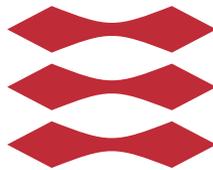


# Efficient biometric identification in large-scale iris databases

Pawel Drozdowski

DTU



Kongens Lyngby 2016

Technical University of Denmark  
Department of Applied Mathematics and Computer Science  
Richard Petersens Plads, building 324,  
2800 Kongens Lyngby, Denmark  
Phone +45 4525 3031  
[compute@compute.dtu.dk](mailto:compute@compute.dtu.dk)  
[www.compute.dtu.dk](http://www.compute.dtu.dk)

# Summary (English)

---

The growing interest and acceptance of biometrics has resulted in an increasing number and size of biometric systems around the world. These systems use measurable and distinctive human characteristics for, among others, the purpose of automatised recognition of individuals. Several country-wide deployments, such as the Indian AADHAAR project [Ind15], were spawned in recent years. The daily operation of these systems faces an immense computational load, which can contribute to increased system costs and reduced system usability. The goal of this thesis is to perform research in the area of biometric workload reduction in identification scenarios for large (human) iris databases. The iris has been selected as the biometric characteristic for the project, due to it being well-suited for use in large systems and is commonly used in actual deployments around the world.

The research in this thesis was carried out using a recently proposed, novel biometric indexing approach based on Bloom filters and binary search trees. During the course of this thesis, said approach was thoroughly analysed and expanded upon. In particular, several key improvements that ensure the system's scalability were proposed, implemented and tested quantitatively in terms of biometric performance and workload reduction. The system was shown to achieve an excellent biometric performance and a substantial workload reduction on a dataset of images with low intra-class variation. It was also discovered, that the biometric performance was severely impaired in the tests on a dataset of images with high intra-class variation. The results suggest that the approach is fully scalable in terms of enrolled database size. The biometric sample quality, however, may be a limiting factor. Furthermore, a brief investigation into multi-iris indexing has been conducted and shows great promise for future research. It is, best to

this author's knowledge, the first such study in the scientific literature.

In addition to the empirical testing, a general statistical model for Bloom filter based biometric indexing was presented. Based on several variables, the model allows for a theoretical assessment of the viability of a system configuration. Lastly, due to the incomprehensibility of current biometric workload reduction result reporting methods in the scientific literature, a unified and transparent methodology of result reporting was proposed. The aim of this undertaking was to elucidate this important issue and to serve as a basis for a better scientific process henceforth.

# Summary (Danish)

---

Den voksende interesse for, og generelle folkelige accept af, biometri har ført til et stigende antal og vækst i størrelser af biometriske systemer i den moderne verden. Disse systemer benytter målbare og karakteristiske personlige træk til automatiseret unik genkendelse af mennesker. Der findes, allerede nu, talrige nationale udrulninger, såsom det Indiske AADHAAR projekt [Ind15]. Den daglige drift af sådanne systemer præges dog af en meget stor beregningsmæssig belastning, hvilket kan være en vigtig faktor i forhold til omkostninger for, og brugervenlighed af systemet. Formålet med denne afhandling er forskning i reduktion af den biometriske arbejdsbyrde, specifikt i forhold til identifikationsscenerier for store (menneskelige) iris databaser. Iris blev valgt, da den er velegnet til brug i store systemer og ofte anvendt i systemer i drift rundt om i verden.

Forskningen i denne afhandling blev udført med fundament i en nyligt foreslået metode til biometrisk indeksering, baseret på Bloom filtre og binære søgetræer. Denne metode blev analyseret grundigt for dernæst at blive udvidet med adskillige forbedringer, som sikrer Bloom filter systemets skalerbarhed. Dette udvidede system opnåede en framragende biometrisk ydeevne og en betydelig reduktion af arbejdsbyrden på et dataset af billeder med lav intra-klasse variation. Det blev også observeret, at den biometriske ydeevne var nedsat signifikant i forbindelse med anvendelse på et dataset af billeder med høj intra-klasse variation. Disse resultater antyder kraftigt, at systemet er fuldt skalerbart for en database af arbitrær størrelse. Der er en indikation af, specifikt for Bloom filtre, at kvaliteten af billeder er en begrænsende faktor for den biometriske ydeevne. I denne afhandling er der også udført en kort undersøgelse i multi-iris indeksering, som viser stort potentiale for videre forskning i emnet. Der er ikke fundet videnskabelig litteratur der omhandler multi-iris indeksering, hvorfor studiet i

denne afhandling må betragtes som det første af sin slags.

Som en tilføjelse til den empiriske undersøgelse, introducerede denne forfatter en generel statistisk model for Bloom filtre baseret biometrisk indeksering. Denne model bygger på flere variabler og muliggør en teoretisk vurdering af hvorvidt et system med en sådan konfiguration giver udbytte. Endeligt, på grund af divergens i metodik for rapportering af resultater af arbejdsbyrdereduktion i den videnskabelige litteratur, foreslåes en forenet og transparent metodik til dette specifikke formål. Dette har til formål at kaste lys på et vigtigt problem og fremme en bedre videnskabelig proces, inden for dette felt, fremover.

# Preface

---

This thesis was prepared at DTU Compute and Center for Advanced Security Research Darmstadt (CASED) in fulfilment of the requirements for acquiring an M.Sc. in Engineering. It was supervised by Professor Rasmus Larsen, Professor Christoph Busch and Doctor Christian Rathgeb.

The thesis deals with methods of workload reduction for biometric identification scenarios in large-scale biometric systems, focusing on the iris modality.

Lyngby, 25-June-2016

*P. Drozdowski*

Pawel Drozdowski



# Acknowledgements

---

I would like to thank my supervisors: Professor Rasmus Larsen, Professor Christoph Busch and Doctor Christian Rathgeb for making this project possible and for the excellent communication, guidance and advice throughout the entire project period. I am especially grateful to Doctor Christian Rathgeb for the day-to-day supervision and feedback. I also thank Kim Rostgaard Christensen and Małgorzata Tyczyńska for several critical comments and help with proofreading the thesis document. Thanks to the Hochschule Darmstadt for sponsoring my stay in Germany, where the project work was carried out. I owe my gratitude to Esbern Andersen-Hoppe for lending a helping hand during my dire straits with living accommodation and for numerous discussions regarding the thesis project.

Portions of the research in this project use the "Interval" and "Thousand" subsets of the CASIA-IrisV4 database collected by the Chinese Academy of Sciences' Institute of Automation (CASIA) and the iris database collected by the IIT Delhi. The data processing and visualisation in this project was done using the R software ecosystem and the matplotlib package. The sources of the re-used images are attributed directly in the text.



# Contents

---

<b>Summary (English)</b>	<b>i</b>
<b>Summary (Danish)</b>	<b>iii</b>
<b>Preface</b>	<b>v</b>
<b>Acknowledgements</b>	<b>vii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Thesis Contribution . . . . .	2
1.2 Thesis Organisation . . . . .	2
<b>2 Biometric Systems Fundamentals</b>	<b>3</b>
2.1 Iris as a Biometric Characteristic . . . . .	3
2.2 Generic Biometric System . . . . .	5
2.2.1 Workflow . . . . .	5
2.2.2 Operation Modes . . . . .	7
<b>3 Related Work</b>	<b>9</b>
3.1 Types of Workload Reduction Approaches . . . . .	9
3.2 Serial Combination of Algorithms . . . . .	10
3.3 Classification . . . . .	11
3.4 Indexing . . . . .	12
3.5 Workload Reduction Reporting . . . . .	13
3.6 Summary . . . . .	16
<b>4 Bloom Filter Approach</b>	<b>17</b>
4.1 Bloom Filter . . . . .	17
4.2 System Basics . . . . .	18

4.2.1	Template Transformation . . . . .	18
4.2.2	Tree Construction . . . . .	20
4.2.3	Tree Traversal . . . . .	21
4.2.4	Relevant Configurations . . . . .	24
4.3	General Model of Bloom Filter Based Indexing . . . . .	25
4.3.1	Random Template and Tree Root Emulation . . . . .	25
4.3.2	Overlap between Root and Random Template . . . . .	27
4.3.3	Validation . . . . .	28
4.4	Summary . . . . .	29
<b>5</b>	<b>Improvements for the Bloom Filter Approach</b>	<b>31</b>
5.1	Multiple Trees . . . . .	31
5.2	Traversal Direction Decision . . . . .	35
5.3	Other . . . . .	36
5.4	Multibiometrics . . . . .	36
5.5	Summary . . . . .	37
<b>6</b>	<b>Experimental Setup</b>	<b>39</b>
6.1	Chosen Datasets . . . . .	39
6.2	Iris Biometric Processing Chain . . . . .	40
6.3	Excluded Images . . . . .	42
6.3.1	Dataset Errors . . . . .	42
6.3.2	Preprocessing Failures . . . . .	43
6.3.3	Automated Quality Check . . . . .	44
6.4	Experiments . . . . .	45
6.5	Summary . . . . .	46
<b>7</b>	<b>Results</b>	<b>47</b>
7.1	Baseline . . . . .	47
7.1.1	Score Distribution . . . . .	47
7.1.2	EER and ROC . . . . .	48
7.1.3	Workload . . . . .	50
7.2	Bloom Filter Approach . . . . .	50
7.2.1	Verification . . . . .	50
7.2.2	Model . . . . .	52
7.2.3	Identification . . . . .	56
7.2.4	Workload . . . . .	57
7.3	Improvements . . . . .	58
7.3.1	Multiple Trees . . . . .	58
7.3.2	Selective Tree Traversal . . . . .	59
7.3.3	Quick Traversal Direction Decision . . . . .	61
7.3.4	Score Sequence Ordering . . . . .	62
7.3.5	Duplicate Enrolment Check . . . . .	63
7.3.6	Multi-Iris Approach . . . . .	65

---

7.4 Summary . . . . .	66
<b>8 Discussion</b>	<b>69</b>
8.1 Results . . . . .	69
8.1.1 Bad Performance on the Thousand Dataset . . . . .	69
8.1.2 Scalability . . . . .	71
8.2 Future Research . . . . .	72
8.2.1 Model . . . . .	72
8.2.2 Tree Construction . . . . .	72
8.2.3 Multibiometrics . . . . .	73
8.2.4 Compact Storage . . . . .	73
8.2.5 Data Protection . . . . .	74
8.3 Summary . . . . .	74
<b>9 Conclusions</b>	<b>75</b>
<b>A List of Symbols</b>	<b>77</b>
<b>B Results</b>	<b>79</b>
<b>Bibliography</b>	<b>85</b>



# Introduction

---

The interest in biometrics has been steadily growing in the last two decades. Currently, biometrics are ubiquitously and reliably used in a wide range of applications: as an alternative or extension to traditional knowledge and token based access control systems, identity documents, criminal surveillance and identification of mass-disaster victims. A study into the biometrics market has estimated its worth to approximately 14 billion USD in 2015 [Tra14]. In recent years several large-scale biometric deployments appeared. The chief among these are the Indian AADHAAR project, which soon will have acquired biometric data and enrolled the entire Indian population of 1.3 billion, as well as American, European and Middle Eastern immigration programmes. Detailed information about these and further examples can be found in [Ind15], [Dau16] and [GAH+08]. Operations of such large deployments face tremendous computational load. Therefore, the efficiency of the underlying system implementations is crucially important - the naïve solutions are insufficient. Furthermore, the traditional approaches for computational load reduction cannot be applied due to fuzziness of the biometric data. This matter is the key motivation behind and the main focus of this thesis. To limit the otherwise immense scope of different biometric characteristics, this thesis shall pertain only to the iris and focus on the challenging biometric identification scenario. As shown in [Dau00], success of such scenario in the large databases depends on the chosen biometric characteristic to be highly resilient against false matches. The iris satisfies this requirement and is thereby well-suited for purposes of large scale deployments.

## 1.1 Thesis Contribution

The contributions made by this project are listed below.

- A thorough state-of-the-art survey of existing approaches within iris biometric workload reduction.
- A proposal of a unified and transparent methodology for biometric workload reduction reporting in iris identification systems.
- Re-implementation and open-set evaluation of one of the recently developed approaches based on Bloom filters and binary trees. Said approach has hitherto only been tested on small, moderately challenging datasets. A study into its scalability and performance on larger and/or more noisy datasets is therefore desirable.
- Seeking out possible improvements for the above scheme, implementation thereof and empirical testing for feasibility assessment.
- Exploring the limitations of the approach through development of a general statistical model for Bloom filter based biometric indexing.
- Preliminary investigation into multibiometric capabilities of the Bloom filter based approach.

## 1.2 Thesis Organisation

This document is organised as follows:

- Chapters 2 and 3 outline the fundamentals of biometric systems, relevant related work and propose a methodology for biometric workload reduction reporting.
- Chapters 4 and 5 describe the Bloom filter based biometric indexing scheme used in this project and propose improvements for it.
- Chapters 6 and 7 present the experimental set-up and results.
- Chapters 8 and 9 conclude the thesis with a discussion of the results, further research proposals and summarise the achievements of this project.

# Biometric Systems Fundamentals

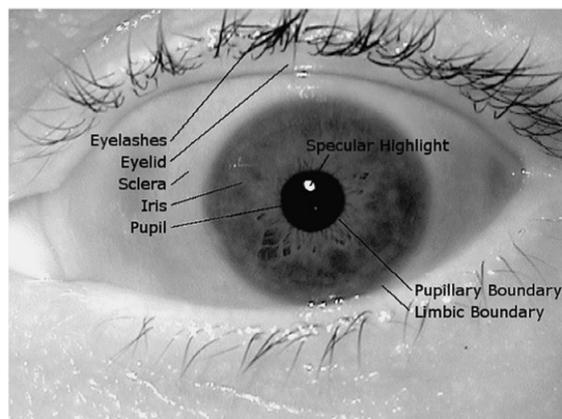
---

This chapter provides a general introduction to biometric systems with emphasis on the iris as the biometric characteristic of choice.

## 2.1 Iris as a Biometric Characteristic

The iris is a part of the eye, located in its frontal part. One of its main responsibilities is adjustment of the pupil dilation, thus controlling the amount of light reaching the retina. The iris also gives eyes their colour. Figure 2.1 shows an image of an eye with labels put on the important anatomic features.

The use of iris patterns and/or colour for personal identification has been proposed already back in 1936 by Frank Burch, an ophthalmologist. However, first patents appeared fairly recently: [FS87] and [Dau94]. Let's now explore the suitability of the iris as a biometric characteristic. A common way for such assessment is to use the properties proposed in [JBP99]. A quantitative assessment, especially on the issue of uniqueness and random false matches can be found in [Dau06].



**Figure 2.1:** A human eye viewed from the front (from [BHF08])

**Universality** Vast majority of the population has an undamaged iris that can be used for biometric purposes. Certain illnesses and alcoholism can affect the iris appearance and thereby cause problems in the biometric contexts (see e.g. [AT09]).

**Distinctiveness** Different irides are highly distinguishable due to high randomness degree of the iris pattern emergence. Even irides from the left and right eye of the same person are independent [BLF10].

**Performance** The biometric performance of iris recognition systems is proven to be very high. A key advantage of iris over other modalities is its high resilience against false matches [Dau00] and [Dau06].

**Permanence** The iris texture patterns are claimed to be stable over time (e.g. [Dau04a]), although both biometric template and biometric characteristic ageing, are controversial and insufficiently researched issues in the biometric community. For example, [FB12] indicates that biometric templates may indeed be subject to ageing and thus result in reduced biometric performance over time. This survey, however, was performed on a very small, non-representative dataset. Iris pattern permanence remains an open issue.

**Collectability** It is easy to acquire a sample of an iris; however, achieving near-optimal image quality usually requires subject cooperation and specialised equipment. The sensor technology is constantly improving - recent developments indicate that iris recognition in visible light spectrum may soon be a viable option. This would mean that potentially any device with a camera (e.g. a smartphone) could be capable of performing sample acquisition [RRKB15].

**Acceptability** Societal acceptance of biometrics in general is rising and the iris is no exception. It is used in the largest deployments, such as the Indian AADHAAR project and the Middle Eastern border control programmes. The recent invention of systems such as Iris at a Distance [Mor] can make the data acquisition completely unobtrusive, thus making it more attractive for the users.

**Circumvention** It is possible to produce fake artefacts for presentation attacks; however, these can be prevented by liveness detection mechanisms. In scenarios where a human operator supervises the process, such attacks are not a big issue, since in most cases the attacker's efforts would be easily spotted by the human operator.

Overall, the iris is an excellent choice for a biometric characteristic; substantiating this assertion is the fact of its use in the largest and most successful biometric deployments around the world. The following section provides a description of working details of such systems.

## 2.2 Generic Biometric System

This section introduces basic operational details of a generic biometric system.

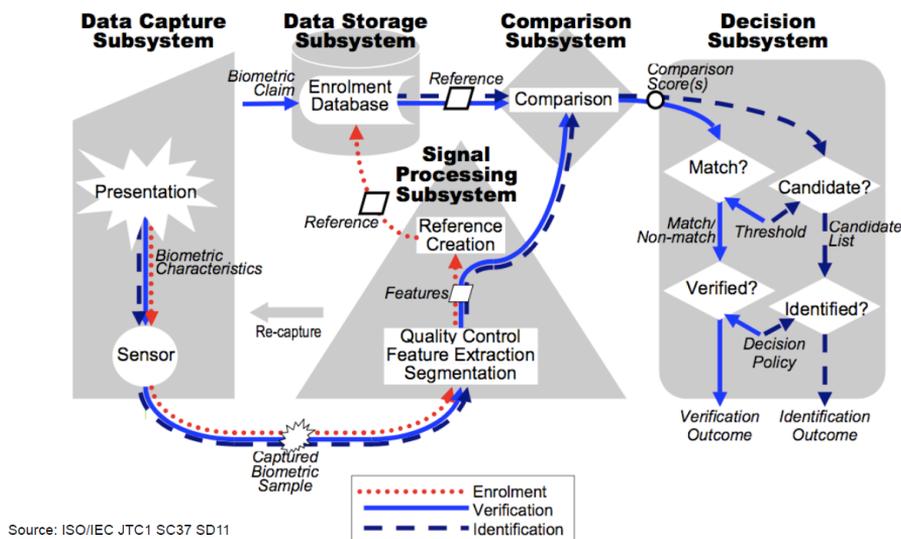
### 2.2.1 Workflow

Regardless of the chosen biometric characteristic, the basic workflow of a biometric system can be generalised as shown in figure 2.2. The figure and the concepts of biometric framework generalisation come from an ISO/IEC standard on biometric performance testing and reporting [ISO11].

The key steps of the process are briefly outlined below, using the iris as the biometric characteristic for concrete examples.

**Data capture** The process of acquiring a sample from a subject through a sensor. For an iris, this is typically a monochromatic image captured with a near infra-red sensitive camera.

**Signal processing** The process of transforming the acquired sample into a standardised biometric template form for the given biometric characteristic. For an iris, most often used is an iris code [Dau04a] representation,



**Figure 2.2:** A generic biometric system workflow (from [ISO11])

which is a two-dimensional binary matrix. The signal processing may consist of multiple stages, as listed below.

**Segmentation** The process of distinguishing the biometric characteristic signal from the rest of the acquired sample. In our case, this involves locating the outline of the iris, pupil, eyelids and eyelashes in an eye image, as well as a normalisation step.

**Feature extraction** The process of obtaining a feature set from a sample. The key idea is for that feature set to have low intra-class variation (i.e. remain largely invariant in different samples from the same subject) and high inter-class variation (i.e. have enough discriminatory power to reliably distinguish between different subjects). For the iris, the feature set is based on the iris texture - the rich patterns of arching ligaments, corona, crypts, freckles, furrows, ridges, rings and zigzag collarette.

**Quality control** In some cases, the poor quality of an acquired sample or a segmentation error can make the template unusable. Automated quality assessment can be implemented (both for raw images and produced templates). For an iris, one possibility is to check how large fraction of the iris was masked out in the segmentation step. This simple measure rejects templates with too much information loss from the signal processing. Another commonly used method is checking the diameter of the iris. Manual expert assessment is

sometimes required despite the aid of automated checks.

**Comparison and decision** The process of comparing a new template against existing records of enrolled templates. The results are then used to determine the final outcome of a query.

### 2.2.2 Operation Modes

Figure 2.2 illustrated that there are two modes a biometric system can operate in. They determine the flow of information in the system and how the outcome decision is made. From the practical point of view, the *open-set* scenario (i.e. where there may be attempts from users not enrolled in the system) is the only sensible one to consider.

**Verification** The subject has to present a claim to an identity. Subsequently, a biometric sample is acquired from the subject, transformed to a template and compared against the enrolled template of the claimed identity. It is thus a trivial case, which merely requires a 1:1 template comparison to reach a decision.

**Identification** Unlike in the case above, there is no identity claim. The system is just presented with the sample acquired from the subject and has to ascertain whether the subject has previously been enrolled and if so, what their identity is. This is a non-trivial case, since, if approached naïvely, at worst  $O(S)$  template comparisons are necessary to reach a decision (i.e. comparing the new template against every enrolled template). In case of the iris, this additionally means accounting for image misalignment by considering several rotations of each template.

The naïve approach in the identification mode, demonstrated in algorithm 2.1, is simply not viable due to a prohibitive computational cost for any large-scale system. Furthermore, in terms of the biometric performance, such a system will suffer from a high false positive risk, as demonstrated in [Dau00]. There, an identification scenario is demonstrated to have the probability of *at least one* false match occurring ( $P_S$ ) as shown in equation 2.1, where  $S$  denotes the number of subjects in the database and  $P_1$  the probability of a false match in a single comparison.

$$P_S = 1 - (1 - P_1)^S \quad (2.1)$$

---

**Algorithm 2.1** Lookup in the naïve implementation of an identification system

---

```

1: procedure LOOKUP(probe, enrolled, threshold)
2:   candidates  $\leftarrow$  empty list
3:   for all reference in enrolled do
4:     dissimilarityscore  $\leftarrow$  COMPARE(probe, reference)       $\triangleright$  Modality
       specific method
5:     if dissimilarityscore  $<$  threshold then
6:       add reference to candidates
7:     end if
8:   end for
9:   return candidates
10: end procedure

```

---

As a concrete example, consider a biometric system with only 500 subjects and low single comparison false match probability of  $P_1 = 0.005$ . In an identification scenario, the probability of at least one false match is then  $P_{500} = 1 - (1 - 0.005)^{500} \approx 91\%$ ! It is immediately clear, that even for relatively small  $S$  values, the probability of false match occurrences quickly becomes high. Based on the above discourse, it is obvious that decreasing the the number of necessary template comparisons for the lookup in identification and duplicate enrolment check (DEC) scenarios is of utmost importance for any sizeable biometric system deployment. Otherwise, such a system will suffer both in terms of the required computational workload and the false positive occurrence chances.

The task solved by the system in the identification mode can be generalised to the nearest neighbour search (NNS) problem as defined in [Kmu73]. Substantial research effort has been devoted to finding efficient solutions for this problem; however, the traditional approaches are often ill-suited for biometrics. The key issue is the fuzziness of the biometric data. Observe, that there are numerous noise sources in the acquisition process (e.g. in the case of the iris, distance and angle of the camera, lighting conditions, eyelid occlusion etc.). These will inevitably vary slightly between individual image acquisitions (of the same subject), and although much of the noise can be effectively eliminated, the process will nevertheless result in biometric templates that are *very similar, but not identical*. This makes a huge difference: it immediately invalidates many traditional approaches (e.g. simple hashing), where the database reference and the probe are expected to be exactly the same. Another problem is the high dimensionality of the iris data - the traditional approaches often perform poorly in such spaces [HDZ08].

The next chapter will present a state-of-the-art survey of existing approaches to workload reduction in the identification and DEC scenarios.

# Related Work

---

This chapter describes the current state-of-the-art in iris biometric identification workload reduction. First, a brief overview of the workload reduction approach types is presented. Subsequently, a literature survey for each of the types is performed. Lastly, a methodology for biometric workload reduction reporting is proposed.

## 3.1 Types of Workload Reduction Approaches

In this section, the workload reduction approach types are described.

**Serial combination of algorithms** A two-step approach that first utilises a computationally efficient algorithm to prepare a short-list of most likely template matches. Subsequently, the slow and much more accurate template comparison by the second algorithm is only performed on the templates from the short-list.

**Classification/Binning** The enrolled template database is split into several subsets with low intra-class variation and high inter-class variation. In an identification scenario, the class of the probe template is determined and

actual comparisons are performed only with database templates of that class.

**Indexing** Techniques that seek to decrease the system load in terms of the big-O notation. They usually utilise probabilistic or hierarchical data structures to reduce the search space.

**Hardware acceleration and parallelism** The identification scenario can be handled very efficiently by using many CPUs/threads (e.g. by using a GPU). The processes can work on disjoint parts of the database and in the end the results are aggregated. Note, that the workload, instead of being *reduced*, is merely *distributed*. Nonetheless, it is important to mention this for completeness sake, as it is a key ingredient of real-world deployments and often can be combined with other approaches that indeed do reduce the workload.

The approaches can additionally be divided into:

**Encoding independent** Techniques that work irrespective of the used image encoding method.

**Encoding dependent** Techniques that are developed only for a certain image encoding method. The most commonly used method in iris biometrics is the, so-called, iris code.

## 3.2 Serial Combination of Algorithms

In one of the first works for this category, [GRC09a] propose a system based on the standard, full-length iris code (FLIC) representation and the short-length iris codes (SLIC). The SLIC representation was introduced in [GRC09b]; it relies on finding the most discriminative regions of the iris code and thus reducing its size more than tenfold. In the scheme for a biometric identification system, both the FLIC and SLIC representations are stored for enrolled subjects. A short-list of candidate templates is produced by performing comparisons only on the stored short-length iris codes. Subsequently, full-length iris code comparisons are performed only on the produced candidate templates. Thereby, the computational workload is reduced - the authors report a 12-fold reduction in the required workload compared to the naïve system implementation. Unfortunately, the biometric performance of this system is impaired. In experiments for genuine attempts, the correct identity was not present in the short-list in

around 7% of cases. This is a severely limiting factor on the true positive identification rate of the system. In [RUW10], the proposed idea is to compute the comparison score (Hamming Distance) incrementally and reject unlikely templates early. The workload is then reduced, since the comparisons are performed only on a fraction of the iris code bits in most cases. The authors report an experimental decrease of about 95% overall bit comparisons and lower storage requirements for iris code masks.

[KSUW10] propose an approach, which tackles the rotation compensation workload. The short-list of candidate templates is produced based on rotation invariant comparisons. Thereafter, in the second stage, the standard iris code comparisons (with rotation compensation) are applied to the templates from the short-list. The authors report a workload reduction of around 80% compared to the naïve system implementation.

A possible problem with the two-stage approaches is the potential negative influence on the genuine matching attempts. This is due to the fact that the probe template now has to successfully pass two tests (i.e. pre-selection, followed by template comparison), rather than just one test (i.e. template comparison), as is the case in the naïve system implementation. It appears that most of the approaches presented in this section manage to overcome this issue and do not experience a loss of biometric performance. In some cases, it is even increased - the two-stage schemes and schemes which only use a fraction of template bits make it more difficult for impostors to be falsely matched. In summary, the size of the short-list produced in the pre-selection step becomes an important variable in such systems. By changing the size of the short-list, the trade-off between biometric performance and workload reduction can be adjusted to satisfy the needs of a given biometric system deployment.

### 3.3 Classification

The simplest example of this category of approaches is the separation of templates in two classes - for left and right eyes. By doing so, the workload can be approximately cut in half. An example algorithm for this can be found in [Li09]. Other intuitive classification methods involve the eye colour and the race of the subjects [QST05]. The authors of [ZSTW14] present a flexible scheme based on hierarchical visual codebook. In the article, it is shown to be accurate in classifying the race of the subjects based on the iris texture; furthermore, it is shown to be useful in the detection of presentation attacks. Instead of creating tangible, human-understandable classes, it is also possible to rely on signal-based and statistical analysis. [RS10] and [YZWY05] assess the viability of such an

approach positively and propose classification schemes based thereupon. In a recently published article [NC15], the authors achieve promising results with de-duplication using a scheme based on online dictionary learning.

The serial combination of algorithms and classification are seemingly identical - they both include a pre-selection step that produces a subset of candidates and a second step, where actual comparisons take place. The key difference is, that in the serial combination, the pre-selection comparator runs over an entire database to produce candidates, while in the classification, the feature (e.g. eye colour) is extracted from the probe and from it, the appropriate bin is chosen. The two limiting factors of classification schemes are:

- Thus far, the demonstrated number of classes the data can be split into is quite low. The resulting workload reduction will therefore be insufficient for large-scale deployments. Furthermore, some of the schemes operate under the assumption that the enrolled subjects are evenly distributed among the classes, which is often not the case. For example, a classification scheme based on race will not yield much workload reduction in a database where the vast majority of subjects are of the same ethnic background.
- Similarly to the serial combination approaches, the additional pre-selection stage in the decision process can negatively affect the genuine attempts.

The second of the above issues appears to have been overcome, since almost all of the referenced articles report very high or near-optimal rates of correct classification. This means that the moderate workload reduction offered by these schemes can be taken advantage of without sacrificing the biometric performance.

## 3.4 Indexing

In one of the first papers for this category [MR08], two indexing schemes are presented. The workload reduction there is not high; nonetheless, the results prove that indexing of data without inherent ordering is feasible. Some early work in the area of iris indexing can be found in [Muk07]. Since then, many approaches have been proposed. [HDZ08] introduces a method called "beacon guided search" as a general method for indexing large fuzzy datasets. They report significant search space reduction with only a small impact on the biometric performance. [MSM09] present an interesting scheme, which uses energy histograms and a B-tree data-structure to significantly reduce the penetration

rate. In [GAR10] an approach based on Burrows-Wheeler Transform is shown to achieve very good results in terms of hit rate, although with still relatively high penetration rate. In [RU10] and [JG13] hash generation schemes are proposed with very good workload reduction results. A scheme based on the iris colour and texture indexing in a kd-tree is demonstrated in [JPG12] and shown to have a high hit rate and a low penetration rate. Finally, [Pro13] provides a more complete survey of indexing approaches; it is also a very important paper in the area of indexing data with bad quality, which, overall, is an insufficiently researched topic.

While the indexing schemes offer significant workload reduction, often trade-offs are associated with them. Typically, these come as offline computational costs (i.e. preparation of the scheme) or additional storage requirements for the used datastructures.

Article	Dataset	Biometric performance	Workload
[GRC09a]	MMU	max. TPIR 93%	12-fold reduction
[RUW10]	CASIA-V3-Interval	97.2-99.2% RR-1	5% bit comparisons
[KSUW10]	CASIA-V1	92% IR, 0% FAR	70-80% time reduction
	CASIA-V3-Interval MMU	89% IR, 0.85% FAR 79% IR, 0.85% FAR	
[QST05]	CASIA-V2, UPOL, UBIRIS	86% CCR	number of classes
[ZSTW14]	CASIA-V4	0% EER	number of classes
	ND	0.9% EER	
	Clarkson	0.54% EER	
[RS10]	UPOL	0% EER	30% bits used
[NC15]	UPOL	100% CCR	number of classes
[MR08]	CASIA-V3	80-84% hit rate	8-30% penetration rate
[GAR10]	CASIA-V3	99.8% hit rate	17.2% penetration rate
[JPG12]	UBIRIS	98.7% hit rate	7.1-8.3% penetration rate
[JG13]	CASIA-V3-Interval	94% hit rate	10.6% penetration rate
[RU10]	CASIA-V3-Interval	variable w.r.t. penetration rate	3% penetration rate
[MSM09]	CASIA	1.6-43.6% bin miss rate	39.96-0.63% penetration rate
	BATH	4-72% bin miss rate	26.14-0.06% penetration rate
	IITK	1.5-44% bin miss rate	41.4-0.2% penetration rate
[HDZ08]	UAE	0% FAR, 0.64% FRR	0.006% penetration rate
[Pro13]	CASIA-V4-Thousand	~ 94% TPIR at 0.1% FPIR	~ 2.5 – 7.5% av. penetration rate
	UBIRIS	~ 85% TPIR at 10.0% FPIR	~ 38 – 65% av. penetration rate

**Table 3.1:** The experimental results achieved by the methods presented in the surveyed articles (as reported by the author’s themselves, or if unavailable, extracted from the presented plots)

## 3.5 Workload Reduction Reporting

The table 3.1 summarises the results of the survey. Notice the variety of ways in which the results are reported. While the biometric performance reporting often adheres to the standard [ISO11], the differences are particularly evident

in the case of the workload reduction metrics, where there is no standardised way in place. These have been reported in a wide variety of ways. Additionally, some of these ways depend on the intrinsic properties of the data format used in the respective methods. In yet other cases, they provide no direct point of reference to the basic, naïve scheme. Furthermore, for some of the schemes, the impact on the biometric performance of the system is not clearly reported. Obviously, all this makes a direct comparison and evaluation of the presented approaches rather cumbersome.

As a potential remedy for this issue, this author would like to propose a unified methodology of reporting results of an iris biometric workload reduction scheme, by posing seven key requirements, as listed below.

- R1 The baseline workload must be explicitly stated.** This is to be expressed in terms of template size in bits, with the rotation compensation costs accounted for, the number of enrolled subjects and the penetration rate in an open-set identification scenario. Otherwise, there is no clear and direct point of reference for the workload of the proposed system.
- R2 The baseline biometric performance of a state-of-the-art algorithm on the used dataset must be explicitly stated, in a manner described in the ISO standard.** Otherwise, as in **R1**, it will not be possible to establish potential biometric performance costs incurred by the workload reduction of the proposed scheme.
- R3 The workload of the proposed scheme is to be stated in the manner described in R1.** If these parameters vary (e.g. due to different scheme configurations or non-determinism), then a range or an upper bound should be given. If a pre-selection step is involved, then it should be accounted for within the above parameters; if that is not feasible, then its cost should be stated separately.
- R4 The biometric performance of the proposed scheme must be reported according to the ISO standard.** This is necessary, because without regard for biometric performance, arbitrarily high workload reduction can be claimed. A scheme will, for the most part, only be viable if the biometric performance does not become drastically lowered; in any case, the trade-offs must to be mentioned.
- R5 The additional costs and benefits of the proposed scheme should be listed** (e.g. offline costs, storage requirements, rotation invariance). It should also be stated whether or not the template comparisons can be performed using the fast CPU instructions (bitwise operators in particular). This is important to allow a general, well-informed evaluation of the system and the trade-offs associated with the workload reduction.

**R6 The total workload for both the baseline and the proposed system is to be computed using equation 3.1.** By doing so, the total workload of the proposed system can be succinctly and precisely stated as a fraction ( $F$ ) of that of the baseline (e.g. " $F = 0.4$  of the baseline workload") in the worst and average case. Using this metric to summarise the results is advantageous, as it provides the readers with a single value, with which they can immediately and reliably assess the workload reduction conferred by the proposed system. The reasoning behind this requirement is including all the workload related variables in for the sake of accuracy and transparency.

A formula for the total system workload in a single lookup during an identification scenario ( $\omega$ ) is derived from the parameters stated in the requirements above:  $\mathcal{S}$  - the number of subjects enrolled,  $\rho$  - the penetration rate (as defined in the [ISO11] standard) and  $\tau$  - the cost of a single step (i.e. one comparison). In case of the iris, the templates are represented as binary vectors; the cost of a single step can be then expressed in terms of bit comparisons or simply the size of the iris biometric template in bits<sup>1</sup>. As an example, consider a naïve iris identification system, which has 100 enrolled subjects, which does not perform rotation compensation and uses the iris code template representation. The workload for a single lookup is then:  $\omega = 100 * 1.0 * 10240 = 1.02 * 10^6$  bit comparisons.

$$\omega = \mathcal{S} * \rho * \tau \quad (3.1)$$

The advantages of the proposed methodology are twofold. First, in a transparent manner, it takes into consideration all factors that determine the actual computational workload faced by a biometric identification system. Furthermore, by enforcing that biometric performance, other factors (e.g. offline costs) and the baseline results be explicitly included, it allows for assessment of a biometric system from a more holistic perspective. In the table 3.2, a summary of how the surveyed articles report their results can be seen. For **R2** a relaxed evaluation is given - if the article explicitly reports the baseline workload, not necessarily in the way proposed above, it is deemed to satisfy the requirement. For **R3** and **R4**, if inconsistent with the presented requirements, the methods used in the article are given. **R6** is naturally omitted, since it only now has been proposed.

This thesis will strive to adhere to the aforesated methodology of results reporting.

<sup>1</sup>Throughout this document several symbols with specific meanings are introduced. A full list is available in appendix A.

Article	R1	R2	R3	R4	R5
[GRC09a]	Yes	Yes	Yes	Yes	Yes
[RUW10]	Yes	Yes	Bit comparisons	Yes	Yes
[KSUW10]	Yes	Yes	Time	Yes	Yes
[QST05]	No	No	Number of classes	CCR	No
[ZSTW14]	Yes	Yes	Number of classes	Yes	Yes
[RS10]	No	No	Fraction of iris code	Yes	No
[NC15]	No	Yes	Number of classes	CCR	No
[MR08]	Yes	No	Penetration rate	Hit rate	No
[GAR10]	Yes	No	Penetration rate	Hit rate	No
[JPG12]	Yes	No	Penetration rate	Hit rate	No
[JG13]	No	No	Penetration rate	Hit rate	Yes
[RU10]	Yes	No	Penetration rate	Yes	Yes
[MSM09]	No	No	Penetration rate	Hit rate	No
[HDZ08]	Yes	Yes	Time and search space	Yes	Yes
[Pro13]	Yes	Yes	Penetration rate	Yes	Yes

**Table 3.2:** The adherence to the proposed reporting requirements in the surveyed articles

### 3.6 Summary

This chapter has presented the current state-of-the-art in biometric workload reduction. The used approaches can be categorised into three types: serial combination of algorithms, classification and indexing. The methods and results shown in the surveyed articles were briefly outlined. Most of the presented schemes are capable of reducing the computational workload to a small fraction of that required in the naïve biometric system implementation (see table 3.1).

Whilst this literature research was conducted, a major issue became apparent - the overwhelmingly inconsistent way of reporting the results in the area of biometric workload reduction. In response to this, this author proposes a simple, unified way of workload reduction reporting in (iris) identification systems. It can be used until a more thorough and comprehensive investigation of this issue (also including other modalities) by the biometric reporting ISO standard committee, for which it may serve as an inspiration.

In the next chapter, a novel biometric indexing approach based on Bloom filters and binary search trees will be described in detail; it serves as a foundation for the practical work conducted during this project.

# Bloom Filter Approach

---

This chapter presents the Bloom filter based biometric indexing scheme, which serves as a basis for the practical work of this thesis.

## 4.1 Bloom Filter

Bloom filters are applied extensively to solve various tasks in computer science (see e.g. a survey in [BM05]). This probabilistic datastructure was conceived in 1970 [Blo70] for the purpose of efficient membership queries. A Bloom filter (denoted  $\mathcal{B}$ ) is represented as a binary vector of fixed length ( $l$ ). In an empty Bloom filter, all the bits are set to 0. New elements can be added and checked for by following the procedure outlined below<sup>1</sup>.

1. The new element is fed to a beforehand fixed number ( $k$ ) of independent hash functions  $H_1 \dots H'_k$ . These functions always produce numbers in the range  $0 \dots l$ . The result of this step can be simply denoted as a set of hash values  $h_1 \dots h_k$ .

---

<sup>1</sup>An interactive presentation can be found at [Mil16].

2. These hash values directly correspond to indices of  $\mathcal{B}$ . The items at the computed indices are set to 1 (i.e.  $\mathcal{B}[h_i] = 1$  for  $i \in 0 \dots k$ ). If the item at a given index already is set to 1, it remains unchanged.

A membership query for an element is performed in this way:

1. The hash values for the element are produced as outlined above, i.e. produced is a set of hash values  $h'_1 \dots h'_k$ .
2. The corresponding indices in  $\mathcal{B}$  are checked. If and only if the relation  $\forall \mathcal{B}[h'_i] = 1$  for  $i \in 0 \dots k$  is satisfied, the membership check response is positive. That is, if any of the checked indices is set to 0, then the element can with 100% certainty be deemed not present in the Bloom filter; in other words, false negatives cannot occur. It is, however, possible for false positives to occur as more items are added to  $\mathcal{B}$  and the number of indices set to 1 increases. When  $\mathcal{B}$  is full (i.e. all bits set to 1), then any membership query will yield a positive response.

In its basic form, the Bloom filters are very successful in efficient membership tests for traditional data (e.g. where the probe and reference are identical). With certain modifications, the concept can be applied in biometric identification scenarios. In particular, the fuzziness of the biometric data has to be accounted for. In the iris biometrics, a very common comparator uses the Hamming distance to compute the normalised dissimilarity of two iris code templates. The concept can be seamlessly extended to Bloom filters and replace their simple binary decision pattern. The next section describes such a system in detail.

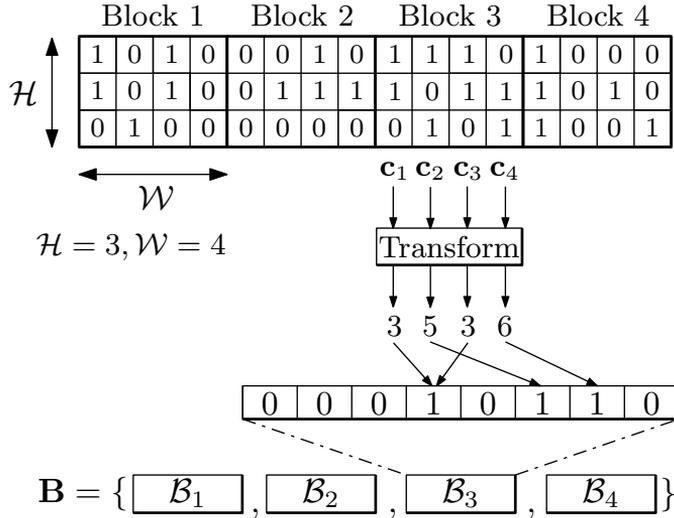
## 4.2 System Basics

This section presents operational details of the Bloom filter based scheme presented in [RBBB15], which is the basis of work performed during the course of this thesis project. Based on the specifications from that article, the system has been re-implemented (and later expanded upon) for use in this project.

### 4.2.1 Template Transformation

The biometric templates are transformed from the iris code representation as described below and conceptually illustrated in figure 4.1.

1. The iris code matrix is divided into equally sized blocks of certain width and height (denoted as  $\mathcal{W}$  and  $\mathcal{H}$ , respectively).
2. Instead of employing multiple hash functions, a simpler mapping is used. A block is mapped into a Bloom filter ( $\mathcal{B}$ ) by converting its columns (denoted as  $\mathbf{c}_1 \dots \mathbf{c}_W$ ) of binary digits into decimal format. This step differs from the traditional Bloom filter element addition: instead of a single item being processed through multiple hash functions, multiple elements (columns) are processed through a single hash function. The end result, however, is the same - a set of values  $h_1 \dots h_W$ . Subsequently, the corresponding indices in  $\mathcal{B}$  are set to 1.
3. Performing step 2 for all the iris code blocks results in a representation, where a template is a set of Bloom filters (denoted as  $\mathbf{B}$ ).



**Figure 4.1:** The process of generation of a Bloom filter set from an iris code (adapted from [RBBB15])

Due to rotation-compensating properties, in most height and width configurations, this representation requires fewer bit comparisons than the iris code. In the iris code, the possible misalignments must be accounted for by pre-computing and storing or computing on the fly multiple rotations of the original template. The size of a Bloom filter based template is calculated as shown in equation 4.1, where  $IC_W$  is the width of an iris code based template (here, 512).

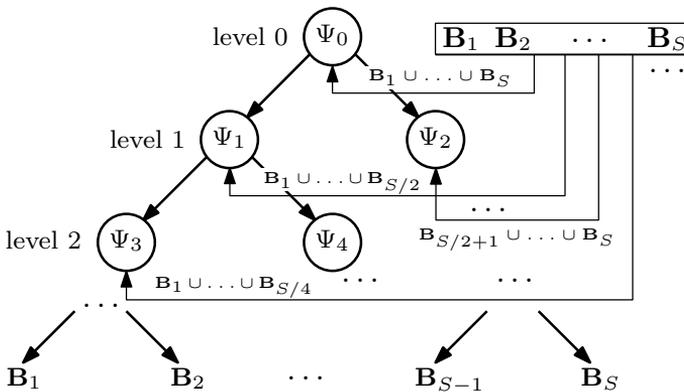
$$\tau = 2^{\mathcal{H}} * \frac{IC_W}{\mathcal{W} * 2} \quad (4.1)$$

Note, that at this point, the system can be used in a verification scenario. The only change thus far has been the transformation of the iris code based templates to sets of Bloom filters. In the next section, the set-up for an identification scenario is described.

### 4.2.2 Tree Construction

For ( $S$ ) templates enrolled into the system, a binary search tree is constructed. The tree nodes in breadth first ordering are denoted  $\Psi_0 \dots \Psi_{S-1}$ . This process is described below and shown conceptually in figure 4.2.

1. The tree root is created as an element-wise union of all enrolled templates (i.e  $\Psi_0 = \mathbf{B}_1 \cup \mathbf{B}_2 \cup \dots \cup \mathbf{B}_S$ ).
2. The child nodes at subsequent tree levels are generated recursively by taking the element-wise union of half of the templates of the parent node.
3. The templates themselves ( $\mathbf{B}_1 \dots \mathbf{B}_S$ ) are inserted as the tree leaves.
4. Insertion of new templates into an already built tree is possible in  $O(\log S)$  steps. A removal of template(s) requires the tree to be fully rebuilt, and thus  $O(S \log S)$  steps are needed.



**Figure 4.2:** The process of constructing a binary search tree of Bloom filter based templates (adapted from [RBBB15])

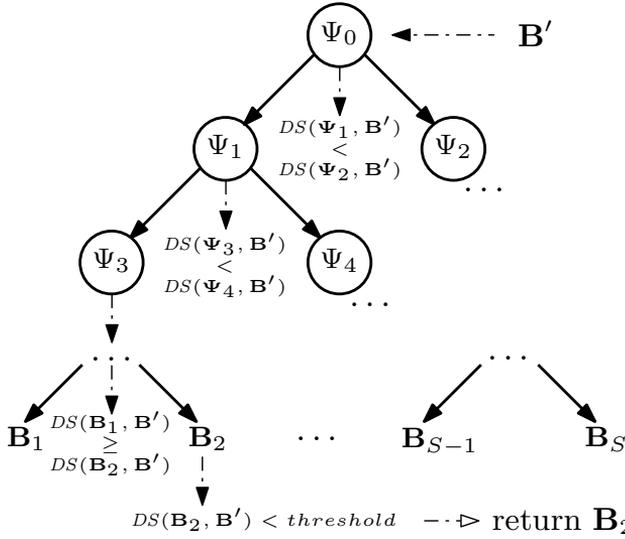
### 4.2.3 Tree Traversal

Given a membership query ( $\mathbf{B}'$ ) for a new Bloom filter based template, a tree traversal takes place beginning at the root ( $\Psi_0$ ). At each step, dissimilarity scores for both children of the current node ( $\Psi_{current+1}$  and  $\Psi_{current+2}$ ) are computed and the direction of the traversal is decided based on comparison of these scores. Upon reaching a leaf, the final decision is made by comparing the dissimilarity score there against a threshold, which has been previously computed using a training set of templates disjoint from the enrolled templates set. As shown in equations 4.2 and 4.3, the dissimilarity of two Bloom filter sets ( $\mathbf{B}$  and  $\mathbf{B}'$ ) is calculated as a score-level fusion of the pairwise dissimilarities between all the individual Bloom filters in the sets ( $\mathcal{B}_i$  and  $\mathcal{B}'_i$ ). The intuition is to count the number of agreeing bits in each Bloom filter pair and, since the number of bits set to 1 can vary, normalise by the Hamming weight of these Bloom filters. Then, the average dissimilarity score for the whole set of Bloom filters can be calculated. Observe, that in a concrete implementation,  $DS(\mathcal{B}, \mathcal{B}')$  can be reduced to only 6 efficient CPU instructions.

$$DS(\mathbf{B}, \mathbf{B}') = \frac{1}{N} \sum_{i=1}^N DS(\mathcal{B}_i, \mathcal{B}'_i) \quad (4.2)$$

$$DS(\mathcal{B}, \mathcal{B}') = 1 - \frac{|\mathcal{B} \wedge \mathcal{B}'|}{\frac{1}{2}(|\mathcal{B}| + |\mathcal{B}'|)} \quad (4.3)$$

Optionally, the sequence of scores obtained during the tree traversal can be required to be ordered, since at each subsequent level lower scores are generally expected (i.e.  $DS_{level_0} > DS_{level_1} > \dots > DS_{level_{log_2}}$ ). This makes it much more difficult for the impostor attempts to get accepted, potentially giving a lower false acceptance rate and significantly reduces the workload for the impostor attempts, since the ordering is checked on the fly during the traversal and impostors can be rejected early. However, the genuine attempts may get rejected this way too, potentially lowering the true acceptance rate. The whole lookup process for one Bloom filter tree is demonstrated formally in algorithm 4.1 and conceptually in figure 4.3.



**Figure 4.3:** The lookup process in a tree constructed from the enrolled Bloom filter templates (adapted from [RBBB15])

As more templates are stored in one tree, the risk of falsely matching bits increases. This affects the capability of making a correct traversal direction decision for membership queries of subjects that are enrolled in the system (i.e. genuine attempts). Therefore, it becomes necessary to build multiple trees and spread the enrolled templates among them in order to alleviate this issue. This change means that the lookup process is slightly different, as shown conceptually in figure 4.4 and formally in algorithm 4.2. Unfortunately, this change raises new concerns, particularly:

1. Building more trees causes higher workload, as more template comparisons are required per lookup. Observe, that with the scheme, as presented here, all the constructed trees *must* be traversed. Currently, there is no way of knowing beforehand if a tree is likely to contain a match.
2. As more templates are added to a tree, the Bloom filters fill up with 1's. This data denseness negatively affects the accuracy of the system; this issue can be alleviated by building more trees for template storage. It is therefore necessary to determine precisely when additional trees are needed; in other words, when the data becomes too dense.

Chapter 5 addresses the first issue; the latter is looked into in section 4.3.

---

**Algorithm 4.1** Lookup in the Bloom filter based scheme - single tree

---

```

1: procedure TREE LOOKUP(tree, probe, decisionthreshold)
2:   previousscore  $\leftarrow \infty$ 
3:   currentscore  $\leftarrow \infty$ 
4:   currentnode  $\leftarrow$  tree root
5:   repeat
6:     scoreleft  $\leftarrow$  dissimilarity(probe, leftchild)  $\triangleright$  See equation 4.2
7:     scoreright  $\leftarrow$  dissimilarity(probe, rightchild)
8:     if scoreleft > scoreright then
9:       currentnode  $\leftarrow$  leftchild
10:      currentscore  $\leftarrow$  scoreleft
11:     else
12:       currentnode  $\leftarrow$  rightchild
13:       currentscore  $\leftarrow$  scoreright
14:     end if
15:     if currentscore > previousscore then  $\triangleright$  Optional
16:       return nil
17:     else
18:       previousscore  $\leftarrow$  currentscore
19:     end if
20:   until isleaf(currentnode)
21:   if currentscore < decisionthreshold then
22:     return currentnode  $\triangleright$  Likely identity found
23:   else
24:     return nil
25:   end if
26: end procedure

```

---



---

**Algorithm 4.2** Lookup in the Bloom filter based scheme - multiple trees

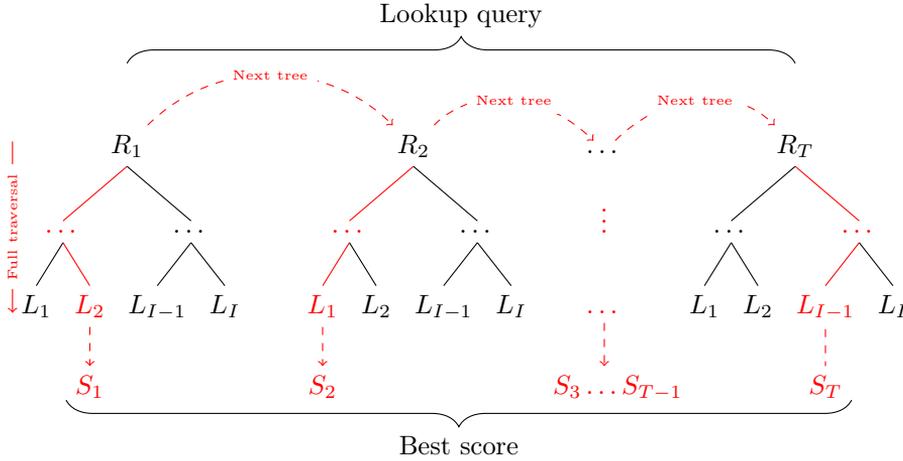
---

```

1: procedure LOOKUP(trees, probe, decisionthreshold)
2:   candidates  $\leftarrow$  empty list
3:   for all trees do
4:     identity  $\leftarrow$  TREE LOOKUP(tree, probe, decisionthreshold)
5:     if identity  $\neq$  nil then
6:       add identity to candidates
7:     end if
8:   end for
9:   return candidates
10: end procedure

```

---



**Figure 4.4:** Lookup in a system with multiple trees constructed. First, *all* the constructed trees are *fully* traversed. Subsequently, from among the individual leaf scores, the best one is chosen. In the figure,  $R$  stands for a tree root,  $L$  for a leaf and  $S$  for a leaf score. The red lines show example paths taken during a lookup scenario, starting at  $R_1$ .

#### 4.2.4 Relevant Configurations

Three important variables in the system are the block height, block width and the number of constructed trees. These can be adjusted within a certain range, minding the issues outlined below.

- Higher height values will make the Bloom filter template size too large. On the other hand, lower heights are not very well suited for the identification scenario - there, bit collisions would occur frequently and the Bloom filters would fill up too quickly as more templates are added.
- Wider blocks would have too many bit collisions upon the transformation from the iris code representation. Narrower block size would mean a total template size increase. Additionally, narrowing the blocks diminishes the rotational invariance properties of the templates.
- Constructing more trees reduces their individual number of levels. Very few large trees imply many templates stored in each one of them and potential problems with Bloom filter overfilling. Very small trees imply few templates per tree, thus diminishing the workload reduction gain.

## 4.3 General Model of Bloom Filter Based Indexing

A potential problem in the Bloom filter based approach is storing too many templates in a tree; doing so can cause the true positive performance to deteriorate. The proposed solution is to build more trees as needed. However, it is not immediately obvious when exactly the additional trees are necessary.

The model presented in this section allows an estimation of the overlap between a random template ( $\mathbf{B}$ ) and a tree root ( $\Psi_0$ ), which is a crucial piece of information for deciding when it is necessary to build more trees for template storage. The reason for modelling the root node specifically is clear - it consists of more templates than any other node in the tree. Recall (section 4.2.2), that the root was constructed as a union of all templates added to the tree, while subsequent nodes only contained fractions thereof. Consequently, the root is where the probability of false bit matches occurring is highest. Thus, it can be confidently assumed, that if correct decisions can be made at the tree root level, then it also is the case at the subsequent, lower tree levels.

### 4.3.1 Random Template and Tree Root Emulation

A Bloom filter based template is essentially a set of sets of unique integer values within a certain range. Recall that these are obtained via a simple transformation as shown in section 4.2.1. During a matching attempt, a pairwise comparison of the sets from probe and reference templates takes place (see section 4.2.3). Given the assumption that all sets exhibit similar characteristics, for the purposes of the model, the discourse is simplified to looking at a single set of integers (i.e. one member of the set of sets).

Let  $\mathcal{B}$  denote a Bloom filter created from an iris code block of width  $\mathcal{W}$  and height  $\mathcal{H}$ . Assuming that all values in the block are *mutually independent* and drawn from a *uniform distribution*, then:

$$\mathcal{B} = \{x \in \mathbb{N}_0 \mid 0 \leq x < 2^{\mathcal{H}}\}, |\mathcal{B}| = \mathcal{W} - \mathcal{D}(\mathcal{W}, 2^{\mathcal{H}}) \quad (4.4)$$

Here,  $\mathcal{D}(\mathcal{W}, 2^{\mathcal{H}})$  is an invocation of a function  $\mathcal{D}(n, m)$ , which calculates the (mean) expected number of duplicates when drawing, with replacement,  $n$  values from a uniform distribution of  $m$  possible values, as shown in the equation 4.5. The steps included in derivation of this formula are not of interest for

the purposes of this thesis; it is merely a variation on the well-known general birthday problem<sup>2</sup>.

$$\mathcal{D}(n, m) = n - m * \left(1 - \left(1 - \frac{1}{m}\right)^n\right) \quad (4.5)$$

However, iris code columns are *not mutually independent* - neighbouring columns have a high probability of being equal [RBB13]. Thus, a block of data from an iris code will have fewer unique values than shown in the equations above. Let  $\epsilon$  denote the difference between expected number of duplicate values in an iris code and randomly generated, mutually independent values. Then, a Bloom filter created from an arbitrary iris code block can be denoted as follows:

$$\mathcal{B} = \{x \in \mathbb{N}_0 \mid 0 \leq x < 2^{\mathcal{H}}\}, |\mathcal{B}| = \mathcal{W} - \mathcal{D}(\mathcal{W}, 2^{\mathcal{H}}) - \epsilon \quad (4.6)$$

$\epsilon$  varies depending on parameters such as the dataset itself, feature extractor and block sizes. It can be readily approximated using a training set, or potentially by a more elaborate analysis of the nature of the iris code.

This representation is not ideal, since in real data certain Bloom filter index values turn out to be more likely to occur than others (i.e. the distribution is not completely uniform; see appendix B.1). Furthermore, in the real data, the  $\epsilon$  value is subject to small variations between different templates, while it is fixed in the model. For the sake of simplicity, let's accept these minor imperfections as potential sources of inaccuracy in the model and proceed.

Finally, by extension of the above reasoning, a root of a Bloom filter template tree can also be modelled as a set of unique integers. Recall (section 4.2.2), that it is created by taking the union of multiple Bloom filter templates. Let  $\mathcal{R}_{\mathcal{K}}$  denote a tree root consisting of  $\mathcal{K}$  individual Bloom filters.

$$\mathcal{R}_{\mathcal{K}} = \mathcal{B}_1 \cup \mathcal{B}_2 \cup \mathcal{B}_3 \cup \dots \cup \mathcal{B}_{\mathcal{K}} \quad (4.7)$$

It follows trivially, that  $\mathcal{N}_{\mathcal{K}} = \mathcal{K} * |\mathcal{B}|$  is the expected number of *non-unique*

---

<sup>2</sup>An observant reader will notice, that the presented formula actually computes the expected number of distinct values and subtracts that from the total number of samples, thus obtaining the expected number of duplicates. This means, that the expected number of unique items could have been used directly in the equation 4.4. The purpose of choosing the indirect route was that it allows to make the high-level reasoning about the model more apparent.

items in  $\mathcal{B}_1 \dots \mathcal{B}_K$ . Then, the expected number of *unique* items in a root is:

$$|\mathcal{R}_{\mathcal{K}}| = \mathcal{N}_{\mathcal{K}} - \mathcal{D}(\mathcal{N}_{\mathcal{K}}, 2^{\mathcal{H}}) \quad (4.8)$$

As a concrete example, consider the following system configuration:  $\mathcal{W} = 16$ ,  $\mathcal{H} = 8$ ,  $\mathcal{T} = 1$ ,  $\epsilon = 8$  and  $\mathcal{S} = \mathcal{K} = 25$ . From the equation 4.6, the expected number of unique items in a single Bloom filter is calculated:  $|\mathcal{B}| = 16 - \mathcal{D}(16, 2^8) - 8 \approx 7.5$ . Then, using the equation 4.8, the number of unique items in the root is calculated:  $|\mathcal{R}_{\mathcal{K}}| = 7.5 * 25 - \mathcal{D}(7.5 * 25, 2^8) \approx 133$ , i.e. the tree root is  $\frac{133}{2^8} \approx 52\%$  full.

With this reasonable approximation of a random (impostor) Bloom filter template (equation 4.6) and a tree root (equation 4.8), the final task can now be tackled. This task is estimating the expected relative overlap between the two. In other words, the item of interest is now a probability distribution for different numbers of items being identical in a tree root and a random template.

### 4.3.2 Overlap between Root and Random Template

The final step is estimating the overlap between a tree root  $\mathcal{R}_{\mathcal{K}}$  and an arbitrary, random (impostor) Bloom filter template  $\mathcal{B}$ . This simply means computing the expected length of a set intersection of these two. Let  $\mathcal{O}$  denote this overlap:

$$\mathcal{O} = |\mathcal{R}_{\mathcal{K}} \cap \mathcal{B}| \quad (4.9)$$

The expected overlap outcome follows a hypergeometric distribution with parameters listed below.

$$P(\mathcal{O} = k) = \frac{\binom{|\mathcal{B}|}{k} \binom{2^{\mathcal{H}} - |\mathcal{B}|}{|\mathcal{R}_{\mathcal{K}}| - k}}{\binom{2^{\mathcal{H}}}{|\mathcal{R}_{\mathcal{K}}|}} \quad (4.10)$$

**Population size** The number of possible Bloom filter values:  $2^{\mathcal{H}}$ .

**Successes** The number of expected unique values in a random template:  $|\mathcal{B}|$ .

**Draws** The number of expected unique values in a tree root:  $|\mathcal{R}_{\mathcal{K}}|$ .

**Observed successes** The number ( $k$ ) of overlapping items in range  $1 \dots |\mathcal{B}|$ .

Later on, the mean of that distribution (equation 4.11) will be used where a single number metric is needed instead of an entire distribution. Let  $\Theta$  denote said metric.

$$\Theta = |\mathcal{R}_K| * \frac{|\mathcal{B}|}{2^{\mathcal{H}}} \quad (4.11)$$

The distribution itself will be used to validate the fit of the model to data - first against some pseudo-randomly generated values (section 4.3.3) and then against real iris data (section 7.2.2.1).

### 4.3.3 Validation

The empirical validation can be performed by an experiment, where two sets that emulate the template and root are generated from a pseudo-random distribution and the size of their intersection is computed. Performing this experiment many times (in this case, 100.000) reveals that, at a glance, the resulting distributions fit closely, as shown in figure 4.5. Plotted are, on the x-axis, the relative overlap (i.e. what percentage of values in the two sets is expected to overlap), against the probability of that occurrence on the y-axis. Naturally, one cannot reliably assess the fit of the model upon visual inspection of the distributions - these serve only for illustration purposes. A simple metric to assess the closeness of the fit of the two distributions is therefore necessary. One such metric, which can be used for discrete distributions, is the Hellinger Distance, defined as follows:

$$\mathcal{H}_{\mathcal{D}} = \sqrt{1 - \mathcal{B}_C} \quad (4.12)$$

Where  $\mathcal{B}_C$  stands for the Bhattacharyya coefficient for distributions  $d$  and  $d'$ :

$$\mathcal{B}_C = \sum_{i=1}^K \sqrt{(d_i * d'_i)} \quad (4.13)$$

In this particular case, the possible values of  $\mathcal{H}_{\mathcal{D}}$  are  $\{x \in \mathbb{R} \mid 0.0 \leq x \leq 1.0\}$ . For all the considered system configurations, the resulting distances  $\mathcal{H}_{\mathcal{D}}$  have a mean  $\mu = 0.11$ , with a standard deviation of  $\sigma = 0.07$ . Figure 4.5c shows a histogram of the obtained  $\mathcal{H}_{\mathcal{D}}$  values. This signifies an excellent fit between the distributions from the theoretical model and the ones generated empirically, thus validating the logic behind the approach. The fit between the model and real iris data will be presented in chapter 7.

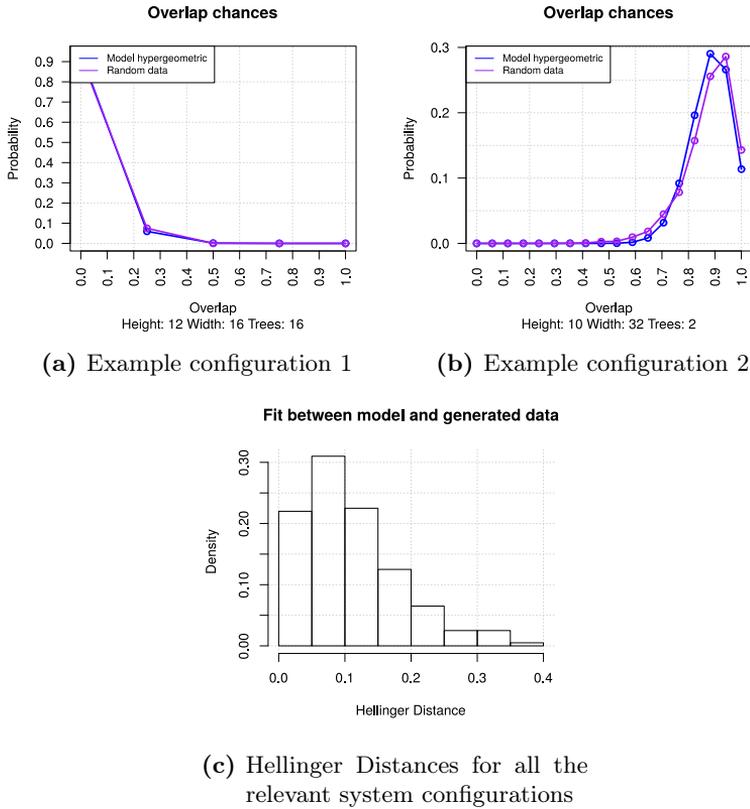


Figure 4.5: The fit between the model and randomly generated data

## 4.4 Summary

In the first two sections of this chapter, the fundamental details of a Bloom filter based biometric system have been outlined, partially based on the original article in which this approach was proposed [RBBB15].

The third section contains a new contribution: a proposal for a way to emulate an impostor Bloom filter based template and a root of a tree of enrolled templates. A method for computing the expected overlap between these two has also been introduced. Based on this expected overlap value, one can determine at which point a tree root becomes overfilled and construction of additional trees is needed to maintain good biometric performance for genuine attempts. It is important to be able to estimate this, since the consequence of building more

trees is increased workload - one would therefore seek to do so only when absolutely necessary. The theoretical model has been shown to fit well with data generated from a pseudo-random distribution, where the special characteristics of the biometric data are accounted for by one variable,  $\epsilon$ , for now estimated from a training set. Since only minor simplifications and potential error sources have been introduced by the model, it is expected to fit the real iris data well. The next chapter will present improvement proposals for the Bloom filter based approach.

# Improvements for the Bloom Filter Approach

---

This chapter presents changes that can be made in order to decrease the computational workload and ensure the scalability of the Bloom filter based system for biometric identification.

## 5.1 Multiple Trees

The biggest limiting factor of the Bloom filter based approach is the computational complexity increase when multiple trees are constructed. Let  $\mathcal{C}$  denote the number of necessary Bloom filter template comparisons, for a single lookup in an identification scenario. Note, that  $\mathcal{C}$  is directly related to  $\rho$ :  $\rho = \frac{\mathcal{C}}{\mathcal{S}}$ . In the simple case, where all the constructed trees are traversed, this cost can be computed using equation 5.1, where  $\mathcal{S}$  is the number of subjects enrolled in the system, and  $\mathcal{T}$  the number of constructed trees.

$$\mathcal{C} = \begin{cases} 2 * \log \mathcal{S} - 1 & \text{if } \mathcal{T} = 1 \\ \mathcal{T} * (2 * (\log \frac{\mathcal{S}}{\mathcal{T}} - 1)) & \text{if } 1 < \mathcal{T} < \frac{\mathcal{S}}{2} \\ \mathcal{S} & \text{if } \mathcal{T} \geq \frac{\mathcal{S}}{2} \end{cases} \quad (5.1)$$

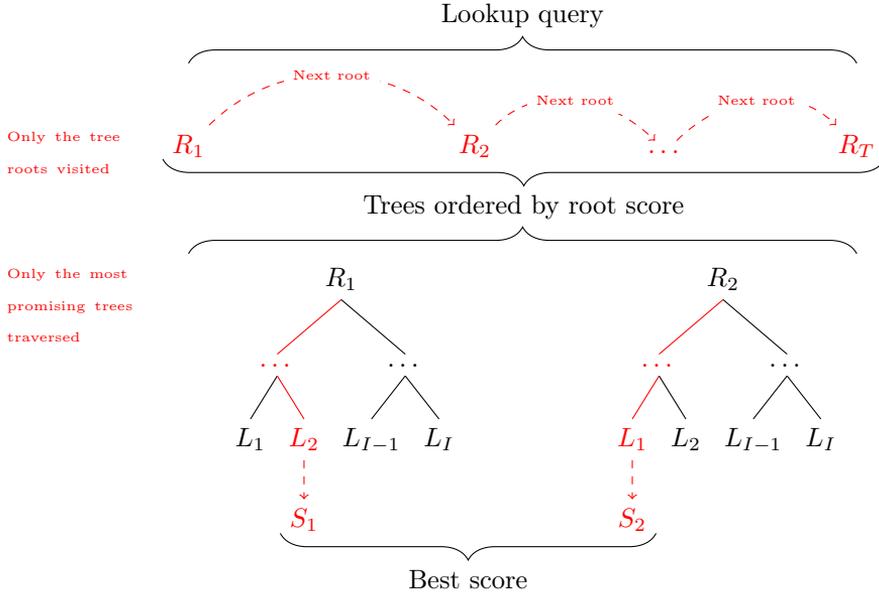
The basic case is the traversal of one tree ( $\mathcal{T} = 1$ ).  $\log S$  represents the number of tree levels and the factor of 2 is included, since at each level both nodes are checked. Finally, one node is subtracted, since the root is skipped during the traversal. In the second case, the templates are spread among multiple trees. Thus, the number of levels in each tree decreases to  $\log \frac{S}{\mathcal{T}}$  and the procedure must be repeated multiple times ( $\mathcal{T}$ ). The last case, where  $\mathcal{T} \geq \frac{S}{2}$  is not of interest, since it corresponds to an exhaustive search of the database. It was included in the above equation for the completeness sake and will be omitted in other equations in this chapter. For larger databases, where many trees have to be constructed to avoid problems outlined in section 4.3, the associated workload increase is non-trivial. It is an issue that must be resolved in order for the system to be truly scalable.

Often, in order to reduce the workload, one is willing to concede receiving a full list of candidate identities and instead stop algorithm execution when the first plausible identity is returned. This will, on average, happen after traversing half of the trees, thus decreasing the number of traversed trees by a factor of 2, as shown in equation 5.2.

$$c = \begin{cases} 2 * \log S - 1 & \text{if } \mathcal{T} = 1 \\ \frac{\mathcal{T}}{2} * (2 * (\log \frac{S}{\mathcal{T}} - 1)) & \text{if } 1 < \mathcal{T} < \frac{S}{2} \end{cases} \quad (5.2)$$

Further decrease in the number of necessary template comparisons is achieved by the proposed improvement demonstrated conceptually in figure 5.1 and formally in algorithm 5.1. The key idea is to only traverse  $\mathcal{N}$  most promising tree(s) instead of them all;  $\mathcal{N} \ll \mathcal{T}$ . *In other words, a pre-selection step is added to the indexing scheme.* The trees to traverse are selected based on computing the dissimilarity scores between the probe and all the tree roots, thus allowing to quickly find the most promising trees and only traverse these. The number of template comparisons in the worst case can be calculated using equation 5.3. The cost of the pre-selection step is simply the number of tree roots checked ( $\mathcal{T}$ ) and it is added onto the formula for multiple tree traversal. The significant change in that formula is the replacement of the  $\mathcal{T}$  factor with the much smaller  $\mathcal{N}$  factor. Additionally, due to the traversal being ordered (most promising trees first), the actual number of necessary tree traversals will (non-deterministically) tend to be lower than in the presented formula.

$$c = \begin{cases} 2 * \log S - 1 & \text{if } \mathcal{T} = 1 \\ \mathcal{T} + \mathcal{N} * (2 * (\log \frac{S}{\mathcal{T}} - 1)) & \text{if } 1 < \mathcal{T} < \frac{S}{2} \end{cases} \quad (5.3)$$



**Figure 5.1:** Lookup in a system with multiple trees constructed and selective tree traversal. First, *only* the dissimilarity scores between the probe and the tree roots are computed, resulting in an ordered list of the trees. Subsequently, only the most promising  $N$  trees are traversed. Amongst these, the best leaf score is chosen.

---

**Algorithm 5.1** Lookup in the Bloom filter based scheme - improved

---

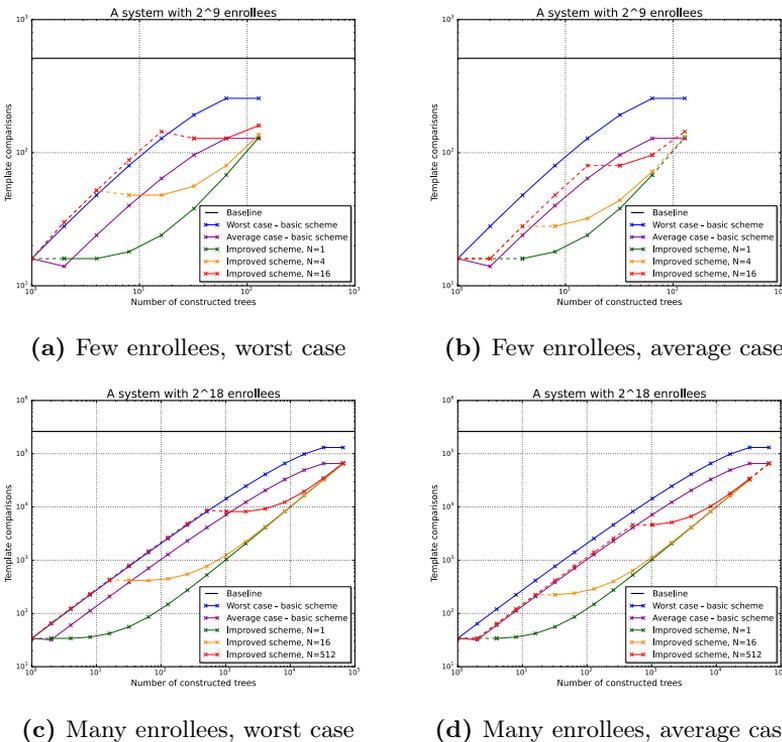
```

1: procedure LOOKUP( $trees, N, probe, decisionthreshold$ )
2:    $candidates \leftarrow$  empty list
3:   for all  $trees$  do
4:     compute  $dissimilarity(probe, treeroot)$ 
5:   end for
6:    $chosentrees \leftarrow$  select  $N$  trees with lowest dissimilarity scores at root
7:   for all  $chosentrees$  do
8:      $identity \leftarrow$  TREE LOOKUP( $tree, probe, decisionthreshold$ )
9:     if  $identity \neq nil$  then
10:      add  $identity$  to  $candidates$ 
11:    end if
12:  end for
13:  return  $candidates$ 
14: end procedure

```

---

The figure 5.2 illustrates the numbers of template comparisons computed by the equations above. For the proposed scheme, three values of most promising tree(s) selected for traversal ( $\mathcal{N}$ ) are displayed: only a single, best tree (green), a moderate number (orange) and an unnecessarily large number (red). The ideal value of  $\mathcal{N}$  is expected to lie between the green and the orange line. The dashed lines denote the configurations in which the proposed system is infeasible (i.e. where the corresponding configuration of a basic scheme requires fewer template comparisons). This is the case in situations when the number of constructed trees and number of trees chosen for traversal are similar - then, intuitively, the proposed approach obviously does not make sense. Consider, for instance, a worst-case scenario in a system of  $2^9$  enrollees (figure 5.2a), 16 trees constructed and 4 trees traversed (orange line). The number of template comparisons needed for a lookup is then 56, which is less than a half of that of the basic scheme (blue line), where all the 16 constructed trees are traversed.



**Figure 5.2:** Template comparisons per lookup in the proposed selective traversal scheme with small and large numbers of enrolled subjects. The baseline, as well as worst and average cases of the basic Bloom filter scheme are plotted for comparison.

## 5.2 Traversal Direction Decision

Another non-trivial factor in the computational cost of the scheme is the way in which the tree traversal direction decisions are made. Recall (see previous section and algorithm 4.1, between lines 6 and 19), that in the basic scheme, the dissimilarity scores at each tree level for *both* nodes are computed and compared against each other to make a direction decision. This means that a single tree is traversed in  $2 * (\log \frac{S}{T} - 1)$  steps instead of  $\log \frac{S}{T} - 1$  steps.

One can take advantage of the fact that genuine score sequences are expected to decrease and make a quick decision about the traversal direction, as demonstrated in algorithm 5.2. The key idea is that one of the two nodes is checked first and if the score has decreased in comparison with the parent, then that node is chosen immediately instead of also computing the score for the second node. If the score has not decreased, then it is necessary to compute the score for the second node as well. Effectively, on average in half the traversal direction decision cases, it will only be necessary to compute the dissimilarity for only *one* of the child nodes. Furthermore, observe that:

- In the configurations without the selective tree traversal scheme, the root score will now have to be computed in order to be able to make a quick traversal direction decision at the first tree level. By implication, the highest relative decrease of required template comparisons will occur in the configurations with a few trees constructed.
- In the selective tree traversal scheme, all the root scores are computed beforehand. Thus, no additional computations have to be made.
- While the factor of 2 is not entirely eliminated from the equation, it is significantly reduced, albeit non-deterministically. The efficacy of this scheme will be tested empirically later on.

---

**Algorithm 5.2** Lookup direction decision (fragment) - improved

---

```

1: scoreleft ← dissimilarity(probe, leftchild)
2: if scoreleft < previouscore then
3:   currentnode ← leftchild
4:   currentscore ← scoreleft
5: else
6:   currentnode ← rightchild
7:   currentscore ← dissimilarity(probe, leftchild)
8: end if

```

---

### 5.3 Other

A couple additional, non-deterministic improvements can be added for the duplicate enrolment check (DEC) scenarios. In these cases, the identity of the probe template subject is not necessarily important - of interest is whether or not the subject has already been enrolled in the system. This means that sometimes it may be possible to make a decision without the need of traversing the entire tree to reach a leaf. Two possibilities are:

- If there is a very large decrease in the dissimilarity score between two consecutive tree levels, then it is extremely likely that the subject's template is in the tree. Impostor templates hardly ever exhibit such behaviour.
- If a dissimilarity score at one of the non-leaf levels is already below the final acceptance threshold, then it is very likely that the subject's template is present in the tree. This approach works, since for genuine attempts the dissimilarity score sequence is expected to decrease at each consecutive tree level. Thus, continuing tree traversal would most likely only cause the score to improve.

Finally, as has been shown earlier, some indexes in Bloom filters are more likely to occur than others. Most notably, the first (0) and last ( $2^h - 1$ ) index are frequent. These correspond to iris columns consisting entirely of 0's and 1's, respectively. These indexes carry very little discriminative power and can be therefore eliminated from consideration during matching attempts. In so doing, the dissimilarity scores will be slightly impaired, but this impairment would have much larger effect on the impostor attempts than the genuine attempts. As a consequence, the false positive rate is expected to be a bit lower.

### 5.4 Multibiometrics

In [HJP99] the concept of combining multiple biometric characteristics into one system is proposed and evaluated. The key idea is that fusing information from separate sources should improve the biometric performance. A more comprehensive look at research and developments in multibiometrics can be found in [RNJ06].

In this project, an early study into feasibility of multi-iris indexing is performed. Although, in itself, multi-iris biometrics is not a new idea (see e.g. [WWZQ07]),

best to this author's knowledge, it is the first such attempt in the scientific literature for biometric indexing schemes. This early study may serve as a basis for research aiming at integrating other biometric characteristics (e.g. face) into a multibiometric system utilising the Bloom filter based scheme. The proposed approach is based on a feature level fusion of two irides from a single subject upon enrolment and prior to a matching attempt. Let  $\mathbf{B}_M$  denote a multi-iris template of a subject. It consists of the left eye template ( $\mathbf{B}_{\text{left}}$ ) and the right eye template ( $\mathbf{B}_{\text{right}}$ ) fused together, as shown in equation 5.4. In the concrete implementation, this is simply an element-wise union of two Bloom filter sets.

$$\mathbf{B}_M = \mathbf{B}_{\text{left}} \cup \mathbf{B}_{\text{right}} \quad (5.4)$$

The tree construction and the identification scenario then follow in the same way as described earlier in this chapter. Due to more information carried by the fused template, one can expect an increase in the true positive identification rate and a decrease in false positive identification rate. Overall, the biometric performance is expected to be superior to that based on the single iris templates. The only additional (negligible) cost of the Bloom filter based multi-iris scheme is the fusion of the templates. Observe also, that the system can take advantage of all the workload reducing improvements proposed in the previous sections.

## 5.5 Summary

Previous sections have demonstrated multiple possible improvements for the basic Bloom filter scheme presented in chapter 4. Their aim is to reduce the workload in lookup scenarios in terms of required template comparisons. These changes will necessarily also have a non-negligible (positive or negative) influence on the biometric performance of the system. Additionally, a multi-iris extension of the scheme, which is expected to be beneficial for the biometric performance, has been proposed. Chapter 7 will, among other matters, present experimental results of the scheme with and without utilising the proposed changes.



# Experimental Setup

---

This chapter contains an overview of the raw data selected for use during this project, its preparation for the experiments and a description of how the experiments were conducted.

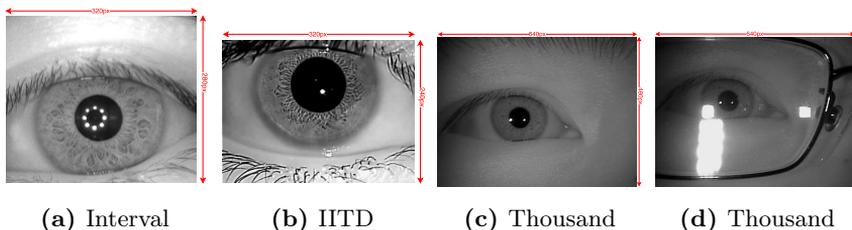
## 6.1 Chosen Datasets

Many publicly available iris datasets exist (see, for example, a summary in [BHF08]). For use during the experimental phase of this project, three distinct datasets were selected. Example images are shown in figure 6.1, while brief descriptions are provided below. Notice the differences in the image quality, image size and iris diameter.

**CASIA-V4 Iris Thousand** At the time of this writing, it is the largest publicly available dataset. It consists of images from 1000 subjects - 10 for each eye, thus making the total number of images 20.000. The variability of the data is high: the extra illumination source sometimes is on/off and in many cases, the subjects are wearing glasses. These properties make it a challenging dataset in the context of biometric performance [Chi].

**CASIA-V4 Iris Interval** Contains 2639 images from 249 subjects. The images are of very high quality, with few occlusions and reflections; the iris texture details are clearly visible [Chi].

**IITD** Contains 2240 images from 224 subjects. The quality of most images is very high [IIT].



**Figure 6.1:** Example images from each dataset

Since the IITD and Interval datasets have similar properties, they have been combined into one larger dataset. The purpose of this is for the Bloom filter system to be able to have more enrolled subjects during identification scenario experiments. Henceforth, this dataset will be referred to as "Combined".

## 6.2 Iris Biometric Processing Chain

This section describes the process of converting a raw eye image into the representation worked with throughout this project. It is a concrete example for the abstract, generic description of the process presented in section 2.2.

The data preparation steps and concrete methods involved are:

**Segmentation** For the **Thousand** dataset, the OSIRIS software package is employed [Tel], since it uses the Viterbi algorithm, which is well-suited for noisy images [SGSD12]. For the **Combined** dataset, the USIT software package [RUWH16] is used. It provides an implementation of the contrast-adjusted Hough transform, which is well-suited for high quality images. The purpose of this step is finding the boundaries of the iris and the pupil. Additionally, a mask is produced to cover image occlusions, reflections and other noise (e.g. eyelashes).

**Normalisation** Preparation for the feature extraction step. The differences in pupil dilation and distance from the camera are compensated for. To

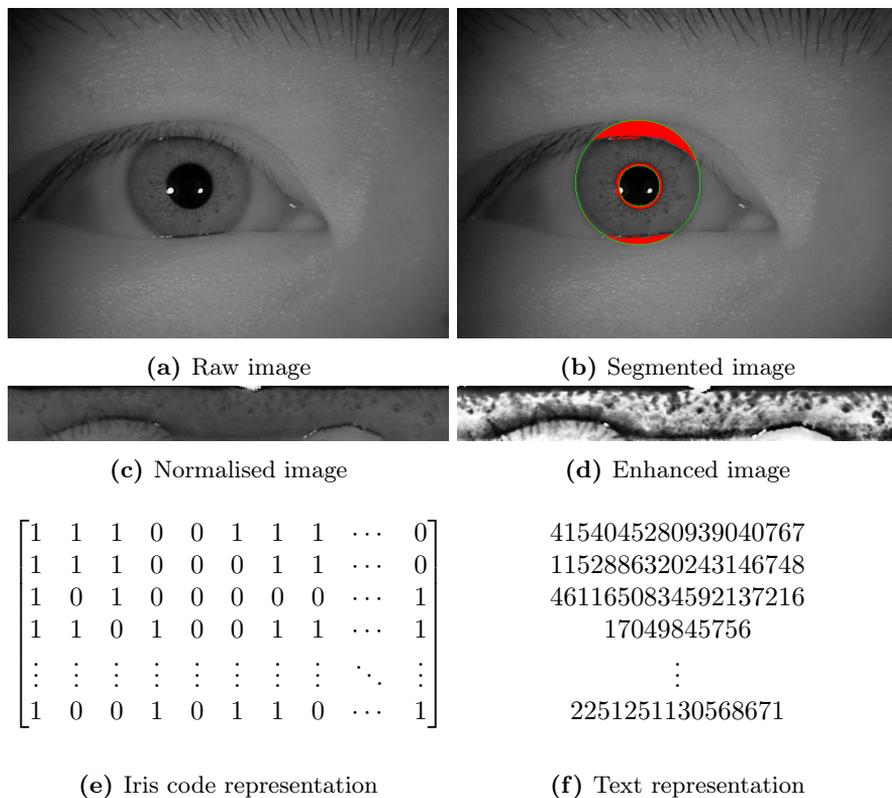
achieve this, Daugman's rubber sheet model is used. In this process, the Cartesian points of the iris image are mapped to a polar coordinate system [Dau04a].

**Enhancement** Further preparation for the feature extraction step. Contrast-limited adaptive histogram equalization from the USIT software package is used. The goal is to make the iris texture features more prominent. By doing so, the recognition accuracy of the system is improved later on.

**Feature extraction** Two methods which follow similar processing steps are utilised. First, the normalised iris texture signal is decomposed to one-dimensional representation. Then, filters are applied: the quadratic spline wavelet (QSW) or the complex Log-Gabor convolution (LG). The last step is binary encoding of the data into the iris code matrix format. For QSW, a fixed number of sub-bands is produced, from which the responses' local extrema are found. The LG approach discretizes the produced complex values' phase angle. In both cases, the final result is a binary matrix with a width of 512 bits and a height of 20 bits. Implementations of both feature extractors provided by the USIT software package are used.

**Conversion to text** An optional, albeit very useful step. An iris code is stored in a text file containing 160 unsigned integer values (`uint64`). Thereby, in the experiments, the iris code can be loaded directly from this file, instead of repeating (parts of) the above process. The `uint64` representation also confers benefits of being able to use the fast operations of the CPU, such as the logical operators and bitcounts. These are utilised in the comparison step of the naïve baseline system implementation (see section 2.2.2) and during and after transformation to the Bloom filter representation (see section 4.2).

Figure 6.2 illustrates how the data looks at each step of the processing chain.



**Figure 6.2:** Data at each step of the iris biometric processing chain

## 6.3 Excluded Images

In the quality control step, some images had to be excluded from use in this project. This section outlines the reasoning behind such decisions.

### 6.3.1 Dataset Errors

It turns out that there are some errors in the datasets. Three types of errors can be distinguished.

- Eye images from same subject are used multiple times and labelled as separate subjects. This is a problem especially when one of the subjects is

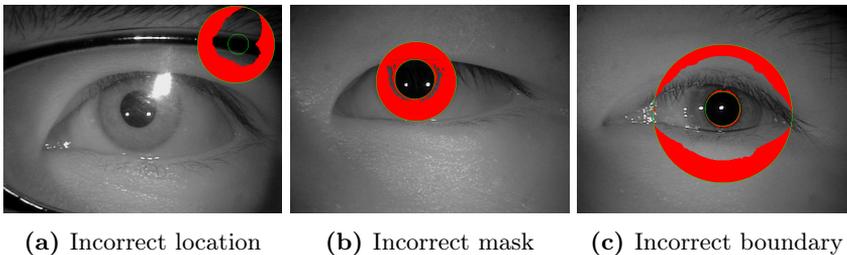
used as an enrollee and another as an impostor, yielding unjustly increased false positive rates.

- Same eye image is used multiple times with different labels for the same subject. For such images, the comparison will result in a 0 dissimilarity score, thus skewing the score distribution.
- Images from one eye only are present for a subject, but are mislabelled as coming from both left and right eye. This is an issue, since for experimental purposes, images from left and right eye are treated as separate subjects. Experiments would then yield incorrectly higher false positive rates.

These errors are notoriously difficult to discover, although several cases have been found and excluded.

### 6.3.2 Preprocessing Failures

It is important to be aware that the preprocessing is not flawless and failures may occur. The segmentation step is of interest here, since it is vulnerable to errors which then propagate further down the processing chain. Furthermore, segmentation failures can actually be spotted upon visual inspection, as shown in figure 6.3 and by using an automated check outlined in section 6.3.3.



**Figure 6.3:** The typical segmentation failures

Through a visual inspection the worst and most obvious failures were found. This has resulted in removal of approximately 500 images (around 2.5% of all) from the **Thousand** dataset. The images from the **Combined** dataset are of much better quality - there were only a couple cases of failed segmentation.

### 6.3.3 Automated Quality Check

In addition to the manual filtering via visual inspection, an automated quality check has been put in place. As shown in algorithm 6.1, it uses the generated masks to determine the fraction of the iris code that which had not been occluded or otherwise deficient. Subsequently, templates with quality score below certain threshold can be rejected.

---

**Algorithm 6.1** A simple segmentation quality check

---

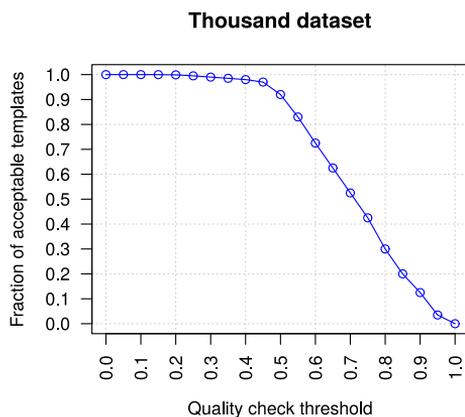
```

1: procedure QUALITY CHECK(mask, threshold)
2:   iriscodebits  $\leftarrow$  10240                                ▷ 20 rows and 512 columns
3:   usedbits  $\leftarrow$  0
4:   for all uint64 in mask do
5:     usedbits  $\leftarrow$  usedbits + popcount(uint64)
6:   end for
7:   usedbitsfraction  $\leftarrow$  usedbits/iriscodebits
8:   return usedbitsfraction > threshold
9: end procedure

```

---

Since the data in the **Combined** dataset is very clean, it was only necessary to apply this procedure for the **Thousand** dataset. Figure 6.4 shows that a quality check threshold of 0.5 is reasonable - around 10% of the worst quality templates are excluded. This is the threshold used for all the **Thousand** dataset results presented in chapter 7.



**Figure 6.4:** Fraction of acceptable templates depending on the quality check threshold

Table 6.1 provides an overview of the raw data used in this project.

Dataset	Instances	Images	Excluded	Resolution	Av. iris diameter	Quality
Thousand	2000	20000	1643	640x480 px	~185 px	Moderate
Interval	395	2639	1	320x280 px	~210 px	High
IITD	448	2240	100	320x240 px	~205 px	High

**Table 6.1:** Raw data overview

## 6.4 Experiments

The dataset has been split into 4 groups: enrolled, genuine, impostor and training (for the final threshold estimation in the Bloom filter scheme). Table 6.2 shows the number of templates in each group. The conducted experiments are listed below.

**Baseline** The basic implementation, which performs comparisons on iris codes. In an identification scenario the database is searched exhaustively, as shown in algorithm 2.1.

**Verification** All possible cross-comparisons for templates from each subject to obtain genuine scores. Impostor scores are obtained by performing all possible comparisons of templates with differing subjects.

**Identification** One template per subject is enrolled (reference). Remaining templates (probes) are compared against the enrolled template to obtain genuine and impostor scores in an open-set scenario.

**Bloom filter** The system as described in chapter 4.

**Verification** As in the baseline experiment above; the only difference is that now the templates have been transformed to the Bloom filter representation.

**Identification** One template per subject is enrolled and from these, the Bloom filter trees are constructed. Genuine and impostor scores are obtained by comparing the probe templates against the tree(s) in an open-set scenario.

**Workload reduction** A repetition of the above identification scenario, with addition of the proposed system improvements described in chapter 5.

**Multi-iris** A repetition of the above identification scenario, using fused templates as shown in section 5.4.

Dataset	Enrolled	Genuine	Impostor	Training
Thousand	1024	8126	9079	128
Combined	512	2157	2045	64
Combined multi-iris	256	914	604	64

**Table 6.2:** Dataset split (templates) for the experiments

## 6.5 Summary

This chapter presented the details of the experimental set-up for this project. This includes the used data, how it was processed and curated, as well as what experiments were performed.

Three datasets were chosen - CASIA-V4 Iris Thousand (large variability, not high quality), CASIA-V4 Interval and IITD (low variability, high quality); the latter two were combined into one larger dataset. The raw eye images from these datasets were transformed to a textual representation of an iris code by a multi-step process, which consists of segmentation, normalisation, enhancement and feature extraction. Due to various errors in the datasets themselves and in the processing chain, a fraction of the images had to be excluded through visual inspection. Additionally, a simple, automated quality check has been implemented. The chapter concluded with a list and short descriptions of empirical experiments that were carried out during the course of this project. Their results are presented in the next chapter.

# Results

---

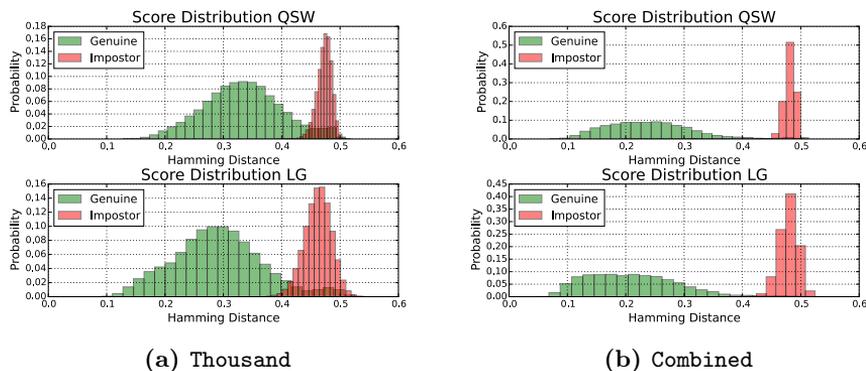
Experimental results are presented in this chapter. The labelling nomenclature in this chapter is as follows: "*baseline*" refers to the naïve, iris code based biometric system. Any mention of the "*basic*" or "*single-tree*" Bloom filter system refers to the system from [RBBB15]. Other Bloom filter system versions are the proposed improvements (e.g. "*selective traversal*"), with the exception of the score sequence ordering from the article cited above. The Bloom filter results are further denoted by the system configuration  $(\mathcal{H}, \mathcal{W}, \mathcal{T}, \mathcal{N})$ .

## 7.1 Baseline

The baseline results come from the naïve biometric system implementation. It is important to establish these, since they will serve as a reference point for the Bloom filter system results.

### 7.1.1 Score Distribution

A distribution of the comparison scores provides valuable insights into the biometric data.



**Figure 7.1:** The Hamming Distance distributions for the verification experiment

By examining figure 7.1, the following can be readily observed:

- Compared with the **Combined** dataset, the genuine score distribution in the **Thousand** dataset is significantly shifted to the right. In other words, the difference between two templates from the same subject (intra-class variation) is higher for the **Thousand** dataset.
- For the **Thousand** dataset, the median of the genuine distribution is around 0.3. Consequently, approximately every third bit in the iris code can be expected to flip between two separate templates from the same subject.
- In the **Combined** dataset, there is nearly no overlap between the genuine and impostor distributions. However, as a consequence of the genuine distribution shift, the overlap between the genuine and impostor distributions in the **Thousand** dataset is significant.

These observations suggest the expected biometric performance of the **Combined** dataset to be very high, while it should be lower for the **Thousand** dataset.

### 7.1.2 EER and ROC

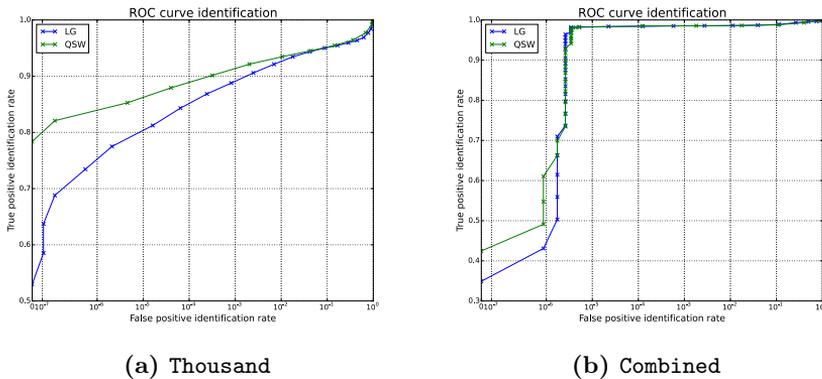
A commonly used metric for reporting performance of a biometric verification system is the equal error rate (EER). It is the point at which the false non-match rate and the false match rate (as defined in the [ISO11] standard) are equal. In table 7.1, it can be seen that the performance of the **Combined** dataset

is excellent, while the performance of the **Thousand** dataset is much worse, although still within the acceptable range. These results are in agreement with the expectations based on the score distributions from the previous section.

Dataset	Extractor	EER
Thousand	LG	5.2%
Thousand	QSW	5.1%
Combined	LG	1.12%
Combined	QSW	1.37%

**Table 7.1:** The EERs achieved during the verification experiment

For a biometric identification system, the commonly used metrics are the true positive identification rate (TPIR) and the false positive identification rate (FPIR) (as defined in the [ISO11] standard). For a fixed database size, as is the case here, these can be plotted as a ROC curve. Figure 7.2 shows these curves in the case where the size of the returned candidate list is 1 (i.e. only the candidate with the best comparison score). In addition to the ROC curves, two specific biometric performance points are defined:  $TP_0$  and  $TP_{0.1}$ , which correspond to the true positive identification rate at 0% and 0.1% false positive identification rate, respectively. Where it is possible to compute these, they will be used as single-value biometric performance indicators and posted together with workload reduction results to give an instant overview of the system configuration outcomes. Otherwise, a single pair of TPIR and FPIR values is reported.



**Figure 7.2:** The ROC curves for the baseline system

As could be expected based on the results of the verification experiment, the **Combined** dataset achieves a very high biometric performance, while that of the **Thousand** dataset is much lower.

### 7.1.3 Workload

An iris code template consists of 10240 bits. In this project rotation compensation is performed for 8 bits in each direction, which gives 17 rotations of the iris code template in total. Overall size of one template can therefore be considered to be  $10240 * 17 = 174080$  bits. A single identification attempt performs an exhaustive search of the database, which means that the penetration rate is 1.0. Finally, the number of subjects enrolled is 1024 in the case of the **Thousand** dataset and 512 in the case of the **Combined** dataset. Using these values, the total workload per lookup can be then computed using the formula presented in equation 3.1. Table 7.2 summarises the baseline system results.

Dataset	Workload		Performance	
	Worst	Average	LG	QSW
Thousand	$\omega \approx 1.78 * 10^8$	$\omega \approx 8.91 * 10^7$	$TP_0 = 53.22\%$ $TP_{0.1} = 88.76\%$	$TP_0 = 78.46\%$ $TP_{0.1} = 92.31\%$
Combined	$\omega \approx 8.91 * 10^7$	$\omega \approx 4.46 * 10^7$	$TP_0 = 34.67\%$ $TP_{0.1} = 98.51\%$	$TP_0 = 42.05\%$ $TP_{0.1} = 98.56\%$

**Table 7.2:** The results of the baseline system

## 7.2 Bloom Filter Approach

This section contains the results of the experiments performed with the system described in chapters 4 and 5.

### 7.2.1 Verification

The first experiment is the verification scenario. It allows to see whether or not the mere template representation change from iris code to Bloom filter has had any effect on the performance. Table 7.3 shows the equal error rates for this experiment. Since the Bloom filter system can be run in many configurations of block width and height (and later, in the identification mode, trees), only the results for the top 5 such configurations are presented in this section. Full results are available in appendix B.1 and B.2.

The effects of the template representation transformation are twofold:

- A significant amount of information is lost, since columns of multiple bits

are consolidated into single indexes in Bloom filters. This makes it more difficult for genuine attempts. Suppose a column has 10 bits and just one of these flips between two separate templates from the same subject. In the iris code representation, this would still mean that 9 out of 10 bits match. In the Bloom filter representation, however, this would mean that the column would be assigned a different index, thus simply resulting in a mismatch.

- The templates become rotation-invariant to a certain degree.

In comparison to the baseline, the **Combined** dataset maintains the high performance in most configurations; in some cases, it is even slightly improved. The performance of the **Thousand** dataset becomes much worse. An explanation for this can be derived from the shifted genuine distribution, shown earlier in figure 7.1. Given an average 1 in 3 chance of a bit flip between two templates, it quickly becomes unlikely for entire columns of many bits to be completely identical. It is therefore not surprising that the comparatively best results for this dataset are obtained at relatively low column sizes.

Dataset	Extractor	EER	Configuration
Thousand	LG	7.43%	$\mathcal{H} : 5 \mathcal{W} : 16$
		7.93%	$\mathcal{H} : 4 \mathcal{W} : 16$
		7.94%	$\mathcal{H} : 5 \mathcal{W} : 8$
		8.04%	$\mathcal{H} : 6 \mathcal{W} : 8$
		8.11%	$\mathcal{H} : 4 \mathcal{W} : 8$
Thousand	QSW	9.13%	$\mathcal{H} : 5 \mathcal{W} : 16$
		9.80%	$\mathcal{H} : 4 \mathcal{W} : 16$
		10.05%	$\mathcal{H} : 4 \mathcal{W} : 8$
		10.06%	$\mathcal{H} : 5 \mathcal{W} : 8$
		10.81%	$\mathcal{H} : 6 \mathcal{W} : 16$
Combined	LG	0.89%	$\mathcal{H} : 11 \mathcal{W} : 32$
		0.90%	$\mathcal{H} : 12 \mathcal{W} : 32$
		0.93%	$\mathcal{H} : 10 \mathcal{W} : 32$
		0.93%	$\mathcal{H} : 8 \mathcal{W} : 32$
		0.94%	$\mathcal{H} : 9 \mathcal{W} : 32$
Combined	QSW	0.96%	$\mathcal{H} : 9 \mathcal{W} : 32$
		0.96%	$\mathcal{H} : 10 \mathcal{W} : 32$
		0.97%	$\mathcal{H} : 11 \mathcal{W} : 32$
		0.99%	$\mathcal{H} : 12 \mathcal{W} : 32$
		0.99%	$\mathcal{H} : 8 \mathcal{W} : 32$

**Table 7.3:** The best EERs achieved during the verification experiment

It should also be noted, that the reported performance has only been reached after implementing additional measures for bit flip compensation, such as feature extractor response reordering and majority voting on the iris code bits before transformation to the Bloom filter representation.

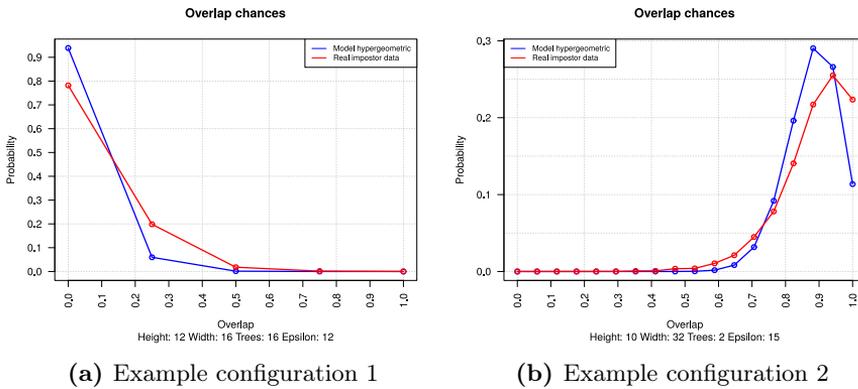
Due to the poor results of the **Thousand** dataset already in the verification mode, it makes little sense to use it in the identification mode, where the results can only deteriorate even further. *Therefore, only the results of the **Combined** dataset are shown and described in depth in the subsequent sections.* The results of the **Thousand** dataset, which substantiate the exclusion decision can be viewed in appendix B.2. The reasons for the inferior results on the **Thousand** dataset are discussed in section 8.1.

## 7.2.2 Model

Before proceeding to the identification mode results, let's revisit the model described in section 4.3. The main thought behind its development was the assessment of feasibility of a system configuration in the identification mode based on just a few parameters: the number of enrolled subjects ( $\mathcal{S}$ ), the Bloom filter block size ( $\mathcal{H}$  and  $\mathcal{W}$ ), the number of constructed trees ( $\mathcal{T}$ ) and the factor accounting for iris code entropy in relation to randomly generated data ( $\epsilon$ ). The purpose of this section is twofold: examining the applicability of the model to real data and presenting the use of the model for finding well-performing system configurations.

### 7.2.2.1 Validation

The first and foremost matter of interest for the model is, whether or not it is a reasonable representation of real iris data. Figure 7.3 shows model and real distributions for an example system configuration. The plot is identical to that in figure 4.5, except that now real iris data distribution is also plotted. It can be seen, that the fit between the model and the real distributions is good. As mentioned earlier, due to the multitude of possible system configurations, a quantitative metric is used to assess the model fit. This metric is the Hellinger Distance (see equation 4.12). Computing it for all the relevant configurations results in a mean of  $\mu = 0.16$  and a standard deviation of  $\sigma = 0.08$ . Figure 7.3c shows a histogram of the obtained  $\mathcal{H}_{\mathcal{D}}$  values. The model appears to fit the real data well. The likely source of discrepancies, aside from the assumptions that were made to simplify the model, is the rigid  $\epsilon$  value, which therefore fails to account for real data variability.



Fit between model and real data

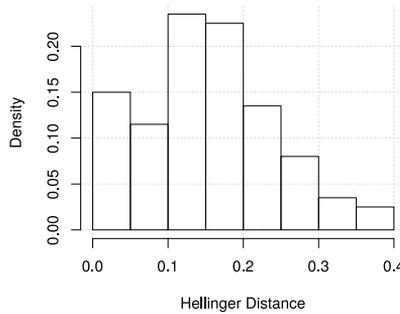


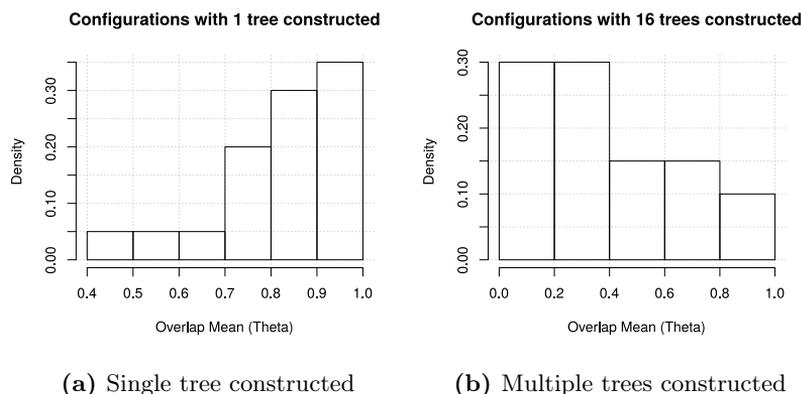
Figure 7.3: The fit between the model and real impostor data

7.2.2.2 Finding Good Configurations

The subsequent sections in this chapter will present empirical results of the Bloom filter system. However, there is a multitude of possible configurations the system can be run in - different block widths, heights and number of constructed trees. Instead of blindly and exhaustively plotting them all for the possible system set-ups, the model can be used to discover which configurations can be expected to achieve good performance. This can be done by feeding the variables  $(S, H, W, T, \epsilon)$  to the model and computing the mean of the expected overlap distribution between a tree root and a random (impostor) template (see equation 4.11). The hypothesis is, that configurations with lowest  $\Theta$  values will achieve

the highest biometric performance in terms of true positive identification rate due to lowest risk of false bit matches in the tree root and the top levels of the tree. The two items of interest are:

- Configurations with just a single tree constructed. These will serve as a basic result of the Bloom filter system in its original form and be used for comparison purposes with the results of the system with application of the proposed improvements.
- Configurations with multiple trees constructed. For the sake of brevity, let's consider all possible (20) configurations with 16 trees constructed.



**Figure 7.4:** The  $\Theta$  values for all relevant system configurations

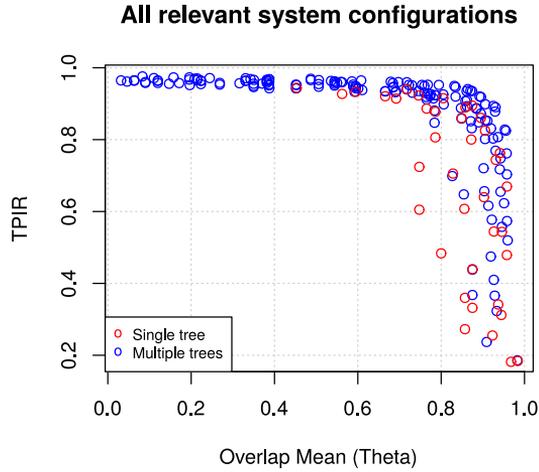
The histograms of computed  $\Theta$  values for all the relevant configurations (see section 4.2.4) are shown in figure 7.4. Observe, that  $\Theta$  is in general much higher for the configurations with single tree constructed in comparison with the configurations with multiple trees constructed. This is obvious - with more trees, there are fewer templates in each tree root, thus making it less likely for random bit matches to occur. Now, for both the single and multiple trees case, a few configurations with low and high  $\Theta$  values are chosen for the experiments in the following sections. The chosen configurations are listed in tables 7.4 and 7.5, respectively. Having established the configurations which are likely to perform well (and bad) in the identification mode, the empirical test results can now be looked into. In figure 7.5, the  $\Theta$  values of all relevant system configurations both in the single and multiple tree mode are plotted against the achieved TPIR at 0% FPIR. Notice the strong correlation - it appears, that upon crossing a certain threshold (around  $\Theta = 0.75$ ), the TPIR begins to drop dramatically due to the inability to make correct traversal direction decisions for genuine attempts.

Configuration	Overlap mean
$\mathcal{H} : 12 \mathcal{W} : 8 \mathcal{T} : 1$	$\Theta \approx 0.45$
$\mathcal{H} : 12 \mathcal{W} : 16 \mathcal{T} : 1$	$\Theta \approx 0.71$
$\mathcal{H} : 11 \mathcal{W} : 8 \mathcal{T} : 1$	$\Theta \approx 0.59$
$\mathcal{H} : 11 \mathcal{W} : 16 \mathcal{T} : 1$	$\Theta \approx 0.76$
$\mathcal{H} : 10 \mathcal{W} : 8 \mathcal{T} : 1$	$\Theta \approx 0.69$
$\mathcal{H} : 12 \mathcal{W} : 32 \mathcal{T} : 16$	$\Theta \approx 0.16$
$\mathcal{H} : 12 \mathcal{W} : 16 \mathcal{T} : 16$	$\Theta \approx 0.08$
$\mathcal{H} : 10 \mathcal{W} : 16 \mathcal{T} : 16$	$\Theta \approx 0.24$
$\mathcal{H} : 12 \mathcal{W} : 64 \mathcal{T} : 16$	$\Theta \approx 0.72$
$\mathcal{H} : 8 \mathcal{W} : 16 \mathcal{T} : 16$	$\Theta \approx 0.56$

**Table 7.4:** Several potentially feasible system configurations

Configuration	Overlap mean
$\mathcal{H} : 12 \mathcal{W} : 64 \mathcal{T} : 1$	$\Theta \approx 0.94$
$\mathcal{H} : 10 \mathcal{W} : 16 \mathcal{T} : 1$	$\Theta \approx 0.85$
$\mathcal{H} : 8 \mathcal{W} : 32 \mathcal{T} : 1$	$\Theta \approx 0.86$
$\mathcal{H} : 10 \mathcal{W} : 32 \mathcal{T} : 1$	$\Theta \approx 0.90$
$\mathcal{H} : 8 \mathcal{W} : 64 \mathcal{T} : 1$	$\Theta \approx 0.98$

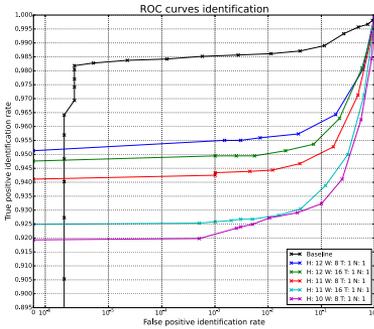
**Table 7.5:** Several potentially infeasible system configurations



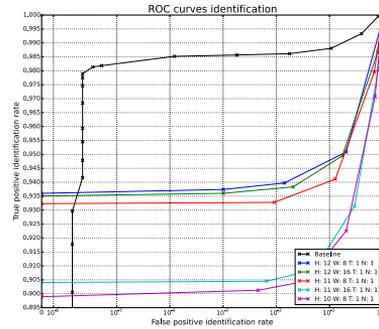
**Figure 7.5:** The correlation between  $\Theta$  and TPIR

### 7.2.3 Identification

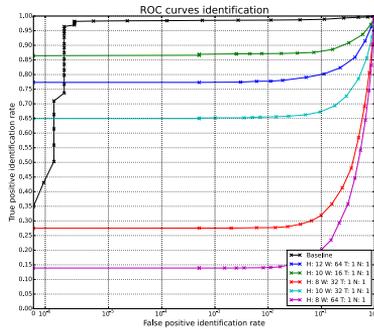
This section presents the results for the system where the probe templates are compared against *a single* Bloom filter tree (i.e. the basic scheme from [RBBB15]), without multiple trees and other improvements described in chapter 5). In figure 7.6, the results for the top 5 system configurations are shown.



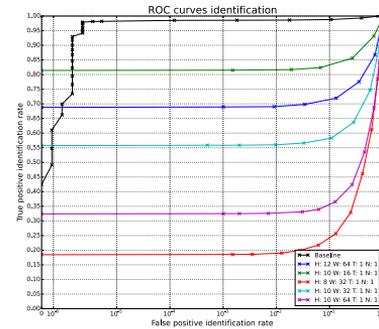
(a) Low  $\Theta$  configurations - LG



(b) Low  $\Theta$  configurations - QSW



(c) High  $\Theta$  configurations - LG

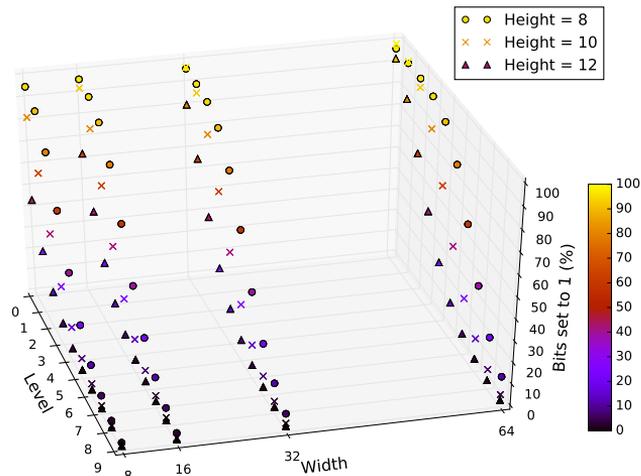


(d) High  $\Theta$  configurations - QSW

**Figure 7.6:** The best ROC curves for the basic Bloom filter scheme

It can be observed, that the achieved true positive identification rate is lower than in the baseline case. On the other hand, the false positive identification rate of flat 0% is achieved at very high true positive identification rate in the case of the Bloom filter system, whereas this is not the case in the baseline (the black line in this and subsequent plots is cut off - it actually goes way down as was shown in figure 7.2). Notice also, that the good biometric performance is achieved using the configurations with low  $\Theta$  values, while configurations with high  $\Theta$  values perform poorly - as suggested by the model.

Lastly, it appears that the best configurations from the verification experiment are not necessarily the best in the identification scenario. This is because in the identification scenario, the Bloom filters have to store more information due to the union of templates (see section 4.2.2). In the verification experiment, the best configurations had relatively small sizes of the Bloom filters. These become filled up in the identification scenario, as can be seen in figure 7.7. The tree traversal decision making is thus severely impaired. Consequently, the biometric performance, especially in terms of true positives, would decline in these cases. Consider, for example, the configuration  $\mathcal{H} = 8$  and  $\mathcal{W} = 32$ ; it was one of the best performing in the verification experiment, but in the identification experiment its performance is atrociously bad.



**Figure 7.7:** The amount of bits set to 1 (in %) at different tree levels for several relevant system configurations

### 7.2.4 Workload

The single lookup workload and biometric performance for the configurations presented in the previous section is presented in table 7.6. The number of enrolled subjects is 512, the template size is calculated using equation 4.1, the number of template comparisons (and from it, the penetration rate) is obtained using equations 5.1 and 5.2. Observe, that the workload has been reduced to a

small fraction of the baseline. Naturally, this came at the cost of a significantly reduced biometric performance.

Configuration	Workload		Performance	
	Worst	Average	LG	QSW
$\mathcal{H}$ : 12 $\mathcal{W}$ : 8 $\mathcal{T}$ : 1	$\omega \approx 8.39 * 10^6$ $F \approx 0.094$	$\omega \approx 8.39 * 10^6$ $F \approx 0.188$	$TP_0 = 95.17\%$ $TP_{0.1} = 95.50\%$	$TP_0 = 93.61\%$ $TP_{0.1} = 93.74\%$
$\mathcal{H}$ : 12 $\mathcal{W}$ : 16 $\mathcal{T}$ : 1	$\omega \approx 4.19 * 10^6$ $F \approx 0.047$	$\omega \approx 4.19 * 10^6$ $F \approx 0.094$	$TP_0 = 94.81\%$ $TP_{0.1} = 94.95\%$	$TP_0 = 93.51\%$ $TP_{0.1} = 93.60\%$
$\mathcal{H}$ : 11 $\mathcal{W}$ : 8 $\mathcal{T}$ : 1	$\omega \approx 4.19 * 10^6$ $F \approx 0.047$	$\omega \approx 4.19 * 10^6$ $F \approx 0.094$	$TP_0 = 94.11\%$ $TP_{0.1} = 94.34\%$	$TP_0 = 93.23\%$ $TP_{0.1} = 93.23\%$
$\mathcal{H}$ : 11 $\mathcal{W}$ : 16 $\mathcal{T}$ : 1	$\omega \approx 2.10 * 10^6$ $F \approx 0.024$	$\omega \approx 2.10 * 10^6$ $F \approx 0.047$	$TP_0 = 92.44\%$ $TP_{0.1} = 92.58\%$	$TP_0 = 90.40\%$ $TP_{0.1} = 90.40\%$
$\mathcal{H}$ : 10 $\mathcal{W}$ : 8 $\mathcal{T}$ : 1	$\omega \approx 2.10 * 10^6$ $F \approx 0.024$	$\omega \approx 2.10 * 10^6$ $F \approx 0.047$	$TP_0 = 91.98\%$ $TP_{0.1} = 92.30\%$	$TP_0 = 89.89\%$ $TP_{0.1} = 90.07\%$

**Table 7.6:** The results of the Bloom filter scheme with a single tree constructed

## 7.3 Improvements

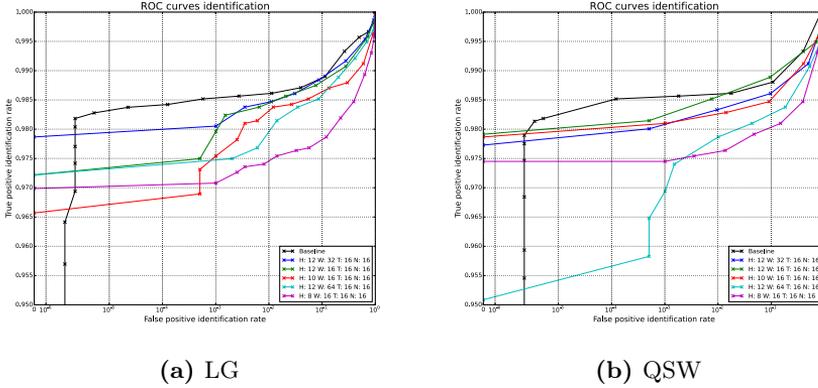
This section presents the results of the various system improvements presented in chapter 5.

### 7.3.1 Multiple Trees

This is the first and the most important improvement of the Bloom filter approach. First matter to investigate is its effect on the biometric performance. Figure 7.8 shows the biometric performance for the best performing system configurations. It is clear, that spreading the templates out positively affects the biometric performance in comparison with the basic Bloom filter scheme. Now, the biometric is only minimally lower than that of the baseline. The reason for this is obvious - with more trees constructed, each one holds fewer templates, thus reducing the probability of false bit matches in the top levels of the tree. The results validate the predictions of the model - all of the well-performing configurations shown in the figure below have low or very low  $\Theta$  values.

#### 7.3.1.1 Workload

The single lookup workload is computed in the same way as in section 7.2.4. As can be seen in table 7.7, the workload has now increased compared to that of the



**Figure 7.8:** The best ROC curves for the Bloom filter scheme with multiple trees constructed

basic scheme. This was expected, since now multiple trees have to be traversed instead of just one; thus resulting in a higher penetration rate. Notice, however, that the workload is still significantly lower than that of the baseline - only now, the biometric performance is nearly identical to that of the baseline.

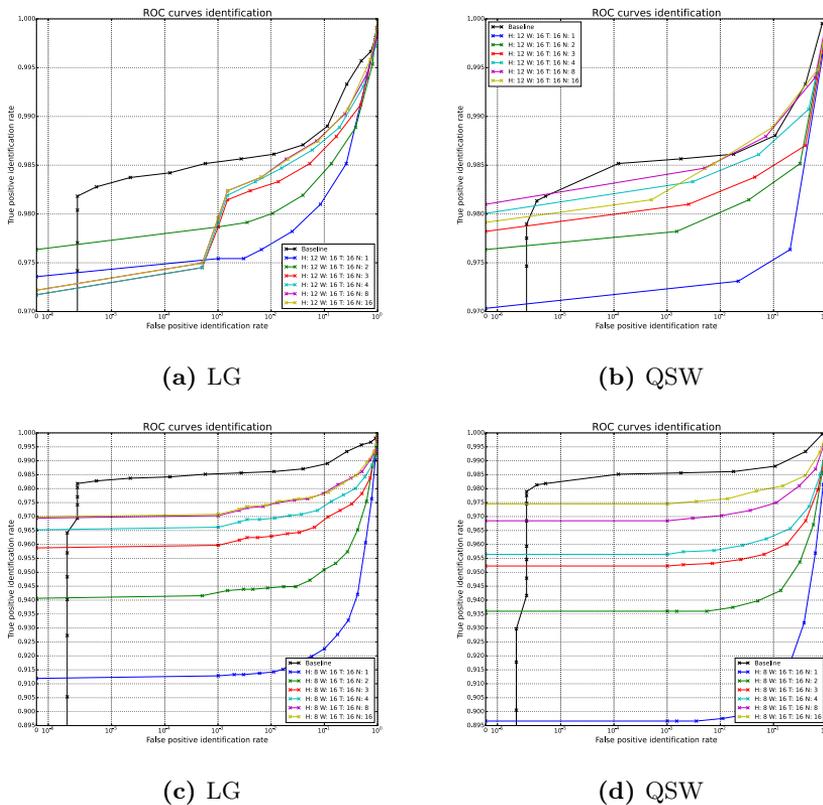
Configuration	Workload		Performance	
	Worst	Average	LG	QSW
$\mathcal{H}: 12 \mathcal{W}: 32 \mathcal{T}: 16$	$\omega \approx 1.68 * 10^7$ $F \approx 0.188$	$\omega \approx 1.05 * 10^7$ $F \approx 0.235$	$TP_0 = 97.85\%$ $TP_{0.1} = 98.09\%$	$TP_0 = 97.75\%$ $TP_{0.1} = 97.98\%$
$\mathcal{H}: 12 \mathcal{W}: 16 \mathcal{T}: 16$	$\omega \approx 3.36 * 10^7$ $F \approx 0.377$	$\omega \approx 2.10 * 10^7$ $F \approx 0.470$	$TP_0 = 97.25\%$ $TP_{0.1} = 97.92\%$	$TP_0 = 97.94\%$ $TP_{0.1} = 98.33\%$
$\mathcal{H}: 10 \mathcal{W}: 16 \mathcal{T}: 16$	$\omega \approx 8.39 * 10^6$ $F \approx 0.094$	$\omega \approx 5.24 * 10^6$ $F \approx 0.118$	$TP_0 = 96.56\%$ $TP_{0.1} = 97.52\%$	$TP_0 = 97.86\%$ $TP_{0.1} = 98.05\%$
$\mathcal{H}: 12 \mathcal{W}: 64 \mathcal{T}: 16$	$\omega \approx 8.39 * 10^6$ $F \approx 0.094$	$\omega \approx 5.24 * 10^6$ $F \approx 0.118$	$TP_0 = 97.25\%$ $TP_{0.1} = 97.49\%$	$TP_0 = 95.15\%$ $TP_{0.1} = 96.87\%$
$\mathcal{H}: 8 \mathcal{W}: 16 \mathcal{T}: 16$	$\omega \approx 2.10 * 10^6$ $F \approx 0.024$	$\omega \approx 1.31 * 10^6$ $F \approx 0.029$	$TP_0 = 97.01\%$ $TP_{0.1} = 97.15\%$	$TP_0 = 97.45\%$ $TP_{0.1} = 97.45\%$

**Table 7.7:** The results of the Bloom filter scheme with many trees constructed

### 7.3.2 Selective Tree Traversal

While reduced in comparison with the baseline, the single lookup workload is still quite high in the scheme in which all constructed trees are traversed. To mitigate this issue, it was proposed to selectively traverse a few most promising trees. Figure 7.9 shows the results of this approach for two of the configurations from the earlier example (figure 7.8); the other three are available in appendix

**B.3.** Observe, that there is a trade-off associated with the workload reduction. Fewer traversed trees result in a decreased true positive identification rate. Note, that this decrease is not excessive - even when just the single most promising tree is traversed, the biometric performance is acceptable. Upon selecting merely a few additional promising trees for traversal, the performance converges with that of traversing all the constructed trees.



**Figure 7.9:** The ROC curves for two configurations in the selective tree traversal experiment

### 7.3.2.1 Workload

The single lookup workload is calculated in almost the same way as before, only now equation 5.3 is used to compute the number of template comparisons (and from it, the penetration rate). Table 7.8 shows an overview of the workload

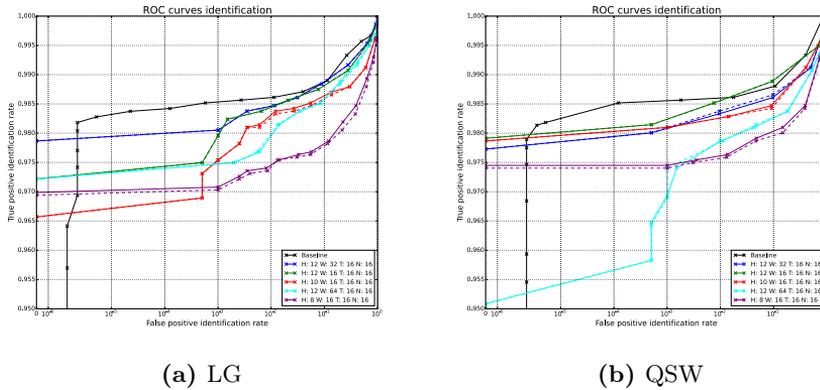
and biometric performance results for the configurations presented in figure 7.9, while the remaining results can be found in appendix B.3.

Configuration	Workload		Performance	
	Worst	Average	LG	QSW
$\mathcal{H}$ : 8 $\mathcal{W}$ : 16 $\mathcal{T}$ : 16 $\mathcal{N}$ : 8	$\omega \approx 1.31 * 10^6$ $F \approx 0.015$	$\omega \approx 7.86 * 10^5$ $F \approx 0.018$	$TP_0 = 96.95\%$ $TP_{0.1} = 97.02\%$	$TP_0 = 96.78\%$ $TP_{0.1} = 96.78\%$
$\mathcal{H}$ : 8 $\mathcal{W}$ : 16 $\mathcal{T}$ : 16 $\mathcal{N}$ : 4	$\omega \approx 7.86 * 10^5$ $F \approx 0.009$	$\omega \approx 5.24 * 10^5$ $F \approx 0.012$	$TP_0 = 96.50\%$ $TP_{0.1} = 96.59\%$	$TP_0 = 95.61\%$ $TP_{0.1} = 95.61\%$
$\mathcal{H}$ : 8 $\mathcal{W}$ : 16 $\mathcal{T}$ : 16 $\mathcal{N}$ : 2	$\omega \approx 5.24 * 10^5$ $F \approx 0.006$	$\omega \approx 3.93 * 10^5$ $F \approx 0.009$	$TP_0 = 94.15\%$ $TP_{0.1} = 94.22\%$	$TP_0 = 93.61\%$ $TP_{0.1} = 93.61\%$
$\mathcal{H}$ : 8 $\mathcal{W}$ : 16 $\mathcal{T}$ : 16 $\mathcal{N}$ : 1	$\omega \approx 3.93 * 10^5$ $F \approx 0.004$	$\omega \approx 3.93 * 10^5$ $F \approx 0.009$	$TP_0 = 91.19\%$ $TP_{0.1} = 91.17\%$	$TP_0 = 89.60\%$ $TP_{0.1} = 89.60\%$
$\mathcal{H}$ : 12 $\mathcal{W}$ : 16 $\mathcal{T}$ : 16 $\mathcal{N}$ : 8	$\omega \approx 2.10 * 10^7$ $F \approx 0.235$	$\omega \approx 1.26 * 10^7$ $F \approx 0.282$	$TP_0 = 97.23\%$ $TP_{0.1} = 97.89\%$	$TP_0 = 98.15\%$ $TP_{0.1} = 98.43\%$
$\mathcal{H}$ : 12 $\mathcal{W}$ : 16 $\mathcal{T}$ : 16 $\mathcal{N}$ : 4	$\omega \approx 1.26 * 10^7$ $F \approx 0.141$	$\omega \approx 8.39 * 10^6$ $F \approx 0.188$	$TP_0 = 97.19\%$ $TP_{0.1} = 97.88\%$	$TP_0 = 98.00\%$ $TP_{0.1} = 98.24\%$
$\mathcal{H}$ : 12 $\mathcal{W}$ : 16 $\mathcal{T}$ : 16 $\mathcal{N}$ : 2	$\omega \approx 8.39 * 10^6$ $F \approx 0.094$	$\omega \approx 6.29 * 10^6$ $F \approx 0.141$	$TP_0 = 97.66\%$ $TP_{0.1} = 97.79\%$	$TP_0 = 97.68\%$ $TP_{0.1} = 97.76\%$
$\mathcal{H}$ : 12 $\mathcal{W}$ : 16 $\mathcal{T}$ : 16 $\mathcal{N}$ : 1	$\omega \approx 6.29 * 10^6$ $F \approx 0.071$	$\omega \approx 6.29 * 10^6$ $F \approx 0.141$	$TP_0 = 97.38\%$ $TP_{0.1} = 97.53\%$	$TP_0 = 97.09\%$ $TP_{0.1} = 97.21\%$

**Table 7.8:** The results of the Bloom filter scheme with selective tree traversal

### 7.3.3 Quick Traversal Direction Decision

In section 5.2 a scheme for further reduction of necessary template comparisons in a tree traversal has been proposed. By relying on the nature of the score sequence obtained through a tree traversal, on average half of the traversal direction decisions can be taken quickly. This is done by only computing the dissimilarity score for one of the child nodes instead of them both. The first matter to address is whether or not this scheme has an effect on the biometric performance. Figure 7.10 shows the biometric performance of the system with and without application of the quick traversal direction decision scheme. The biometric performance with the proposed scheme applied is virtually indistinguishable from that without the application of the proposed scheme. In other words, the benefits of workload reduction in this case does not have a negative impact on the biometric performance of the system. The workload reduction is non-deterministic; in the experiments, it has been consistently reducing the number of necessary template comparisons by around 10%-15% from that of the Bloom filter scheme with multiple constructed trees.



**Figure 7.10:** Comparison of the ROC curves for the normal (solid lines) and quick (dashed lines) traversal decision experiments

### 7.3.4 Score Sequence Ordering

This improvement was proposed in the original paper presenting the Bloom filter based scheme. The idea is to immediately reject attempts, which do not have a decreasing score sequence during tree traversal (see algorithm 4.1, lines 15 to 19). In all the previous sections, this setting was not activated, as plotting ROC curves would not be possible, since in many configurations, most impostors are actually rejected based on the sequence ordering, rather than their final score. The assessment of this change will therefore be assessed based on a single TPIR/FPIR pair, where the final decision threshold is obtained from a training set. In terms of workload, the average result will be reported, since the effects of the change are non-deterministic and can only be measured in an empirical experiment with many matching attempts. Observe, that this change will only have significant impact on the workload for the impostor attempts, since that is where the score sequence can be expected to fluctuate. The score sequence in the vast majority of genuine attempts exhibits the decreasing ordering. Table 7.9 shows the results for several configurations from previous sections. It can be seen, that with the sequence ordering check applied, the biometric performance in terms of true positive identification rate is slightly reduced, albeit still very high. The workload is decreased as well, although by not very much. This change can, however, prove useful in high security scenarios, since it makes it more difficult for the impostors to be accepted.

Configuration	Performance	Workload
$\mathcal{H}: 8 \mathcal{W}: 16 \mathcal{T}: 16 \mathcal{N}: 8$	TPIR: 95.9% FPIR: 0.35%	$\omega \approx 1.15 * 10^6$ $F \approx 0.013$
$\mathcal{H}: 8 \mathcal{W}: 16 \mathcal{T}: 16 \mathcal{N}: 4$	TPIR: 95.5% FPIR: 0.30%	$\omega \approx 7.22 * 10^5$ $F \approx 0.008$
$\mathcal{H}: 8 \mathcal{W}: 16 \mathcal{T}: 16 \mathcal{N}: 2$	TPIR: 93.0% FPIR: 0.25%	$\omega \approx 5.07 * 10^5$ $F \approx 0.005$
$\mathcal{H}: 12 \mathcal{W}: 16 \mathcal{T}: 16 \mathcal{N}: 8$	TPIR: 97.0% FPIR: 0.15%	$\omega \approx 2.02 * 10^7$ $F \approx 0.226$
$\mathcal{H}: 12 \mathcal{W}: 16 \mathcal{T}: 16 \mathcal{N}: 4$	TPIR: 97.0% FPIR: 0.15%	$\omega \approx 1.23 * 10^7$ $F \approx 0.139$
$\mathcal{H}: 12 \mathcal{W}: 16 \mathcal{T}: 16 \mathcal{N}: 2$	TPIR: 96.7% FPIR: 0.05%	$\omega \approx 8.22 * 10^6$ $F \approx 0.092$

**Table 7.9:** The results of the system in several configurations with the sequence ordering scheme active

### 7.3.5 Duplicate Enrolment Check

The proposed schemes for duplicate enrolment check scenarios allow to accept matching attempts before a tree leaf is reached.

#### 7.3.5.1 Quick Threshold Acceptance

When traversing a tree, a situation can occur, where the node score is below the final acceptance threshold before reaching a leaf. In such cases, one can, with a high level of confidence, accept such a matching attempt, since the score is expected to be even lower at subsequent tree levels. From this, it follows that the biometric performance of this scheme can never be lower than that of a full tree traversal. It can, naturally, only be used in duplicate enrolment check scenarios and not in the identification mode. This is because no actual identity is returned (since the traversal is abandoned before reaching a leaf) - the reply only consists of confirming that the matching attempt was successful (i.e. the subject's template is present in the tree). The workload reduction is non-deterministic; the experimental results can be seen in table 7.10. Note, that the workload figure represents an average for a genuine attempt only - the scheme does not affect the workload for the impostor attempts.

Configuration	Performance	Workload
$\mathcal{H}: 8 \mathcal{W}: 16 \mathcal{T}: 16 \mathcal{N}: 8$	TPIR: 97.1% FPIR: 0.10%	$\omega \approx 3.87 * 10^5$ $F \approx 0.0087$
$\mathcal{H}: 8 \mathcal{W}: 16 \mathcal{T}: 16 \mathcal{N}: 4$	TPIR: 96.7% FPIR: 0.10%	$\omega \approx 3.69 * 10^5$ $F \approx 0.0083$
$\mathcal{H}: 8 \mathcal{W}: 16 \mathcal{T}: 16 \mathcal{N}: 2$	TPIR: 94.2% FPIR: 0.05%	$\omega \approx 3.55 * 10^5$ $F \approx 0.0079$
$\mathcal{H}: 12 \mathcal{W}: 16 \mathcal{T}: 16 \mathcal{N}: 8$	TPIR: 98.3% FPIR: 0.25%	$\omega \approx 5.60 * 10^6$ $F \approx 0.125$
$\mathcal{H}: 12 \mathcal{W}: 16 \mathcal{T}: 16 \mathcal{N}: 4$	TPIR: 98.2% FPIR: 0.20%	$\omega \approx 5.46 * 10^6$ $F \approx 0.122$
$\mathcal{H}: 12 \mathcal{W}: 16 \mathcal{T}: 16 \mathcal{N}: 2$	TPIR: 97.9% FPIR: 0.15%	$\omega \approx 5.38 * 10^6$ $F \approx 0.120$

**Table 7.10:** The results of the system in several configurations with the quick threshold acceptance scheme active

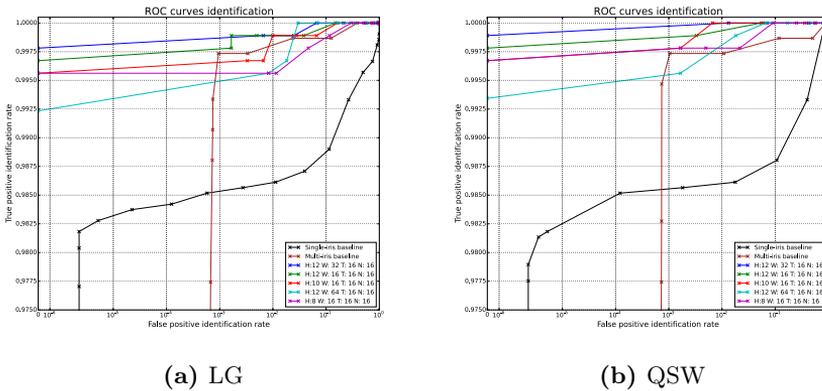
### 7.3.5.2 Score Sequence Difference Acceptance

The genuine attempts often exhibit large decreases in the score sequence from one tree level to the next. This is very rarely the case with the impostor attempts. One can therefore deem an attempt as successful, if a large enough score decrease takes place (the threshold for this can be determined from a training set). From this specification, it follows that the true positive identification rate will never be lower than that of a standard identification mode, since the only difference is that the genuine attempts can happen to be accepted a bit earlier. On the other hand, the false positive identification rate might increase - in the rare cases where the impostor attempts exhibits the drastically decreasing score sequence, they would be deemed as genuine attempts.

The lack of guarantees about the biometric performance makes this scheme inferior in comparison to the one described in the previous section. The experimental results showed no greater workload reduction than that of the scheme from the previous section. While the false positive risk can be minimised by fine-tuning the difference threshold, the risk can never be fully, provably eliminated. Therefore, the quick acceptance scheme should be based on the final decision threshold, rather than the score sequence difference. Finally, using both approaches simultaneously will only confer negligible extra workload reduction compared to when a single one is used - their effects cancel each other out (when a score sequence decreases rapidly, it will very likely also simultaneously drop beneath the final acceptance threshold).

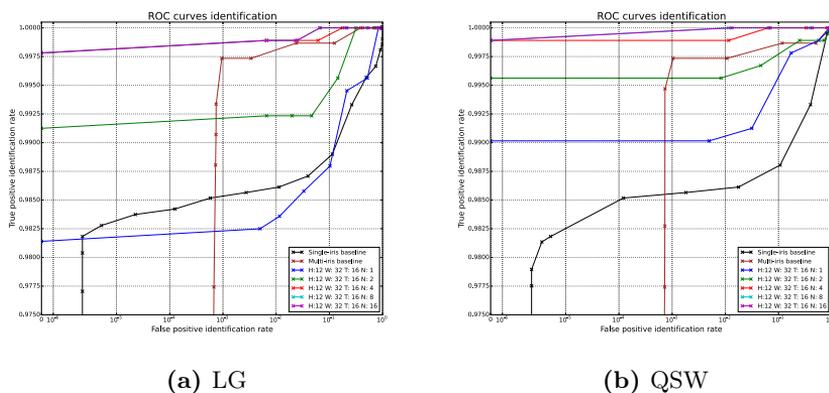
### 7.3.6 Multi-Iris Approach

Figure 7.11 shows the results achieved for the multi-iris scheme described in the section 5.4. The biometric performance is superior to that of a single-iris system. In particular, the true positive identification rate has been increased substantially. The overall biometric performance is at a near-optimal level; it is even significantly better than the single-iris baseline and at similar level to the multi-iris baseline. Although, it should be noted that, due to the template fusion, the number of enrolled templates and the test sample size are smaller than in the previous experiments (see table 6.2). In the case of the Bloom filter system, the multi-iris improvement comes at a very small additional computational cost during lookup (the template fusion). In the baseline case, however, the workload is significantly increased due to much higher rotation compensation costs. Observe, that while the single-iris baseline considers  $r$  possible rotations of one template, in the multi-iris baseline  $r^2$  rotation combinations of two templates have to be considered. Additionally, due to much higher number of template comparisons required, the baseline is much less resilient to false matches than the Bloom filter based system.



**Figure 7.11:** The ROC curves for the multi-iris identification experiment

Finally, all the other improvements proposed earlier (e.g. selective traversal, quick traversal direction decision etc.) can be seamlessly applied to the multi-iris scheme as well. Figure 7.12 shows, that even with only one most promising tree traversed, the system is capable of achieving an excellent biometric performance.



**Figure 7.12:** The ROC curves for an example configuration in the multi-iris selective traversal experiment

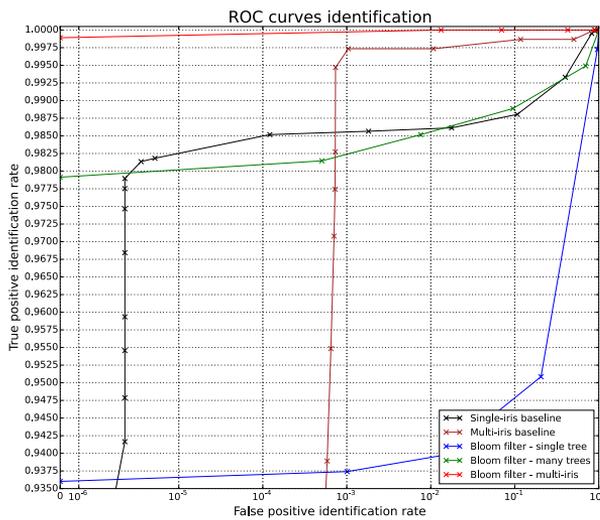
## 7.4 Summary

This chapter has presented the experimental results of the project. It has been demonstrated, that the Bloom filter based approach and the proposed improvements constitute a viable biometric system. Furthermore, the model has been shown to reflect the real iris data well and has been subsequently demonstrated to be capable of accurately determining the feasible system configurations for the identification mode. In the empirical experiments, the workload has been reduced to a mere fraction of that of the baseline, with only a small decrease in the biometric performance. The approach also shows great promise for research into multibiometrics. These successful results are, however, confined only to the **Combined** dataset. The **Thousand** dataset performed poorly, especially in the identification mode; this was presumably due to high intra-class variation in that dataset. Table 7.11 shows a summary of the best results achieved by different systems and improvements. Figure 7.13 shows the ROC curves for the best performing configuration in each (major) system version.

In the next chapter, the results and future outlooks of the Bloom filter based approach will be discussed in more depth.

System/Improvement	Biometric performance	Workload
Baseline	$TP_0 = 42.05\%$ $TP_{0.1} = 98.56\%$	$F = 1.0$
Single tree	$TP_0 = 92.0 - 95.0\%$ $TP_{0.1} = 92.5 - 95.5\%$	$F = 0.024 - 0.094$
Multiple trees	$TP_0 = 96.5 - 98.0\%$ $TP_{0.1} = 97.0 - 98.3\%$	$F = 0.024 - 0.377$
Selective traversal	$TP_0 = 96.5 - 98.1\%$ $TP_{0.1} = 94.0 - 98.4\%$	$F = 0.006 - 0.235$
Multi-iris	$TP_0 = 99.3 - 99.9\%$ $TP_{0.1} = 99.5 - 100\%$	Same as selective traversal
Quick traversal decision	Same as selective traversal	$F = 0.0053 - 0.197$
Quick threshold acceptance	Same as selective traversal	$F = 0.0079 - 0.125$
Score sequence ordering	$TPIR = 93.0 - 97.0\%$ $FPIR = 0.05 - 0.35\%$	$F = 0.005 - 0.226$

**Table 7.11:** A summary of the results for the Combined dataset achieved by the systems and improvements presented in this chapter



**Figure 7.13:** The ROC curves for the best configuration of each of the major system versions



# Discussion

---

In this chapter, the obtained results are discussed. Additionally, several future research possibilities for the Bloom filter based biometric indexing are briefly described.

## 8.1 Results

In this section two key issues are discussed: the poor results achieved on the **Thousand** dataset and the scalability assessment of the Bloom filter based indexing scheme.

### 8.1.1 Bad Performance on the Thousand Dataset

The poor biometric performance of the **Thousand** dataset has been partially explained by the Hamming Distance distributions in figure 7.1. There, it can be seen, that the genuine distribution is shifted to the right, meaning more frequent bit flips. This is a huge issue for the Bloom filter system, where the entire columns of many bits have to be a complete match between two templates. The overlap between the impostor and genuine distribution is also increased due

to said shift. This overlap essentially constitutes the intrinsic decidability of the decision problem faced by the biometric system. As the overlap increases, distinguishing between the impostors and genuines becomes harder. One of the existing metrics for quantitatively measuring this decidability is  $d'$  (used e.g. in [Dau00]). It is computed based on the means ( $\mu$ ) and standard deviations ( $\sigma$ ) of the two distributions, as shown in equation 8.1.

$$d' = \frac{|\mu_1 - \mu_2|}{\sqrt{\frac{1}{2} * (\sigma_1^2 + \sigma_2^2)}} \quad (8.1)$$

Tables 8.1 and 8.2 show the decidability metric computed for the score distributions of the two datasets used in this project. In the identification mode, the given values are for the best system configuration.

Dataset	Extractor	Decidability	
		Baseline	Bloom filter
Combined	LG	$d' \approx 4.74$	$d' \approx 3.34$
	QSW	$d' \approx 4.59$	$d' \approx 3.39$
Thousand	LG	$d' \approx 3.08$	$d' \approx 2.30$
	QSW	$d' \approx 2.98$	$d' \approx 2.06$

**Table 8.1:** The decidability of the score distributions obtained from the experiments in the verification mode

Dataset	Extractor	Decidability	
		Baseline	Bloom filter
Combined	LG	$d' \approx 5.76$	$d' \approx 3.52$
	QSW	$d' \approx 5.24$	$d' \approx 3.36$
Thousand	LG	$d' \approx 3.17$	$d' \approx 1.31$
	QSW	$d' \approx 3.10$	$d' \approx 1.19$

**Table 8.2:** The decidability of the score distributions obtained from the experiments in the identification mode

Unfortunately, there is no reference scale for the metric - the values can range from 0 to infinity, where higher values signify better decidability. The obtained values can be compared to others from the scientific literature. For instance, [Dau04b] reports  $d' = 14.1$  in an ideal case, and  $d' = 7.3$  in a very good, albeit non-ideal case. Elsewhere in the literature, values between  $d' \approx 3$  and  $d' \approx 7$  are generally associated with well-performing systems. The obtained values for the **Thousand** dataset are very low, which translates to a poor biometric performance, as has been seen in the results chapter. Of interest is also the

direct comparison of the obtained values. It can be observed, that they are consistently higher for the **Combined** dataset.

The underlying reason for the issues faced when using the **Thousand** dataset may be generalised to high intra-class variation. The images from that dataset have many reflections and occlusions; additionally, some of them are taken with the subjects wearing eyeglasses. Based on the decidability discourse above and the empirical results, it appears justified to claim, that the Bloom filter based scheme is suitable only for data with relatively low intra-class variation. This would mean, that the system would be viable for real-world application scenarios, where the data acquisition process can be undertaken in a constrained environment, thus consistently yielding biometric samples of good quality. Bearing in mind, that the data acquisition in most real applications is performed under controlled conditions and with supervision, this limitation of the system does not appear to be as serious as it may seem.

### 8.1.2 Scalability

In order to accommodate a larger number of enrolled templates, the basic approach was expanded with construction of multiple search trees and intelligent selection of the trees to traverse upon lookup.

The empirical experiments on the **Combined** dataset (512 enrollees) show, that the system is capable of achieving biometric performance comparable with the naïve baseline implementation. This performance has been achieved by having to traverse only around 25% of the constructed search trees. It can be reasonably assumed, that this property will hold when more subjects are enrolled (and more trees built), thus maintaining the low workload requirements for larger systems.

The overall workload in the identification mode is vastly reduced - in some configurations to around  $F = 0.01$  of the baseline. Even further workload reductions are possible, although at the cost of a decreased biometric performance in terms of true positive identification rate. For the duplicate enrolment check scenarios, the system constraints are slightly looser (yes/no query instead of returning a list of identities), which allows to take advantage of certain behaviours of the data and thus reduce the overall workload even more.

The vast array of possible system configurations and proposed improvements which can be activated and tuned at will, allow for a great flexibility in adjusting the system's biometric performance and workload to the individual needs of a given application. This is beneficial in terms of feasibility for real-world deployments, since their varying requirements could be accommodated by fine-tuning

the system's parameters. In addition to the empirical results, the developed model can be used to instantly assess whether or not a given system configuration is likely to perform well. This is immensely useful, since no empirical tests are required.

Based on the above discourse, it can be confidently asserted that the Bloom filter based indexing scheme is scalable both in terms of the biometric performance and the workload. Furthermore, the system is on par with, or exceeds the current state of the art (see table 3.1).

## 8.2 Future Research

In this section several proposals for future work with the Bloom filter based scheme are presented.

### 8.2.1 Model

The model operates on several variables, one of which is  $\epsilon$ , which accounts for the entropy loss between the iris code and randomly generated data. Currently, this value has to be estimated from a real data training set. It is conceivable, however, that with a thorough statistical analysis of the iris code, the empirical training set estimation could be dispensed with in favour of some theoretical method. Additionally, instead of using rigid  $\epsilon$  values, the model could be expanded to account for the variance in the real data. Lastly, the current version of the model operates based on a few simplifying assumptions (e.g. all Bloom filter index values being equally likely). Again, the model could be expanded to account for these variations.

### 8.2.2 Tree Construction

Currently, the Bloom filter templates are not assigned to trees in a specific manner - they are merely added in the order of their appearance in the dataset. It may be possible to assign and position them within the trees more intelligently. The aim would be to minimise the number of bit collisions in the Bloom filters.

The optimal, exhaustive solutions are unfortunately likely to have a very high computational complexity. It may also be possible to develop an evolutionary

approach. It can be argued, that since the tree construction is an offline cost, the constraints could be loosened. Thus, the slower solutions could still be viable, as long as insertion of new templates to an already constructed forest can be handled efficiently.

The potential implications of this implementation would be an improved biometric performance and lower lookup workload.

### 8.2.3 Multibiometrics

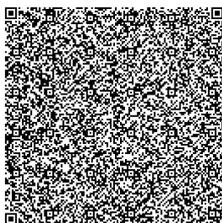
In this project, a groundwork in this area has been carried out. As a first study of its kind for biometric indexing, it has been demonstrated, that the Bloom filter based scheme is well suited for implementation of a multibiometric system based on a fusion of the left and right irides of a subject. The biometric performance of this approach has been shown to exceed that of the single-iris system - both baseline and Bloom filter based.

It would therefore be interesting to investigate how other biometric characteristics (e.g. face or fingerprint) could be integrated into this approach. The challenge here would be a different biometric template representation for these characteristics (e.g. minutiae points for fingerprints) - one would have to find a way to map them to a Bloom filter representation.

Such a multibiometric system would be a very good fit for the high-security scenarios, where it is crucial to achieve an extremely low false positive rate, while maintaining a relatively high usability of the system in terms of true positive rate and a low lookup time.

### 8.2.4 Compact Storage

The Bloom filter based biometric templates are largely rotation invariant and substantially reduce the size of the original iris code representation. One could investigate how far the biometric data can be compressed, whilst still remaining usable in authentication contexts. One interesting utilisation of compact templates could be embedding them in a QR code [Den] and conducting a market study into potential real-world applications of this technology. Already now, the Bloom filter based templates can be stored in a QR code (albeit a large one). Figure 8.1 shows a QR code generated from one such template.



**Figure 8.1:** A QR code generated from a Bloom filter based iris template

### 8.2.5 Data Protection

Privacy and protection of sensitive data (e.g. biometrics) has been a topic of interest lately. The Bloom filter representation of the iris template is *seemingly* irreversible. Some column location information is lost upon putting the data into a Bloom filter set - one can only tell which block an iris code column belongs to; furthermore, duplicate filter index entries are lost. However, it may not be impossible to reconstruct an iris code from a Bloom filter template. Therefore, a study into this matter would be desirable. It would be interesting to quantitatively show whether or not such a reconstruction is a feasible undertaking and if so, what can be done to prevent it. A successfully conducted study could result in a biometric system which offers high biometric performance, requires low workload and offers provable privacy and template protection benefits.

## 8.3 Summary

In this chapter, the results of the project have been discussed in detail. The reason for poor Bloom filter system performance on the **Thousand** dataset was examined and the scalability of the Bloom filter based approach was argued for.

The Bloom filter based scheme offers a multitude of future research possibilities. Several of these have been briefly outlined in this chapter. Examples include additional improvement of the system in terms of biometric performance and workload, developments in the areas of multibiometrics, biometric data compression and biometric data protection.

# Conclusions

---

In recent years, the public, commercial and governmental interest in biometrics has been steadily growing. Several huge deployments of biometric systems around the world have appeared. Said systems face the tremendous challenge of delivering consistently high biometric performance at a low computational workload. These trends make biometrics, and particularly workload reduction, a decidedly relevant and attractive research area nowadays. This thesis has pertained to the topic of workload reduction for biometric identification in large-scale iris databases.

A related work survey has demonstrated large discrepancies in how the biometric workload reduction is reported. This lack of uniformity and transparency made it incredibly cumbersome to compare the methods proposed in the surveyed articles. In response to this, a methodology for biometric workload reporting has been formulated as a set of six key requirements. The result reporting of this thesis has adhered to said requirements. Should this methodology become widely adopted in its current form or at least serve as an inspiration for the ISO biometric reporting committee, it could ameliorate the incongruity of the scientific process in this area.

A recently proposed biometric indexing approach based on Bloom filters and binary search trees was used for practical work throughout this thesis. Said approach was until now only in a proof-of-concept stage - without a proper the-

oretical framework and many open questions regarding biometric performance, workload reduction and scalability. All of these have been addressed during the course of this thesis. After implementing the system along with several key improvements, it has been demonstrated, that the system is capable of achieving a biometric performance comparable to that of a naïve implementation baseline, while reducing the necessary workload to around 1%. With the proposed improvements in place, the system appears to be scalable in terms of biometric performance and workload for any number of enrollees. Additionally, the system is very flexible due to a number of parameters that can be changed, thus adjusting the trade-off between the necessary workload and the biometric performance. To help cope with this configurational complexity, a statistical model has been developed; it instantly assesses the feasibility of a given configuration based only on the selected parameters. Lastly, in a, best to the author's knowledge, first experiment of its kind in the scientific literature, a multi-iris indexing scheme has been presented. The experimental results show a near-optimal biometric performance, while still taking advantage of the proposed workload reduction schemes. These results are very promising in terms of a prospective study into multibiometric capabilities of the Bloom filter based approach.

The excellent results and flexibility of the Bloom filter based approach suggest its suitability for real-world system deployments and open numerous propitious avenues of future research.

## APPENDIX *A*

# List of Symbols

---

The specific symbol meanings defined for use in this thesis are listed below.

$S$  A number of enrolled subjects in a biometric system

$\rho$  A penetration rate in an identification scenario

$\tau$  A cost of a single step in an identification scenario lookup. In this thesis, this is equivalent to the biometric template size in bits

$\omega$  A total workload for a single lookup in an identification scenario

$F$  A proposed system's workload in relation to a baseline workload

$\mathbf{B}$  A Bloom filter based template

$\mathbf{B}_M$  A multi-iris Bloom filter based template

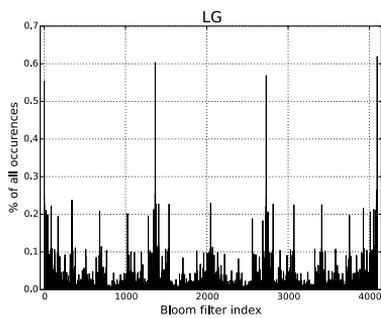
$\mathcal{B}$  An individual Bloom filter

$\mathcal{H}$  A Bloom filter block height

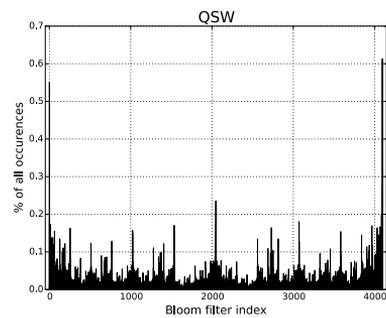
$\mathcal{W}$  A Bloom filter block width

$\mathcal{T}$  A number of constructed trees in a Bloom filter identification system

- $\mathcal{N}$  A number of trees chosen in the selective traversal scenario
- $\mathcal{R}_{\mathcal{K}}$  A tree root consisting of  $\mathcal{K}$  Bloom filters
- $\mathcal{O}$  An overlap between a tree root and a Bloom filter
- $P(\mathcal{O} = k)$  An expected overlap distribution
- $\Theta$  The mean of an expected overlap distribution
- $\epsilon$  A difference between expected number of duplicate values in an iris code and randomly generated, mutually independent values (uniform distribution)
- $\mathcal{C}$  A number of Bloom filter template comparisons required in a single identification lookup
- $\mathcal{D}(n, m)$  The expected (mean) number of duplicate items when drawing, with replacement,  $n$  values from a uniform distribution of  $m$  possible values
- $\mathcal{H}_{\mathcal{D}}$  The Hellinger distance
- $\mathcal{B}_{\mathcal{C}}$  The Bhattacharyya coefficient
- $TP_0$  A true positive identification rate at 0% false positive identification rate
- $TP_{0.1}$  A true positive identification rate at 0.1% false positive identification rate
- $d'$  A decidability metric proposed in [\[Dau04b\]](#)



(a) LG feature extraction



(b) QSW feature extraction

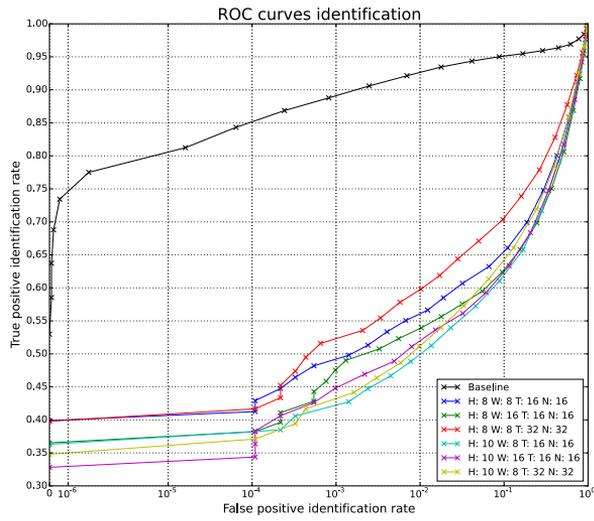
**Figure B.1:** The distribution of Bloom filter index occurrences for  $\mathcal{H} = 12$

LG		QSW	
EER	Configuration	EER	Configuration
0.89%	$\mathcal{H} : 11 \mathcal{W} : 32$	0.96%	$\mathcal{H} : 9 \mathcal{W} : 32$
0.90%	$\mathcal{H} : 12 \mathcal{W} : 32$	0.96%	$\mathcal{H} : 10 \mathcal{W} : 32$
0.93%	$\mathcal{H} : 10 \mathcal{W} : 32$	0.97%	$\mathcal{H} : 11 \mathcal{W} : 32$
0.93%	$\mathcal{H} : 8 \mathcal{W} : 32$	0.99%	$\mathcal{H} : 12 \mathcal{W} : 32$
0.94%	$\mathcal{H} : 9 \mathcal{W} : 32$	0.99%	$\mathcal{H} : 8 \mathcal{W} : 32$
0.98%	$\mathcal{H} : 11 \mathcal{W} : 64$	1.12%	$\mathcal{H} : 10 \mathcal{W} : 64$
1.03%	$\mathcal{H} : 10 \mathcal{W} : 64$	1.14%	$\mathcal{H} : 11 \mathcal{W} : 64$
1.04%	$\mathcal{H} : 12 \mathcal{W} : 64$	1.15%	$\mathcal{H} : 9 \mathcal{W} : 64$
1.07%	$\mathcal{H} : 9 \mathcal{W} : 64$	1.15%	$\mathcal{H} : 12 \mathcal{W} : 64$
1.18%	$\mathcal{H} : 8 \mathcal{W} : 64$	1.16%	$\mathcal{H} : 12 \mathcal{W} : 16$
1.27%	$\mathcal{H} : 9 \mathcal{W} : 16$	1.19%	$\mathcal{H} : 11 \mathcal{W} : 16$
1.13%	$\mathcal{H} : 11 \mathcal{W} : 16$	1.19%	$\mathcal{H} : 9 \mathcal{W} : 16$
1.16%	$\mathcal{H} : 12 \mathcal{W} : 16$	1.22%	$\mathcal{H} : 10 \mathcal{W} : 16$
1.20%	$\mathcal{H} : 10 \mathcal{W} : 16$	1.31%	$\mathcal{H} : 8 \mathcal{W} : 64$
1.30%	$\mathcal{H} : 8 \mathcal{W} : 16$	1.32%	$\mathcal{H} : 8 \mathcal{W} : 16$
2.57%	$\mathcal{H} : 11 \mathcal{W} : 8$	3.40%	$\mathcal{H} : 11 \mathcal{W} : 8$
2.60%	$\mathcal{H} : 12 \mathcal{W} : 8$	3.44%	$\mathcal{H} : 12 \mathcal{W} : 8$
2.77%	$\mathcal{H} : 10 \mathcal{W} : 8$	3.58%	$\mathcal{H} : 10 \mathcal{W} : 8$
2.85%	$\mathcal{H} : 9 \mathcal{W} : 8$	3.68%	$\mathcal{H} : 9 \mathcal{W} : 8$
3.03%	$\mathcal{H} : 8 \mathcal{W} : 8$	3.88%	$\mathcal{H} : 8 \mathcal{W} : 8$

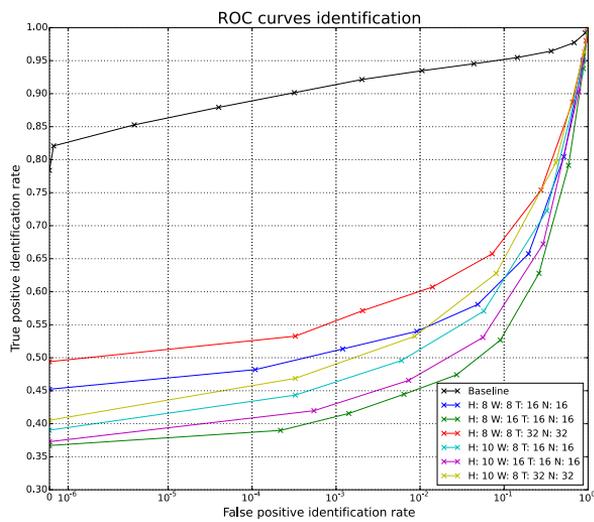
**Table B.1:** All the EERs for the verification experiment on the Combined dataset

LG		QSW	
EER	Configuration	EER	Configuration
9.13%	$\mathcal{H} : 5 \mathcal{W} : 16$	7.43%	$\mathcal{H} : 5 \mathcal{W} : 16$
9.80%	$\mathcal{H} : 4 \mathcal{W} : 16$	7.93%	$\mathcal{H} : 4 \mathcal{W} : 16$
10.05%	$\mathcal{H} : 4 \mathcal{W} : 8$	7.94%	$\mathcal{H} : 5 \mathcal{W} : 8$
10.06%	$\mathcal{H} : 5 \mathcal{W} : 8$	8.04%	$\mathcal{H} : 6 \mathcal{W} : 16$
10.81%	$\mathcal{H} : 6 \mathcal{W} : 16$	8.11%	$\mathcal{H} : 4 \mathcal{W} : 8$
10.93%	$\mathcal{H} : 7 \mathcal{W} : 16$	8.40%	$\mathcal{H} : 7 \mathcal{W} : 16$
11.97%	$\mathcal{H} : 7 \mathcal{W} : 8$	9.14%	$\mathcal{H} : 8 \mathcal{W} : 16$
12.13%	$\mathcal{H} : 5 \mathcal{W} : 32$	9.32%	$\mathcal{H} : 7 \mathcal{W} : 8$
12.55%	$\mathcal{H} : 8 \mathcal{W} : 16$	9.57%	$\mathcal{H} : 5 \mathcal{W} : 32$
13.36%	$\mathcal{H} : 7 \mathcal{W} : 32$	9.74%	$\mathcal{H} : 7 \mathcal{W} : 32$
13.68%	$\mathcal{H} : 8 \mathcal{W} : 8$	10.06%	$\mathcal{H} : 8 \mathcal{W} : 8$
14.24%	$\mathcal{H} : 6 \mathcal{W} : 32$	10.25%	$\mathcal{H} : 8 \mathcal{W} : 8$
14.40%	$\mathcal{H} : 9 \mathcal{W} : 16$	10.29%	$\mathcal{H} : 6 \mathcal{W} : 32$
14.63%	$\mathcal{H} : 8 \mathcal{W} : 32$	10.73%	$\mathcal{H} : 9 \mathcal{W} : 16$
15.02%	$\mathcal{H} : 4 \mathcal{W} : 32$	11.20%	$\mathcal{H} : 4 \mathcal{W} : 32$
15.92%	$\mathcal{H} : 9 \mathcal{W} : 8$	11.45%	$\mathcal{H} : 9 \mathcal{W} : 32$
16.08%	$\mathcal{H} : 9 \mathcal{W} : 32$	11.84%	$\mathcal{H} : 9 \mathcal{W} : 8$
17.29%	$\mathcal{H} : 7 \mathcal{W} : 64$	11.99%	$\mathcal{H} : 8 \mathcal{W} : 64$
17.40%	$\mathcal{H} : 10 \mathcal{W} : 16$	12.35%	$\mathcal{H} : 7 \mathcal{W} : 64$
17.92%	$\mathcal{H} : 8 \mathcal{W} : 64$	12.48%	$\mathcal{H} : 10 \mathcal{W} : 16$
18.34%	$\mathcal{H} : 5 \mathcal{W} : 64$	12.96%	$\mathcal{H} : 10 \mathcal{W} : 32$
18.63%	$\mathcal{H} : 10 \mathcal{W} : 32$	12.98%	$\mathcal{H} : 9 \mathcal{W} : 64$
18.77%	$\mathcal{H} : 9 \mathcal{W} : 64$	13.59%	$\mathcal{H} : 10 \mathcal{W} : 8$
19.35%	$\mathcal{H} : 10 \mathcal{W} : 8$	13.84%	$\mathcal{H} : 11 \mathcal{W} : 16$
19.58%	$\mathcal{H} : 6 \mathcal{W} : 64$	13.97%	$\mathcal{H} : 5 \mathcal{W} : 64$
20.11%	$\mathcal{H} : 11 \mathcal{W} : 16$	14.21%	$\mathcal{H} : 6 \mathcal{W} : 64$
21.03%	$\mathcal{H} : 10 \mathcal{W} : 64$	14.30%	$\mathcal{H} : 11 \mathcal{W} : 32$
21.06%	$\mathcal{H} : 11 \mathcal{W} : 32$	14.54%	$\mathcal{H} : 10 \mathcal{W} : 64$
22.33%	$\mathcal{H} : 11 \mathcal{W} : 8$	14.77%	$\mathcal{H} : 11 \mathcal{W} : 8$
22.59%	$\mathcal{H} : 12 \mathcal{W} : 16$	15.15%	$\mathcal{H} : 12 \mathcal{W} : 16$
23.00%	$\mathcal{H} : 11 \mathcal{W} : 64$	15.74%	$\mathcal{H} : 12 \mathcal{W} : 32$
23.35%	$\mathcal{H} : 12 \mathcal{W} : 32$	16.05%	$\mathcal{H} : 11 \mathcal{W} : 64$
24.44%	$\mathcal{H} : 12 \mathcal{W} : 8$	16.08%	$\mathcal{H} : 12 \mathcal{W} : 8$
24.99%	$\mathcal{H} : 12 \mathcal{W} : 64$	17.68%	$\mathcal{H} : 12 \mathcal{W} : 64$
26.69%	$\mathcal{H} : 4 \mathcal{W} : 64$	18.54%	$\mathcal{H} : 4 \mathcal{W} : 64$

**Table B.2:** All the EERs for the verification experiment on the Thousand dataset

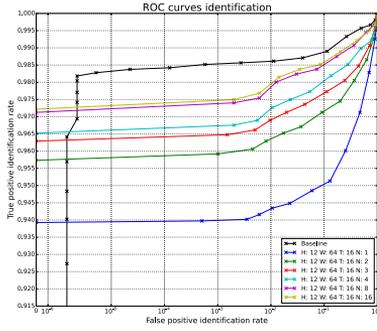


(a) LG

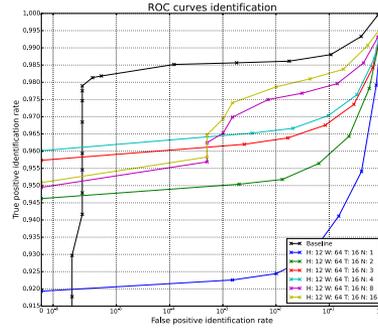


(b) QSW

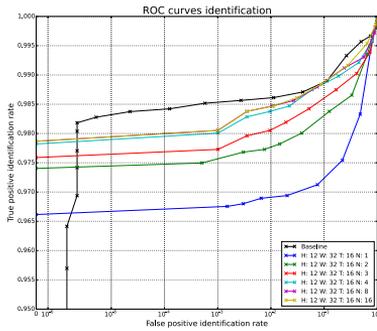
**Figure B.2:** The ROC curves for the comparatively best performing system configurations on the **Thousand** dataset



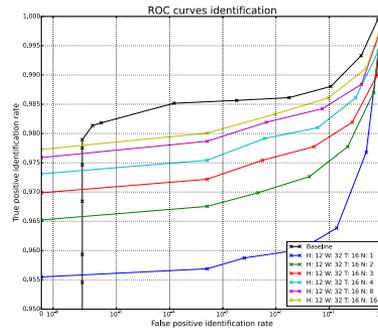
(a) LG



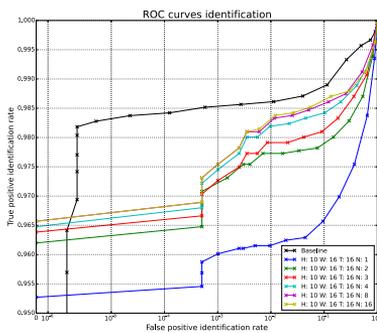
(b) QSW



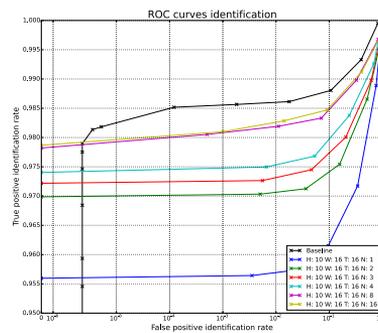
(c) LG



(d) QSW



(e) LG



(f) QSW

**Figure B.3:** The ROC curves for several well-performing system configurations in the selective tree traversal experiment

Configuration	Workload		Performance	
	Worst	Average	LG	QSW
$\mathcal{H}: 12 \mathcal{W}: 64 \mathcal{T}: 16 \mathcal{N}: 8$	$\omega \approx 5.24 * 10^6$ $F \approx 0.059$	$\omega \approx 3.15 * 10^6$ $F \approx 0.071$	$TP_0 = 97.13\%$ $TP_{0.1} = 97.42\%$	$TP_0 = 94.94\%$ $TP_{0.1} = 96.50\%$
$\mathcal{H}: 12 \mathcal{W}: 64 \mathcal{T}: 16 \mathcal{N}: 4$	$\omega \approx 3.15 * 10^6$ $F \approx 0.035$	$\omega \approx 2.10 * 10^6$ $F \approx 0.047$	$TP_0 = 96.52\%$ $TP_{0.1} = 96.62\%$	$TP_0 = 96.00\%$ $TP_{0.1} = 96.50\%$
$\mathcal{H}: 12 \mathcal{W}: 64 \mathcal{T}: 16 \mathcal{N}: 2$	$\omega \approx 2.10 * 10^6$ $F \approx 0.024$	$\omega \approx 1.57 * 10^6$ $F \approx 0.035$	$TP_0 = 95.73\%$ $TP_{0.1} = 95.90\%$	$TP_0 = 94.54\%$ $TP_{0.1} = 95.00\%$
$\mathcal{H}: 12 \mathcal{W}: 64 \mathcal{T}: 16 \mathcal{N}: 1$	$\omega \approx 1.57 * 10^6$ $F \approx 0.018$	$\omega \approx 1.57 * 10^6$ $F \approx 0.035$	$TP_0 = 93.91\%$ $TP_{0.1} = 94.02\%$	$TP_0 = 91.93\%$ $TP_{0.1} = 92.21\%$
$\mathcal{H}: 12 \mathcal{W}: 32 \mathcal{T}: 16 \mathcal{N}: 8$	$\omega \approx 1.05 * 10^6$ $F \approx 0.118$	$\omega \approx 6.29 * 10^6$ $F \approx 0.141$	$TP_0 = 97.76\%$ $TP_{0.1} = 98.05\%$	$TP_0 = 97.59\%$ $TP_{0.1} = 97.84\%$
$\mathcal{H}: 12 \mathcal{W}: 32 \mathcal{T}: 16 \mathcal{N}: 4$	$\omega \approx 6.29 * 10^6$ $F \approx 0.071$	$\omega \approx 4.19 * 10^6$ $F \approx 0.094$	$TP_0 = 97.72\%$ $TP_{0.1} = 98.01\%$	$TP_0 = 97.28\%$ $TP_{0.1} = 97.52\%$
$\mathcal{H}: 12 \mathcal{W}: 32 \mathcal{T}: 16 \mathcal{N}: 2$	$\omega \approx 4.19 * 10^6$ $F \approx 0.047$	$\omega \approx 3.15 * 10^6$ $F \approx 0.071$	$TP_0 = 97.46\%$ $TP_{0.1} = 97.54\%$	$TP_0 = 96.51\%$ $TP_{0.1} = 96.75\%$
$\mathcal{H}: 12 \mathcal{W}: 32 \mathcal{T}: 16 \mathcal{N}: 1$	$\omega \approx 3.15 * 10^6$ $F \approx 0.035$	$\omega \approx 3.15 * 10^6$ $F \approx 0.071$	$TP_0 = 96.56\%$ $TP_{0.1} = 96.72\%$	$TP_0 = 95.53\%$ $TP_{0.1} = 95.71\%$
$\mathcal{H}: 10 \mathcal{W}: 16 \mathcal{T}: 16 \mathcal{N}: 8$	$\omega \approx 5.24 * 10^6$ $F \approx 0.059$	$\omega \approx 3.15 * 10^6$ $F \approx 0.071$	$TP_0 = 96.53\%$ $TP_{0.1} = 97.51\%$	$TP_0 = 97.73\%$ $TP_{0.1} = 98.02\%$
$\mathcal{H}: 10 \mathcal{W}: 16 \mathcal{T}: 16 \mathcal{N}: 4$	$\omega \approx 3.15 * 10^6$ $F \approx 0.035$	$\omega \approx 2.10 * 10^6$ $F \approx 0.047$	$TP_0 = 96.49\%$ $TP_{0.1} = 97.46\%$	$TP_0 = 97.42\%$ $TP_{0.1} = 97.50\%$
$\mathcal{H}: 10 \mathcal{W}: 16 \mathcal{T}: 16 \mathcal{N}: 2$	$\omega \approx 2.10 * 10^6$ $F \approx 0.024$	$\omega \approx 1.57 * 10^6$ $F \approx 0.035$	$TP_0 = 96.23\%$ $TP_{0.1} = 97.28\%$	$TP_0 = 97.00\%$ $TP_{0.1} = 97.08\%$
$\mathcal{H}: 10 \mathcal{W}: 16 \mathcal{T}: 16 \mathcal{N}: 1$	$\omega \approx 1.57 * 10^6$ $F \approx 0.018$	$\omega \approx 1.57 * 10^6$ $F \approx 0.035$	$TP_0 = 95.29\%$ $TP_{0.1} = 96.00\%$	$TP_0 = 95.56\%$ $TP_{0.1} = 95.56\%$

**Table B.3:** The results of several well-performing configurations of the Bloom filter scheme with selective tree traversal

# Bibliography

---

- [AT09] Tariq Mehmood Aslam and Shi Zhuan Tan. Iris recognition in the presence of ocular disease. *Journal of the Royal Society Interface*, 6(34):489–493, 2009.
- [BHF08] Kevin W Bowyer, Karen Hollingsworth, and Patrick J Flynn. Image understanding for iris biometrics: A survey. *Computer vision and image understanding*, 110(2):281–307, 2008.
- [BLF10] Kevin W Bowyer, Stephen Lagree, and Samuel Fenker. Human Versus Biometric Detection of Texture Similarity In Left and Right Irises. In *IEEE International Carnahan Conference on Security Technology (ICCST)*, pages 323–329, 2010.
- [Blo70] Burton H. Bloom. Space/time trade-offs in hash coding with allowable errors. *Communications of the ACM*, 13(7):422–426, 1970.
- [BM05] Andrei Broder and Michael Mitzenmacher. Network Applications of Bloom Filters : A Survey. *Internet Mathematics*, 1(4):485–509, 2005.
- [Chi] Chinese Academy of Sciences. CASIA Iris Image Database. <http://biometrics.idealtest.org/>. [Online; accessed June 21, 2016].
- [Dau94] John Daugman. Biometric personal identification system based on iris analysis, March 1 1994. US Patent 5,291,560.
- [Dau00] John Daugman. Biometric decision landscapes. 2000.
- [Dau04a] John Daugman. How Iris Recognition Works. *IEEE Transactions on circuits and systems for video technology*, 4(1), 2004.

- [Dau04b] John Daugman. Recognising Persons by Their Iris Patterns. In *Advances in Biometric Person Authentication: 5th Chinese Conference on Biometric Recognition*, pages 5–25, 2004.
- [Dau06] John Daugman. Probing the uniqueness and randomness of iris-codes: Results from 200 billion iris pair comparisons. *Proceedings of the IEEE*, 94(11):1927–1934, 2006.
- [Dau16] John Daugman. Major International Deployments of the Iris Recognition Algorithms: a Billion Persons. <http://www.cl.cam.ac.uk/~jgd1000/national-ID-deployments.html>, 2016. [Online; accessed June 21, 2016].
- [Den] Denso Wave Incorporated. QR Code. <http://www.qrcode.com/en/>. [Online; accessed June 21, 2016].
- [FB12] Samuel P Fenker and Kevin W Bowyer. Analysis of Template Aging in Iris Biometrics. In *IEEE Computer Society Biometrics Workshop*, pages 1–7, 2012.
- [FS87] L. Flom and A. Safir. Iris recognition system, February 3 1987. US Patent 4,641,349.
- [GAH<sup>+</sup>08] James Goldstein, Rina Angeletti, Manfred Holzbach, Daniel Konrad, and Max Snijder. *Large-scale Biometrics Deployment in Europe : Identifying Challenges and Threats*. 2008.
- [GAR10] Ravindra B. Gadde, Donald Adjeroh, and Arun Ross. Indexing iris images using the Burrows-Wheeler Transform. *2010 IEEE International Workshop on Information Forensics and Security*, (December):1–6, 2010.
- [GRC09a] J E Gentile, N Ratha, and J Connell. An Efficient, Two-Stage Iris Recognition System. *2009 Ieee 3rd International Conference on Biometrics: Theory, Applications and Systems*, pages 211–215, 2009.
- [GRC09b] J E Gentile, N Ratha, and J Connell. SLIC: Short-Length Iris Codes. *2009 Ieee 3rd International Conference on Biometrics: Theory, Applications and Systems*, pages 171–175, 2009.
- [HDZ08] Feng Hao, J. Daugman, and P. Zielinski. A Fast Search Algorithm for a Large Fuzzy Database. *IEEE Transactions on Information Forensics and Security*, 3(2):203–212, 2008.
- [HJP99] Lin Hong, Anil Jain, and Sharath Pankanti. Can Multibiometrics Improve Performance? In *Proceedings of IEEE AutoID*, pages 59–64, 1999.

- [Hun07] J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing In Science & Engineering*, 9(3):90–95, 2007.
- [IIT] IIT Delhi. IIT Delhi Iris Database. [http://www4.comp.polyu.edu.hk/~csajaykr/IITD/Database\\_Iris.htm](http://www4.comp.polyu.edu.hk/~csajaykr/IITD/Database_Iris.htm). [Online; accessed June 21, 2016].
- [Ind15] India. Aadhaar issued summary. <https://portal.uidai.gov.in/uidwebportal/dashboard.do>, 2015. [Online; accessed June 21, 2016].
- [ISO11] ISO/IEC. INTERNATIONAL STANDARD Performance testing and reporting. 2011.
- [JBP99] A.K. Jain, R. Bolle, and S. Pankanti. *Biometrics: Personal Identification in Networked Society*. Kluwer Academic Publications, 1999.
- [JG13] Umarani Jayaraman and Phalguni Gupta. Iris Code Hashing. *2013 IEEE International Conference on Communications (ICC)*, pages 716–720, 2013.
- [JPG12] Umarani Jayaraman, Surya Prakash, and Phalguni Gupta. An efficient color and texture based iris image retrieval technique. *Expert Systems with Applications*, 39(5):4915–4926, 2012.
- [Knu73] Donald E. Knuth. *The Art of Computer Programming, Volume 3: Sorting and Searching*. Addison-Wesley, 1973.
- [KP10] Ajay Kumar and Arun Passi. Comparison and Combination of Iris Matchers for Reliable Personal Authentication. *Pattern Recognition*, 43(3):1016–1026, 2010.
- [KSUW10] Mario Konrad, Herbert Stogner, Andreas Uhl, and Peter Wild. Rotation-Invariant and Rotation Compensating Iris Recognition Algorithms. *Proceedings of the 5th International Conference on Computer Vision Theory and Applications*, 1:85–90, 2010.
- [Li09] Stan Z. Li. *Encyclopedia of Biometrics*. Springer Publishing Company, 1st edition, 2009.
- [Mil16] Bill Mill. Bloom Filters by Example. <http://billmill.org/bloomfilter-tutorial/>, 2016. [Online; accessed June 21, 2016].
- [Mor] Morpho (Safran). Morpho IAD. <http://www.morpho.com/en/public-security/smart-borders/automated-solutions/acquisition-devices/morpho-iad>. [Online; accessed June 21, 2016].

- [MR08] R. Mukherjee and a. Ross. Indexing iris images. *2008 19th International Conference on Pattern Recognition*, pages 1–3, 2008.
- [MSM09] Hunny Mehrotra, Badrinath G Srinivas, and Banshidhar Majhi. Indexing Iris Biometric Database Using Energy Histogram of DCT Subbands. *Journal of Communications in Computer and Information Science*, 40:194–204, 2009.
- [Muk07] Rajiv Mukherjee. Indexing Techniques for Fingerprint and Iris Databases. *MSc Thesis*, 2007.
- [NC15] Pattabhi Ramaiah Nalla and Krishna Mohan Chalavadi. Iris classification based on sparse representations using on-line dictionary learning for large-scale de-duplication applications. *SpringerPlus*, 4:1–10, 2015.
- [Pro13] Hugo Proença. Iris biometrics: Indexing and retrieving heavily degraded data. *IEEE Transactions on Information Forensics and Security*, 8(12):1975–1985, 2013.
- [QST05] Xianchao Qiu, Zhenan Sun, and Tieniu Tan. Global Texture Analysis of Iris Images for Ethnic Classification. *Advances in Biometrics*, 3832:411–418, 2005.
- [R C14] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2014.
- [RBB13] C. Rathgeb, F. Breitingner, and C. Busch. Alignment-free cancelable iris biometric templates based on adaptive bloom filters. In *2013 International Conference on Biometrics (ICB)*, pages 1–8, 2013.
- [RBBB15] C Rathgeb, F. Breitingner, H Baier, and C Busch. Towards Bloom filter-based indexing of iris biometric data. In *2015 International Conference on Biometrics (ICB)*, pages 422–429, Phuket, May 2015. IEEE.
- [RNJ06] Arun Ross, K Nandakumar, and A Jain. *Handbook of multibiometrics*, volume 6. 2006.
- [RRKB15] Kiran B Raja, R Raghavendra, Vinay Krishna, and Christoph Busch. Smartphone based visible iris recognition using deep sparse filtering. *Pattern Recognition Letters*, 57:33–42, 2015.
- [RS10] Arun Ross and Manisha Sam Sunder. Block based texture analysis for iris classification and matching. *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops, CVPRW 2010*, pages 30–37, 2010.

- [RSH<sup>+</sup>13] Petru Radu, Konstantinos Sirlantzis, Gareth Howells, Sanaul Hoque, and Farzin Deravi. A Colour Iris Recognition System Employing Multiple Classifier Techniques. 12(2):54–65, 2013.
- [RU10] Christian Rathgeb and Andreas Uhl. Iris-Biometric Hash Generation for Biometric Database Indexing. *2010 20th International Conference on Pattern Recognition*, pages 2848–2851, 2010.
- [RUW10] Christian Rathgeb, Andreas Uhl, and Peter Wild. Incremental iris recognition: A single-algorithm serial fusion strategy to optimize time complexity. *IEEE 4th International Conference on Biometrics: Theory, Applications and Systems*, 2010.
- [RUWH16] Christian Rathgeb, Andreas Uhl, Peter Wild, and Heinz Hofbauer. Design decisions for an iris recognition sdk. In Kevin Bowyer and Mark J. Burge, editors, *Handbook of Iris Recognition*, Advances in Computer Vision and Pattern Recognition. Springer, second edition, 2016.
- [SGSD12] Guillaume Sutra, Sonia Garcia-Salicetti, and Bernadette Dorizzi. The Viterbi Algorithm at Different Resolutions for Enhanced Iris Segmentation. *5th IAPR International Conference on Biometrics*, pages 310–316, 2012.
- [Tel] Telecom Sud Paris. BioSecure project. [http://svnext.it-sudparis.eu/svnview2-eph/ref\\_syst//Iris\\_Osiris\\_v4.1/](http://svnext.it-sudparis.eu/svnview2-eph/ref_syst//Iris_Osiris_v4.1/). [Online; accessed June 21, 2016].
- [Tra14] Transparency. Biometrics Technology (Face, Hand geometry, Voice, Signature, Iris, AFIS, Non-AFIS and Others) Market - Global Industry Analysis, Size, Share, Growth, Trends and Forecast 2013 - 2019. <http://www.transparencymarketresearch.com/biometrics-technology-market.html>, 2014.
- [WWZQ07] Xiangqian Wu, Kuanquan Wang, David Zhang, and Ning Qi. Combining Left and Right Irises for Personal Authentication. In *Proceedings of the Energy Minimization Methods in Computer Vision and Pattern Recognition: 6th International Conference*, pages 145–152, 2007.
- [YZWY05] Li Yu, David Zhang, Kuanquan Wang, and Wen Yang. Coarse iris classification using box-counting to estimate fractal dimensions. *Pattern Recognition*, 38:1791–1798, 2005.
- [ZSTW14] Hui Zhang, Zhenan Sun, Tieniu Tan, and Jianyu Wang. Iris Image Classification Based on Hierarchical Visual Codebook. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(6), 2014.