable data scatter, we found significant age changes: anterior chamber depth decreased 0.011 mm/year . . . " [AMK+08]. The changes in the corneal shape and in the distance from the cornea to the iris surface with the passage of time might impose that the iris images at different age might have variations with different wrapped up versions of iris textures. Therefore, this can potentially contribute to template ageing.

Even though the assertion that the iris texture do not change with age is still conceived, it is crucial to consider these factors which can evidently contribute to template ageing in iris biometrics. Certain factors that have to be taken into consideration while accounting for template ageing are decision threshold of the biometric system, variation in pupil dilation, accuracy level of the segmentation algorithm and presence of contact lenses.

Iris template ageing effect has always taken the form of an increase in the false non-match rate with an increase in time [SBF]. Another metric is analysing the physical effects of ageing on iris [FE11]. But the question is, how reliable are these metrics which are used for demonstrating ageing effects. It is important to consider algorithm weaknesses into account which might account for the high dis-similarity scores. In most cases, it is possible that the images under different pose and illumination conditions can be analogous to ageing pattern. Therefore, it is important to carefully examine the ageing pattern change in iris which is not substantiated by any of these factors. It is of-course not possible to completely ignore these factors, but measures have to be taken to keep these variation at a minimum level, so that a much more realistic result would be obtained.

7.3.2 Iris ageing

Textural changes in iris is still under investigation. While dealing with this dataset, there are iris images with the variations in pose and illumination. This could have possibly hindered the ageing effect on irises. While critically analysing the variations in iris images, the textural changes go unnoticed. Therefore, it is significant to take measures to minimise the variations present in the images considered for experimenting iris ageing.

It is also vital to understand the speed of ageing in irises (if ageing exists). It will be interesting to analyse if the ageing occurs faster at a specific range of age or the textural change is a linear process over time. Even though we have observed an decrease in 1-FNMR at 0.01% of FMR for long comparisons of two years old iris images, it will be interesting to see the trend for 10 years or a longer time lapse dataset. It is also possible that the iris ageing effects stated by different scientists is specific to the corresponding datasets. Therefore, further investigation on diverse datasets is demanded.

One of the challenges pertaining to existence of iris ageing would be to develop algorithms which are age-resistant and robust to ageing factors. Considering the amount of deployments, there is a need for upgrading the iris recognition systems. Upgradation on large-scale would have huge cost effects.

Based on the statistics [cat], Asian countries such as India and China are estimated to be having most cataract (eye disease) affected cases. One of the reasons being ageing, diet habits, etc. of the person. It is vital to notice that the illness is geographically spread. In the similar manner, there is a possibility that ageing process differs based on population. Therefore, study of ageing for groups with different geographical location might yield interesting results. It is crucial to gain detailed information of the subjects during acquisition, for instance chain smoker, in order to figure out the factors that may contribute to faster ageing. In a nutshell, there is lots to explore in the field of iris ageing. These initial ideas pave a way to conduct new analysis in iris ageing.

Chapter 8

Conclusion and Future Work

This thesis provides a comprehensive experimental analysis of ageing effects in iris biometrics with images acquired over one and two year time lapse using six different feature extraction algorithms. The experimental results indicate that across all the feature extraction algorithms present in USIT, the long time lapse comparisons yields less 1-FNMR values at 0.01% FMR compared to the short comparisons. This demonstrates there is a performance degradation across different years irrespective of the iris feature extration algorithm used. Therefore, the performance degradation known as 'iris ageing' is not assumed to be algorithm specific.

Further work included determining if iris ageing is subject-specific. Experiments involved examination of individual subject's iris image pairs present in the dataset, finding the average fractional Hamming distance for short and long time comparisons, comparing them and identifying the specific subjects having large difference in the average fractional Hamming. But since the results yielded different results across different feature extraction algorithms, it was difficult to conclude that the iris ageing is subject-specific.

Next phase was to determine the reason for bad comparison scores of certain iris images. Multi-instance image comparison analysis was designed to analyse the effects of ageing on a subset of the original dataset. The results of this phase revealed that the performance of the FNMR of the subset did not remain the same as that of the original dataset. This questions the metrics used for proving ageing in iris as it is expected that performance of the subset remain the same as the original dataset's performance.

8.1 Future work

Although a comprehensive study of subject-specific iris ageing was performed, it only serves as a starting point for further investigations. It is recommended that the future investigations consider the detailed information of the subjects present in the dataset. It is important to consider factors such as, ethnicity, diet habits, tabacco intake habits etc., to analyse if ageing is seen to be much faster for subjects with smoking habit etc. It is also recommended that the study of ageing is done on different population groups having to different geographical origin. It is possible that ageing varies based on geographical location.

Another aspect related to the pace of ageing is an interesting factor to consider for further study. It would be helpful to find out that ageing is a linear process or there is a range of age where it is seen to be faster or slower. For understanding this feature, it is important to have details about the age of the subjects. An important direction for future work is to conduct ageing study across various age groups such as old and young.

Even though the conducted analysis shows that the metrics on which iris ageing is proved is debatable, the future work would include coming up with new metrics for proving ageing. One of them is a visual proof, but until now, no one has claimed to have found one as they all refer to binary level changes in the iris code. The future work would deal with investigating textural changes in iris using visual proof and also pinpointing the location where these changes occur. Once the ageing is proved, it is also important to develop age-resisting algorithms.

Appendix A

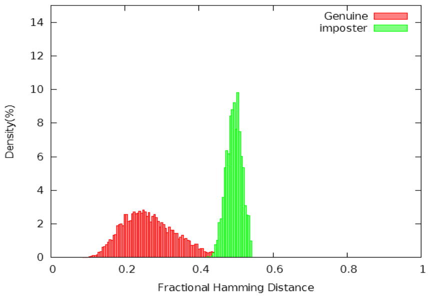
Evaluation Plots

A.1 Appendices

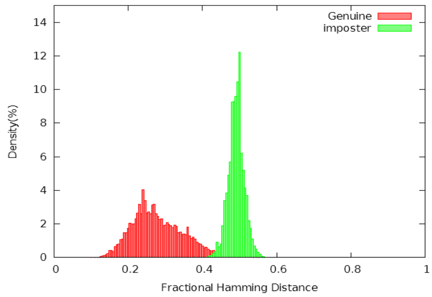
These appendices are also provided in [ER13].

A.1.1 Appendix A

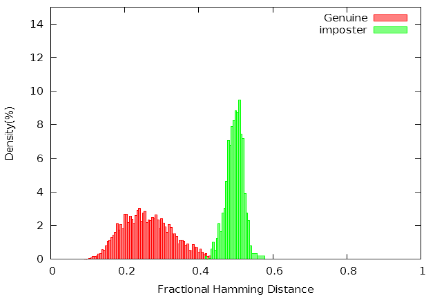
Figure A.1: Graph of Density Distribution Histograms of 1D-LogGabor.



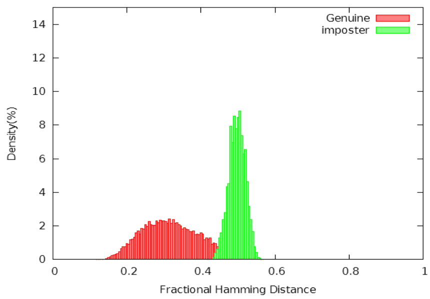
(a) Density distribution histogram of similarity scores for 1D-LogGabor 2008-2009 short comparisons.



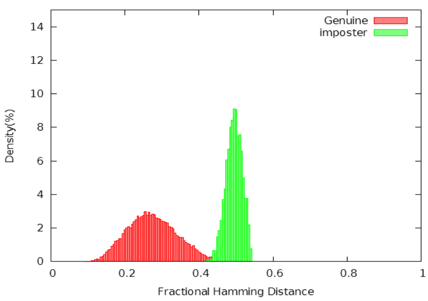
(b) Density distribution histogram of similarity scores for 1D-LogGabor 2008-2009 long comparisons.



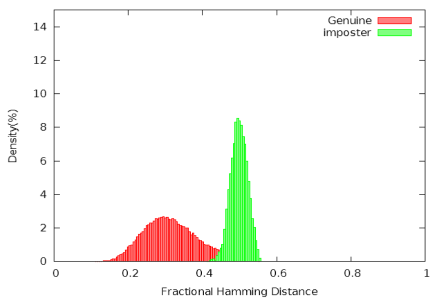
(c) Density distribution histogram of similarity scores for 1D-LogGabor 2008-2010 short comparisons.



(d) Density distribution histogram of similarity scores for 1D-LogGabor 2008-2010 long comparisons.

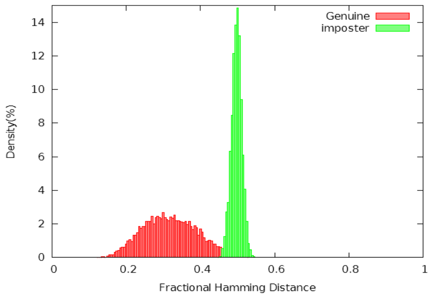


(e) Density distribution histogram of similarity scores for 1D-LogGabor 2009-2010 short comparisons.

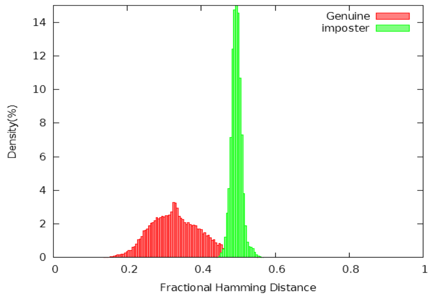


(f) Density distribution histogram of similarity scores for 1D-LogGabor 2009-2010 long comparisons.

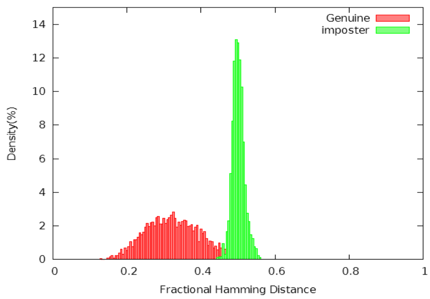
Figure A.2: Graph of Density Distribution Histograms of Algorithm Ma *et al.*



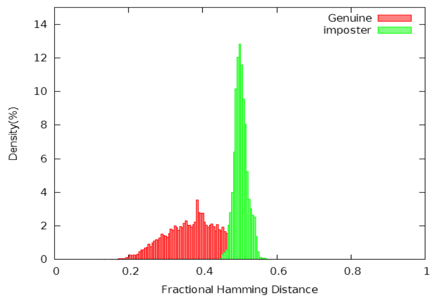
(a) Density distribution histogram of similarity scores for Ma *et al.* 2008-2009 short comparisons.



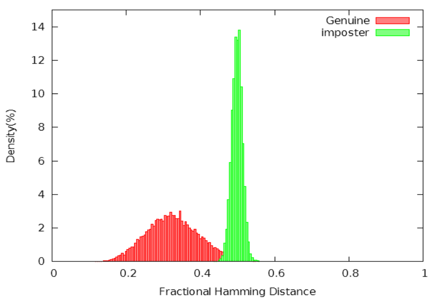
(b) Density distribution histogram of similarity scores for Ma *et al.* 2008-2009 long comparisons.



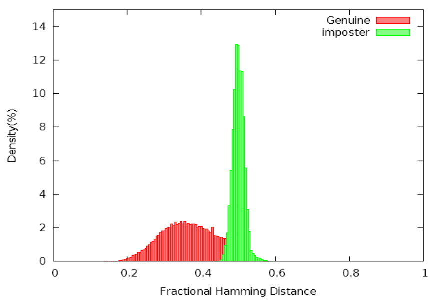
(c) Density distribution histogram of similarity scores for Ma *et al.* 2008-2010 short comparisons.



(d) Density distribution histogram of similarity scores for Ma *et al.* 2008-2010 long comparisons.

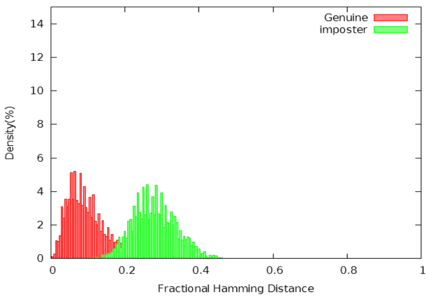


(e) Density distribution histogram of similarity scores for Ma *et al.* 2009-2010 short comparisons.

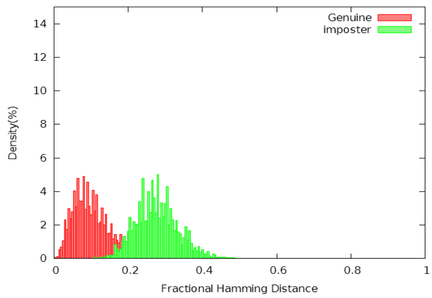


(f) Density distribution histogram of similarity scores for Ma *et al.* 2009-2010 long comparisons.

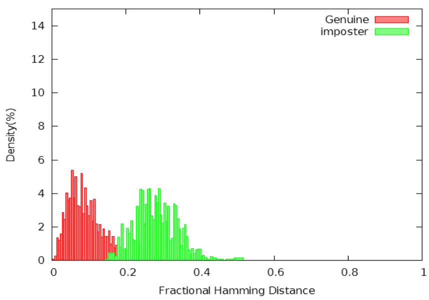
Figure A.3: Graph of Density Distribution Histograms of Algorithm Ko *et al.*



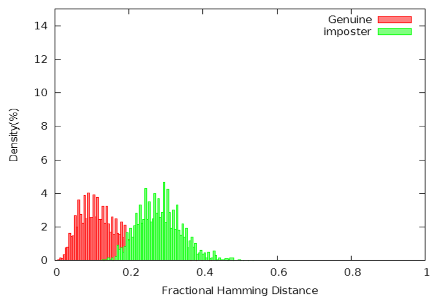
(a) Density distribution histogram of similarity scores for Ko *et al.* 2008-2009 short comparisons.



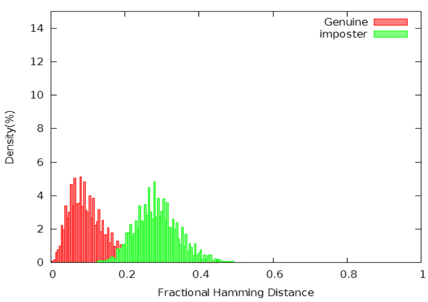
(b) Density distribution histogram of similarity scores for Ko *et al.* 2008-2009 long comparisons.



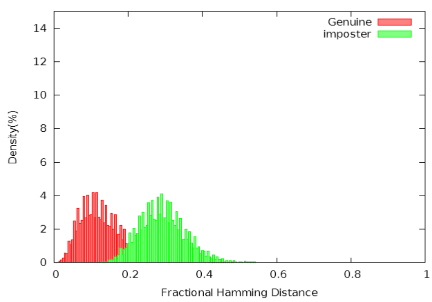
(c) Density distribution histogram of similarity scores for Ko *et al.* 2008-2010 short comparisons.



(d) Density distribution histogram of similarity scores for Ko *et al.* 2008-2010 long comparisons.

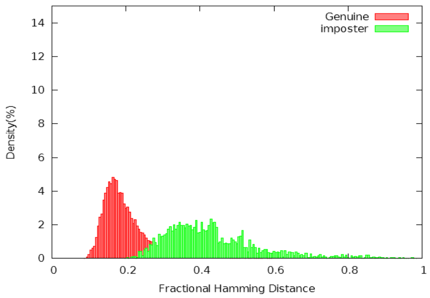


(e) Density distribution histogram of similarity scores for Ko *et al.* 2009-2010 short comparisons.

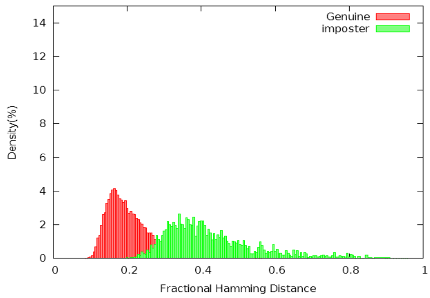


(f) Density distribution histogram of similarity scores for Ko *et al.* 2009-2010 long comparisons.

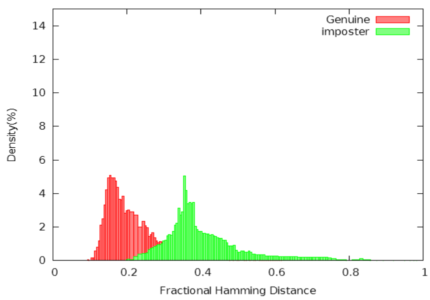
Figure A.4: Graph of Density Distribution Histograms of Context-based algorithm



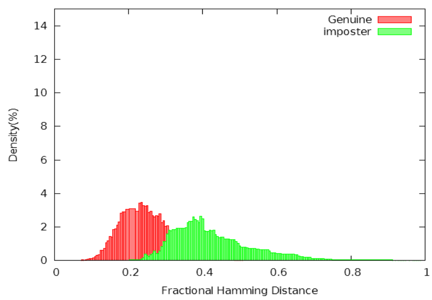
(a) Density distribution histogram of similarity scores for Context-based algorithm 2008-2009 short comparisons.



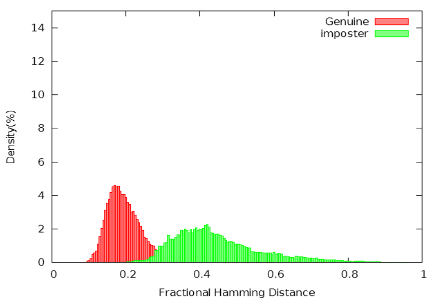
(b) Density distribution histogram of similarity scores for Context-based algorithm 2008-2009 long comparisons.



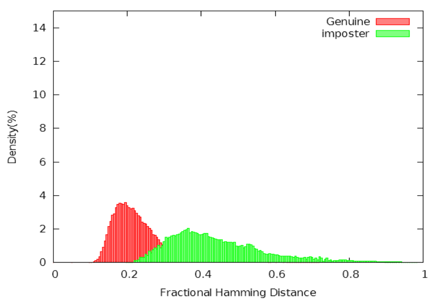
(c) Density distribution histogram of similarity scores for Context-based algorithm 2008-2010 short comparisons.



(d) Density distribution histogram of similarity scores for Context-based algorithm 2008-2010 long comparisons.

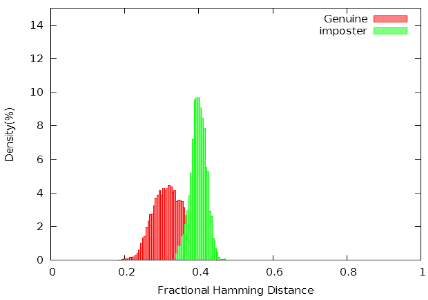


(e) Density distribution histogram of similarity scores for Context-based algorithm 2009-2010 short comparisons.

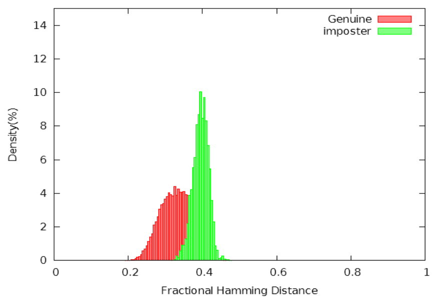


(f) Density distribution histogram of similarity scores for Context-based algorithm 2009-2010 long comparisons.

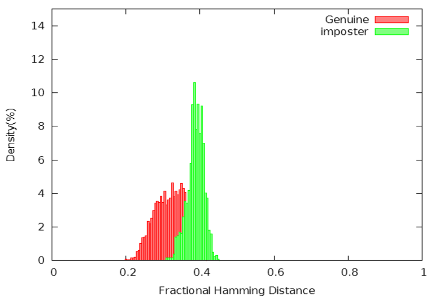
Figure A.5: Graph of Density Distribution Histograms of Algorithm Rathgeb and Uhl



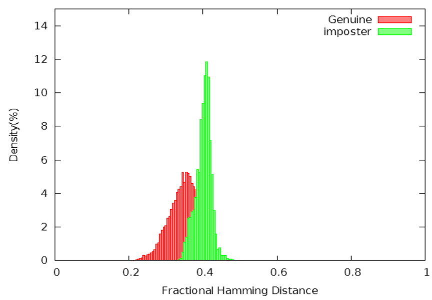
(a) Density distribution histogram of similarity scores for algorithm Rathgeb and Uhl 2008-2009 short comparisons.



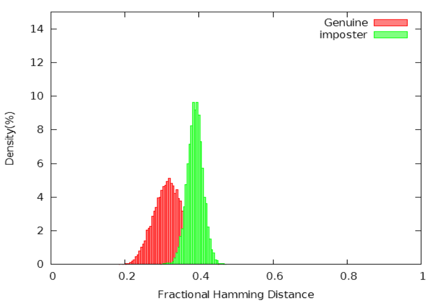
(b) Density distribution histogram of similarity scores for algorithm Rathgeb and Uhl 2008-2009 long comparisons.



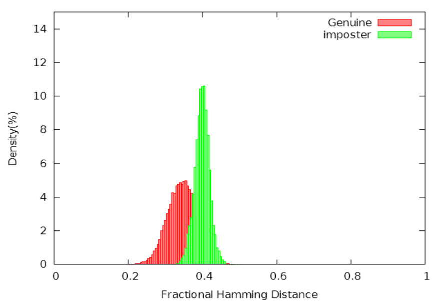
(c) Density distribution histogram of similarity scores for algorithm Rathgeb and Uhl 2008-2010 short comparisons.



(d) Density distribution histogram of similarity scores for algorithm Rathgeb and Uhl 2008-2010 long comparisons.

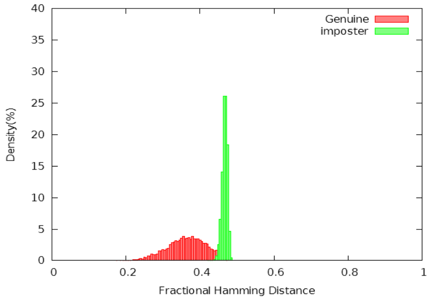


(e) Density distribution histogram of similarity scores for algorithm Rathgeb and Uhl 2009-2010 short comparisons.

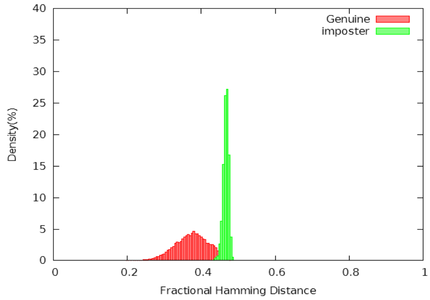


(f) Density distribution histogram of similarity scores for algorithm Rathgeb and Uhl 2009-2010 long comparisons.

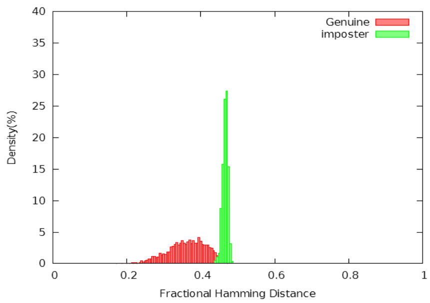
Figure A.6: Graph of Density Distribution Histograms of Algorithm Monro *et al.*



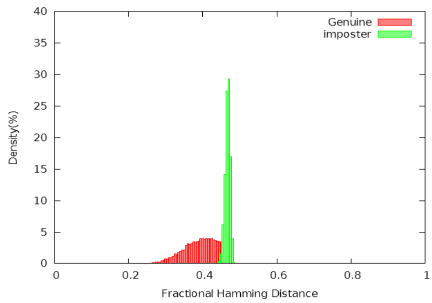
(a) Density distribution histogram of similarity scores for algorithm Monro *et al.* 2008-2009 short comparisons.



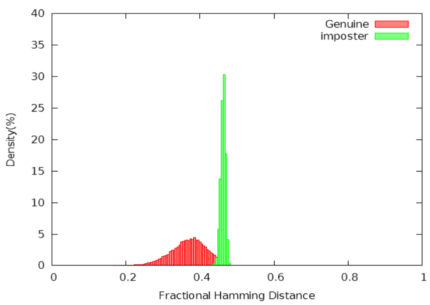
(b) Density distribution histogram of similarity scores for algorithm Monro *et al.* 2008-2009 long comparisons.



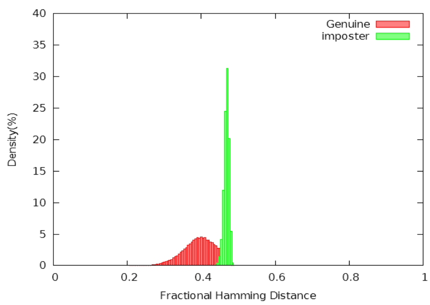
(c) Density distribution histogram of similarity scores for algorithm Monro *et al.* 2008-2010 short comparisons.



(d) Density distribution histogram of similarity scores for algorithm Monro *et al.* 2008-2010 long comparisons.



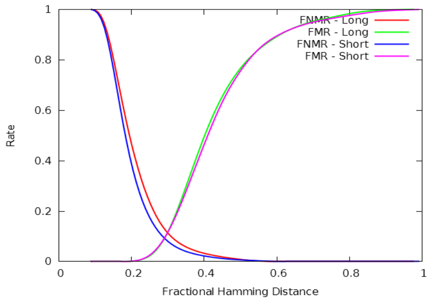
(e) Density distribution histogram of similarity scores for algorithm Monro *et al.* 2009-2010 short comparisons.



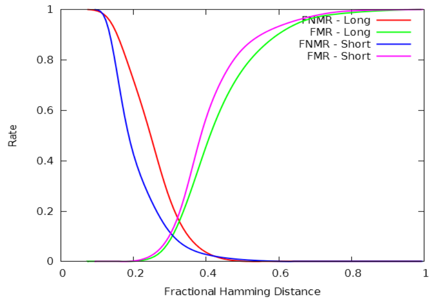
(f) Density distribution histogram of similarity scores for algorithm Monro *et al.* 2009-2010 long comparisons.

A.1.2 Appendix B

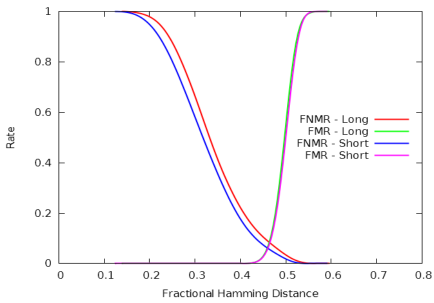
Figure A.7: Graph FNMR/FMR versus Fractional Hamming Distance



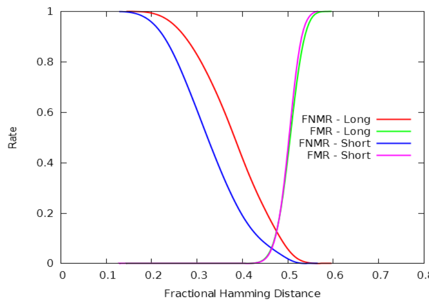
(a) Context-based algorithm: FNMR and FMR versus Fractional Hamming Distance Short and Long comparisons 2008-2009.



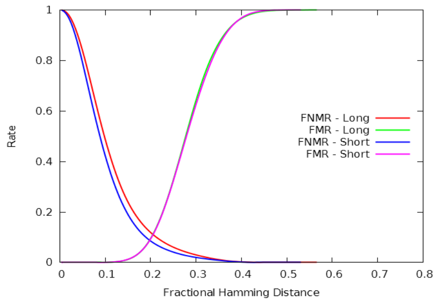
(b) Context-based algorithm: FNMR and FMR versus Fractional Hamming Distance Short and Long comparisons 2008-2010.



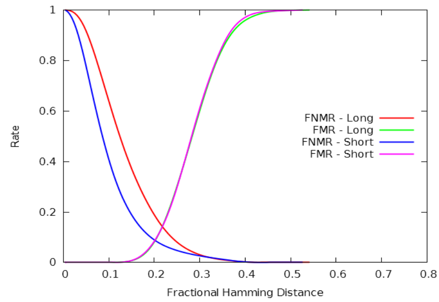
(c) Ma *et al.*: FNMR and FMR versus Fractional Hamming Distance Short and Long comparisons 2008-2009.



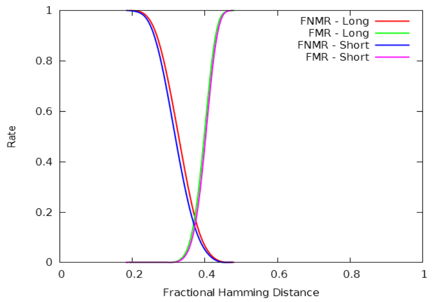
(d) Ma *et al.*: FNMR and FMR versus Fractional Hamming Distance Short and Long comparisons 2008-2010.



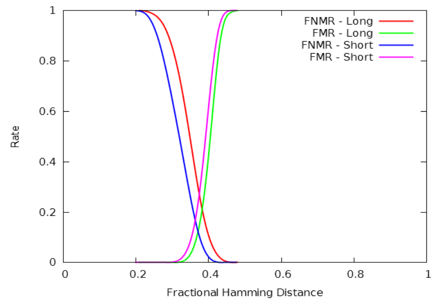
(e) Ko *et al.*: FNMR and FMR versus Fractional Hamming Distance Short and Long comparisons 2008-2009.



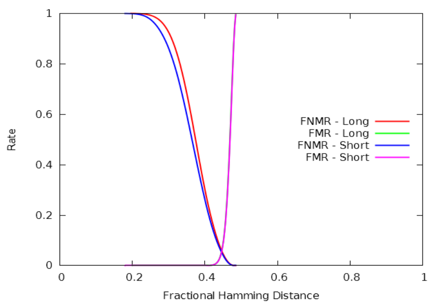
(f) Ko *et al.*: FNMR and FMR versus Fractional Hamming Distance Short and Long comparisons 2008-2010.



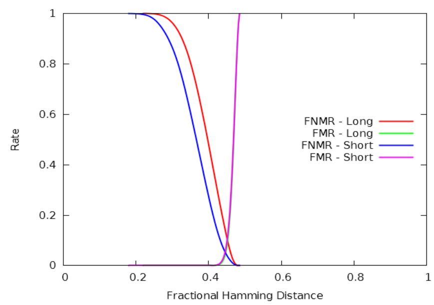
(g) Algorithm Rathgeb and Uhl: FNMR and FMR versus Fractional Hamming Distance Short and Long comparisons 2008-2009.



(h) Algorithm Rathgeb and Uhl: FNMR and FMR versus Fractional Hamming Distance Short and Long comparisons 2008-2010.



(i) Algorithm Monro *et al.*: FNMR and FMR versus Fractional Hamming Distance Short and Long comparisons 2008-2009.



(j) Algorithm Monro *et al.*: FNMR and FMR versus Fractional Hamming Distance Short and Long comparisons 2008-2010.

A.1.3 Appendix C

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `2008_2009_long`()
BEGIN
declare imageOne varchar(12);
declare median_row_index int;
declare len INT;
declare minScore int;
declare no_of_subset INT;
declare input_parameter INT;
DECLARE done INT DEFAULT FALSE;
DECLARE a INT Default 0 ;
declare image1s cursor for select distinct image1 from 2008_2009_long;
DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = 1;

delete from 2008_2009_long_best_of_two;
delete from 2008_2009_long_mean_of_two;
delete from 2008_2009_long_best_of_three;
delete from 2008_2009_long_mean_of_three;
delete from 2008_2009_long_best_of_five;
delete from 2008_2009_long_mean_of_five;
delete from 2008_2009_long_median_of_three;
delete from 2008_2009_long_median_of_five;
delete from 2008_2009_long_mean_of_best_three;

open image1s;
REPEAT
        FETCH image1s INTO imageOne;
        select count(*) INTO len from 2008_2009_long where image1=imageOne;
        IF(len = 2) then

                -- insert best value into best_of_two
                insert into 2008_2009_long_best_of_two (image1, image2, score) (
                     select image1, image2, min(score) from 2008_2009_long where
                     image1=imageOne);
                -- insert mean value into mean_of_two
                insert into 2008_2009_long_mean_of_two (image1, image2, score) (
                     select image1, image2, round(avg(score),7) from 2008
                     _2009_long where image1=imageOne);

        ELSEIF (len = 3) then

                -- insert best value into best_of_three
                insert into 2008_2009_long_best_of_three (image1, image2, score) (
                     select image1, image2, min(score) from 2008_2009_long where
                     image1=imageOne);
                -- insert mean value into mean_of_three
                insert into 2008_2009_long_mean_of_three (image1, image2, score) (
                     select image1, image2, round(avg(score),7) from 2008
                     _2009_long where image1=imageOne);
                -- insert median value into median_of_three
                insert into 2008_2009_long_median_of_three (image1, image2, score)
                      (select image1, image2, score from 2008_2009_long where
                     image1=imageOne order by score  limit 1,1);
                -- insert average of first two values (median better)
                insert into 2008_2009_long_mean_of_best_three (image1, image2,
                     score) (select image1, image2,round(avg(score),7) from 2008
                     _2009_long where image1=imageOne order by score limit 2);

                -- subset processing

                        -- input parameter as 2
                        set input_parameter =2;
                        set no_of_subset= (len+1)/2;
                        set a=0;
```

```
                simple_loop: LOOP
                         SET a=a+1;

                        -- this line generates random subset of length 2.
                        CREATE TEMPORARY TABLE subset1 AS (select image1,
                             image2, score from 2008_2009_long where
                             image1=imageOne order by rand() limit 2);
                        -- insert best value into best_of_two
                        insert into 2008_2009_long_best_of_two (image1,
                             image2, score) (select image1,image2, min(
                             score) from subset1);
                        -- insert mean value into mean_of_two
                        insert into 2008_2009_long_mean_of_two (image1,
                             image2, score) (select image1,image2, round(
                             avg(score),7) from subset1);
                         DROP TABLE subset1;
                                IF a >= no_of_subset THEN
                                        LEAVE simple_loop;
                                 END IF;
                                END LOOP simple_loop;

ELSEIF (len = 4) then

-- subset processing

        -- input parameter as 2
        set input_parameter =2;
        set no_of_subset= (len+1)/2;
        set a=0;
        simple_loop: LOOP
                 SET a=a+1;

                -- this line generates random subset of length 2.
                CREATE TEMPORARY TABLE subset1 AS (select image1, image2,
                     score from 2008_2009_long where image1=imageOne order
                      by rand() limit 2);
                -- insert best value into best_of_two
                insert into 2008_2009_long_best_of_two (image1, image2,
                     score) (select image1,image2, min(score) from subset1
                     );
                -- insert mean value into mean_of_two
                insert into 2008_2009_long_mean_of_two (image1, image2,
                     score) (select image1,image2, round(avg(score),7)
                     from subset1);
                 DROP TABLE subset1;
                        IF a >= no_of_subset THEN
                                LEAVE simple_loop;
                         END IF;
                        END LOOP simple_loop;

-- input parameter as 3
        set input_parameter =3;
        set no_of_subset= (len+1)/3;
        set a=0;
        simple_loop: LOOP
                 SET a=a+1;

                -- this line generates random subset of length 2.
                CREATE TEMPORARY TABLE subset1 AS (select image1, image2,
                     score from 2008_2009_long where image1=imageOne order
                      by rand() limit 3);
                -- insert best value into best_of_three
                insert into 2008_2009_long_best_of_three (image1, image2,
                     score) (select image1,image2, min(score) from subset1
```

```
                );
            -- insert mean value into mean_of_three
            insert into 2008_2009_long_mean_of_three (image1, image2,
                 score) (select image1,image2, round(avg(score),7)
                 from subset1);
            -- insert median value into median_of_three
            insert into 2008_2009_long_median_of_three (image1, image2
                 , score) (select image1, image2, score from subset1
                 order by score  limit 1,1);
            -- insert average of first two values (median better)
            insert into 2008_2009_long_mean_of_best_three (image1,
                 image2, score) (select image1, image2,round(avg(score
                 ),7) from subset1  order by score limit 2);

            DROP TABLE subset1;
                    IF a >= no_of_subset THEN
                            LEAVE simple_loop;
                     END IF;
                    END LOOP simple_loop;

ELSEIF (len = 5) then

        -- insert best value into best_of_five
        insert into 2008_2009_long_best_of_five (image1, image2, score) (
             select image1, image2, min(score) from 2008_2009_long where
             image1=imageOne);
        -- insert mean value into mean_of_five
        insert into 2008_2009_long_mean_of_five (image1, image2, score) (
             select image1, image2, round(avg(score),7) from 2008
             _2009_long where image1=imageOne);
        -- insert median value into median_of_five
        insert into 2008_2009_long_median_of_five (image1, image2, score)
             (select image1, image2, score from 2008_2009_long where
             image1=imageOne order by score  limit 1,1);
        -- insert average of best 3 values (median better)
        insert into 2008_2009_long_mean_of_best_five (image1, image2,
             score) (select image1, image2,round(avg(score),7) from 2008
             _2009_long where image1=imageOne order by score limit 2);


        -- subset processing

                -- input parameter as 2
                set input_parameter =2;
                set no_of_subset= (len+1)/2;
                set a=0;
                simple_loop: LOOP
                         SET a=a+1;

                        -- this line generates random subset of length 2.
                        CREATE TEMPORARY TABLE subset1 AS (select image1,
                             image2, score from 2008_2009_long where
                             image1=imageOne order by rand() limit 2);
                        -- insert best value into best_of_two
                        insert into 2008_2009_long_best_of_two (image1,
                             image2, score) (select image1,image2, min(
                             score) from subset1);
                        -- insert mean value into mean_of_two
                        insert into 2008_2009_long_mean_of_two (image1,
                             image2, score) (select image1,image2, round(
                             avg(score),7) from subset1);
                         DROP TABLE subset1;
                                IF a >= no_of_subset THEN
                                        LEAVE simple_loop;
                                 END IF;
```

```
                        END LOOP simple_loop;

-- input parameter as 3
        set input_parameter =3;
        set no_of_subset= (len+1)/3;
        set a=0;
        simple_loop: LOOP
                 SET a=a+1;

                -- this line generates random subset of length 2.
                CREATE TEMPORARY TABLE subset1 AS (select image1,
                     image2, score from 2008_2009_long where
                     image1=imageOne order by rand() limit 3);
                -- insert best value into best_of_three
                insert into 2008_2009_long_best_of_three (image1,
                     image2, score) (select image1,image2, min(
                     score) from subset1);
                -- insert mean value into mean_of_three
                insert into 2008_2009_long_mean_of_three (image1,
                     image2, score) (select image1,image2, round(
                     avg(score),7) from subset1);
                -- insert median value into median_of_three
                insert into 2008_2009_long_median_of_three (image1
                     , image2, score) (select image1, image2,
                     score from subset1 order by score  limit 1,1)
                     ;
                -- insert average of first two values (median
                     better)
                insert into 2008_2009_long_mean_of_best_three (
                     image1, image2, score) (select image1, image2
                     ,round(avg(score),7) from subset1  order by
                     score limit 2);

                DROP TABLE subset1;
                        IF a >= no_of_subset THEN
                                LEAVE simple_loop;
                         END IF;
                        END LOOP simple_loop;

ELSEIF (len > 5) then
-- subset processing

        -- input parameter as 2
        set input_parameter =2;
        set no_of_subset= (len+1)/2;
        set a=0;
        simple_loop: LOOP
                 SET a=a+1;

                -- this line generates random subset of length 2.
                CREATE TEMPORARY TABLE subset1 AS (select image1,
                     image2, score from 2008_2009_long where
                     image1=imageOne order by rand() limit 2);
                -- insert best value into best_of_two
                insert into 2008_2009_long_best_of_two (image1,
                     image2, score) (select image1,image2, min(
                     score) from subset1);
                -- insert mean value into mean_of_two
                insert into 2008_2009_long_mean_of_two (image1,
                     image2, score) (select image1,image2, round(
                     avg(score),7) from subset1);
                 DROP TABLE subset1;
                        IF a >= no_of_subset THEN
                                LEAVE simple_loop;
                         END IF;
```

```
                        END LOOP simple_loop;

-- input parameter as 3
        set input_parameter =3;
        set no_of_subset= (len+1)/3;
        set a=0;
        simple_loop: LOOP
                 SET a=a+1;

                -- this line generates random subset of length 2.
                CREATE TEMPORARY TABLE subset1 AS (select image1,
                     image2, score from 2008_2009_long where
                     image1=imageOne order by rand() limit 3);
                -- insert best value into best_of_three
                insert into 2008_2009_long_best_of_three (image1,
                     image2, score) (select image1,image2, min(
                     score) from subset1);
                -- insert mean value into mean_of_three
                insert into 2008_2009_long_mean_of_three (image1,
                     image2, score) (select image1,image2, round(
                     avg(score),7) from subset1);
                -- insert median value into median_of_three
                insert into 2008_2009_long_median_of_three (image1
                     , image2, score) (select image1, image2,
                     score from subset1 order by score  limit 1,1)
                     ;
                -- insert average of first two values (median
                     better)
                insert into 2008_2009_long_mean_of_best_three (
                     image1, image2, score) (select image1, image2
                     ,round(avg(score),7) from subset1  order by
                     score limit 2);

                DROP TABLE subset1;
                        IF a >= no_of_subset THEN
                                LEAVE simple_loop;
                         END IF;
                        END LOOP simple_loop;

-- input parameter as 5

        set input_parameter =5;
        set no_of_subset= (len+1)/5;
        set a=0;
        simple_loop: LOOP
                 SET a=a+1;

                -- this line generates random subset of length 2.
                CREATE TEMPORARY TABLE subset1 AS (select image1,
                     image2, score from 2008_2009_long where
                     image1=imageOne order by rand() limit 5);
                -- insert best value into best_of_five
                insert into 2008_2009_long_best_of_five (image1,
                     image2, score) (select image1,image2, min(
                     score) from subset1);
                -- insert mean value into mean_of_five
                insert into 2008_2009_long_mean_of_five (image1,
                     image2, score) (select image1,image2, round(
                     avg(score),7) from subset1);
                -- insert median value into median_of_five
                insert into 2008_2009_long_median_of_five (image1,
                      image2, score) (select image1, image2, score
                      from subset1 order by score  limit 1,1);
                -- insert average of first two values (median
                     better)
```

```
                                insert into 2008_2009_long_mean_of_best_five (
                                     image1, image2, score) (select image1, image2
                                     ,round(avg(score),7) from subset1  order by
                                     score limit 2);

                                DROP TABLE subset1;
                                        IF a >= no_of_subset THEN
                                                LEAVE simple_loop;
                                         END IF;
                                        END LOOP simple_loop;
                END IF;

        UNTIL done END REPEAT;
        CLOSE image1s;

END ;;
DELIMITER ;
/*!50003 SET sql_mode              = @saved_sql_mode */ ;
/*!50003 SET character_set_client  = @saved_cs_client */ ;
/*!50003 SET character_set_results = @saved_cs_results */ ;
/*!50003 SET collation_connection  = @saved_col_connection */ ;
/*!50003 DROP PROCEDURE IF EXISTS `2008_2009_short` */;
/*!50003 SET @saved_cs_client      = @@character_set_client */ ;
/*!50003 SET @saved_cs_results     = @@character_set_results */ ;
/*!50003 SET @saved_col_connection = @@collation_connection */ ;
/*!50003 SET character_set_client  = utf8 */ ;
/*!50003 SET character_set_results = utf8 */ ;
/*!50003 SET collation_connection  = utf8_general_ci */ ;
/*!50003 SET @saved_sql_mode       = @@sql_mode */ ;
/*!50003 SET sql_mode              = 'STRICT_TRANS_TABLES,NO_AUTO_CREATE_USER,
     NO_ENGINE_SUBSTITUTION' */ ;
DELIMITER ;;
CREATE DEFINER=`root`@`localhost` PROCEDURE `2008_2009_short`()
BEGIN
declare imageOne varchar(12);
declare median_row_index int;
declare len INT;
declare minScore int;
declare no_of_subset INT;
declare input_parameter INT;
DECLARE done INT DEFAULT FALSE;
DECLARE a INT Default 0 ;
declare image1s cursor for select distinct image1 from 2008_2009_short;
DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = 1;

delete from 2008_2009_short_best_of_two;
delete from 2008_2009_short_mean_of_two;
delete from 2008_2009_short_best_of_three;
delete from 2008_2009_short_mean_of_three;
delete from 2008_2009_short_best_of_five;
delete from 2008_2009_short_mean_of_five;
delete from 2008_2009_short_median_of_three;
delete from 2008_2009_short_median_of_five;
delete from 2008_2009_short_mean_of_best_three;

open image1s;
REPEAT
        FETCH image1s INTO imageOne;
        select count(*) INTO len from 2008_2009_short where image1=imageOne;
        IF(len = 2) then

                -- insert best value into best_of_two
                insert into 2008_2009_short_best_of_two (image1, image2, score) (
                     select image1, image2, min(score) from 2008_2009_short where
                     image1=imageOne);
```

```
        -- insert mean value into mean_of_two
        insert into 2008_2009_short_mean_of_two (image1, image2, score) (
             select image1, image2, round(avg(score),7) from 2008
             _2009_short where image1=imageOne);

ELSEIF (len = 3) then

        -- insert best value into best_of_three
        insert into 2008_2009_short_best_of_three (image1, image2, score)
             (select image1, image2, min(score) from 2008_2009_short where
              image1=imageOne);
        -- insert mean value into mean_of_three
        insert into 2008_2009_short_mean_of_three (image1, image2, score)
             (select image1, image2, round(avg(score),7) from 2008
             _2009_short where image1=imageOne);
        -- insert median value into median_of_three
        insert into 2008_2009_short_median_of_three (image1, image2, score
             ) (select image1, image2, score from 2008_2009_short where
             image1=imageOne order by score  limit 1,1);
        -- insert average of first two values (median better)
        insert into 2008_2009_short_mean_of_best_three (image1, image2,
             score) (select image1, image2,round(avg(score),7) from 2008
             _2009_short where image1=imageOne order by score limit 2);

        -- subset processing

                -- input parameter as 2
                set input_parameter =2;
                set no_of_subset= (len+1)/2;
                set a=0;

                simple_loop: LOOP
                         SET a=a+1;

                        -- this line generates random subset of length 2.
                        CREATE TEMPORARY TABLE subset1 AS (select image1,
                             image2, score from 2008_2009_short where
                             image1=imageOne order by rand() limit 2);
                        -- insert best value into best_of_two
                        insert into 2008_2009_short_best_of_two (image1,
                             image2, score) (select image1,image2, min(
                             score) from subset1);
                        -- insert mean value into mean_of_two
                        insert into 2008_2009_short_mean_of_two (image1,
                             image2, score) (select image1,image2, round(
                             avg(score),7) from subset1);
                         DROP TABLE subset1;
                                IF a >= no_of_subset THEN
                                        LEAVE simple_loop;
                                 END IF;
                                END LOOP simple_loop;

ELSEIF (len = 4) then

-- subset processing

        -- input parameter as 2
        set input_parameter =2;
        set no_of_subset= (len+1)/2;
        set a=0;
        simple_loop: LOOP
                 SET a=a+1;

                -- this line generates random subset of length 2.
```

```
                CREATE TEMPORARY TABLE subset1 AS (select image1, image2,
                     score from 2008_2009_short where image1=imageOne
                     order by rand() limit 2);
                -- insert best value into best_of_two
                insert into 2008_2009_short_best_of_two (image1, image2,
                     score) (select image1,image2, min(score) from subset1
                     );
                -- insert mean value into mean_of_two
                insert into 2008_2009_short_mean_of_two (image1, image2,
                     score) (select image1,image2, round(avg(score),7)
                     from subset1);
                 DROP TABLE subset1;
                        IF a >= no_of_subset THEN
                                LEAVE simple_loop;
                         END IF;
                        END LOOP simple_loop;

-- input parameter as 3
        set input_parameter =3;
        set no_of_subset= (len+1)/3;
        set a=0;
        simple_loop: LOOP
                 SET a=a+1;

                -- this line generates random subset of length 2.
                CREATE TEMPORARY TABLE subset1 AS (select image1, image2,
                     score from 2008_2009_short where image1=imageOne
                     order by rand() limit 3);
                select * from subset1;
                select min(score) from subset1;
                -- insert best value into best_of_three
                insert into 2008_2009_short_best_of_three (image1, image2,
                      score) (select image1,image2, min(score) from
                     subset1);
                -- insert mean value into mean_of_three
                insert into 2008_2009_short_mean_of_three (image1, image2,
                      score) (select image1,image2, round(avg(score),7)
                     from subset1);
                -- insert median value into median_of_three
                insert into 2008_2009_short_median_of_three (image1,
                     image2, score) (select image1, image2, score from
                     subset1 order by score  limit 1,1);
                -- insert average of first two values (median better)
                insert into 2008_2009_short_mean_of_best_three (image1,
                     image2, score) (select image1, image2,round(avg(score
                     ),7) from subset1  order by score limit 2);

                DROP TABLE subset1;
                        IF a >= no_of_subset THEN
                                LEAVE simple_loop;
                         END IF;
                        END LOOP simple_loop;

ELSEIF (len = 5) then

        -- insert best value into best_of_five
        insert into 2008_2009_short_best_of_five (image1, image2, score) (
             select image1, image2, min(score) from 2008_2009_short where
             image1=imageOne);
        -- insert mean value into mean_of_five
        insert into 2008_2009_short_mean_of_five (image1, image2, score) (
             select image1, image2, round(avg(score),7) from 2008
             _2009_short where image1=imageOne);
        -- insert median value into median_of_five
```

```
insert into 2008_2009_short_median_of_five (image1, image2, score)
      (select image1, image2, score from 2008_2009_short where
     image1=imageOne order by score  limit 1,1);
-- insert average of best 3 values (median better)
insert into 2008_2009_short_mean_of_best_five (image1, image2,
     score) (select image1, image2,round(avg(score),7) from 2008
     _2009_short where image1=imageOne order by score limit 2);


-- subset processing

        -- input parameter as 2
        set input_parameter =2;
        set no_of_subset= (len+1)/2;
        set a=0;
        simple_loop: LOOP
                 SET a=a+1;

                -- this line generates random subset of length 2.
                CREATE TEMPORARY TABLE subset1 AS (select image1,
                     image2, score from 2008_2009_short where
                     image1=imageOne order by rand() limit 2);
                -- insert best value into best_of_two
                insert into 2008_2009_short_best_of_two (image1,
                     image2, score) (select image1,image2, min(
                     score) from subset1);
                -- insert mean value into mean_of_two
                insert into 2008_2009_short_mean_of_two (image1,
                     image2, score) (select image1,image2, round(
                     avg(score),7) from subset1);
                 DROP TABLE subset1;
                        IF a >= no_of_subset THEN
                                LEAVE simple_loop;
                         END IF;
                        END LOOP simple_loop;

-- input parameter as 3
        set input_parameter =3;
        set no_of_subset= (len+1)/3;
        set a=0;
        simple_loop: LOOP
                 SET a=a+1;

                -- this line generates random subset of length 2.
                CREATE TEMPORARY TABLE subset1 AS (select image1,
                     image2, score from 2008_2009_short where
                     image1=imageOne order by rand() limit 3);
                select * from subset1;
                select min(score) from subset1;
                -- insert best value into best_of_three
                insert into 2008_2009_short_best_of_three (image1,
                      image2, score) (select image1,image2, min(
                     score) from subset1);
                -- insert mean value into mean_of_three
                insert into 2008_2009_short_mean_of_three (image1,
                      image2, score) (select image1,image2, round(
                     avg(score),7) from subset1);
                -- insert median value into median_of_three
                insert into 2008_2009_short_median_of_three (
                     image1, image2, score) (select image1, image2
                     , score from subset1 order by score  limit
                     1,1);
                -- insert average of first two values (median
                     better)
```

```
                insert into 2008_2009_short_mean_of_best_three (
                     image1, image2, score) (select image1, image2
                     ,round(avg(score),7) from subset1  order by
                     score limit 2);

                DROP TABLE subset1;
                        IF a >= no_of_subset THEN
                                LEAVE simple_loop;
                         END IF;
                        END LOOP simple_loop;

ELSEIF (len > 5) then
-- subset processing

        -- input parameter as 2
        set input_parameter =2;
        set no_of_subset= (len+1)/2;
        set a=0;
        simple_loop: LOOP
                 SET a=a+1;

                -- this line generates random subset of length 2.
                CREATE TEMPORARY TABLE subset1 AS (select image1,
                     image2, score from 2008_2009_short where
                     image1=imageOne order by rand() limit 2);
                -- insert best value into best_of_two
                insert into 2008_2009_short_best_of_two (image1,
                     image2, score) (select image1,image2, min(
                     score) from subset1);
                -- insert mean value into mean_of_two
                insert into 2008_2009_short_mean_of_two (image1,
                     image2, score) (select image1,image2, round(
                     avg(score),7) from subset1);
                 DROP TABLE subset1;
                        IF a >= no_of_subset THEN
                                LEAVE simple_loop;
                         END IF;
                        END LOOP simple_loop;

-- input parameter as 3
        set input_parameter =3;
        set no_of_subset= (len+1)/3;
        set a=0;
        simple_loop: LOOP
                 SET a=a+1;

                -- this line generates random subset of length 2.
                CREATE TEMPORARY TABLE subset1 AS (select image1,
                     image2, score from 2008_2009_short where
                     image1=imageOne order by rand() limit 3);
                select * from subset1;
                select min(score) from subset1;
                -- insert best value into best_of_three
                insert into 2008_2009_short_best_of_three (image1,
                      image2, score) (select image1,image2, min(
                     score) from subset1);
                -- insert mean value into mean_of_three
                insert into 2008_2009_short_mean_of_three (image1,
                      image2, score) (select image1,image2, round(
                     avg(score),7) from subset1);
                -- insert median value into median_of_three
                insert into 2008_2009_short_median_of_three (
                     image1, image2, score) (select image1, image2
                     , score from subset1 order by score  limit
                     1,1);
```

```
                                -- insert average of first two values (median
                                     better)
                                insert into 2008_2009_short_mean_of_best_three (
                                     image1, image2, score) (select image1, image2
                                     ,round(avg(score),7) from subset1  order by
                                     score limit 2);

                                DROP TABLE subset1;
                                        IF a >= no_of_subset THEN
                                                LEAVE simple_loop;
                                         END IF;
                                        END LOOP simple_loop;

                -- input parameter as 5

                        set input_parameter =5;
                        set no_of_subset= (len+1)/5;
                        set a=0;
                        simple_loop: LOOP
                                 SET a=a+1;

                                -- this line generates random subset of length 2.
                                CREATE TEMPORARY TABLE subset1 AS (select image1,
                                     image2, score from 2008_2009_short where
                                     image1=imageOne order by rand() limit 5);
                                -- insert best value into best_of_five
                                insert into 2008_2009_short_best_of_five (image1,
                                     image2, score) (select image1,image2, min(
                                     score) from subset1);
                                -- insert mean value into mean_of_five
                                insert into 2008_2009_short_mean_of_five (image1,
                                     image2, score) (select image1,image2, round(
                                     avg(score),7) from subset1);
                                -- insert median value into median_of_five
                                insert into 2008_2009_short_median_of_five (image1
                                     , image2, score) (select image1, image2,
                                     score from subset1 order by score  limit 1,1)
                                     ;
                                -- insert average of first two values (median
                                     better)
                                insert into 2008_2009_short_mean_of_best_five (
                                     image1, image2, score) (select image1, image2
                                     ,round(avg(score),7) from subset1  order by
                                     score limit 2);

                                DROP TABLE subset1;
                                        IF a >= no_of_subset THEN
                                                LEAVE simple_loop;
                                         END IF;
                                        END LOOP simple_loop;
                END IF;

        UNTIL done END REPEAT;
        CLOSE image1s;

END ;;
DELIMITER ;
/*!50003 SET sql_mode              = @saved_sql_mode */ ;
/*!50003 SET character_set_client  = @saved_cs_client */ ;
/*!50003 SET character_set_results = @saved_cs_results */ ;
/*!50003 SET collation_connection  = @saved_col_connection */ ;
/*!50003 DROP PROCEDURE IF EXISTS `2008_2010_long` */;
/*!50003 SET @saved_cs_client      = @@character_set_client */ ;
/*!50003 SET @saved_cs_results     = @@character_set_results */ ;
/*!50003 SET @saved_col_connection = @@collation_connection */ ;
```

```
/*!50003 SET character_set_client  = utf8 */ ;
/*!50003 SET character_set_results = utf8 */ ;
/*!50003 SET collation_connection  = utf8_general_ci */ ;
/*!50003 SET @saved_sql_mode       = @@sql_mode */ ;
/*!50003 SET sql_mode              = 'STRICT_TRANS_TABLES,NO_AUTO_CREATE_USER,
     NO_ENGINE_SUBSTITUTION' */ ;
DELIMITER ;;
CREATE DEFINER=`root`@`localhost` PROCEDURE `2008_2010_long`()
BEGIN
declare imageOne varchar(12);
declare median_row_index int;
declare len INT;
declare minScore int;
declare no_of_subset INT;
declare input_parameter INT;
DECLARE done INT DEFAULT FALSE;
DECLARE a INT Default 0 ;
declare image1s cursor for select distinct image1 from 2008_2010_long;
DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = 1;

delete from 2008_2010_long_best_of_two;
delete from 2008_2010_long_mean_of_two;
delete from 2008_2010_long_best_of_three;
delete from 2008_2010_long_mean_of_three;
delete from 2008_2010_long_best_of_five;
delete from 2008_2010_long_mean_of_five;
delete from 2008_2010_long_median_of_three;
delete from 2008_2010_long_median_of_five;
delete from 2008_2010_long_mean_of_best_three;

open image1s;
REPEAT
        FETCH image1s INTO imageOne;
        select count(*) INTO len from 2008_2010_long where image1=imageOne;
        IF(len = 2) then

                -- insert best value into best_of_two
                insert into 2008_2010_long_best_of_two (image1, image2, score) (
                     select image1, image2, min(score) from 2008_2010_long where
                     image1=imageOne);
                -- insert mean value into mean_of_two
                insert into 2008_2010_long_mean_of_two (image1, image2, score) (
                     select image1, image2, round(avg(score),7) from 2008
                     _2010_long where image1=imageOne);

        ELSEIF (len = 3) then

                -- insert best value into best_of_three
                insert into 2008_2010_long_best_of_three (image1, image2, score) (
                     select image1, image2, min(score) from 2008_2010_long where
                     image1=imageOne);
                -- insert mean value into mean_of_three
                insert into 2008_2010_long_mean_of_three (image1, image2, score) (
                     select image1, image2, round(avg(score),7) from 2008
                     _2010_long where image1=imageOne);
                -- insert median value into median_of_three
                insert into 2008_2010_long_median_of_three (image1, image2, score)
                      (select image1, image2, score from 2008_2010_long where
                     image1=imageOne order by score  limit 1,1);
                -- insert average of first two values (median better)
                insert into 2008_2010_long_mean_of_best_three (image1, image2,
                     score) (select image1, image2,round(avg(score),7) from 2008
                     _2010_long where image1=imageOne order by score limit 2);

                -- subset processing
```

```
                -- input parameter as 2
                set input_parameter =2;
                set no_of_subset= (len+1)/2;
                set a=0;

                simple_loop: LOOP
                         SET a=a+1;

                        -- this line generates random subset of length 2.
                        CREATE TEMPORARY TABLE subset1 AS (select image1,
                             image2, score from 2008_2010_long where
                             image1=imageOne order by rand() limit 2);
                        -- insert best value into best_of_two
                        insert into 2008_2010_long_best_of_two (image1,
                             image2, score) (select image1,image2, min(
                             score) from subset1);
                        -- insert mean value into mean_of_two
                        insert into 2008_2010_long_mean_of_two (image1,
                             image2, score) (select image1,image2, round(
                             avg(score),7) from subset1);
                         DROP TABLE subset1;
                                IF a >= no_of_subset THEN
                                        LEAVE simple_loop;
                                 END IF;
                                END LOOP simple_loop;

ELSEIF (len = 4) then

-- subset processing

        -- input parameter as 2
        set input_parameter =2;
        set no_of_subset= (len+1)/2;
        set a=0;
        simple_loop: LOOP
                 SET a=a+1;

                -- this line generates random subset of length 2.
                CREATE TEMPORARY TABLE subset1 AS (select image1, image2,
                     score from 2008_2010_long where image1=imageOne order
                      by rand() limit 2);
                -- insert best value into best_of_two
                insert into 2008_2010_long_best_of_two (image1, image2,
                     score) (select image1,image2, min(score) from subset1
                     );
                -- insert mean value into mean_of_two
                insert into 2008_2010_long_mean_of_two (image1, image2,
                     score) (select image1,image2, round(avg(score),7)
                     from subset1);
                 DROP TABLE subset1;
                        IF a >= no_of_subset THEN
                                LEAVE simple_loop;
                         END IF;
                        END LOOP simple_loop;

-- input parameter as 3
        set input_parameter =3;
        set no_of_subset= (len+1)/3;
        set a=0;
        simple_loop: LOOP
                 SET a=a+1;

                -- this line generates random subset of length 2.
```

```
                CREATE TEMPORARY TABLE subset1 AS (select image1, image2,
                     score from 2008_2010_long where image1=imageOne order
                      by rand() limit 3);
                -- insert best value into best_of_three
                insert into 2008_2010_long_best_of_three (image1, image2,
                     score) (select image1,image2, min(score) from subset1
                     );
                -- insert mean value into mean_of_three
                insert into 2008_2010_long_mean_of_three (image1, image2,
                     score) (select image1,image2, round(avg(score),7)
                     from subset1);
                -- insert median value into median_of_three
                insert into 2008_2010_long_median_of_three (image1, image2
                     , score) (select image1, image2, score from subset1
                     order by score  limit 1,1);
                -- insert average of first two values (median better)
                insert into 2008_2010_long_mean_of_best_three (image1,
                     image2, score) (select image1, image2,round(avg(score
                     ),7) from subset1  order by score limit 2);

                DROP TABLE subset1;
                        IF a >= no_of_subset THEN
                                LEAVE simple_loop;
                         END IF;
                        END LOOP simple_loop;

ELSEIF (len = 5) then

        -- insert best value into best_of_five
        insert into 2008_2010_long_best_of_five (image1, image2, score) (
             select image1, image2, min(score) from 2008_2010_long where
             image1=imageOne);
        -- insert mean value into mean_of_five
        insert into 2008_2010_long_mean_of_five (image1, image2, score) (
             select image1, image2, round(avg(score),7) from 2008
             _2010_long where image1=imageOne);
        -- insert median value into median_of_five
        insert into 2008_2010_long_median_of_five (image1, image2, score)
             (select image1, image2, score from 2008_2010_long where
             image1=imageOne order by score  limit 1,1);
        -- insert average of best 3 values (median better)
        insert into 2008_2010_long_mean_of_best_five (image1, image2,
             score) (select image1, image2,round(avg(score),7) from 2008
             _2010_long where image1=imageOne order by score limit 2);


        -- subset processing

                -- input parameter as 2
                set input_parameter =2;
                set no_of_subset= (len+1)/2;
                set a=0;
                simple_loop: LOOP
                         SET a=a+1;

                        -- this line generates random subset of length 2.
                        CREATE TEMPORARY TABLE subset1 AS (select image1,
                             image2, score from 2008_2010_long where
                             image1=imageOne order by rand() limit 2);
                        -- insert best value into best_of_two
                        insert into 2008_2010_long_best_of_two (image1,
                             image2, score) (select image1,image2, min(
                             score) from subset1);
                        -- insert mean value into mean_of_two
```

```
                insert into 2008_2010_long_mean_of_two (image1,
                     image2, score) (select image1,image2, round(
                     avg(score),7) from subset1);
                 DROP TABLE subset1;
                        IF a >= no_of_subset THEN
                                LEAVE simple_loop;
                         END IF;
                        END LOOP simple_loop;

-- input parameter as 3
        set input_parameter =3;
        set no_of_subset= (len+1)/3;
        set a=0;
        simple_loop: LOOP
                 SET a=a+1;

                -- this line generates random subset of length 2.
                CREATE TEMPORARY TABLE subset1 AS (select image1,
                     image2, score from 2008_2010_long where
                     image1=imageOne order by rand() limit 3);
                -- insert best value into best_of_three
                insert into 2008_2010_long_best_of_three (image1,
                     image2, score) (select image1,image2, min(
                     score) from subset1);
                -- insert mean value into mean_of_three
                insert into 2008_2010_long_mean_of_three (image1,
                     image2, score) (select image1,image2, round(
                     avg(score),7) from subset1);
                -- insert median value into median_of_three
                insert into 2008_2010_long_median_of_three (image1
                     , image2, score) (select image1, image2,
                     score from subset1 order by score  limit 1,1)
                     ;
                -- insert average of first two values (median
                     better)
                insert into 2008_2010_long_mean_of_best_three (
                     image1, image2, score) (select image1, image2
                     ,round(avg(score),7) from subset1  order by
                     score limit 2);

                DROP TABLE subset1;
                        IF a >= no_of_subset THEN
                                LEAVE simple_loop;
                         END IF;
                        END LOOP simple_loop;

ELSEIF (len > 5) then
-- subset processing

        -- input parameter as 2
        set input_parameter =2;
        set no_of_subset= (len+1)/2;
        set a=0;
        simple_loop: LOOP
                 SET a=a+1;

                -- this line generates random subset of length 2.
                CREATE TEMPORARY TABLE subset1 AS (select image1,
                     image2, score from 2008_2010_long where
                     image1=imageOne order by rand() limit 2);
                -- insert best value into best_of_two
                insert into 2008_2010_long_best_of_two (image1,
                     image2, score) (select image1,image2, min(
                     score) from subset1);
                -- insert mean value into mean_of_two
```

```
                insert into 2008_2010_long_mean_of_two (image1,
                     image2, score) (select image1,image2, round(
                     avg(score),7) from subset1);
                 DROP TABLE subset1;
                        IF a >= no_of_subset THEN
                                LEAVE simple_loop;
                         END IF;
                        END LOOP simple_loop;

-- input parameter as 3
        set input_parameter =3;
        set no_of_subset= (len+1)/3;
        set a=0;
        simple_loop: LOOP
                 SET a=a+1;

                -- this line generates random subset of length 2.
                CREATE TEMPORARY TABLE subset1 AS (select image1,
                     image2, score from 2008_2010_long where
                     image1=imageOne order by rand() limit 3);
                -- insert best value into best_of_three
                insert into 2008_2010_long_best_of_three (image1,
                     image2, score) (select image1,image2, min(
                     score) from subset1);
                -- insert mean value into mean_of_three
                insert into 2008_2010_long_mean_of_three (image1,
                     image2, score) (select image1,image2, round(
                     avg(score),7) from subset1);
                -- insert median value into median_of_three
                insert into 2008_2010_long_median_of_three (image1
                     , image2, score) (select image1, image2,
                     score from subset1 order by score  limit 1,1)
                     ;
                -- insert average of first two values (median
                     better)
                insert into 2008_2010_long_mean_of_best_three (
                     image1, image2, score) (select image1, image2
                     ,round(avg(score),7) from subset1  order by
                     score limit 2);

                DROP TABLE subset1;
                        IF a >= no_of_subset THEN
                                LEAVE simple_loop;
                         END IF;
                        END LOOP simple_loop;

-- input parameter as 5

        set input_parameter =5;
        set no_of_subset= (len+1)/5;
        set a=0;
        simple_loop: LOOP
                 SET a=a+1;

                -- this line generates random subset of length 2.
                CREATE TEMPORARY TABLE subset1 AS (select image1,
                     image2, score from 2008_2010_long where
                     image1=imageOne order by rand() limit 5);
                -- insert best value into best_of_five
                insert into 2008_2010_long_best_of_five (image1,
                     image2, score) (select image1,image2, min(
                     score) from subset1);
                -- insert mean value into mean_of_five
                insert into 2008_2010_long_mean_of_five (image1,
                     image2, score) (select image1,image2, round(
```

```
                                 avg(score),7) from subset1);
                             -- insert median value into median_of_five
                             insert into 2008_2010_long_median_of_five (image1,
                                  image2, score) (select image1, image2, score
                                  from subset1 order by score  limit 1,1);
                             -- insert average of first two values (median
                                 better)
                             insert into 2008_2010_long_mean_of_best_five (
                                 image1, image2, score) (select image1, image2
                                 ,round(avg(score),7) from subset1  order by
                                 score limit 2);

                             DROP TABLE subset1;
                                     IF a >= no_of_subset THEN
                                             LEAVE simple_loop;
                                      END IF;
                                     END LOOP simple_loop;
                END IF;

        UNTIL done END REPEAT;
        CLOSE image1s;

END ;;
DELIMITER ;
/*!50003 SET sql_mode              = @saved_sql_mode */ ;
/*!50003 SET character_set_client  = @saved_cs_client */ ;
/*!50003 SET character_set_results = @saved_cs_results */ ;
/*!50003 SET collation_connection  = @saved_col_connection */ ;
/*!50003 DROP PROCEDURE IF EXISTS `2008_2010_short` */;
/*!50003 SET @saved_cs_client      = @@character_set_client */ ;
/*!50003 SET @saved_cs_results     = @@character_set_results */ ;
/*!50003 SET @saved_col_connection = @@collation_connection */ ;
/*!50003 SET character_set_client  = utf8 */ ;
/*!50003 SET character_set_results = utf8 */ ;
/*!50003 SET collation_connection  = utf8_general_ci */ ;
/*!50003 SET @saved_sql_mode       = @@sql_mode */ ;
/*!50003 SET sql_mode              = 'STRICT_TRANS_TABLES,NO_AUTO_CREATE_USER,
     NO_ENGINE_SUBSTITUTION' */ ;
DELIMITER ;;
CREATE DEFINER=`root`@`localhost` PROCEDURE `2008_2010_short`()
BEGIN
declare imageOne varchar(12);
declare median_row_index int;
declare len INT;
declare minScore int;
declare no_of_subset INT;
declare input_parameter INT;
DECLARE done INT DEFAULT FALSE;
DECLARE a INT Default 0 ;
declare image1s cursor for select distinct image1 from 2008_2010_short;
DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = 1;

delete from 2008_2010_short_best_of_two;
delete from 2008_2010_short_mean_of_two;
delete from 2008_2010_short_best_of_three;
delete from 2008_2010_short_mean_of_three;
delete from 2008_2010_short_best_of_five;
delete from 2008_2010_short_mean_of_five;
delete from 2008_2010_short_median_of_three;
delete from 2008_2010_short_median_of_five;
delete from 2008_2010_short_mean_of_best_three;

open image1s;
REPEAT
        FETCH image1s INTO imageOne;
```

```
select count(*) INTO len from 2008_2010_short where image1=imageOne;
IF(len = 2) then

        -- insert best value into best_of_two
        insert into 2008_2010_short_best_of_two (image1, image2, score) (
             select image1, image2, min(score) from 2008_2010_short where
             image1=imageOne);
        -- insert mean value into mean_of_two
        insert into 2008_2010_short_mean_of_two (image1, image2, score) (
             select image1, image2, round(avg(score),7) from 2008
             _2010_short where image1=imageOne);

ELSEIF (len = 3) then

        -- insert best value into best_of_three
        insert into 2008_2010_short_best_of_three (image1, image2, score)
             (select image1, image2, min(score) from 2008_2010_short where
              image1=imageOne);
        -- insert mean value into mean_of_three
        insert into 2008_2010_short_mean_of_three (image1, image2, score)
             (select image1, image2, round(avg(score),7) from 2008
             _2010_short where image1=imageOne);
        -- insert median value into median_of_three
        insert into 2008_2010_short_median_of_three (image1, image2, score
             ) (select image1, image2, score from 2008_2010_short where
             image1=imageOne order by score  limit 1,1);
        -- insert average of first two values (median better)
        insert into 2008_2010_short_mean_of_best_three (image1, image2,
             score) (select image1, image2,round(avg(score),7) from 2008
             _2010_short where image1=imageOne order by score limit 2);

        -- subset processing

                -- input parameter as 2
                set input_parameter =2;
                set no_of_subset= (len+1)/2;
                set a=0;

                simple_loop: LOOP
                         SET a=a+1;

                        -- this line generates random subset of length 2.
                        CREATE TEMPORARY TABLE subset1 AS (select image1,
                             image2, score from 2008_2010_short where
                             image1=imageOne order by rand() limit 2);
                        -- insert best value into best_of_two
                        insert into 2008_2010_short_best_of_two (image1,
                             image2, score) (select image1,image2, min(
                             score) from subset1);
                        -- insert mean value into mean_of_two
                        insert into 2008_2010_short_mean_of_two (image1,
                             image2, score) (select image1,image2, round(
                             avg(score),7) from subset1);
                         DROP TABLE subset1;
                                IF a >= no_of_subset THEN
                                        LEAVE simple_loop;
                                 END IF;
                                END LOOP simple_loop;

ELSEIF (len = 4) then

-- subset processing

        -- input parameter as 2
        set input_parameter =2;
```

```
        set no_of_subset= (len+1)/2;
        set a=0;
        simple_loop: LOOP
                 SET a=a+1;

                -- this line generates random subset of length 2.
                CREATE TEMPORARY TABLE subset1 AS (select image1, image2,
                     score from 2008_2010_short where image1=imageOne
                     order by rand() limit 2);
                -- insert best value into best_of_two
                insert into 2008_2010_short_best_of_two (image1, image2,
                     score) (select image1,image2, min(score) from subset1
                     );
                -- insert mean value into mean_of_two
                insert into 2008_2010_short_mean_of_two (image1, image2,
                     score) (select image1,image2, round(avg(score),7)
                     from subset1);
                 DROP TABLE subset1;
                        IF a >= no_of_subset THEN
                                LEAVE simple_loop;
                         END IF;
                        END LOOP simple_loop;

-- input parameter as 3
        set input_parameter =3;
        set no_of_subset= (len+1)/3;
        set a=0;
        simple_loop: LOOP
                 SET a=a+1;

                -- this line generates random subset of length 2.
                CREATE TEMPORARY TABLE subset1 AS (select image1, image2,
                     score from 2008_2010_short where image1=imageOne
                     order by rand() limit 3);
                -- insert best value into best_of_three
                insert into 2008_2010_short_best_of_three (image1, image2,
                      score) (select image1,image2, min(score) from
                     subset1);
                -- insert mean value into mean_of_three
                insert into 2008_2010_short_mean_of_three (image1, image2,
                      score) (select image1,image2, round(avg(score),7)
                     from subset1);
                -- insert median value into median_of_three
                insert into 2008_2010_short_median_of_three (image1,
                     image2, score) (select image1, image2, score from
                     subset1 order by score  limit 1,1);
                -- insert average of first two values (median better)
                insert into 2008_2010_short_mean_of_best_three (image1,
                     image2, score) (select image1, image2,round(avg(score
                     ),7) from subset1  order by score limit 2);

                DROP TABLE subset1;
                        IF a >= no_of_subset THEN
                                LEAVE simple_loop;
                         END IF;
                        END LOOP simple_loop;

ELSEIF (len = 5) then

        -- insert best value into best_of_five
        insert into 2008_2010_short_best_of_five (image1, image2, score) (
             select image1, image2, min(score) from 2008_2010_short where
             image1=imageOne);
        -- insert mean value into mean_of_five
```

```
insert into 2008_2010_short_mean_of_five (image1, image2, score) (
     select image1, image2, round(avg(score),7) from 2008
     _2010_short where image1=imageOne);
-- insert median value into median_of_five
insert into 2008_2010_short_median_of_five (image1, image2, score)
      (select image1, image2, score from 2008_2010_short where
     image1=imageOne order by score  limit 1,1);
-- insert average of best 3 values (median better)
insert into 2008_2010_short_mean_of_best_five (image1, image2,
     score) (select image1, image2,round(avg(score),7) from 2008
     _2010_short where image1=imageOne order by score limit 2);


-- subset processing

        -- input parameter as 2
        set input_parameter =2;
        set no_of_subset= (len+1)/2;
        set a=0;
        simple_loop: LOOP
                 SET a=a+1;

                -- this line generates random subset of length 2.
                CREATE TEMPORARY TABLE subset1 AS (select image1,
                     image2, score from 2008_2010_short where
                     image1=imageOne order by rand() limit 2);
                -- insert best value into best_of_two
                insert into 2008_2010_short_best_of_two (image1,
                     image2, score) (select image1,image2, min(
                     score) from subset1);
                -- insert mean value into mean_of_two
                insert into 2008_2010_short_mean_of_two (image1,
                     image2, score) (select image1,image2, round(
                     avg(score),7) from subset1);
                 DROP TABLE subset1;
                        IF a >= no_of_subset THEN
                                LEAVE simple_loop;
                         END IF;
                        END LOOP simple_loop;

-- input parameter as 3
        set input_parameter =3;
        set no_of_subset= (len+1)/3;
        set a=0;
        simple_loop: LOOP
                 SET a=a+1;

                -- this line generates random subset of length 2.
                CREATE TEMPORARY TABLE subset1 AS (select image1,
                     image2, score from 2008_2010_short where
                     image1=imageOne order by rand() limit 3);
                -- insert best value into best_of_three
                insert into 2008_2010_short_best_of_three (image1,
                      image2, score) (select image1,image2, min(
                     score) from subset1);
                -- insert mean value into mean_of_three
                insert into 2008_2010_short_mean_of_three (image1,
                      image2, score) (select image1,image2, round(
                     avg(score),7) from subset1);
                -- insert median value into median_of_three
                insert into 2008_2010_short_median_of_three (
                     image1, image2, score) (select image1, image2
                     , score from subset1 order by score  limit
                     1,1);
```

```
                -- insert average of first two values (median
                     better)
                insert into 2008_2010_short_mean_of_best_three (
                     image1, image2, score) (select image1, image2
                     ,round(avg(score),7) from subset1  order by
                     score limit 2);

                DROP TABLE subset1;
                        IF a >= no_of_subset THEN
                                LEAVE simple_loop;
                         END IF;
                        END LOOP simple_loop;

ELSEIF (len > 5) then
-- subset processing

        -- input parameter as 2
        set input_parameter =2;
        set no_of_subset= (len+1)/2;
        set a=0;
        simple_loop: LOOP
                 SET a=a+1;

                -- this line generates random subset of length 2.
                CREATE TEMPORARY TABLE subset1 AS (select image1,
                     image2, score from 2008_2010_short where
                     image1=imageOne order by rand() limit 2);
                -- insert best value into best_of_two
                insert into 2008_2010_short_best_of_two (image1,
                     image2, score) (select image1,image2, min(
                     score) from subset1);
                -- insert mean value into mean_of_two
                insert into 2008_2010_short_mean_of_two (image1,
                     image2, score) (select image1,image2, round(
                     avg(score),7) from subset1);
                 DROP TABLE subset1;
                        IF a >= no_of_subset THEN
                                LEAVE simple_loop;
                         END IF;
                        END LOOP simple_loop;

-- input parameter as 3
        set input_parameter =3;
        set no_of_subset= (len+1)/3;
        set a=0;
        simple_loop: LOOP
                 SET a=a+1;

                -- this line generates random subset of length 2.
                CREATE TEMPORARY TABLE subset1 AS (select image1,
                     image2, score from 2008_2010_short where
                     image1=imageOne order by rand() limit 3);
                -- insert best value into best_of_three
                insert into 2008_2010_short_best_of_three (image1,
                      image2, score) (select image1,image2, min(
                     score) from subset1);
                -- insert mean value into mean_of_three
                insert into 2008_2010_short_mean_of_three (image1,
                      image2, score) (select image1,image2, round(
                     avg(score),7) from subset1);
                -- insert median value into median_of_three
                insert into 2008_2010_short_median_of_three (
                     image1, image2, score) (select image1, image2
                     , score from subset1 order by score  limit
                     1,1);
```

```
                                -- insert average of first two values (median
                                     better)
                                insert into 2008_2010_short_mean_of_best_three (
                                     image1, image2, score) (select image1, image2
                                     ,round(avg(score),7) from subset1  order by
                                     score limit 2);

                                DROP TABLE subset1;
                                        IF a >= no_of_subset THEN
                                                LEAVE simple_loop;
                                         END IF;
                                        END LOOP simple_loop;

                -- input parameter as 5

                        set input_parameter =5;
                        set no_of_subset= (len+1)/5;
                        set a=0;
                        simple_loop: LOOP
                                 SET a=a+1;

                                -- this line generates random subset of length 2.
                                CREATE TEMPORARY TABLE subset1 AS (select image1,
                                     image2, score from 2008_2010_short where
                                     image1=imageOne order by rand() limit 5);
                                -- insert best value into best_of_five
                                insert into 2008_2010_short_best_of_five (image1,
                                     image2, score) (select image1,image2, min(
                                     score) from subset1);
                                -- insert mean value into mean_of_five
                                insert into 2008_2010_short_mean_of_five (image1,
                                     image2, score) (select image1,image2, round(
                                     avg(score),7) from subset1);
                                -- insert median value into median_of_five
                                insert into 2008_2010_short_median_of_five (image1
                                     , image2, score) (select image1, image2,
                                     score from subset1 order by score  limit 1,1)
                                     ;
                                -- insert average of first two values (median
                                     better)
                                insert into 2008_2010_short_mean_of_best_five (
                                     image1, image2, score) (select image1, image2
                                     ,round(avg(score),7) from subset1  order by
                                     score limit 2);

                                DROP TABLE subset1;
                                        IF a >= no_of_subset THEN
                                                LEAVE simple_loop;
                                         END IF;
                                        END LOOP simple_loop;
                END IF;

        UNTIL done END REPEAT;
        CLOSE image1s;

END ;;
DELIMITER ;
/*!50003 SET sql_mode              = @saved_sql_mode */ ;
/*!50003 SET character_set_client  = @saved_cs_client */ ;
/*!50003 SET character_set_results = @saved_cs_results */ ;
/*!50003 SET collation_connection  = @saved_col_connection */ ;
/*!50003 DROP PROCEDURE IF EXISTS `density_distribution1` */;
/*!50003 SET @saved_cs_client      = @@character_set_client */ ;
/*!50003 SET @saved_cs_results     = @@character_set_results */ ;
/*!50003 SET @saved_col_connection = @@collation_connection */ ;
```

```
/*!50003 SET character_set_client  = utf8 */ ;
/*!50003 SET character_set_results = utf8 */ ;
/*!50003 SET collation_connection  = utf8_general_ci */ ;
/*!50003 SET @saved_sql_mode       = @@sql_mode */ ;
/*!50003 SET sql_mode              = 'STRICT_TRANS_TABLES,NO_AUTO_CREATE_USER,
     NO_ENGINE_SUBSTITUTION' */ ;
DELIMITER ;;
CREATE DEFINER=`root`@`localhost` PROCEDURE `density_distribution1`(in table_name
     text)
BEGIN
        declare table_length int;
        SET @t1 =concat(CONCAT('SELECT SQL_CALC_FOUND_ROWS * FROM ',table_name ),
             ' limit 0');


        PREPARE stmt1 FROM @t1;
        EXECUTE stmt1;
        SELECT FOUND_ROWS() into table_length;
        set @t2 =concat(concat(concat(CONCAT('CREATE TEMPORARY TABLE loggabor.
             subsetOne AS (SELECT catagory, count(catagory) as category_tot, (
             count(*)/',table_length),')*100 as density FROM '), table_name),'
             group by catagory);');

        DROP TEMPORARY TABLE IF EXISTS logGabor.subsetOne;
        PREPARE stmt2 FROM @t2;
         EXECUTE stmt2;
        set @csum := 0;

        DROP TEMPORARY TABLE IF EXISTS logGabor.subsetTwo;
        create temporary table loggabor.subsetTwo(select catagory, category_tot, (
             @csum := @csum + category_tot) as cumulative_sum,  density from
             logGabor.subsetOne);
        select *, cumulative_sum/table_length,(cumulative_sum/table_length) *100
             from logGabor.subsetTwo;

        DEALLOCATE PREPARE stmt1;
        DEALLOCATE PREPARE stmt2;
END ;;
DELIMITER ;
/*!50003 SET sql_mode              = @saved_sql_mode */ ;
/*!50003 SET character_set_client  = @saved_cs_client */ ;
/*!50003 SET character_set_results = @saved_cs_results */ ;
/*!50003 SET collation_connection  = @saved_col_connection */ ;
/*!50003 DROP PROCEDURE IF EXISTS `I_2008_2009_long` */;
/*!50003 SET @saved_cs_client      = @@character_set_client */ ;
/*!50003 SET @saved_cs_results     = @@character_set_results */ ;
/*!50003 SET @saved_col_connection = @@collation_connection */ ;
/*!50003 SET character_set_client  = utf8 */ ;
/*!50003 SET character_set_results = utf8 */ ;
/*!50003 SET collation_connection  = utf8_general_ci */ ;
/*!50003 SET @saved_sql_mode       = @@sql_mode */ ;
/*!50003 SET sql_mode              = 'STRICT_TRANS_TABLES,NO_AUTO_CREATE_USER,
     NO_ENGINE_SUBSTITUTION' */ ;
DELIMITER ;;
CREATE DEFINER=`root`@`localhost` PROCEDURE `I_2008_2009_long`()
BEGIN
declare imageOne varchar(12);
declare median_row_index int;
declare len INT;
declare minScore int;
declare no_of_subset INT;
declare input_parameter INT;
DECLARE done INT DEFAULT FALSE;
DECLARE a INT Default 0 ;
declare image1s cursor for select distinct image1 from I_2008_2009_long;
```

```
DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = 1;

delete from I_2008_2009_long_best_of_two;
delete from I_2008_2009_long_mean_of_two;
delete from I_2008_2009_long_best_of_three;
delete from I_2008_2009_long_mean_of_three;
delete from I_2008_2009_long_best_of_five;
delete from I_2008_2009_long_mean_of_five;
delete from I_2008_2009_long_median_of_three;
delete from I_2008_2009_long_median_of_five;
delete from I_2008_2009_long_mean_of_best_three;

open image1s;
REPEAT
        FETCH image1s INTO imageOne;
        select count(*) INTO len from I_2008_2009_long where image1=imageOne;
        IF(len = 2) then

                -- insert best value into best_of_two
                insert into I_2008_2009_long_best_of_two (image1, image2, score) (
                     select image1, image2, min(score) from I_2008_2009_long where
                      image1=imageOne);
                -- insert mean value into mean_of_two
                insert into I_2008_2009_long_mean_of_two (image1, image2, score) (
                     select image1, image2, round(avg(score),7) from
                     I_2008_2009_long where image1=imageOne);

        ELSEIF (len = 3) then

                -- insert best value into best_of_three
                insert into I_2008_2009_long_best_of_three (image1, image2, score)
                      (select image1, image2, min(score) from I_2008_2009_long
                     where image1=imageOne);
                -- insert mean value into mean_of_three
                insert into I_2008_2009_long_mean_of_three (image1, image2, score)
                      (select image1, image2, round(avg(score),7) from
                     I_2008_2009_long where image1=imageOne);
                -- insert median value into median_of_three
                insert into I_2008_2009_long_median_of_three (image1, image2,
                     score) (select image1, image2, score from I_2008_2009_long
                     where image1=imageOne order by score  limit 1,1);
                -- insert average of first two values (median better)
                insert into I_2008_2009_long_mean_of_best_three (image1, image2,
                     score) (select image1, image2,round(avg(score),7) from
                     I_2008_2009_long where image1=imageOne order by score limit
                     2);

                -- subset processing

                        -- input parameter as 2
                        set input_parameter =2;
                        set no_of_subset= (len+1)/2;
                        set a=0;

                        simple_loop: LOOP
                                 SET a=a+1;

                                -- this line generates random subset of length 2.
                                CREATE TEMPORARY TABLE subset1 AS (select image1,
                                     image2, score from I_2008_2009_long where
                                     image1=imageOne order by rand() limit 2);
                                -- insert best value into best_of_two
                                insert into I_2008_2009_long_best_of_two (image1,
                                     image2, score) (select image1,image2, min(
                                     score) from subset1);
```

```
                        -- insert mean value into mean_of_two
                        insert into I_2008_2009_long_mean_of_two (image1,
                             image2, score) (select image1,image2, round(
                             avg(score),7) from subset1);
                         DROP TABLE subset1;
                                IF a >= no_of_subset THEN
                                        LEAVE simple_loop;
                                 END IF;
                                END LOOP simple_loop;

ELSEIF (len = 4) then

-- subset processing

        -- input parameter as 2
        set input_parameter =2;
        set no_of_subset= (len+1)/2;
        set a=0;
        simple_loop: LOOP
                 SET a=a+1;

                -- this line generates random subset of length 2.
                CREATE TEMPORARY TABLE subset1 AS (select image1, image2,
                     score from I_2008_2009_long where image1=imageOne
                     order by rand() limit 2);
                -- insert best value into best_of_two
                insert into I_2008_2009_long_best_of_two (image1, image2,
                     score) (select image1,image2, min(score) from subset1
                     );
                -- insert mean value into mean_of_two
                insert into I_2008_2009_long_mean_of_two (image1, image2,
                     score) (select image1,image2, round(avg(score),7)
                     from subset1);
                 DROP TABLE subset1;
                        IF a >= no_of_subset THEN
                                LEAVE simple_loop;
                         END IF;
                        END LOOP simple_loop;

-- input parameter as 3
        set input_parameter =3;
        set no_of_subset= (len+1)/3;
        set a=0;
        simple_loop: LOOP
                 SET a=a+1;

                -- this line generates random subset of length 2.
                CREATE TEMPORARY TABLE subset1 AS (select image1, image2,
                     score from I_2008_2009_long where image1=imageOne
                     order by rand() limit 3);
                -- insert best value into best_of_three
                insert into I_2008_2009_long_best_of_three (image1, image2
                     , score) (select image1,image2, min(score) from
                     subset1);
                -- insert mean value into mean_of_three
                insert into I_2008_2009_long_mean_of_three (image1, image2
                     , score) (select image1,image2, round(avg(score),7)
                     from subset1);
                -- insert median value into median_of_three
                insert into I_2008_2009_long_median_of_three (image1,
                     image2, score) (select image1, image2, score from
                     subset1 order by score  limit 1,1);
                -- insert average of first two values (median better)
                insert into I_2008_2009_long_mean_of_best_three (image1,
                     image2, score) (select image1, image2,round(avg(score
```

```
                    ),7) from subset1  order by score limit 2);

                DROP TABLE subset1;
                        IF a >= no_of_subset THEN
                                LEAVE simple_loop;
                         END IF;
                        END LOOP simple_loop;

ELSEIF (len = 5) then

        -- insert best value into best_of_five
        insert into I_2008_2009_long_best_of_five (image1, image2, score)
             (select image1, image2, min(score) from I_2008_2009_long
             where image1=imageOne);
        -- insert mean value into mean_of_five
        insert into I_2008_2009_long_mean_of_five (image1, image2, score)
             (select image1, image2, round(avg(score),7) from
             I_2008_2009_long where image1=imageOne);
        -- insert median value into median_of_five
        insert into I_2008_2009_long_median_of_five (image1, image2, score
             ) (select image1, image2, score from I_2008_2009_long where
             image1=imageOne order by score  limit 1,1);
        -- insert average of best 3 values (median better)
        insert into I_2008_2009_long_mean_of_best_five (image1, image2,
             score) (select image1, image2,round(avg(score),7) from
             I_2008_2009_long where image1=imageOne order by score limit
             2);


        -- subset processing

                -- input parameter as 2
                set input_parameter =2;
                set no_of_subset= (len+1)/2;
                set a=0;
                simple_loop: LOOP
                         SET a=a+1;

                        -- this line generates random subset of length 2.
                        CREATE TEMPORARY TABLE subset1 AS (select image1,
                             image2, score from I_2008_2009_long where
                             image1=imageOne order by rand() limit 2);
                        -- insert best value into best_of_two
                        insert into I_2008_2009_long_best_of_two (image1,
                             image2, score) (select image1,image2, min(
                             score) from subset1);
                        -- insert mean value into mean_of_two
                        insert into I_2008_2009_long_mean_of_two (image1,
                             image2, score) (select image1,image2, round(
                             avg(score),7) from subset1);
                         DROP TABLE subset1;
                                IF a >= no_of_subset THEN
                                        LEAVE simple_loop;
                                 END IF;
                                END LOOP simple_loop;

        -- input parameter as 3
                set input_parameter =3;
                set no_of_subset= (len+1)/3;
                set a=0;
                simple_loop: LOOP
                         SET a=a+1;

                        -- this line generates random subset of length 2.
```

```
                CREATE TEMPORARY TABLE subset1 AS (select image1,
                     image2, score from I_2008_2009_long where
                     image1=imageOne order by rand() limit 3);
                -- insert best value into best_of_three
                insert into I_2008_2009_long_best_of_three (image1
                     , image2, score) (select image1,image2, min(
                     score) from subset1);
                -- insert mean value into mean_of_three
                insert into I_2008_2009_long_mean_of_three (image1
                     , image2, score) (select image1,image2, round
                     (avg(score),7) from subset1);
                -- insert median value into median_of_three
                insert into I_2008_2009_long_median_of_three (
                     image1, image2, score) (select image1, image2
                     , score from subset1 order by score  limit
                     1,1);
                -- insert average of first two values (median
                     better)
                insert into I_2008_2009_long_mean_of_best_three (
                     image1, image2, score) (select image1, image2
                     ,round(avg(score),7) from subset1  order by
                     score limit 2);

                DROP TABLE subset1;
                        IF a >= no_of_subset THEN
                                LEAVE simple_loop;
                         END IF;
                        END LOOP simple_loop;

ELSEIF (len > 5) then
-- subset processing

        -- input parameter as 2
        set input_parameter =2;
        set no_of_subset= (len+1)/2;
        set a=0;
        simple_loop: LOOP
                 SET a=a+1;

                -- this line generates random subset of length 2.
                CREATE TEMPORARY TABLE subset1 AS (select image1,
                     image2, score from I_2008_2009_long where
                     image1=imageOne order by rand() limit 2);
                -- insert best value into best_of_two
                insert into I_2008_2009_long_best_of_two (image1,
                     image2, score) (select image1,image2, min(
                     score) from subset1);
                -- insert mean value into mean_of_two
                insert into I_2008_2009_long_mean_of_two (image1,
                     image2, score) (select image1,image2, round(
                     avg(score),7) from subset1);
                 DROP TABLE subset1;
                        IF a >= no_of_subset THEN
                                LEAVE simple_loop;
                         END IF;
                        END LOOP simple_loop;

-- input parameter as 3
        set input_parameter =3;
        set no_of_subset= (len+1)/3;
        set a=0;
        simple_loop: LOOP
                 SET a=a+1;

                -- this line generates random subset of length 2.
```

```
                CREATE TEMPORARY TABLE subset1 AS (select image1,
                     image2, score from I_2008_2009_long where
                     image1=imageOne order by rand() limit 3);
                -- insert best value into best_of_three
                insert into I_2008_2009_long_best_of_three (image1
                     , image2, score) (select image1,image2, min(
                     score) from subset1);
                -- insert mean value into mean_of_three
                insert into I_2008_2009_long_mean_of_three (image1
                     , image2, score) (select image1,image2, round
                     (avg(score),7) from subset1);
                -- insert median value into median_of_three
                insert into I_2008_2009_long_median_of_three (
                     image1, image2, score) (select image1, image2
                     , score from subset1 order by score  limit
                     1,1);
                -- insert average of first two values (median
                     better)
                insert into I_2008_2009_long_mean_of_best_three (
                     image1, image2, score) (select image1, image2
                     ,round(avg(score),7) from subset1  order by
                     score limit 2);

                DROP TABLE subset1;
                        IF a >= no_of_subset THEN
                                LEAVE simple_loop;
                         END IF;
                        END LOOP simple_loop;

-- input parameter as 5

        set input_parameter =5;
        set no_of_subset= (len+1)/5;
        set a=0;
        simple_loop: LOOP
                 SET a=a+1;

                -- this line generates random subset of length 2.
                CREATE TEMPORARY TABLE subset1 AS (select image1,
                     image2, score from I_2008_2009_long where
                     image1=imageOne order by rand() limit 5);
                -- insert best value into best_of_five
                insert into I_2008_2009_long_best_of_five (image1,
                      image2, score) (select image1,image2, min(
                     score) from subset1);
                -- insert mean value into mean_of_five
                insert into I_2008_2009_long_mean_of_five (image1,
                      image2, score) (select image1,image2, round(
                     avg(score),7) from subset1);
                -- insert median value into median_of_five
                insert into I_2008_2009_long_median_of_five (
                     image1, image2, score) (select image1, image2
                     , score from subset1 order by score  limit
                     1,1);
                -- insert average of first two values (median
                     better)
                insert into I_2008_2009_long_mean_of_best_five (
                     image1, image2, score) (select image1, image2
                     ,round(avg(score),7) from subset1  order by
                     score limit 2);

                DROP TABLE subset1;
                        IF a >= no_of_subset THEN
                                LEAVE simple_loop;
                         END IF;
```

```
                                        END LOOP simple_loop;
                END IF;

        UNTIL done END REPEAT;
        CLOSE image1s;

END ;;
DELIMITER ;
/*!50003 SET sql_mode              = @saved_sql_mode */ ;
/*!50003 SET character_set_client  = @saved_cs_client */ ;
/*!50003 SET character_set_results = @saved_cs_results */ ;
/*!50003 SET collation_connection  = @saved_col_connection */ ;
/*!50003 DROP PROCEDURE IF EXISTS `I_2008_2009_short` */;
/*!50003 SET @saved_cs_client      = @@character_set_client */ ;
/*!50003 SET @saved_cs_results     = @@character_set_results */ ;
/*!50003 SET @saved_col_connection = @@collation_connection */ ;
/*!50003 SET character_set_client  = utf8 */ ;
/*!50003 SET character_set_results = utf8 */ ;
/*!50003 SET collation_connection  = utf8_general_ci */ ;
/*!50003 SET @saved_sql_mode       = @@sql_mode */ ;
/*!50003 SET sql_mode              = 'STRICT_TRANS_TABLES,NO_AUTO_CREATE_USER,
     NO_ENGINE_SUBSTITUTION' */ ;
DELIMITER ;;
CREATE DEFINER=`root`@`localhost` PROCEDURE `I_2008_2009_short`()
BEGIN
declare imageOne varchar(12);
declare median_row_index int;
declare len INT;
declare minScore int;
declare no_of_subset INT;
declare input_parameter INT;
DECLARE done INT DEFAULT FALSE;
DECLARE a INT Default 0 ;
declare image1s cursor for select distinct image1 from I_2008_2009_short;
DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = 1;

delete from I_2008_2009_short_best_of_two;
delete from I_2008_2009_short_mean_of_two;
delete from I_2008_2009_short_best_of_three;
delete from I_2008_2009_short_mean_of_three;
delete from I_2008_2009_short_best_of_five;
delete from I_2008_2009_short_mean_of_five;
delete from I_2008_2009_short_median_of_three;
delete from I_2008_2009_short_median_of_five;
delete from I_2008_2009_short_mean_of_best_three;

open image1s;
REPEAT
        FETCH image1s INTO imageOne;
        select count(*) INTO len from I_2008_2009_short where image1=imageOne;
        IF(len = 2) then

                -- insert best value into best_of_two
                insert into I_2008_2009_short_best_of_two (image1, image2, score)
                     (select image1, image2, min(score) from I_2008_2009_short
                     where image1=imageOne);
                -- insert mean value into mean_of_two
                insert into I_2008_2009_short_mean_of_two (image1, image2, score)
                     (select image1, image2, round(avg(score),7) from
                     I_2008_2009_short where image1=imageOne);

        ELSEIF (len = 3) then

                -- insert best value into best_of_three
```

```
        insert into I_2008_2009_short_best_of_three (image1, image2, score
             ) (select image1, image2, min(score) from I_2008_2009_short
             where image1=imageOne);
        -- insert mean value into mean_of_three
        insert into I_2008_2009_short_mean_of_three (image1, image2, score
             ) (select image1, image2, round(avg(score),7) from
             I_2008_2009_short where image1=imageOne);
        -- insert median value into median_of_three
        insert into I_2008_2009_short_median_of_three (image1, image2,
             score) (select image1, image2, score from I_2008_2009_short
             where image1=imageOne order by score  limit 1,1);
        -- insert average of first two values (median better)
        insert into I_2008_2009_short_mean_of_best_three (image1, image2,
             score) (select image1, image2,round(avg(score),7) from
             I_2008_2009_short where image1=imageOne order by score limit
             2);

        -- subset processing

                -- input parameter as 2
                set input_parameter =2;
                set no_of_subset= (len+1)/2;
                set a=0;

                simple_loop: LOOP
                         SET a=a+1;

                        -- this line generates random subset of length 2.
                        CREATE TEMPORARY TABLE subset1 AS (select image1,
                             image2, score from I_2008_2009_short where
                             image1=imageOne order by rand() limit 2);
                        -- insert best value into best_of_two
                        insert into I_2008_2009_short_best_of_two (image1,
                              image2, score) (select image1,image2, min(
                             score) from subset1);
                        -- insert mean value into mean_of_two
                        insert into I_2008_2009_short_mean_of_two (image1,
                              image2, score) (select image1,image2, round(
                             avg(score),7) from subset1);
                         DROP TABLE subset1;
                                IF a >= no_of_subset THEN
                                        LEAVE simple_loop;
                                 END IF;
                                END LOOP simple_loop;

ELSEIF (len = 4) then

-- subset processing

        -- input parameter as 2
        set input_parameter =2;
        set no_of_subset= (len+1)/2;
        set a=0;
        simple_loop: LOOP
                 SET a=a+1;

                -- this line generates random subset of length 2.
                CREATE TEMPORARY TABLE subset1 AS (select image1, image2,
                     score from I_2008_2009_short where image1=imageOne
                     order by rand() limit 2);
                -- insert best value into best_of_two
                insert into I_2008_2009_short_best_of_two (image1, image2,
                      score) (select image1,image2, min(score) from
                     subset1);
                -- insert mean value into mean_of_two
```

```
                insert into I_2008_2009_short_mean_of_two (image1, image2,
                     score) (select image1,image2, round(avg(score),7)
                     from subset1);
                 DROP TABLE subset1;
                        IF a >= no_of_subset THEN
                                LEAVE simple_loop;
                         END IF;
                        END LOOP simple_loop;

-- input parameter as 3
        set input_parameter =3;
        set no_of_subset= (len+1)/3;
        set a=0;
        simple_loop: LOOP
                 SET a=a+1;

                -- this line generates random subset of length 2.
                CREATE TEMPORARY TABLE subset1 AS (select image1, image2,
                     score from I_2008_2009_short where image1=imageOne
                     order by rand() limit 3);
                select * from subset1;
                select min(score) from subset1;
                -- insert best value into best_of_three
                insert into I_2008_2009_short_best_of_three (image1,
                     image2, score) (select image1,image2, min(score) from
                      subset1);
                -- insert mean value into mean_of_three
                insert into I_2008_2009_short_mean_of_three (image1,
                     image2, score) (select image1,image2, round(avg(score
                     ),7) from subset1);
                -- insert median value into median_of_three
                insert into I_2008_2009_short_median_of_three (image1,
                     image2, score) (select image1, image2, score from
                     subset1 order by score  limit 1,1);
                -- insert average of first two values (median better)
                insert into I_2008_2009_short_mean_of_best_three (image1,
                     image2, score) (select image1, image2,round(avg(score
                     ),7) from subset1  order by score limit 2);

                DROP TABLE subset1;
                        IF a >= no_of_subset THEN
                                LEAVE simple_loop;
                         END IF;
                        END LOOP simple_loop;

ELSEIF (len = 5) then

        -- insert best value into best_of_five
        insert into I_2008_2009_short_best_of_five (image1, image2, score)
              (select image1, image2, min(score) from I_2008_2009_short
             where image1=imageOne);
        -- insert mean value into mean_of_five
        insert into I_2008_2009_short_mean_of_five (image1, image2, score)
              (select image1, image2, round(avg(score),7) from
             I_2008_2009_short where image1=imageOne);
        -- insert median value into median_of_five
        insert into I_2008_2009_short_median_of_five (image1, image2,
             score) (select image1, image2, score from I_2008_2009_short
             where image1=imageOne order by score  limit 1,1);
        -- insert average of best 3 values (median better)
        insert into I_2008_2009_short_mean_of_best_five (image1, image2,
             score) (select image1, image2,round(avg(score),7) from
             I_2008_2009_short where image1=imageOne order by score limit
             2);
```

```
-- subset processing

        -- input parameter as 2
        set input_parameter =2;
        set no_of_subset= (len+1)/2;
        set a=0;
        simple_loop: LOOP
                 SET a=a+1;

                -- this line generates random subset of length 2.
                CREATE TEMPORARY TABLE subset1 AS (select image1,
                     image2, score from I_2008_2009_short where
                     image1=imageOne order by rand() limit 2);
                -- insert best value into best_of_two
                insert into I_2008_2009_short_best_of_two (image1,
                      image2, score) (select image1,image2, min(
                     score) from subset1);
                -- insert mean value into mean_of_two
                insert into I_2008_2009_short_mean_of_two (image1,
                      image2, score) (select image1,image2, round(
                     avg(score),7) from subset1);
                 DROP TABLE subset1;
                        IF a >= no_of_subset THEN
                                LEAVE simple_loop;
                         END IF;
                        END LOOP simple_loop;

-- input parameter as 3
        set input_parameter =3;
        set no_of_subset= (len+1)/3;
        set a=0;
        simple_loop: LOOP
                 SET a=a+1;

                -- this line generates random subset of length 2.
                CREATE TEMPORARY TABLE subset1 AS (select image1,
                     image2, score from I_2008_2009_short where
                     image1=imageOne order by rand() limit 3);
                select * from subset1;
                select min(score) from subset1;
                -- insert best value into best_of_three
                insert into I_2008_2009_short_best_of_three (
                     image1, image2, score) (select image1,image2,
                      min(score) from subset1);
                -- insert mean value into mean_of_three
                insert into I_2008_2009_short_mean_of_three (
                     image1, image2, score) (select image1,image2,
                      round(avg(score),7) from subset1);
                -- insert median value into median_of_three
                insert into I_2008_2009_short_median_of_three (
                     image1, image2, score) (select image1, image2
                     , score from subset1 order by score  limit
                     1,1);
                -- insert average of first two values (median
                     better)
                insert into I_2008_2009_short_mean_of_best_three (
                     image1, image2, score) (select image1, image2
                     ,round(avg(score),7) from subset1  order by
                     score limit 2);

                DROP TABLE subset1;
                        IF a >= no_of_subset THEN
                                LEAVE simple_loop;
                         END IF;
```

```
                        END LOOP simple_loop;

ELSEIF (len > 5) then
-- subset processing

        -- input parameter as 2
        set input_parameter =2;
        set no_of_subset= (len+1)/2;
        set a=0;
        simple_loop: LOOP
                 SET a=a+1;

                -- this line generates random subset of length 2.
                CREATE TEMPORARY TABLE subset1 AS (select image1,
                     image2, score from I_2008_2009_short where
                     image1=imageOne order by rand() limit 2);
                -- insert best value into best_of_two
                insert into I_2008_2009_short_best_of_two (image1,
                      image2, score) (select image1,image2, min(
                     score) from subset1);
                -- insert mean value into mean_of_two
                insert into I_2008_2009_short_mean_of_two (image1,
                      image2, score) (select image1,image2, round(
                     avg(score),7) from subset1);
                 DROP TABLE subset1;
                        IF a >= no_of_subset THEN
                                LEAVE simple_loop;
                         END IF;
                        END LOOP simple_loop;

-- input parameter as 3
        set input_parameter =3;
        set no_of_subset= (len+1)/3;
        set a=0;
        simple_loop: LOOP
                 SET a=a+1;

                -- this line generates random subset of length 2.
                CREATE TEMPORARY TABLE subset1 AS (select image1,
                     image2, score from I_2008_2009_short where
                     image1=imageOne order by rand() limit 3);
                select * from subset1;
                select min(score) from subset1;
                -- insert best value into best_of_three
                insert into I_2008_2009_short_best_of_three (
                     image1, image2, score) (select image1,image2,
                      min(score) from subset1);
                -- insert mean value into mean_of_three
                insert into I_2008_2009_short_mean_of_three (
                     image1, image2, score) (select image1,image2,
                      round(avg(score),7) from subset1);
                -- insert median value into median_of_three
                insert into I_2008_2009_short_median_of_three (
                     image1, image2, score) (select image1, image2
                     , score from subset1 order by score  limit
                     1,1);
                -- insert average of first two values (median
                     better)
                insert into I_2008_2009_short_mean_of_best_three (
                     image1, image2, score) (select image1, image2
                     ,round(avg(score),7) from subset1  order by
                     score limit 2);

                DROP TABLE subset1;
                        IF a >= no_of_subset THEN
```

```
                                        LEAVE simple_loop;
                                 END IF;
                                END LOOP simple_loop;

                -- input parameter as 5

                        set input_parameter =5;
                        set no_of_subset= (len+1)/5;
                        set a=0;
                        simple_loop: LOOP
                                 SET a=a+1;

                                -- this line generates random subset of length 2.
                                CREATE TEMPORARY TABLE subset1 AS (select image1,
                                     image2, score from I_2008_2009_short where
                                     image1=imageOne order by rand() limit 5);
                                -- insert best value into best_of_five
                                insert into I_2008_2009_short_best_of_five (image1
                                     , image2, score) (select image1,image2, min(
                                     score) from subset1);
                                -- insert mean value into mean_of_five
                                insert into I_2008_2009_short_mean_of_five (image1
                                     , image2, score) (select image1,image2, round
                                     (avg(score),7) from subset1);
                                -- insert median value into median_of_five
                                insert into I_2008_2009_short_median_of_five (
                                     image1, image2, score) (select image1, image2
                                     , score from subset1 order by score  limit
                                     1,1);
                                -- insert average of first two values (median
                                     better)
                                insert into I_2008_2009_short_mean_of_best_five (
                                     image1, image2, score) (select image1, image2
                                     ,round(avg(score),7) from subset1  order by
                                     score limit 2);

                                DROP TABLE subset1;
                                        IF a >= no_of_subset THEN
                                                LEAVE simple_loop;
                                         END IF;
                                        END LOOP simple_loop;
                END IF;

        UNTIL done END REPEAT;
        CLOSE image1s;

END ;;
DELIMITER ;
/*!50003 SET sql_mode              = @saved_sql_mode */ ;
/*!50003 SET character_set_client  = @saved_cs_client */ ;
/*!50003 SET character_set_results = @saved_cs_results */ ;
/*!50003 SET collation_connection  = @saved_col_connection */ ;
/*!50003 DROP PROCEDURE IF EXISTS `I_2008_2010_long` */;
/*!50003 SET @saved_cs_client      = @@character_set_client */ ;
/*!50003 SET @saved_cs_results     = @@character_set_results */ ;
/*!50003 SET @saved_col_connection = @@collation_connection */ ;
/*!50003 SET character_set_client  = utf8 */ ;
/*!50003 SET character_set_results = utf8 */ ;
/*!50003 SET collation_connection  = utf8_general_ci */ ;
/*!50003 SET @saved_sql_mode       = @@sql_mode */ ;
/*!50003 SET sql_mode              = 'STRICT_TRANS_TABLES,NO_AUTO_CREATE_USER,
     NO_ENGINE_SUBSTITUTION' */ ;
DELIMITER ;;
CREATE DEFINER=`root`@`localhost` PROCEDURE `I_2008_2010_long`()
BEGIN
```

```
declare imageOne varchar(12);
declare median_row_index int;
declare len INT;
declare minScore int;
declare no_of_subset INT;
declare input_parameter INT;
DECLARE done INT DEFAULT FALSE;
DECLARE a INT Default 0 ;
declare image1s cursor for select distinct image1 from I_2008_2010_long;
DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = 1;

delete from I_2008_2010_long_best_of_two;
delete from I_2008_2010_long_mean_of_two;
delete from I_2008_2010_long_best_of_three;
delete from I_2008_2010_long_mean_of_three;
delete from I_2008_2010_long_best_of_five;
delete from I_2008_2010_long_mean_of_five;
delete from I_2008_2010_long_median_of_three;
delete from I_2008_2010_long_median_of_five;
delete from I_2008_2010_long_mean_of_best_three;

open image1s;
REPEAT
        FETCH image1s INTO imageOne;
        select count(*) INTO len from I_2008_2010_long where image1=imageOne;
        IF(len = 2) then

                -- insert best value into best_of_two
                insert into I_2008_2010_long_best_of_two (image1, image2, score) (
                     select image1, image2, min(score) from I_2008_2010_long where
                      image1=imageOne);
                -- insert mean value into mean_of_two
                insert into I_2008_2010_long_mean_of_two (image1, image2, score) (
                     select image1, image2, round(avg(score),7) from
                     I_2008_2010_long where image1=imageOne);

        ELSEIF (len = 3) then

                -- insert best value into best_of_three
                insert into I_2008_2010_long_best_of_three (image1, image2, score)
                      (select image1, image2, min(score) from I_2008_2010_long
                     where image1=imageOne);
                -- insert mean value into mean_of_three
                insert into I_2008_2010_long_mean_of_three (image1, image2, score)
                      (select image1, image2, round(avg(score),7) from
                     I_2008_2010_long where image1=imageOne);
                -- insert median value into median_of_three
                insert into I_2008_2010_long_median_of_three (image1, image2,
                     score) (select image1, image2, score from I_2008_2010_long
                     where image1=imageOne order by score  limit 1,1);
                -- insert average of first two values (median better)
                insert into I_2008_2010_long_mean_of_best_three (image1, image2,
                     score) (select image1, image2,round(avg(score),7) from
                     I_2008_2010_long where image1=imageOne order by score limit
                     2);

                -- subset processing

                        -- input parameter as 2
                        set input_parameter =2;
                        set no_of_subset= (len+1)/2;
                        set a=0;

                        simple_loop: LOOP
                                 SET a=a+1;
```

```
                                -- this line generates random subset of length 2.
                                CREATE TEMPORARY TABLE subset1 AS (select image1,
                                     image2, score from I_2008_2010_long where
                                     image1=imageOne order by rand() limit 2);
                                -- insert best value into best_of_two
                                insert into I_2008_2010_long_best_of_two (image1,
                                     image2, score) (select image1,image2, min(
                                     score) from subset1);
                                -- insert mean value into mean_of_two
                                insert into I_2008_2010_long_mean_of_two (image1,
                                     image2, score) (select image1,image2, round(
                                     avg(score),7) from subset1);
                                 DROP TABLE subset1;
                                        IF a >= no_of_subset THEN
                                                LEAVE simple_loop;
                                         END IF;
                                        END LOOP simple_loop;

ELSEIF (len = 4) then

-- subset processing

        -- input parameter as 2
        set input_parameter =2;
        set no_of_subset= (len+1)/2;
        set a=0;
        simple_loop: LOOP
                 SET a=a+1;

                -- this line generates random subset of length 2.
                CREATE TEMPORARY TABLE subset1 AS (select image1, image2,
                     score from I_2008_2010_long where image1=imageOne
                     order by rand() limit 2);
                -- insert best value into best_of_two
                insert into I_2008_2010_long_best_of_two (image1, image2,
                     score) (select image1,image2, min(score) from subset1
                     );
                -- insert mean value into mean_of_two
                insert into I_2008_2010_long_mean_of_two (image1, image2,
                     score) (select image1,image2, round(avg(score),7)
                     from subset1);
                 DROP TABLE subset1;
                        IF a >= no_of_subset THEN
                                LEAVE simple_loop;
                         END IF;
                        END LOOP simple_loop;

-- input parameter as 3
        set input_parameter =3;
        set no_of_subset= (len+1)/3;
        set a=0;
        simple_loop: LOOP
                 SET a=a+1;

                -- this line generates random subset of length 2.
                CREATE TEMPORARY TABLE subset1 AS (select image1, image2,
                     score from I_2008_2010_long where image1=imageOne
                     order by rand() limit 3);
                -- insert best value into best_of_three
                insert into I_2008_2010_long_best_of_three (image1, image2
                     , score) (select image1,image2, min(score) from
                     subset1);
                -- insert mean value into mean_of_three
```

```
                insert into I_2008_2010_long_mean_of_three (image1, image2
                     , score) (select image1,image2, round(avg(score),7)
                     from subset1);
                -- insert median value into median_of_three
                insert into I_2008_2010_long_median_of_three (image1,
                     image2, score) (select image1, image2, score from
                     subset1 order by score  limit 1,1);
                -- insert average of first two values (median better)
                insert into I_2008_2010_long_mean_of_best_three (image1,
                     image2, score) (select image1, image2,round(avg(score
                     ),7) from subset1  order by score limit 2);

                DROP TABLE subset1;
                        IF a >= no_of_subset THEN
                                LEAVE simple_loop;
                         END IF;
                        END LOOP simple_loop;

ELSEIF (len = 5) then

        -- insert best value into best_of_five
        insert into I_2008_2010_long_best_of_five (image1, image2, score)
             (select image1, image2, min(score) from I_2008_2010_long
             where image1=imageOne);
        -- insert mean value into mean_of_five
        insert into I_2008_2010_long_mean_of_five (image1, image2, score)
             (select image1, image2, round(avg(score),7) from
             I_2008_2010_long where image1=imageOne);
        -- insert median value into median_of_five
        insert into I_2008_2010_long_median_of_five (image1, image2, score
             ) (select image1, image2, score from I_2008_2010_long where
             image1=imageOne order by score  limit 1,1);
        -- insert average of best 3 values (median better)
        insert into I_2008_2010_long_mean_of_best_five (image1, image2,
             score) (select image1, image2,round(avg(score),7) from
             I_2008_2010_long where image1=imageOne order by score limit
             2);


        -- subset processing

                -- input parameter as 2
                set input_parameter =2;
                set no_of_subset= (len+1)/2;
                set a=0;
                simple_loop: LOOP
                         SET a=a+1;

                        -- this line generates random subset of length 2.
                        CREATE TEMPORARY TABLE subset1 AS (select image1,
                             image2, score from I_2008_2010_long where
                             image1=imageOne order by rand() limit 2);
                        -- insert best value into best_of_two
                        insert into I_2008_2010_long_best_of_two (image1,
                             image2, score) (select image1,image2, min(
                             score) from subset1);
                        -- insert mean value into mean_of_two
                        insert into I_2008_2010_long_mean_of_two (image1,
                             image2, score) (select image1,image2, round(
                             avg(score),7) from subset1);
                         DROP TABLE subset1;
                                IF a >= no_of_subset THEN
                                        LEAVE simple_loop;
                                 END IF;
                                END LOOP simple_loop;
```

```
-- input parameter as 3
        set input_parameter =3;
        set no_of_subset= (len+1)/3;
        set a=0;
        simple_loop: LOOP
                 SET a=a+1;

                -- this line generates random subset of length 2.
                CREATE TEMPORARY TABLE subset1 AS (select image1,
                     image2, score from I_2008_2010_long where
                     image1=imageOne order by rand() limit 3);
                -- insert best value into best_of_three
                insert into I_2008_2010_long_best_of_three (image1
                     , image2, score) (select image1,image2, min(
                     score) from subset1);
                -- insert mean value into mean_of_three
                insert into I_2008_2010_long_mean_of_three (image1
                     , image2, score) (select image1,image2, round
                     (avg(score),7) from subset1);
                -- insert median value into median_of_three
                insert into I_2008_2010_long_median_of_three (
                     image1, image2, score) (select image1, image2
                     , score from subset1 order by score  limit
                     1,1);
                -- insert average of first two values (median
                     better)
                insert into I_2008_2010_long_mean_of_best_three (
                     image1, image2, score) (select image1, image2
                     ,round(avg(score),7) from subset1  order by
                     score limit 2);

                DROP TABLE subset1;
                        IF a >= no_of_subset THEN
                                LEAVE simple_loop;
                         END IF;
                        END LOOP simple_loop;

ELSEIF (len > 5) then
-- subset processing

        -- input parameter as 2
        set input_parameter =2;
        set no_of_subset= (len+1)/2;
        set a=0;
        simple_loop: LOOP
                 SET a=a+1;

                -- this line generates random subset of length 2.
                CREATE TEMPORARY TABLE subset1 AS (select image1,
                     image2, score from I_2008_2010_long where
                     image1=imageOne order by rand() limit 2);
                -- insert best value into best_of_two
                insert into I_2008_2010_long_best_of_two (image1,
                     image2, score) (select image1,image2, min(
                     score) from subset1);
                -- insert mean value into mean_of_two
                insert into I_2008_2010_long_mean_of_two (image1,
                     image2, score) (select image1,image2, round(
                     avg(score),7) from subset1);
                 DROP TABLE subset1;
                        IF a >= no_of_subset THEN
                                LEAVE simple_loop;
                         END IF;
                        END LOOP simple_loop;
```

```
-- input parameter as 3
        set input_parameter =3;
        set no_of_subset= (len+1)/3;
        set a=0;
        simple_loop: LOOP
                 SET a=a+1;

                -- this line generates random subset of length 2.
                CREATE TEMPORARY TABLE subset1 AS (select image1,
                     image2, score from I_2008_2010_long where
                     image1=imageOne order by rand() limit 3);
                -- insert best value into best_of_three
                insert into I_2008_2010_long_best_of_three (image1
                     , image2, score) (select image1,image2, min(
                     score) from subset1);
                -- insert mean value into mean_of_three
                insert into I_2008_2010_long_mean_of_three (image1
                     , image2, score) (select image1,image2, round
                     (avg(score),7) from subset1);
                -- insert median value into median_of_three
                insert into I_2008_2010_long_median_of_three (
                     image1, image2, score) (select image1, image2
                     , score from subset1 order by score  limit
                     1,1);
                -- insert average of first two values (median
                     better)
                insert into I_2008_2010_long_mean_of_best_three (
                     image1, image2, score) (select image1, image2
                     ,round(avg(score),7) from subset1  order by
                     score limit 2);

                DROP TABLE subset1;
                        IF a >= no_of_subset THEN
                                LEAVE simple_loop;
                         END IF;
                        END LOOP simple_loop;

-- input parameter as 5

        set input_parameter =5;
        set no_of_subset= (len+1)/5;
        set a=0;
        simple_loop: LOOP
                 SET a=a+1;

                -- this line generates random subset of length 2.
                CREATE TEMPORARY TABLE subset1 AS (select image1,
                     image2, score from I_2008_2010_long where
                     image1=imageOne order by rand() limit 5);
                -- insert best value into best_of_five
                insert into I_2008_2010_long_best_of_five (image1,
                      image2, score) (select image1,image2, min(
                     score) from subset1);
                -- insert mean value into mean_of_five
                insert into I_2008_2010_long_mean_of_five (image1,
                      image2, score) (select image1,image2, round(
                     avg(score),7) from subset1);
                -- insert median value into median_of_five
                insert into I_2008_2010_long_median_of_five (
                     image1, image2, score) (select image1, image2
                     , score from subset1 order by score  limit
                     1,1);
                -- insert average of first two values (median
                     better)
```

```
                                insert into I_2008_2010_long_mean_of_best_five (
                                     image1, image2, score) (select image1, image2
                                     ,round(avg(score),7) from subset1  order by
                                     score limit 2);

                                DROP TABLE subset1;
                                        IF a >= no_of_subset THEN
                                                LEAVE simple_loop;
                                         END IF;
                                        END LOOP simple_loop;
                END IF;

        UNTIL done END REPEAT;
        CLOSE image1s;

END ;;
DELIMITER ;
/*!50003 SET sql_mode              = @saved_sql_mode */ ;
/*!50003 SET character_set_client  = @saved_cs_client */ ;
/*!50003 SET character_set_results = @saved_cs_results */ ;
/*!50003 SET collation_connection  = @saved_col_connection */ ;
/*!50003 DROP PROCEDURE IF EXISTS `I_2008_2010_short` */;
/*!50003 SET @saved_cs_client      = @@character_set_client */ ;
/*!50003 SET @saved_cs_results     = @@character_set_results */ ;
/*!50003 SET @saved_col_connection = @@collation_connection */ ;
/*!50003 SET character_set_client  = utf8 */ ;
/*!50003 SET character_set_results = utf8 */ ;
/*!50003 SET collation_connection  = utf8_general_ci */ ;
/*!50003 SET @saved_sql_mode       = @@sql_mode */ ;
/*!50003 SET sql_mode              = 'STRICT_TRANS_TABLES,NO_AUTO_CREATE_USER,
     NO_ENGINE_SUBSTITUTION' */ ;
DELIMITER ;;
CREATE DEFINER=`root`@`localhost` PROCEDURE `I_2008_2010_short`()
BEGIN
declare imageOne varchar(12);
declare median_row_index int;
declare len INT;
declare minScore int;
declare no_of_subset INT;
declare input_parameter INT;
DECLARE done INT DEFAULT FALSE;
DECLARE a INT Default 0 ;
declare image1s cursor for select distinct image1 from I_2008_2010_short;
DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = 1;

delete from I_2008_2010_short_best_of_two;
delete from I_2008_2010_short_mean_of_two;
delete from I_2008_2010_short_best_of_three;
delete from I_2008_2010_short_mean_of_three;
delete from I_2008_2010_short_best_of_five;
delete from I_2008_2010_short_mean_of_five;
delete from I_2008_2010_short_median_of_three;
delete from I_2008_2010_short_median_of_five;
delete from I_2008_2010_short_mean_of_best_three;

open image1s;
REPEAT
        FETCH image1s INTO imageOne;
        select count(*) INTO len from I_2008_2010_short where image1=imageOne;
        IF(len = 2) then

                -- insert best value into best_of_two
                insert into I_2008_2010_short_best_of_two (image1, image2, score)
                     (select image1, image2, min(score) from I_2008_2010_short
                     where image1=imageOne);
```

```
        -- insert mean value into mean_of_two
        insert into I_2008_2010_short_mean_of_two (image1, image2, score)
             (select image1, image2, round(avg(score),7) from
             I_2008_2010_short where image1=imageOne);

ELSEIF (len = 3) then

        -- insert best value into best_of_three
        insert into I_2008_2010_short_best_of_three (image1, image2, score
             ) (select image1, image2, min(score) from I_2008_2010_short
             where image1=imageOne);
        -- insert mean value into mean_of_three
        insert into I_2008_2010_short_mean_of_three (image1, image2, score
             ) (select image1, image2, round(avg(score),7) from
             I_2008_2010_short where image1=imageOne);
        -- insert median value into median_of_three
        insert into I_2008_2010_short_median_of_three (image1, image2,
             score) (select image1, image2, score from I_2008_2010_short
             where image1=imageOne order by score  limit 1,1);
        -- insert average of first two values (median better)
        insert into I_2008_2010_short_mean_of_best_three (image1, image2,
             score) (select image1, image2,round(avg(score),7) from
             I_2008_2010_short where image1=imageOne order by score limit
             2);

        -- subset processing

                -- input parameter as 2
                set input_parameter =2;
                set no_of_subset= (len+1)/2;
                set a=0;

                simple_loop: LOOP
                         SET a=a+1;

                        -- this line generates random subset of length 2.
                        CREATE TEMPORARY TABLE subset1 AS (select image1,
                             image2, score from I_2008_2010_short where
                             image1=imageOne order by rand() limit 2);
                        -- insert best value into best_of_two
                        insert into I_2008_2010_short_best_of_two (image1,
                              image2, score) (select image1,image2, min(
                             score) from subset1);
                        -- insert mean value into mean_of_two
                        insert into I_2008_2010_short_mean_of_two (image1,
                              image2, score) (select image1,image2, round(
                             avg(score),7) from subset1);
                         DROP TABLE subset1;
                                IF a >= no_of_subset THEN
                                        LEAVE simple_loop;
                                 END IF;
                                END LOOP simple_loop;

ELSEIF (len = 4) then

-- subset processing

        -- input parameter as 2
        set input_parameter =2;
        set no_of_subset= (len+1)/2;
        set a=0;
        simple_loop: LOOP
                 SET a=a+1;

                -- this line generates random subset of length 2.
```

```
                CREATE TEMPORARY TABLE subset1 AS (select image1, image2,
                     score from I_2008_2010_short where image1=imageOne
                     order by rand() limit 2);
                -- insert best value into best_of_two
                insert into I_2008_2010_short_best_of_two (image1, image2,
                      score) (select image1,image2, min(score) from
                     subset1);
                -- insert mean value into mean_of_two
                insert into I_2008_2010_short_mean_of_two (image1, image2,
                      score) (select image1,image2, round(avg(score),7)
                     from subset1);
                 DROP TABLE subset1;
                        IF a >= no_of_subset THEN
                                LEAVE simple_loop;
                         END IF;
                        END LOOP simple_loop;

-- input parameter as 3
        set input_parameter =3;
        set no_of_subset= (len+1)/3;
        set a=0;
        simple_loop: LOOP
                 SET a=a+1;

                -- this line generates random subset of length 2.
                CREATE TEMPORARY TABLE subset1 AS (select image1, image2,
                     score from I_2008_2010_short where image1=imageOne
                     order by rand() limit 3);
                -- insert best value into best_of_three
                insert into I_2008_2010_short_best_of_three (image1,
                     image2, score) (select image1,image2, min(score) from
                      subset1);
                -- insert mean value into mean_of_three
                insert into I_2008_2010_short_mean_of_three (image1,
                     image2, score) (select image1,image2, round(avg(score
                     ),7) from subset1);
                -- insert median value into median_of_three
                insert into I_2008_2010_short_median_of_three (image1,
                     image2, score) (select image1, image2, score from
                     subset1 order by score  limit 1,1);
                -- insert average of first two values (median better)
                insert into I_2008_2010_short_mean_of_best_three (image1,
                     image2, score) (select image1, image2,round(avg(score
                     ),7) from subset1  order by score limit 2);

                DROP TABLE subset1;
                        IF a >= no_of_subset THEN
                                LEAVE simple_loop;
                         END IF;
                        END LOOP simple_loop;

ELSEIF (len = 5) then

        -- insert best value into best_of_five
        insert into I_2008_2010_short_best_of_five (image1, image2, score)
              (select image1, image2, min(score) from I_2008_2010_short
             where image1=imageOne);
        -- insert mean value into mean_of_five
        insert into I_2008_2010_short_mean_of_five (image1, image2, score)
              (select image1, image2, round(avg(score),7) from
             I_2008_2010_short where image1=imageOne);
        -- insert median value into median_of_five
        insert into I_2008_2010_short_median_of_five (image1, image2,
             score) (select image1, image2, score from I_2008_2010_short
             where image1=imageOne order by score  limit 1,1);
```

```
-- insert average of best 3 values (median better)
insert into I_2008_2010_short_mean_of_best_five (image1, image2,
     score) (select image1, image2,round(avg(score),7) from
     I_2008_2010_short where image1=imageOne order by score limit
     2);


-- subset processing

        -- input parameter as 2
        set input_parameter =2;
        set no_of_subset= (len+1)/2;
        set a=0;
        simple_loop: LOOP
                 SET a=a+1;

                -- this line generates random subset of length 2.
                CREATE TEMPORARY TABLE subset1 AS (select image1,
                     image2, score from I_2008_2010_short where
                     image1=imageOne order by rand() limit 2);
                -- insert best value into best_of_two
                insert into I_2008_2010_short_best_of_two (image1,
                      image2, score) (select image1,image2, min(
                     score) from subset1);
                -- insert mean value into mean_of_two
                insert into I_2008_2010_short_mean_of_two (image1,
                      image2, score) (select image1,image2, round(
                     avg(score),7) from subset1);
                 DROP TABLE subset1;
                        IF a >= no_of_subset THEN
                                LEAVE simple_loop;
                         END IF;
                        END LOOP simple_loop;

-- input parameter as 3
        set input_parameter =3;
        set no_of_subset= (len+1)/3;
        set a=0;
        simple_loop: LOOP
                 SET a=a+1;

                -- this line generates random subset of length 2.
                CREATE TEMPORARY TABLE subset1 AS (select image1,
                     image2, score from I_2008_2010_short where
                     image1=imageOne order by rand() limit 3);
                -- insert best value into best_of_three
                insert into I_2008_2010_short_best_of_three (
                     image1, image2, score) (select image1,image2,
                      min(score) from subset1);
                -- insert mean value into mean_of_three
                insert into I_2008_2010_short_mean_of_three (
                     image1, image2, score) (select image1,image2,
                      round(avg(score),7) from subset1);
                -- insert median value into median_of_three
                insert into I_2008_2010_short_median_of_three (
                     image1, image2, score) (select image1, image2
                     , score from subset1 order by score  limit
                     1,1);
                -- insert average of first two values (median
                     better)
                insert into I_2008_2010_short_mean_of_best_three (
                     image1, image2, score) (select image1, image2
                     ,round(avg(score),7) from subset1  order by
                     score limit 2);
```

```
                DROP TABLE subset1;
                        IF a >= no_of_subset THEN
                                LEAVE simple_loop;
                         END IF;
                        END LOOP simple_loop;

ELSEIF (len > 5) then
-- subset processing

        -- input parameter as 2
        set input_parameter =2;
        set no_of_subset= (len+1)/2;
        set a=0;
        simple_loop: LOOP
                 SET a=a+1;

                -- this line generates random subset of length 2.
                CREATE TEMPORARY TABLE subset1 AS (select image1,
                     image2, score from I_2008_2010_short where
                     image1=imageOne order by rand() limit 2);
                -- insert best value into best_of_two
                insert into I_2008_2010_short_best_of_two (image1,
                      image2, score) (select image1,image2, min(
                     score) from subset1);
                -- insert mean value into mean_of_two
                insert into I_2008_2010_short_mean_of_two (image1,
                      image2, score) (select image1,image2, round(
                     avg(score),7) from subset1);
                 DROP TABLE subset1;
                        IF a >= no_of_subset THEN
                                LEAVE simple_loop;
                         END IF;
                        END LOOP simple_loop;

-- input parameter as 3
        set input_parameter =3;
        set no_of_subset= (len+1)/3;
        set a=0;
        simple_loop: LOOP
                 SET a=a+1;

                -- this line generates random subset of length 2.
                CREATE TEMPORARY TABLE subset1 AS (select image1,
                     image2, score from I_2008_2010_short where
                     image1=imageOne order by rand() limit 3);
                -- insert best value into best_of_three
                insert into I_2008_2010_short_best_of_three (
                     image1, image2, score) (select image1,image2,
                      min(score) from subset1);
                -- insert mean value into mean_of_three
                insert into I_2008_2010_short_mean_of_three (
                     image1, image2, score) (select image1,image2,
                      round(avg(score),7) from subset1);
                -- insert median value into median_of_three
                insert into I_2008_2010_short_median_of_three (
                     image1, image2, score) (select image1, image2
                     , score from subset1 order by score  limit
                     1,1);
                -- insert average of first two values (median
                     better)
                insert into I_2008_2010_short_mean_of_best_three (
                     image1, image2, score) (select image1, image2
                     ,round(avg(score),7) from subset1  order by
                     score limit 2);
```

```
                        DROP TABLE subset1;
                                IF a >= no_of_subset THEN
                                        LEAVE simple_loop;
                                 END IF;
                                END LOOP simple_loop;

                -- input parameter as 5

                        set input_parameter =5;
                        set no_of_subset= (len+1)/5;
                        set a=0;
                        simple_loop: LOOP
                                 SET a=a+1;

                                -- this line generates random subset of length 2.
                                CREATE TEMPORARY TABLE subset1 AS (select image1,
                                     image2, score from I_2008_2010_short
                                where image1=imageOne order by rand() limit 5);
                                -- insert best value into best_of_five
                                insert into I_2008_2010_short_best_of_five (image1
                                     , image2, score) (select image1,image2,
                                min(score) from subset1);
                                -- insert mean value into mean_of_five
                                insert into I_2008_2010_short_mean_of_five (image1
                                     , image2, score) (select image1,image2,
                                round(avg(score),7) from subset1);
                                -- insert median value into median_of_five
                                insert into I_2008_2010_short_median_of_five (
                                     image1, image2, score) (select image1, image2
                                     , score from subset1 order by score  limit
                                     1,1);
                                -- insert average of first two values (median
                                     better)
                                insert into I_2008_2010_short_mean_of_best_five (
                                     image1, image2, score) (select image1, image2
                                     ,round(avg(score),7) from subset1  order by
                                     score limit 2);

                        DROP TABLE subset1;
                                IF a >= no_of_subset THEN
                                        LEAVE simple_loop;
                                 END IF;
                                END LOOP simple_loop;
                END IF;

        UNTIL done END REPEAT;
        CLOSE image1s;

END ;;
```

Bibliography

[All11] SC Alliance. Smart Cards and Biometrics . http://irisid.com/download/news/Smart_Cards_and_Biometrics_030111.pdf, 2011. [Online; accessed 23-January-2013].

[AMK+08] D. A. Atchison, E. L. Markwell, S. Kasthurirangan, J. M. Pope, G. Smith, and P. G. Swann. Age-related changes in optical and biometric characteristics of emmetropic eyes. *Journal of Vision*, 8(4), 2008.

[ARAK08] A. N. Al-Raisi and A. M. Al-Khouri. Iris recognition and the challenge of homeland and border control security in UAE. *Telematics and Informatics*, 25(2):117–132, 2008.

[ASK05] R. Abiantun, M. Savvides, and P. K. Khosla. Automatic eye-level height system for face and iris recognition systems. In *2005. Fourth IEEE Workshop on Automatic Identification Advanced Technologies*, pages 155–159. IEEE, 2005.

[BBF09] S. E. Baker, K. W. Bowyer, and P. J. Flynn. Empirical evidence for correct iris match score degradation with increased time-lapse between gallery and probe matches. In *Advances in Biometrics*, pages 1170–1179. Springer, 2009.

[BBFP13] S. Baker, K. W. Bowyer, P. J. Flynn, and P. J. Phillips. Template aging in iris biometrics: evidence of increased false reject rate in ICE 2006. *Handbook of Iris Recognition*, 2013.

[BHF08] K. W. Bowyer, K. Hollingsworth, and P. J. Flynn. Image understanding for iris biometrics: A survey. *Computer vision and image understanding*, 110(2):281–307, 2008.

[BHG+05] C. Fancourtand L. Bogoni, K. Hanna, Y. Guo, R.Wildes, N. TakahashI, and U. Jain. Iris recognition at a distance. In *Audio-and Video-Based Biometric Person Authentication*, pages 1–13. Springer, 2005.

[BJ03] Alexandre Cruz Berg and Silvana Cardoso Justo. Aging of orbicularis muscle in virtual human faces. In *IV 2003. Proceedings. Seventh International Conference on Information Visualization*, pages 164–168. IEEE, 2003.

[BPG06] A. C. Berg, F. J. Perales, and L. M. Gonzalez. A facial aging simulation method using flaccidity deformation criteria. In *Tenth International Conference on Information Visualization, 2006. IV 2006.*, pages 791–796. IEEE, 2006.

[BTN04] A. Bastanfard, H. Takahashi, and M. Nakajima. Toward E-appearance of human face and hair by age, expression and rejuvenation. In *2004 International Conference on Cyberworlds*, pages 306–311. IEEE, 2004.

[cat] VISION 2020: the cataract challenge . `http://www.cehjournal.org/0953-6833/13/jceh_13_34_017.html`. [Online; accessed 27-June-2013].

[Dau93] J. Daugman. High confidence visual recognition of persons by a test of statistical independence. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(11):1148–1161, 1993.

[Dau94] J. Daugman. Biometric personal identification system based on iris analysis, March 1 1994. US Patent 5,291,560.

[Dau01] J. Daugman. Iris Recognition - The colored part of the eye contains delicate patterns that vary randomly from person to person, offering a powerful means of identification. *American scientist*, 89(4):326–333, 2001.

[Dau04a] J. Daugman. How iris recognition works. *IEEE Transactions on Circuits and Systems for Video Technology*, 14(1):21–30, 2004.

[Dau04b] John Daugman. How iris recognition works. *IEEE Transactions on Circuits and Systems for Video Technology*, 14(1):21–30, 2004.

[Dau07] John Daugman. New methods in iris recognition. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 37(5):1167–1175, 2007.

[DD13] J. Daugman and C. Downing. No change over time is shown in Rankin et al. Iris recognition failure over time: The effects of texture. *Pattern Recognition*, 46(2):609–610, 2013.

[ddi86] A. Bertillon In: Annales de demographie internationale. *La couleur de l'iris*. Masson, 1886.

[DKF93] R. A. Dixon, D. Kurzman, and I. C. Friesen. Handwriting performance in younger and older adults: Age, familiarity, and practice effects. *Psychology and aging*, 8(3):360–370, 1993.

[DS08] S. Dey and D. Samanta. Improved feature processing for iris biometric authentication system. *International Journal of Computer Science and Engineering*, 4(2):p127–134, 2008.

[EF12] M. Erbilek and M. Fairhurst. Framework for managing ageing effects in signature biometrics. *IET biometrics*, 1(2):136–147, 2012.

[ER13] E. Ellavarason and C. Rathgeb. Template Ageing in Iris Biometrics: An Investigation of the ND-Iris-Template-Ageing-2008-2010 Database. *Biometrics and Internet-Security Research Group, Center for Advanced Security Research, Darmstadt, Germany, Tech. Rep. HDA-da/sec-2013-001*, 2013.

[FB11] S. P. Fenker and K. W. Bowyer. Experimental evidence of a template aging effect in iris biometrics. In *2011 IEEE Workshop on Applications of Computer Vision (WACV)*, pages 232–239. IEEE, 2011.

[FB12] S. P. Fenker and K. W. Bowyer. Analysis of template aging in iris biometrics. In *2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 45–51. IEEE, 2012.

[FE11] M. Fairhurst and M. Erbilek. Analysis of physical ageing effects in iris biometrics. *Computer Vision, IET*, 5(6):358–366, 2011.

[FS87] L. Flom and A. Safir. Iris recognition system, February 3 1987. US Patent 4,641,349.

[HGL$^+$] T. Hotz, C. Gottschlich, R. Lorenz, S. Bernhardt, M. Hantschel, and A. Munk. Statistical Analyses of Fingerprint Growth. In *Gesellschaft für Informatik eV (GI).*, page 11.

[HMM$^+$96] K. Hanna, R. Mandelbaum, D. Mishra, V. Paragano, and L. Wixson. A system for non-intrusive human iris acquisition and identification. In *IAPR Workshop on Machine Vision Applications*, pages 200–203, 1996.

[iso02] ISO/IEC 19794-6:2011 Information technology – Biometric data interchange formats, Part 6: Iris image data . 2002.

[iso06a] ISO/IEC 19795-1: Information technology-Biometric performance testing and reporting-Part 1: Principles and framework. 2006.

[ISO06b] ISO/IEC TC JTC1 SC37 Biometrics. *ISO/IEC 19795-1:2006. Information Technology – Biometric Performance Testing and Reporting – Part 1: Principles and Framework*. International Organization for Standardization and International Electrotechnical Committee, March 2006.

[ISO12] ISO/IEC. Information technology — vocabulary — part 37: Biometrics. Technical report, International Organization for Standardization, 2012.

[JDS] Applications J. Daugman, IEEE Conference on Biometrics: Theory and Systems. Evolving Methods in Iris Recognition . `http://www.cse.nd.edu/BTAS_07/John_Daugman_BTAS.pdf`. [Online; accessed 25-May-2013].

[Joh91] R. G. Johnson. Can iris patterns be used to identify people. *Los Alamos National Laboratory, CA, Chemical and Laser Sciences Division, Rep. LA-12331-PR*, 1991.

[JRP04] A. K. Jain, A. Ross, and S. Prabhakar. An introduction to biometric recognition. *IEEE Transactions on Circuits and Systems for Video Technology*, 14(1):4–20, 2004.

[KGYC07] J. Ko, Y. Gil, J. Yoo, and K. Chung. A novel and efficient feature extraction method for iris recognition. *ETRI journal*, 29(3):399–401, 2007.

[KT10] J. Kieffer and K. Trissell. DOD Biometrics—Lifting the Veil of Insurgent Identity. *Army AL & T*, page 14, 2010.

[LTC95] A. Lanitis, C. J. Taylor, and T. F. Cootes. Automatic face identification system using flexible appearance models. *Image and vision computing*, 13(5):393–401, 1995.

[LTC02] A. Lanitis, C. J. Taylor, and T. F. Cootes. Toward automatic simulation of aging effects on face images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(4):442–455, 2002.

[M+03] L. Masek et al. Recognition of human iris patterns for biometric identification. Master's thesis, University of Western Australia, 2003.

[MNH+06] J. R. Matey, O. Naroditsky, K. Hanna, R. Kolczynski, D. J. LoIacono, S. Mangru, M. Tinker, T. Zappia, and W. Y. Zhao. Iris on the move: Acquisition of images for iris recognition in less constrained environments. *Proceedings of the IEEE*, 94(11):1936–1947, 2006.

[MRZ07] D. M. Monro, S. Rakshit, and D. Zhang. DCT-based iris recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(4):586–595, 2007.

[MTWZ04] L. Ma, T. Tan, Y. Wang, and D. Zhang. Efficient iris recognition by characterizing key local variations. *IEEE Transactions on Image Processing*, 13(6):739–750, 2004.

[NS06] R. Narayanswamy and P. Silveira. Iris recognition at a distance with expanded imaging volume. In *Defense and Security Symposium*, pages 62020G–62020G. International Society for Optics and Photonics, 2006.

[OVV+97] A. J. O'Toole, T. Vetter, H. Volz, E. M. Salter, et al. Three-dimensional caricatures of human heads: distinctiveness and the perception of facial age. *PERCEPTION-LONDON*, 26:719–732, 1997.

[PH] U.S. National Library of Medicine PubMed Health. Eyelid Drooping . `http://www.ncbi.nlm.nih.gov/pubmedhealth/PMH0002013/`. [Online; accessed 10-June-2013].

[PPP10] G. Popa, R. Potorac, and N. Preda. Method for fingerprints age determination. *Romanian Journal of Legal Medicine*, 18(2):149–154, 2010.

[PRAB06] E. Patterson, K. Ricanek, M. Albert, and E. Boone. Automatic representation of adult aging in facial images. In *IASTED'06: Proc. 6th International Conference on Visualization, Imaging, and Image Processing*, pages 171–176, 2006.

[RC06] N. Ramanathan and R. Chellappa. Modeling age progression in young faces. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 387–394. IEEE, 2006.

[RSMP12] D. M. Rankin, B. W. Scotney, P. J. Morrow, and B. K. Pierscionek. Iris recognition-the need to recognise the iris as a dynamic biological system: response to Daugman and Downing. *Pattern Recognition*, 2012.

[RU09] C. Rathgeb and A. Uhl. Context-based texture analysis for secure revocable iris-biometric key generation. In *3rd International Conference on Crime Detection and Prevention (ICDP 2009)*, pages 1–6. IET, 2009.

[RU10] C. Rathgeb and A. Uhl. Secure iris recognition based on local intensity variations. In *Image Analysis and Recognition*, pages 266–275. Springer, 2010.

[RUW13] C. Rathgeb, A. Uhl, and P. Wild. *Iris Biometrics: From Segmentation to Template Security*. Springer, 2013.

[SBF] A. Sgroi, K. W. Bowyer, and P. Flynn. The Prediction of Old and Young Subjects from Iris Texture.

[SCZY02] E. Sung, X. Chen, J. Zhu, and J. Yang. Towards non-cooperative iris recognition systems. In *2002. ICARCV 2002. 7th International Conference on Control, Automation, Robotics and Vision*, volume 2, pages 990–995. IEEE, 2002.

[TGAFOG08] P. Tome-Gonzalez, F. Alonso-Fernandez, and J. Ortega-Garcia. On the effects of time variability in iris recognition. In *2nd IEEE International Conference on Biometrics: Theory, Applications and Systems, 2008. BTAS*, pages 1–6. IEEE, 2008.

[TSK07] J. Thornton, M. Savvides, and V. Kumar. A Bayesian approach to deformed pattern matching of iris images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(4):596–606, 2007.

[UID] AADHAR India . `http://uidai.gov.in/UID_PDF/Working_Papers/UID_and_iris_paper_final.pdf`. [Online; accessed 24-May-2013].

[UKB] Iris recognition immigration system (IRIS). `http://www.ukba.homeoffice.gov.uk/customs-travel/Enteringtheuk/usingiris/`. [Online; accessed 26-May-2013].

[UW12] A. Uhl and P. Wild. Weighted adaptive hough and ellipsopolar transforms for real-time iris segmentation. In *2012 5th IAPR International Conference on Biometrics (ICB)*, pages 283–290. IEEE, 2012.

[WAH+96] R. P. Wildes, J. C. Asmuth, K. J. Hanna, S.C. Hsu, R. J. Kolczynski, J. R. Matey, and S. E. McBride. Automated, non-invasive iris recognition system and method, November 5 1996. US Patent 5,572,596.

[Wal97] J. Walton. Handwriting changes due to aging and Parkinson's syndrome. *Forensic science international*, 88(3):197–214, 1997.

[Way01] J. L. Wayman. Fundamentals of biometric authentication technologies. *International Journal of Image and Graphics*, 1(01):93–113, 2001.

[WBM99] Y. Wu, P. Beylot, and T. Magnenat. Skin aging estimation by facial simulation. In *1999. Proceedings on Computer Animation,*, pages 210–219. IEEE, 1999.

[Wil97] R. P. Wildes. Iris recognition: an emerging biometric technology. *Proceedings of the IEEE*, 85(9):1348–1363, 1997.

[Zel12] F. Zelazny. The Evolution of India's UID Program. *Lessons Learned and Implications for Other Developing Countries at Center For Global Development*, 2012.