# Biometric Systems

Andreas Uhl

Department of Computer Sciences
University of Salzburg

January 26th, 2016

# Overview

1. Formalia
2. Introduction & Motivation
3. Foundations of Biometrics
   - Introduction and Basic Terminology
   - Biometric Traits & their Properties
   - Biometric Systems System Model
   - Assessment of Biometric Systems
   - Security and Privacy in Biometric Systems
4. Non-visual Biometrics
   - Keystroke Dynamics
   - On-Line Signatures
   - Biorhythms
   - Speaker Recognition
   - Odour
5. Visual Biometrics
   - Off-line Signatures
   - Iris Recognition
   - Fingerprint Recognition

# Outline

1. Formalia
2. Introduction & Motivation
3. Foundations of Biometrics
   - Introduction and Basic Terminology
   - Biometric Traits & their Properties
   - Biometric Systems System Model
   - Assessment of Biometric Systems
   - Security and Privacy in Biometric Systems
4. Non-visual Biometrics
   - Keystroke Dynamics
   - On-Line Signatures
   - Biorhythms
   - Speaker Recognition
   - Odour
5. Visual Biometrics
   - Off-line Signatures
   - Iris Recognition
   - Fingerprint Recognition

# Lecturer

Email-Address: `uhl@cosy.sbg.ac.at`.

Basis-URL: `http://www.cosy.sbg.ac.at/~uhl`.

Office: FB Computerwissenschaften (Department of Computer Sciences), Room 1.15, Jakob-Haringer Str. 2, Salzburg-Itzling.

Telefon (Office): (0662) 8044-6303.

Telefon (Secretary): (0662) 8044-6328 or -6343.

# When & Where

Course-URL: http://www.cosy.sbg.ac.at/~uhl/student.html.

When: Di 8:30 - 10:00

Interval: weekly

Where: Lecture Room T02

# This lecture & Exam

Welcome to the lecture on “Biometric Systems”. This lecture is of overview-type but still covers lots of research-related material since the subject-area is rather a recent one.
Biometrics have become a rather larger field, the focus of this lecture emphasises the topics covered locally, especially targeting visual media types (i.e. images, video, 3D data), but covering also on-line signatures and speaker recognition.
We offer 3 variants for the exam:

1. Classical exam (orally, based on these slides mainly)
2. When signing attendence lists: 2 papers according to students interest and some basic knowledge about lectures’ content (increasing the number of papers to be covered by one for each missed lecture up to 4 papers
3. When signing attendence lists: Extending the projects of the lab (Proseminar) and some basic knowledge about lectures’ content

# Outline

1. Formalia
2. **Introduction & Motivation**
3. Foundations of Biometrics
   - Introduction and Basic Terminology
   - Biometric Traits & their Properties
   - Biometric Systems System Model
   - Assessment of Biometric Systems
   - Security and Privacy in Biometric Systems
4. Non-visual Biometrics
   - Keystroke Dynamics
   - On-Line Signatures
   - Biorhythms
   - Speaker Recognition
   - Odour
5. Visual Biometrics
   - Off-line Signatures
   - Iris Recognition
   - Fingerprint Recognition

## What's special about Biometrics ?

Classical Cryptography provides well investigated authentication techniques for all types of security applications. Biometrics provides alternatives for those but is not restricted to the classical security-related authentication application use-case.
Which subject areas / scientific fields are covered in Biometrics ?

- Applications of biometric systems (authentication, human computer interaction, healthcare, etc.)
- Signal- and image/video processing
- Privacy and revocability of data
- Social science, legal questions

# Which biometric modalities are covered by “Biometric Systems” ?

- Non-visual modalities
  - Key-stroke dynamics
  - On-line signatures
  - Biorhythms (heartbeat, ECG, EEG)
  - Speaker recognition
  - Uncommon modalities (odour, keystroke audio, gestures, ....)
- Visual modalities
  - Iris recognition
  - Retina recognition
  - Fingerprint recognition
  - Face recognition
  - Vein recognition
  - Remarks on Gait, Ear, Foot, etc. recognition

# Top Journals

- Specific Journals
    - IEEE Transactions on Information Forensics and Security (TIFS)
    - IET Biometrics
    - Questionable quality: Int. Journal on Biometrics
- Unspecific Journals
    - IEEE Transactions on Pattern Analysis and Machine Intelligence
    - Pattern Recognition
- NOT: Biometrics

.... or in more general purpose Journals in the areas of Multimedia, Signal Processing and Security (“CRYPTOGRAPHY MARRIES MULTIMEDIA SIGNAL PROCESSING”).

# International Conferences

- IAPR Int. Conference on Biometrics ICB (2015 in Phuket, 2016 in Halmstadt)
- IAPR/IEEE Int. Joint Conference on Biometrics IJCB (2014 in Clearwater Bay, FL)
- IEEE Int. Conference on Biometric Technologies and Applications BTAS (always US, mostly Washington DC)
- Int. Conference of the GI Biometrics Special Interest Group BIOSIG (always Darmstadt)
- IEEE Int. Conference on Face and Gesture Recognition ICFG (2015 in Lubljana)
- ACM Workshop on Information Hiding and Multimedia Security (2013 in Montpellier, 2014 in Salzburg, 2015 in Portland)
- IEEE International Workshop on Information Forensics and Security WIFS (2014 in Atlanta, 2015 in Rome)
- SPIE's Technologies for Human Recognition for Homeland Security (in context of EI Symposium, February, Bay area)

Many more smaller meetings ..... and special sessions and special

# Local Projects @ Wavelab

- Privacy-protected Video Surveillance on Scalable Bitstreams (FFG, with Commend International, 200K EUR)
- Biometric Sensor Forensics (FWF, 280K EUR, ongoing)
- Sample Data Compression and Encryption in Biometric Systems (FWF, 210K EUR)
- Assessing Image and Video Encryption Schemes (FWF, 300K EUR, ongoing)
- Biometrics and Forensics in the Digital Age (EU COST Action IC 1106)
- De-identification of Multimedia Data (EU COST Action IC 1206)

# Literature

- Monographs
  - Iris Biometrics: From Segmentation to Template Security (Rathgeb, Uhl, Wild; Springer, 2013).
  - Biometric Systems (Wayman, Kain, Maltoni, Maio; Springer 2005).
  - Handbooks of Fingerprint Recognition (Maltoni, Maio, Jain, Prabhakar; Springer 2009), Face Recognition (Li, Jain; Springer 2004), and Iris Recognition (Bowyer, Philips, Ross; Springer 2015)
  - Palmprint Authentication (Zhang; Kluwer 2004)
  - Biometric User Authentication for IT Security (Vielhauer; Springer 2006)
  - Security and Privacy in Biometrics (Campisi; Springer 2013)
  - Handbook of Biometric Anti-Spoofing (Marcel, Nixon, Li; Springer 2014)
  - Age Factors in Biometric Processing (Fairhurst; IET 2013)

# Outline

1. Formalia
2. Introduction & Motivation
3. Foundations of Biometrics
    - Introduction and Basic Terminology
    - Biometric Traits & their Properties
    - Biometric Systems System Model
    - Assessment of Biometric Systems
    - Security and Privacy in Biometric Systems
4. Non-visual Biometrics
    - Keystroke Dynamics
    - On-Line Signatures
    - Biorhythms
    - Speaker Recognition
    - Odour
5. Visual Biometrics
    - Off-line Signatures
    - Iris Recognition
    - Fingerprint Recognition

# Biometrics Terminology

Term “biometrics”: from ancient Greek – “bios” for life, “metros” for measurement

In the broad sense, biometrics denoted statistical analysis of biological observations and phenomena. In the sense related to security technology, biometrics is the automated recognition of individuals based on biological / physiological or behaviour-related characteristics.

Anthropometrics: measurment techniques for the human body and its parts, e.g. forensic anthropometrics are used to identify criminal offenders based on such measurements.

Identity recognition – authentication – is based on three approaches:

- Token-based: identification by something you have (e.g. document, token, smartcard)
- Knowledge-based: identification by something you know (e.g. password, PIN)
- Biometrics-based: identification by something you are (human body, biometric identifier, human activity)

# Disadvantages of traditional Authentication Techniques

Tokens can get lost, can be stolen, can get robbed. Knowledge can be forgotten, can be guessed or others can find it out (poor quality passwords, write down PINs on smartcards).

GLASBERGEN
"Sorry about the odor. I have all my passwords tattooed between my toes."

Figure: How to cope with many PWDs.

# Application Areas: Medical Biometrics

In bioinformatics (defined as the application of computer science in biology and medicine) the term biometrics is often used as a synonym for “biomedical data sciences”. By anaolgy to other biometric disciplines measurement data of biological or medical phenomena are collected, albeit with an entirely different aim:
Here its not about identifying an *individual*, but about statistical evaluation and description of entire *populations*, e.g. classification of genes, proteins or diseases (e.g. do we find a certain genetic anomaly, we observe in 20% of all individuals a certain clinical symptom).
Corresponding to this aim, the used techniques are different of course.
This is the area which is covered by the journal “Biometrics”.

# Application Areas: Forensic Biometrics

Forensics are the origin of biometrics as we know it today. Aims and used techniques are similiar to user authentication / identification, however, there is a big difference:

- User Authentication: biometric techniques are used by a person to proof hir/her identity
- Forensics: biometric techniques are used by other persons to reveal the identity of a target person. What is different is the perspective here !

Important examples are latent fingerprints or video surveillance combined with face recognition, DNA recognition gait recognition from surveillance and furhter techniques which are used by CSI to identify criminal offenders.

# Application Areas: Convenience Biometrics – HCI

In the area of HCI the aim of biometrics is to identify a user in order to adjust performance and accuracy of user dependent interfaces or simply to enhance usability, e.g.

- If the user is correctly identified, his/her language specifities can be used in speech recognition to improve recognition rate.
- If the user is identified, his/her personal configurations and settings can be loaded (e.g. car settings in case of driver changes)

Aethods are identical to user authentication biometrics, but the aims are different.

# Application areas: Security Biometrics – User Authentication

Here the aim is to determine or confirm the identity of a user. Authentication can be used for providing logical and physical access to some infrastructure (access control) or the binding of digital information to some identity (information authentication).
Access control examples: Fingerprint scanner at front door, fingerprint access to laptops or storage media, face recognition at ABC gates
Information authentication examples: using a biometric hash as a private key to sign electronic documents, copyright definition by embedding of biometric data in e.g. videos (by watermarking), generation of key materail for cryptographic applications in general (**Attention:** in this context we require unique bits, for general biometric authentication is certain similarity is sufficient).
Contrasting to other biometrics application areas, biometrics are employed by to users to reach a specific aim.

# Uncommon Biometrics: Fauna and Flora

## Recognition of animals

- face and retina recognition in sheep
- avian comb of poultry
- nose-prints in cattle
- skin texture: zebra stripes and fish skin (fish farm “surveillance”)
- bird songs to classify species and variations within species

## Recognition of plants, fruits or vegetables

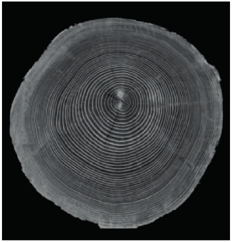
- tree leave identification to determine species
- fruit / vegetable images, for controlling fruits when weighted by customer
- determination of weed and subsequent extinction by agricultural robots

# Uncommon biometrics: Tracability of wood logs

Logs are cut in the forest and transported to a sawmill, pulp mill or another processing company. To cope with financial claims of the forest owner, the provenience of a log in the sawmill has to be known.

## Ways to achieve traceability

- Manual labels (paint, hammer labels) and Badge labels
- Transponders
- Log biometrics

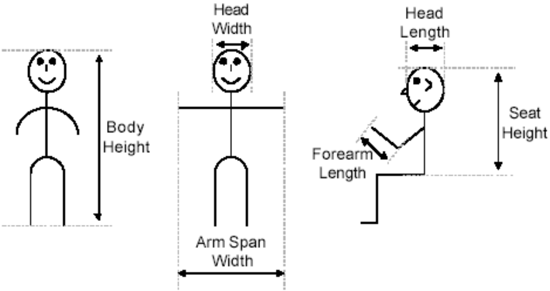


The challenge is to find feature sets in cross section images, which are robust aginst vertical displacement of the cross section (i.e. when the log is cut into smaller segements). Having dendro-chronology in mind (determining a tree's age based on annual ring patterns), this is far from being trivial.

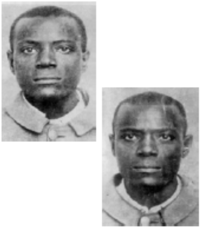
*FWF TRP project "Traceability of Logs by Means of Digital Images"*.

# History of Biometrics: Alfonse Bertillon

Marking prisoners by tatooing got terminated in 1832 in France. This caused to problem how to identify repeat offenders. The resulting system (employing many anthropometric aspects) was the first to use scientific techniques to systematically idententify persons.
Besides body measurements (classified into bins) and eye colour, typical movements as well as local and global skin propoerties got archived. After the case “Will West” the system got replaced by a fingerprint based one.

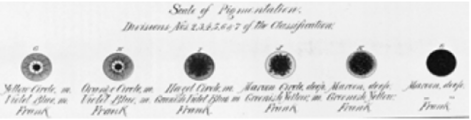


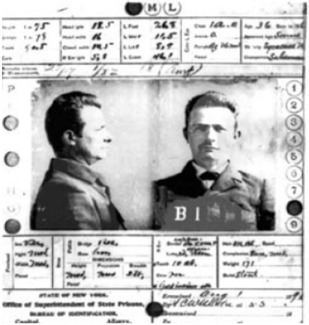
Will West's Bertillon Measurements
178.5; 187.0; 91.2; 19.7; 15.8; 14.8; 6.6; 28.2; 12.3; 9.7

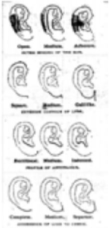


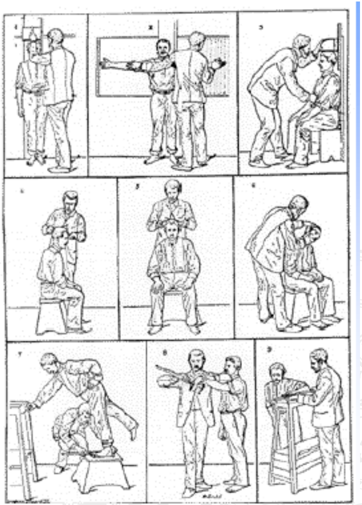
William West's Bertillon Measurements
177.5; 188.0; 91.3; 19.8; 15.9; 14.8; 6.5; 27.5; 12.2; 9.6; 50.3

# Bertillonage


Scale of Pigmentation:
Divisions Nos 2,3,4,5,6 & 7 of the Classification:
Yellow Circle, in Violet Blue, in Frank
Orange Circle, in Violet Blue, in Frank
Hazel Circle, in Greenish Violet Blue in Frank
Maroon Circle, deep, Greenish Yellow, in Frank
Maroon, deep, Greenish Yellow: Frank
Maroon, deep, Frank


M
L
P
1
2
3
4
5
6
7
8
9
B I
STATE OF NEW YORK.
Office of Superintendent of State Prisons,
BUREAU OF IDENTIFICATION.





# Operation Modes in Biometric Authentication

A biometrics-based authentication system (from now on, we restrict to attention to these schemes) is basically a pattern recognition system which recognises a person by verifying the authenticity of a physiological or behavioural characteristic. Depending on the application context, a biometric system may operate in verification or identification mode.

Verification: is to answer the question “does the user actually corresond to the claimed identity” ? The person to be verified claims a certain identity and presents the biometric trait / characteristic, which is compared to the trait stored in the database under this identity. From the computational viewpoint, a 1 to 1 comparison is conducted and the system answer confirms or denies the claimed identity.

Identification: is to answer the question “who is the user” ? A person is recognised by searching through the entire database for a matching trait. From the computational viewpoint, a 1 to N comparison is conducted and sytem answer is either the identity of the user or information about a failure to authenticate.

# Biometric Verification vs. Identification I

Example: a combination lock with 4 digits has 10000 potential combinations.
**Verification**: my lock has combination 2463. In case I find a lock and would like to verify if it is mine, I can test for the combination 2463. If the combination matches, it is probably mine. The probability that I am wrong (that it has the same combination accidentially) is $1/10000 = 0.00001 = 0.01\%$. Thus, the probability that it is actually mine is $1.0 - 0.00001 = 0.9999$ oder 99.99%.
**Identification**: I have a pile of 10000 locks and I am trying to find mine. I have to conduct 10000 tests, the probability to be correct in each single test is 0.9999. For a correct identification I need to be correct in each of 10000 tests, the probability for this is amazing $0.9999^{10000} = 0.37$. The probablity to take the wrong lock (an incorrect identification) is $1.0 - 0.37 = 0.63$ !!!! (although the probability for a single test is 0.9999 !!).

# Biometric Verification vs. Identification II

Probability of false identification rises fast with the size of the database (i.e. the size of the pile of locks).
1000 locks: $1.0 - 0.9999^{1000} = 0.09$. 10000 locks: $1.0 - 0.9999^{10000} = 0.63$. 100000 locks: $1.0 - 0.9999^{100000} = 0.99995$.
Example: FBI Criminal Master File had once fingerprints of 50.000.000 persons. Which accuracy is required from each single comparison to result in an identifaction with 99.99% accuracy ?

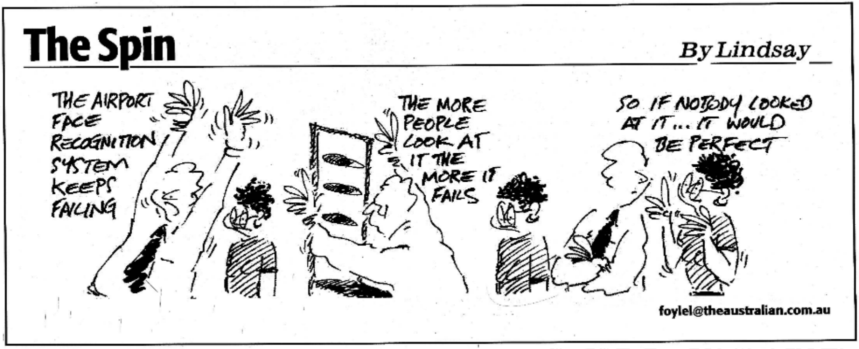
$$X^{50.000.000} = 0.9999$$

$X = 0.9999999999998$ (assuming an identical error probability for each comparison). This means one error per 500,000,000,000 comparisons. Earth population is about 8,000,000,000.

**HOUSTON, WE HAVE A PROBLEM !!!**

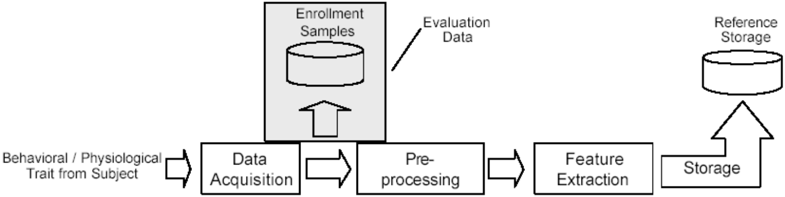
Fortunately, for many applications verification is sufficient, however, identification is often more convenient. However, one should be aware of the general problem when talking about reliable identification.

# Biometric Verification vs. Identification III


The Spin
By Lindsay
THE AIRPORT FACE RECOGNITION SYSTEM KEEPS FAILING
THE MORE PEOPLE LOOK AT IT THE MORE IT FAILS
SO IF NOBODY LOOKED AT IT... IT WOULD BE PERFECT
foylel@theaustralian.com.au

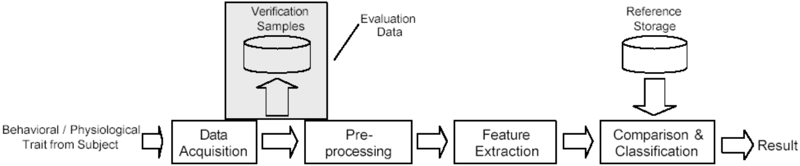
# Enrollment vs. Authentication: Enrollment

Biometric authentication system operate in two modes: In the first stage, all potential users need to register to the system, which is called “enrollment”. Reference traits for each user are stored in the system as “templates” (i.e. feature vectors) and associated with the identity of the user. Enrollment samples (= original data) might be eventually stored in encrypted and compressed manner to support a later alternative feature extraction. Before data is stored, quality control is conduced and in case of low quality a re-enrollment is demanded.



# Enrollment vs. Authentication: Authentication

In authentication, the identity of the user is either confirmed (verification) or determined (identification). In verification, the identity of the user is claimed besides providing the biometric trait to the system. Processing stages correspond to those during enrollment until feature extraction, subsequently comparisons with reference traits in the database are conducted.



# Positive vs. Negative Recognition Mode

- **Positive recognition**: The system verifies if the person really has the claimed (explicitly: verification; implicitely: identification) identity. The aim of positive recogntion is to prevent the use of a single identity by several users. It is tested if the authentication samples originate from a person enrolled in the system. Classical access control is the best example.
- **Negative recognition**: The system verifies if the person has the identity he/she denies to be. The aim of negative recognition is to prevent a single person from using several identities. It is tested if the authentication samples originate from a person not enrolled in the system. It is verified, if the person is not yet enrolled. The best example is "double dipping" for social security payments, which is only done if the person is not enrolled in the system (enrollment is done once a month as soon as a payment is received). In this case, authentication is equal to enrollment !

Positive recognition may be facitlitated with classical authentication means as well, while negative recognition is limited to biometric traits.

# Biometric Application Scenarios I

- *cooperative vs. non-cooperative*: refers to the behaviour of the person to be authenticated when interacting with the system. In case of positive recognition, it is usually in the interest of the authenticating person to act as cooperative as possible (e.g. electronic banking). Negative recognition is different here, since not being recognised in in the interest of the person to be authenticated, thus, for them, it makes sense not to cooperate.
- *overt vs. covert*: in case the person to be authenticated is aware of a biometric system in operation, it is a overt scheme. Face recognition can be operated in covert manner (hidden surveillance cameras, surveillance cameras pretending not to use face recognition), fingerprints have to be always operated in overt manner. In forensic applications this is not necessarily the case.

# Biometric Application Scenarios II

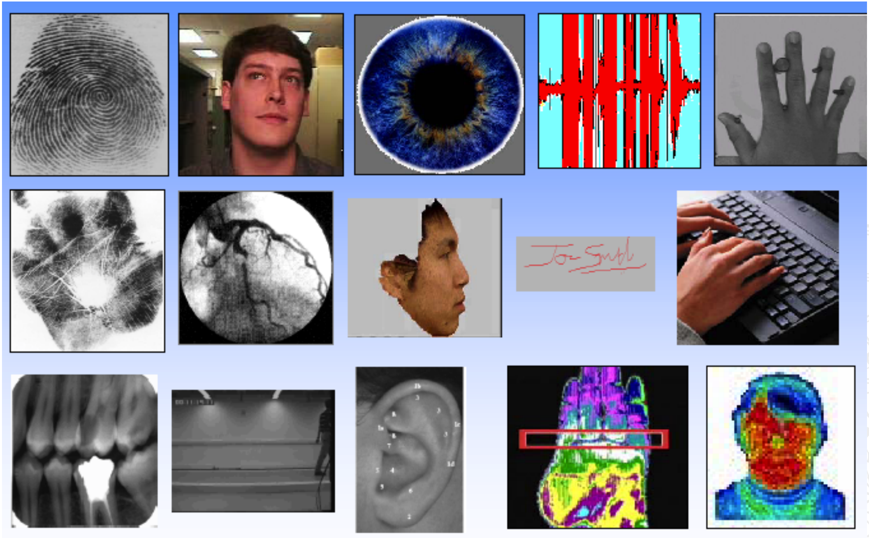
- *habituated vs. non-habituated*: here the question is if the usage of the biometric system is routine for the user. This is important for recognition accuracy, which often increases with the routine of the users. Access control to some office environment is habituated if done on a daily basis, control of driving license or border control is rather non-habituated due to infrequent execution.
- *attended vs. non-attended*: relates to the question if capturing the biometric trait is done under supervision, guided or under surveillance (by security personnel). A biometric system might have attended enrollment (beneficial in terms of quality control) but unattended authentication. Non-cooperative applications need to be attended.
- *standard vs. non-standard*: a standard environment refers to application of the system in a controlled environment ( controlled in terms of temperature, air pressure, humidity, lighting). Quite often we have standard environents indoor, whereas outdoor is mostly non-standard.

# Biometric Application Scenarios III

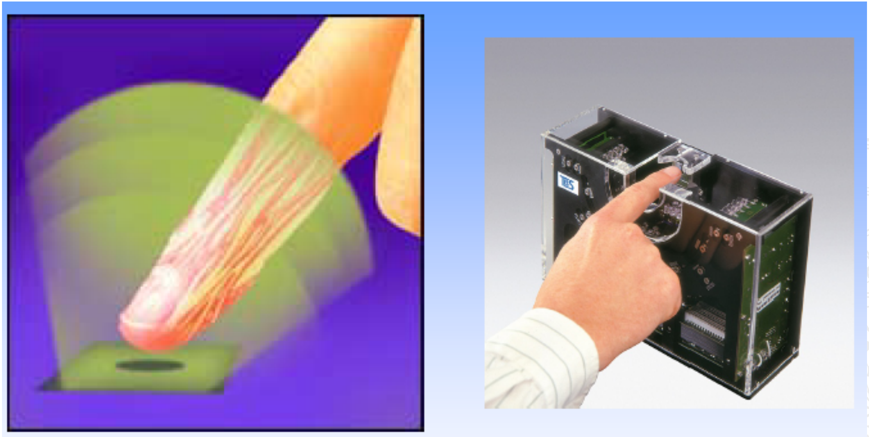
- *public vs. private*: refers to the question if the users of the system are “customers” (public) or employees (private) of the organisation that operates the biometric system. Governmental operated biometrics is always public.
- *open vs. closed*: will it ever be possible to exchange data with another system ? A closed system may employ proprietory data formats (like the FBI WSQ fingerprint representation), while an open system needs to adhere to standards (e.g. JPEG2000).

Example: EU-ABC gates (Face recognition) – positive recognition, cooperative, overt, non-attended, non-habituated, standard, public, open.

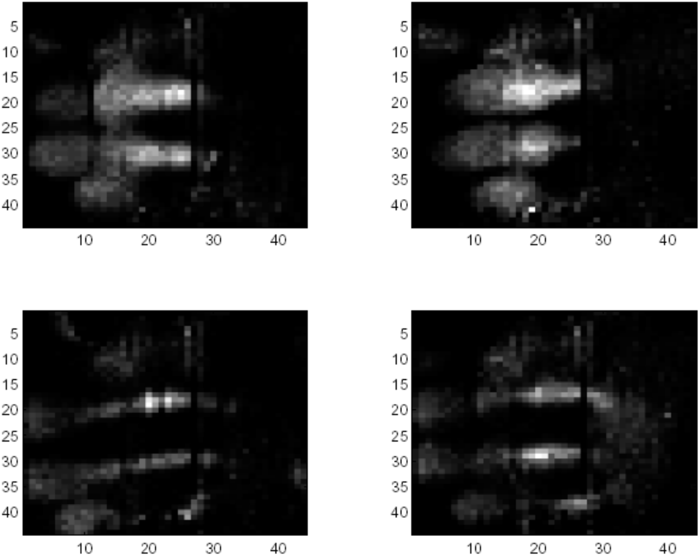
# Classical Biometric Traits



# Recent Traits: 3D finger veins



# Recent Traits: Grip Pattern

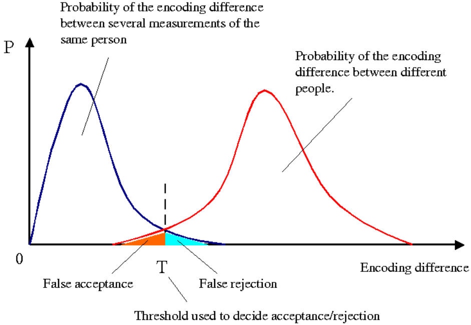


# Properties of Biometric Traits I

- *Universality*: Each person should exhibit a biometric trait
- *Ascernability - Collectability*: Each biometric trait should be measurable quantitatively and of course accessible easily. In this context, sensor technology is of high importance, but also preprocessing plays an important role and has to ascertain a certain quality. Acquisition time and processing time are important as well (problem for DNA-based biometrics).
- *Variability - Permanence*: Each biometric trait is subject to natural variability (physiological traits are affected by contineous cell replacements and ageing, behavioural traits are usually affected even worse). Thus biometric traits differ. This is denoted as Intra-personal or Intra-class variability (in case of interpreting biometrics as a classification problem with many classes). Additional causes are different operational conditions of the sensor and A/D conversion problems. Variability should be kept small to facilitate sufficient trait *permanence*.

# Properties of Biometric Traits II

- *Distinctiveness*: Biometric traits need to exhibit sufficient “discriminative power”, i.e. they need to be different from person to person. This is meant by high Inter-personal or Inter-class variability.



- Left – Intra-personal variability, right – Inter-personal variability
- Distinctiveness is seen at the intersection of the distributions and this should be as small as possible. In the area of the intersection we observe incorrect authentications.

# Properties of Biometric Traits III

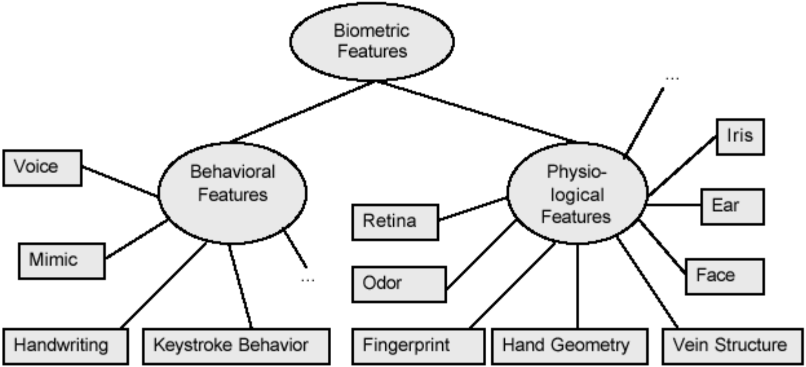
- *Stability*: qualitative propertive of a trait with low variability and high distinctiveness.
- *Performance*: execution speed of acquistion, processing, matching, scalability to large user groups, recogniiton accuracy, required resources to obtain a certain accuracy, etc.
- *Acceptability*: is the question if users are willing to undergo data-acquisition in the required frequency. Taking skin sample is hardly an option, still a retina scan is not really realistic in everyday use.
- *Circumvention*: how difficult is it to fool the system ? Is the system susceptical to spoofing or injection attacks, or to recapturing / presentation attacks ?

# Acquisition of Biometric Traits

- Physical removal of organic material: is mostly done in forensics. Classical examples are hair, skin, saliva, seminal fluid, etc. for DNA analysis. For user authentication this is not feasible due to the time-consuming biochemical extraction of DNA sequences (down to about 10 minutes in fast schemes). Further, organic material can be lost and re-used fraudulently by other persons.
- Person behaviour: is determined by three essential factors – physiological characteristics of the organs producing the behaviour, learnt characteristics how the required behaviour is generated, and the intention of the behariour display.
- Pysiological characteristics: individual biological structures which can be acquired by sensors (camera, microphone) without actually taking physiological samples.

# Behaviour vs. Physiology

When acquiring biometric traits based on physiological properties the authenticating person may remain inactive – this traits are therefor called *passive*. Contrasting to this, activity needs to be conducted for the acquisition of behavioural traits, thus termed *active* biometrics. These observations have important implication wrt. suitability of certain traits in cooperative or covert environments.



# Behaviour Biometrics: Acquisition

In active or behavioural biometrics there are two options how to conduct measurements: Either the entire behaviour is recorded or only the “end result” is captured. In the first case the result of the acquisition are temporally ordered measurment values, which are obtained by sampling the original signal (thus, A/D conversion problems need to be considered). Corresponding techniques are called “on-line”.
“Off-line” techniques only consider the final result but not the generation process. Off-line features can usually be generated from on-line ones, but not vice versa !
A popular example is the human signature, which is used as on-line as well as off-line trait.

# Behaviour vs. Physiology: Liveness

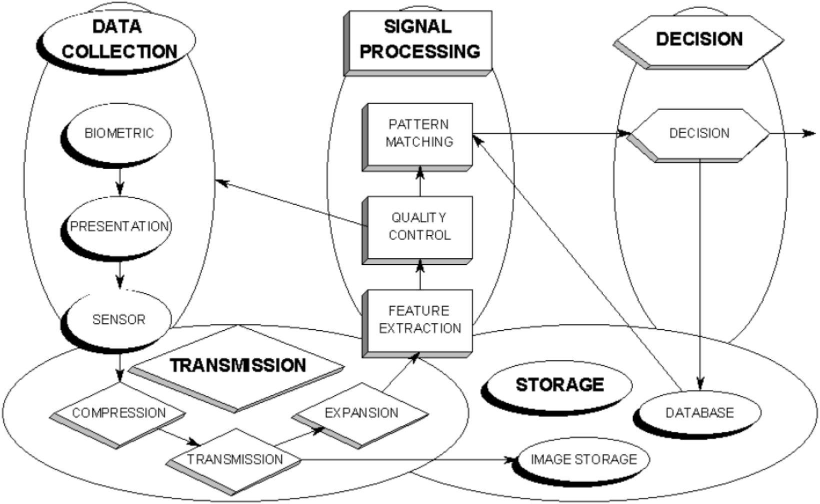
Since for purely physiological acquisition no activity is required, additional appropriate actions are beneficial to confirm the liveness of acquired matrial (to counter presentation attacks like rubber-fingers, cut-off body parts, images or videos on displays or photos).

Strategies include randomisation (multiple acquisitions exhibit slight variances in case of living material), recordings of earlier authentications to investigate eventual evolution, multibiometrics (combination of multiple traits or different sensors), multi-factor authentication (combination of biometrics, tokens, and knowledge) or actual detection of liveness (e.g. pulse, change in skin colouring, hippus, reaction to triggers etc.).

In supervised environments there is no need for liveness detection.

Fraud can also happen in active biometrics, e.g. in case of enforced behaviour, which again can be verified separately.

# System Model of Biometric Systems



# Data Collection

The sensors output is influenced by

- the biometric trait (with various variabilities)
- the manner how the trait is being presented and the properties of the surrounding environment
- the technical characteristics of the sensor (which might differ relativ to environmental conditions)

Trait stability is negatively influenced by changing these items. For an open system, sensor characteristics and presentation manner need to be standardised at least.

# Transmission

In many biometric installations data acquisition is dislocated from data processing or storage of reference data. Thus, acquired data needs to be transmitted and in many cases data are compressed to save bandwidth. Thus, for further processing, data need to be decompressed which introduces artifacts into the data in case lossy compression techniques have been applied.

A recent field of research is the optimisation of compression techniques wrt. specific image material. Eventual transmission errors need to be taken into account by using error correction techniques (FEC).

Existing Standard: ISO/IEC 19794 Biometric data interchange formats, currently suggests JPEG2000 for (lossy) compression; previously suggested was JPEG, other formats include FBI WSQ (wavelet packet based scalar quantisation scheme), and CELP coding for speech data.

# Signal Processing

Signal processing includes preparation (and actual conduct) of the biometric data for comparison with template data in the database.

- Feature Extraction: includes preprocessing (image enhancement techniques) and segmentation, which is the identification of the biometric trait in the captured signal (e.g. detection of phases of active speech and paused speech in a recording, detection of iris texture in an image of the eye, detection of facial landmarks in face recognition etc.). Feature extraction itself is the identification and computation of trait properties which are repeatable and discriminative.

  Non-repeatable artifacts and redundant parts of the data need to be removed, thus, fundamental image and signal processing techniques are employed which are analysed in the following chapters. Feature extraction is a form of non-reversible compression, since a trait cannot be reconstructed from features alone, however, for several traits realistic raw data has been synthetisised from feature data alone (e.g. fingerprints, iris).

# Quality control

Various techniques are used to determine if the acquired signal and/or the extracted features are of sufficient quality to facilitate a sensible matching process. If this is not the case, data collection has to be repeated. This approach has improved biometric systems a lot in the last years and can be considered a crucial stage in accurate recognition systems.

There are several approaches / options:

- Signal domain quality control
  - Generic signal quality assessment (problem: good signal quality does not imply good biometric quality, e.g. off-angle images, however, the opposite is often the case)
  - Trait specific quality assessment: e.g. energy of frequency bands containing trait information is considered, e.g. fingerprints.
- Feature domain quality control: computationally more involved, only realistic if feature extraction is done somehow close to data acquisition; adds more delay to the overall process of eventual repeated data collection; advantage is that this can be done quite specifically.

# Terminology and Matching

In the authentication process, raw data are acquired, often termed probe data. After feature extraction, the resulting “sample” data is reduced in size compared to the raw data and is compared to “templates” stored in the database (also termed “gallery”), which are generated by feature extraction from raw data captured in the enrollment process.

Sample and template are of identical type to enable comparison and are often of vector-type. In case the data resulting from feature extraction is stored in a more complex mathemtical formulation (as often used in speaker or face recognition), the template is rather called “model”, as opposed to templates in e.g. iris & fingerprint.

The aim of the comparison (matching process) is a quantitative result indicating similarity, which is passed on to the decision module.

Depending on whether verification or identification is used, an authentication process consists of a single or multiple matching stages.

The expectation is to obtain small differences in case of data from a single subject and large differences in case of data from different subjects

# Storage

Two types of data can be kept after the enrollment process:

1. Templates / Models: in case of a verification system, a distributed storage system can be used – here, templates are stored on smartcards or other tokens and a central storage is not required. However, even in this case a central database is of advantage, since fraudulent cards may be identified and card duplicates can be issued without re-enrollment. In case of identification, storage should be done in structured manner (index usage or classification) to avoid costly exhaustive search (tradeoff speed vs. accuracy).
2. Raw-Data: in certain environments it is advantageous to also store raw data since these cannot be recovered from templates. This is especially useful to allow changing feature extraction due to performance or license cost problems. Since these data are far more sensitive compared to templates, this is usually done in encrypted (and compressed) manner only, and not included in the operating system (kept off-line).

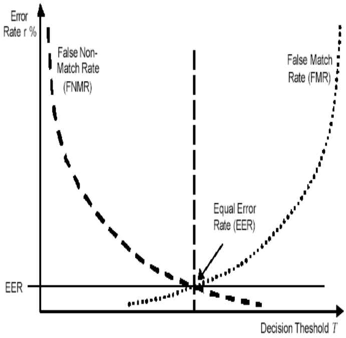
# Decision

The decision module implements the systems policy by steering database search, determins matches found based on the similarity value returned by the matching module and derives the final decision. Policy Examples: users are not authenticated in case the data quality is not high enough (instead of repeated data acquisition). Besides fixed decision thresholds, thresholds can be made dependent on environmental conditions, time, the user, time since enrollment, time since last authentication etc.

In a specific range of similarity values, more than just a single sample can be required. In case of verification, only two failed attemps might be allowed. The decision module should have access to information about the probability of fraud attempts (what is being secured with the system) and decisions determine the expected false positives vs. false negatives (what is more dangereous for the system, depending on application context of course).

# Assessment of Biometric Systems I

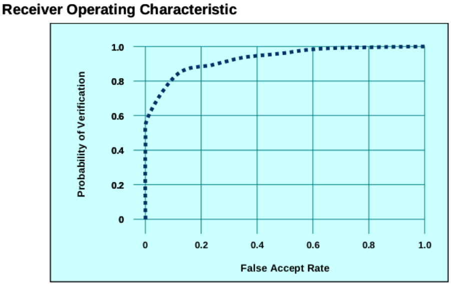
- *False Match Rate (FMR)*: fraction between found correct matches which are actually incorrect and the overall number of conducted matches (false positive verification or identification, type II error)
- *False Non-Match Rate (FNMR)*: fraction between illegitimate matches which would have been actually correct and the overall number of conducted matches (false negative verification or identification, type I error)



* FMR and FNMR are often depicted dependent on a decision threshold T (difference between sample and template accepted for a match).
* Equal-error-rate is seen at the threshold value where FMR and FNMR attain an identical value. EER is often used to assess matching accuracy without discussing its appropriateness.

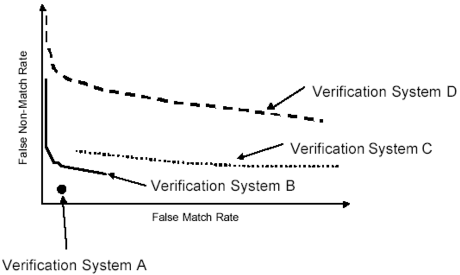
# Assessment of Biometric Systems II

*Receiver operating characteristic (ROC)*: displays the rate of false positives (impostor attempts accepted) on the x-axis as FMR or FAR against the corresponding rate of true positives (genuine attempts accepted) as e.g. probability of verification.



* In biometric systems, it is often preferred to directly relate type I and type II errors since their ranges fit better.

# Assessment of Biometric Systems III



- *Detection error trade-off curve (DET)*: displays FNMR as a function of FMR.
- Optimal value at (0,0); using DET we can compare different techniques and we can select certain techniques with desired properties (e.g. having high FNMR at low FMR).

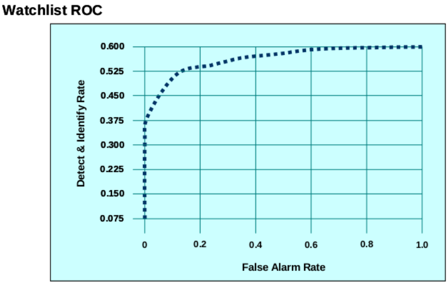
- *Binning Error Rate (BER)*: In case of processing large volumes of data, samples are classified – BER measures the percentage of samples which are assigned to an incorrect bin (class) during the matching process.

# Assessment of Biometric Systems IV

- *Penetration Coefficient (PC)*: average number of matches for each sample in relation to database size (search complexity)
- *Transaction Time (TT*): time demand for a single authentication including data collection and processing
- *FAR und FRR*: acceptance may rely on several matches or non-matches; in simple systems this is identical to FMR and FNMR
- *FIR und CIR*: in case of identification, false positives vs. correct positives divided by the number of matches
- *Threshold trade-off*: how to set the threshold is a compromise between security and usability; a low value reduces the FMR, but increases the FNMR (high intra-class variability leads to incorrectly rejected users)

# Open Set Identification

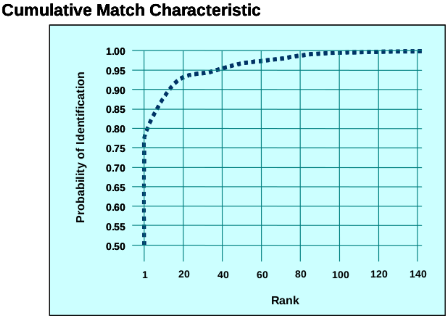
In open-set identification (sometime referred to as a watchlist application), the biometric system determines if the individuals biometric sample matches a biometric template of someone in the database (thus we do not know if the individuum is enrolled – e.g. comparing an individuum against a terrorist database).



- *(Watchlist) ROC*: displays false alarm rate (false positives) vs. detect & identify rate (true positives)

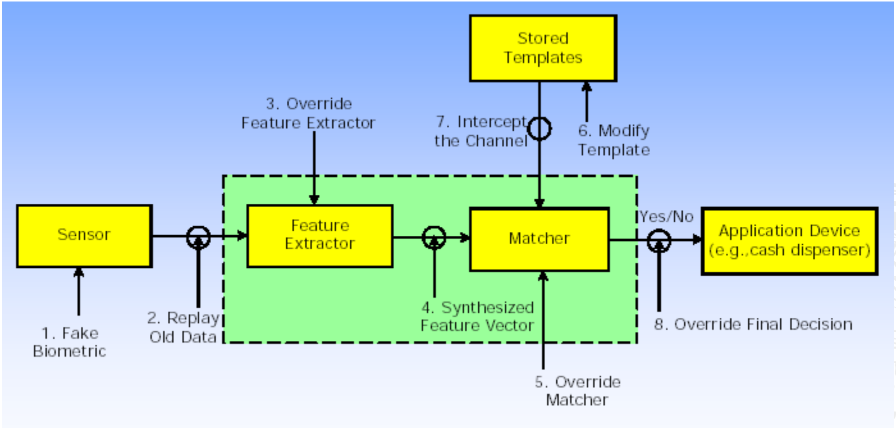
# Closed Set Identification

Closed-set identification is where every sample has a corresponding template match in the database. In practice, there are very few applications that operate under the closed-set identification task, still this scenario is often used in assessment.



- *Cumulative Match Characteristic (CMC)*: displays the probability of identification for numerous ranks – probability of correct identification at rank X means the probability that the correct match is somewhere in the top X similarity scores.

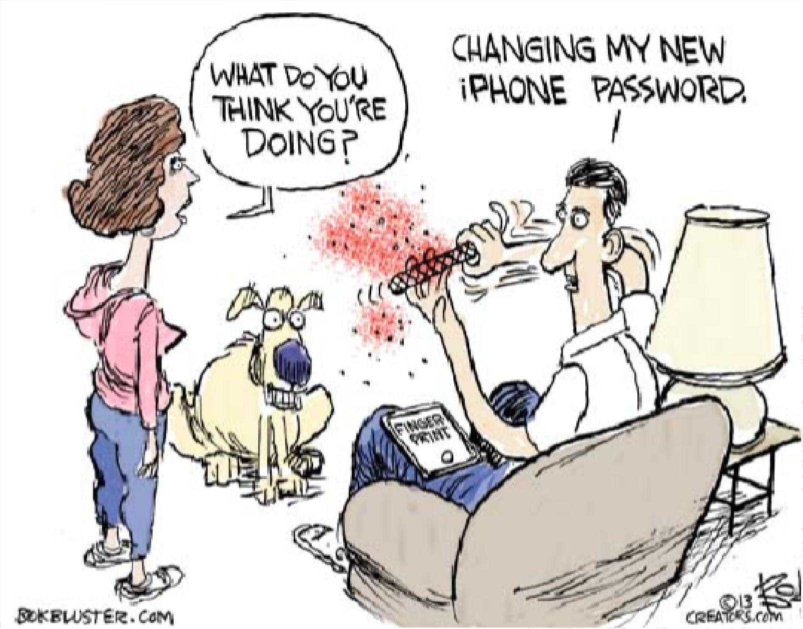
# Security of Biometric Systems



# “ Fake Biometrics” – Presentation Attack / Sensor Spoofing

1. Attacks
   - Fake biometric trait (painted contact lens, rubber finger, 3D face mask, reconstructed trait from template)
   - Image / Video replay attack: replay recorded biometric trait in front of sensor
2. Countermeasures
   - Livenenss detection (pulse, colour changes, temperature, hippus, variability in several samples, etc.)
   - Generic recapturing detection (e.g. interference frame rate / display refresh rate, presence of two optical distortions, colour changes, etc.)

# Practical Smartphone Sensor Spoofing


WHAT DO YOU THINK YOU'RE DOING?
CHANGING MY NEW iPHONE PASSWORD.
FINGER PRINT
BOKBLUSTER.COM
©13 CREATORS.COM

# “Template Security” – The threat of template abuse

While the industry has long claimed that one of the primary benefits of biometric templates is that original biometric signals acquired to enroll a data subject cannot be reconstructed from stored tem- plates, several approaches (in fingerprint, iris, finger vein recognition) have proven this claim wrong. Raw data can be reconstructed from biometric templates allowing for verification and identification, albeit not being identical to the raw data the template has been derived from.

Raw data corresponding to several traits cannot even be assumed to be private, e.g. face images, iris images, (latent) fingerprints.

Since biometric characteristics are largely immutable, a compromise of a biometric trait or of biometric templates results in permanent loss of a subjects biometrics. Thus, this fact endangers secure and sustainable use of biometric authentication systems.

# “Template Security” – Countering Template Compromise

Standard encryption algorithms do not support a comparison of biometric templates in encrypted domain and, thus, leave biometric templates exposed during every authentication attempt.
Three approaches can be found in literature how to cope with this problem (where the third only covers template protection):

## Template Security Schemes

1. Biometric Template Protection
   1. Biometric Cryptosystems
   2. Cancelable Biometrics
2. Matching in Encrypted Domains – (Partially) Homomorphic Encryption, Garbled Circuits

# Irreversibility and Unlinkability

Biometric template protection schemes are designed to meet two requirements of biometric information protection:

- *Irreversibility*: It should be computationally hard to reconstruct the original biometric template from the stored reference data, i.e., the protected template, while it should be easy to generate the protected biometric template.
- *Unlinkability*: Different versions of protected bio- metric templates can be generated based on the same biometric data (renewability – revocability), while protected templates should not allow cross-matching (diversity).

# Biometric Cryptosystems

Biometric cryptosystems (BCSs) are designed to securely bind a digital key to a biometric or generate a digital key from a biometric trait offering solutions to biometric-dependent key-release and biometric template protection.

BCSs are designed to output stable keys which are required to match a 100% at authentication. Original biometric templates are replaced through biometric- dependent public information (termed “helper data”) which assists the key- release process.

Two variants:

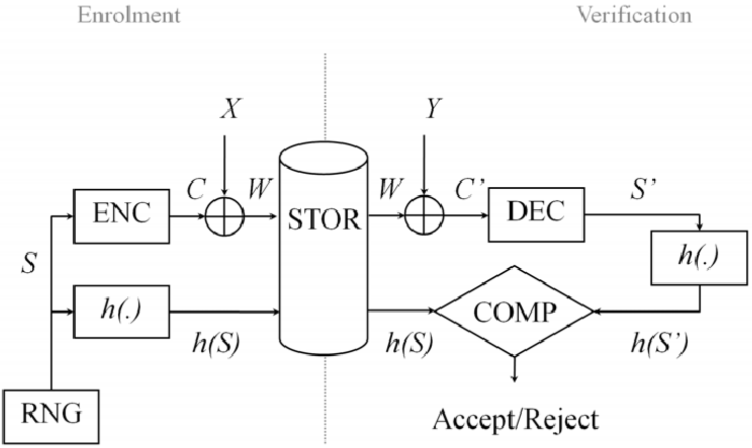
1. Key-binding schemes
2. Key-generation schemes

# Key-binding vs. Key-generation

- *Key-binding schemes*: Helper data are obtained by binding a chosen key to a biometric template. As a result of the binding process a fusion of the secret key and the biometric template is stored as helper data. Applying an appropriate key retrieval algorithm involving a biometric sample, keys are obtained from the helper data at authentication. Since cryptographic keys are independent of biometric features these are revocable
- *Key-generation schemes*: Helper data are derived only from the biometric template. Keys are directly generated from the helper data and a given biometric sample.

# Key-binding: Fuzzy Commitment Scheme (FCS)

Key-binding scheme for binary templates which combines techniques from the area of error correcting codes (ECC) and cryptography. ECC are used to cope with intra-class variability.



# Operation Mode of FCS

During enrollment, a random number $S$ (the key) is generated, and the hash $h(S)$ is stored. S is processed by the encoder ($ENC$) of an ECC to obtain the codeword $C$. The binary biometric template $X$ is combined with $C$ to result in the helper data $W = X \oplus C$ which is also stored. Thus, hte input to the FCS is $X$ and the output is $(W, h(S))$.

During verification, a new biometric sample is captured and the template $Y$ generated. It is combined with $W$ to obtain a candidate codeword $C' = W \oplus Y = C \oplus (X \oplus Y)$. $C'$ is fed into an ECC decoder $DEC$ to obtain a candidate key $S'$. If $h(S') = h(S)$, the obtained key is correct and $X$ and $Y$ have been generated from the identical biometric trait.

# Issues with FCS

- Biometric trait needs to be represented by a binary string and the scheme will work only if the capacity of the ECC is sufficient to cope with the traits intra-class variability (different ECC can be combined to account for different error sources).
- Length of the key (plus ECC) is bound by the length of the biometric template due to the XOR operation.
- Since $W$ is stored, is it secure ? $X$ is obfuscated by the randomness of the codeword $C$ mimicking a OTP construction. The opposite is the case as well. Obviously, the inherent redundancy in $C$ and $X$ implies that the one-time pad is not perfect, and consequently leaks some information.

# Key-generation: Quantisation

Helper data are constructed in a way that they assist in a quantization of biometric features in order to obtain stable keys.

During enrollment, several enrollment samples are analysed and appropriate quantisation intervals (resulting in identical values after quantisation) for each feature element are derived (real-valued feature vectors are required, intervals are defined based on feature variance). These intervals are encoded and stored as helper data.

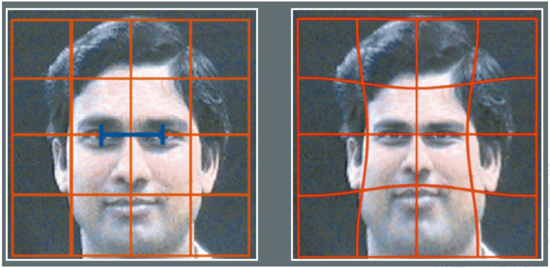
During authentication, biometric characteristics of a subject are measured and mapped into the previously defined intervals, generating a hash or key. In order to provide updateable keys or hashes, most schemes provide a parameterized encoding of intervals.

# Cancelable Biometrics I

Cancelable biometrics (CB) consist of intentional, repeatable distortions of biometric signals or features based on repeatable transforms applied in identical manner during enrollment and authentication which facilitate a comparison of biometric templates in the transformed domain [**?**]. In case of compromise, only the transform needs to be changed. For different biometric installations, different transforms are used.
The inversion of such transformed biometric templates must not be feasible for potential imposters. In contrast to templates protected by standard encryption algorithms, transformed templates are never decrypted since the comparison of biometric templates is performed in transformed space which is the very essence of CB. The application of transforms provides irreversibility and unlinkability of biometric templates.

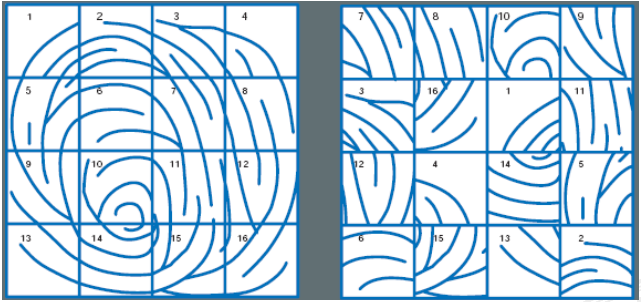
# Cancelable Biometrics II

Important: In case the transform is applied to the raw data, feature extraction still needs to be capable to work properly !



Challenges: Maintain recognition accuracy under distortion & provide reasonable keyspace size (what happens for “close” keys ?).

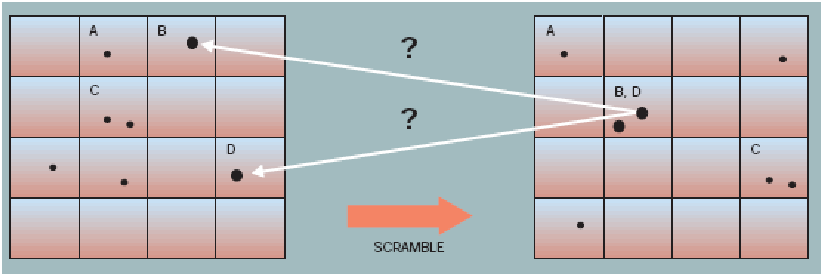
# Cancelable Biometrics III



Distortions must be generated by non-invertible transforms, thus, even in case of leakage of the transform parameters, the raw signal data cannot be reconstructed.

Examples for application at signal level: image morphing (non-invertability due to interpolation) and block-permutation (is invertible and even without transform parameter leakage the origignal signal might be recovered).

# Cancelable Biometrics IV

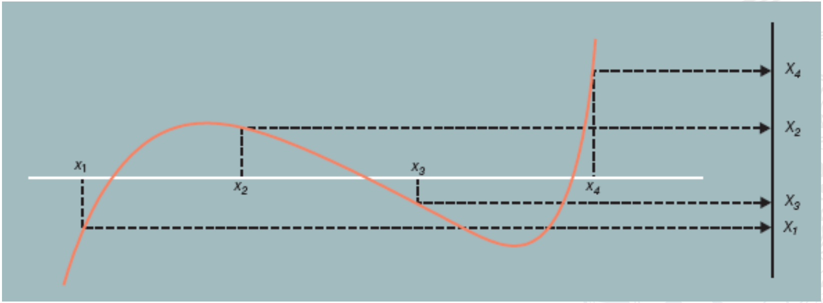


In the figure, a feature domain transformation for fingerprints is displayed, where in addition to the block permutation several blocks are mapped onto a single one. In case there are no feature (i.e. minutiae) overlappings, features are kept (and therefore recognition performance is maintained) while the transform is not invertible.
To result in a repeatbale transformation, the biometric signal and the corresponding features need to be registered – for fingerprints, this can be facilitated based on singular points like “cores” and “deltas”.

# Cancelable Biometrics V

A further example for a non-invertable transform in the feature domain, applicable to point-clouds (like fingerprints, vessel crossings etc.) is given as follows. A set of minutiae $S$ consists of $S = \{(x_i, y_i, \phi_i), i = 1, \ldots, M\}$. A non-invertable function of the x-coordinate is a polynomial of higher order:

$$F(x_i) = \sum_{n=0}^{N} \alpha_n x_i^n = \prod_{n=0}^{N} (x_i - \beta_n)$$



# (Partially) Homomorphic Encryption

The **Paillier encryption** scheme allows two operations in the encrypted domain due to its additively homomorphic property. For any messages $m_1, m_2 \in \mathbb{Z}_n$:

$$\mathcal{D}_P(\mathcal{E}_P(m_1) \cdot \mathcal{E}_P(m_2) \bmod n^2) = m_1 + m_2 \bmod n$$

$$\mathcal{D}_P(\mathcal{E}_P(m_1)^{m_2} \bmod n^2) = m_1 \cdot m_2 \bmod n$$

The additively homomorphic property of the **Goldwasser-Micali encryption** scheme is for any $m_1, m_2 \in \{0, 1\}$ ($\oplus$ denoting *xor*)

$$\mathcal{D}_{GM}(\mathcal{E}_{GM}(m_1) \ \cdot \ \mathcal{E}_{GM}(m_2)) = m_1 \oplus m_2$$

In other words, if $c_1$ and $c_2$ are the encrypted values of $m_1$ and $m_2$, $(c_1 \cdot c_2)$ mod $n$ will be an encryption of $m_1 \oplus m_2$.

# Paillier Encryption

For key generation, $\lambda = lcm(p-1, q-1)$ ($lcm =$ least common multiple) is computed and a random integer $g \in \mathbb{Z}^*_{n^2}$ is selected, resulting in the public/private key pair: $pk_P$: $(n, g)$ and $sk_P$: $(\lambda)$. For encryption, we take a message $m$ with $m \in \mathbb{Z}_n$ and select a random integer $r \in \mathbb{Z}_n$ (the latter provides semantic security and is not required for decryption).

$$c = g^m \cdot r^n \bmod n^2 \ .$$

$$m = \frac{L(c^\lambda \bmod n^2)}{L(g^\lambda \bmod n^2)} \bmod n,$$

where $L(u) = \frac{u-1}{n}$.

# What about RSA ?

Recall that we have

$$c = m^e \bmod n \text{ and } m = c^d \bmod n \text{ with } e \cdot d = 1 \bmod (p-1)(q-1)\ .$$

Obivously, for messages $m_1$ and $m_2$ we get a multiplicative homomorphism since $m_1^e \cdot m_2^e = (m_1 \cdot m_2)^e \bmod n$, thus,

$$\mathcal{D}_P(\mathcal{E}_P(m_1) \cdot \mathcal{E}_P(m_2) \bmod n) = m_1 \cdot m_2 \bmod n\ .$$

However, exactly this nice property is a problem in an adaptive chosen ciphertext attack (CCA) (RSA is said not to be *semantically* secure): To decrypt $c = m^e \bmod n$, compute $c' = c \cdot 2^e \bmod n$, decrypt $c'$ by computing $((m \cdot 2)^e)^d$ which results in having $m$ revealed.

# Comparing Binary Templates with Homomorphic Encryption I

To compare two binary biometric templates $m_1$ and $m_2$ we usually calculate the Hamming distance $h$ of the two binary strings by *xor*-ing $m' = m_1 \oplus m_2$ and then computing the Hamming Weight *HW* of the resulting string (by essentially counting the 1 bits in $m'$): $h = \mathcal{HW}(m')$.

For the Goldwasser-Micali encryption scheme we can directly exploit its homomorphic property. For the Paillier cryptosystem however, calculating the *xor* of two encrypted binary strings is not as trivial. Thus we will have to look at the process of *xor*-ing two bit-strings more closely.

Let $m_1 = (m_1[1] \dots m_1[k]), m_2 = (m_2[1] \dots m_2[k])$ be two binary strings of length $k$. Then

$$m_1 \oplus m_2 = m_1[i] + m_2[i] - 2m_1[i]m_2[i], i = 1, \dots, k\,.$$

# Comparing Binary Templates with Homomorphic Encryption II

As a consequence, we have to use bit-by-bit encryption of the Paillier scheme and can finally perform the encrypted *xor* as follows:

$$\tilde{m_2}[i] = -2m_2[i] \bmod n$$

$$\mathcal{E}_P(m_1[i] \oplus m_2[i]) = \mathcal{E}_P(m_1[i]) \cdot \mathcal{E}_P(m_2[i]) \cdot (\mathcal{E}_P(m_1[i]))^{\tilde{m_2}[i]} \bmod n^2$$

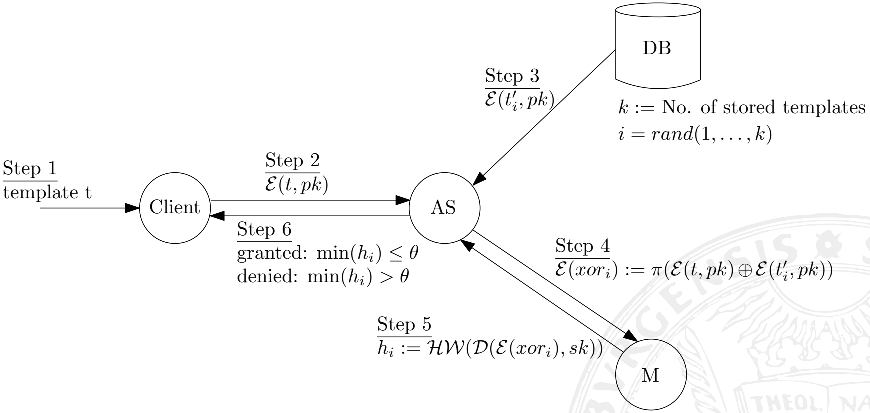
where $n$ is part of the public key for the Paillier cryptosystem and all encryption steps also use the public key.

However, encrypting only a single bit is very inefficient as the Paillier scheme is designed to encrypt messages of length $m < n$. Also, the Goldwasser-Micali is not really efficient since it requires a modular multiplication per bit. Another issue is that the Hamming Weight (the actual difference between two binary templates) can not be computed in the encrypted domain using additively homomorphic encryption.

# System Architecture for Homomorphic Biometric Identification I

1. In enrollment, templates of each individual to be recognised by the system are created, encrypted, and stored in the database (eventually, when using Paillier encryption, also the plaintext template must be stored in the database (as required for the encrypted *xor*($\oplus$) operation).
2. In authentication, the client extracts a template from the sample of the individual requesting authentication.
3. Template is encrypted with the public key at the client and transfered to the server component (authentication server *AS*, database server *DB* storing encrypted reference templates, matcher *M*)
4. *AS* fetches all enrolled templates $t_i'$, $i = 1, \ldots, k$ in random order from the *DB* and *xor*-s against the encrypted template $t$.
5. Calculation of the Hamming weight cannot be accomplished in the encrypted domain using partially homomorphic encryption techniques – *M* computes the Hamming weight using the private key. To prevent *M* from learning about template relations strings are permuted (maintaining Hamming weight).

# System Architecture for Homomorphic Biometric Identification II



DB
Step 3
$\mathcal{E}(t'_i, pk)$
$k :=$ No. of stored templates
$i = rand(1, \ldots, k)$
Step 1
template t
Client
Step 2
$\mathcal{E}(t, pk)$
AS
Step 6
granted: $\min(h_i) \leq \theta$
denied: $\min(h_i) > \theta$
Step 4
$\mathcal{E}(xor_i) := \pi(\mathcal{E}(t, pk) \oplus \mathcal{E}(t'_i, pk))$
Step 5
$h_i := \mathcal{HW}(\mathcal{D}(\mathcal{E}(xor_i), sk))$
M

Figure: Identification architecture

# Privacy in Biometric Systems

In a very specific understanding, privacy is understood of extracting information from data that has been acquired in legitimate manner, for which it have not been intended to. Thus, what about biometric data that is provided to an authentication system ?

- Potential derivation of personal information of the template owner ? E.g. retina – retina diagnosis, diabetes; speech – gender; writing – personal characteristics; gait – injuries; heart and other biorhythms – physical and mental diseases; face – age;
- Cross matching biometric databases facilitates to track peoples itinerary like this can be done with credit card information and telephone calls; however, contrasting to account information or telefone numbers there are hardly large databases mapping templates to names, especially since the type of templates stored often differs. In any case, BCS and CB help here !
- Public availability of some traits (face, finger) makes the biometric information similar to public-keys, but is harder to change once compromised – again, BCS and CB help here !

# Outline

1. Formalia
2. Introduction & Motivation
3. Foundations of Biometrics
   - Introduction and Basic Terminology
   - Biometric Traits & their Properties
   - Biometric Systems System Model
   - Assessment of Biometric Systems
   - Security and Privacy in Biometric Systems
4. Non-visual Biometrics
   - Keystroke Dynamics
   - On-Line Signatures
   - Biorhythms
   - Speaker Recognition
   - Odour
5. Visual Biometrics
   - Off-line Signatures
   - Iris Recognition
   - Fingerprint Recognition

# Keystroke Dynamics: Basics

Pros: No specific additional sensor technology is required since keyboards are available in almost any household, any office etc., however, with decreasing spread due to the proliferation of mobile devices (distributed sensor architecture, high collectability, low price, except the pressure level should be recorded). Accetability is expected to be very high, as many people use keyboards on a daily basis (habituated environmnent). Data produced (which key was pressed, for how long, when released) can be transmitted, stored and processed with low effort.

Cons: Intra-personal variability is expected to be high (person might be tired, hurt, might be standing - especially people not well trained in typing will be affected), the distributed and unattended architectures additionally causes problems (e.g. different keyboard layouts lead to different typing pattern).

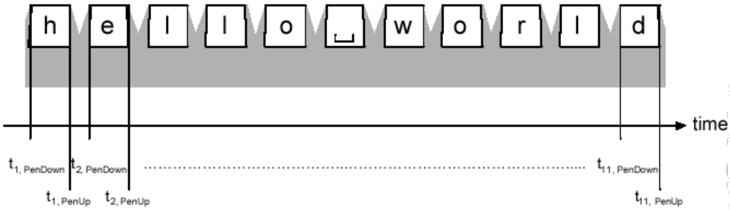
# Keystroke Dynamics: Login vs Monitoring

Login: is the classical “key-stroke enhanced” login scenario. A user types Login and PWD (or only Login), additionally typing pattern is recorded. Login is only granted in case of sufficiently close typing pattern (example for two-factor authentication: knowledge **and** behaviour). For enrollment, Login and PWD need to be entered several times. The amount of data used to characterise typing behaviour is relatively small, however, the fixed text makes things easier.

Monitoring: is constantly being used during a session (“contineous authentication”). While a user is working at the keyboard, a background process collects data and assesses typing behaviour. In case of doubt concerning the identity, the user might get automatically logged out or the administrator gets a warning. The amount of data used is much larger, however, the system is faced with arbitrary text.

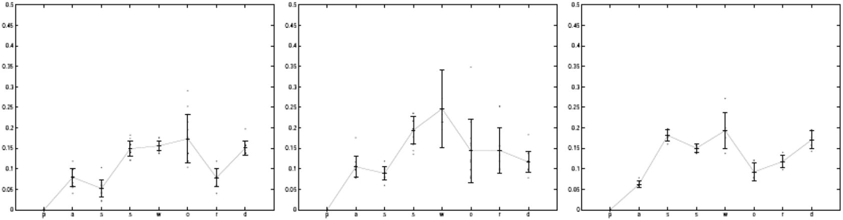
In literature this is sometimes referered to as the difference between verification and identification, in fact both cases are verification indeed !

# Keystroke Dynamics: Setup



# Keystroke Dynamics: Latency

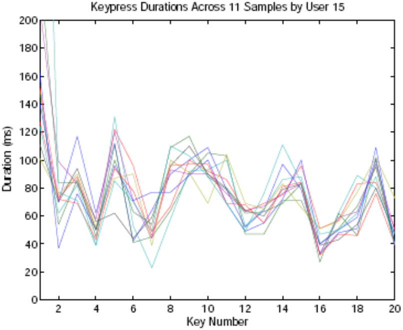
This is **the** classical keystroke feature. In the Login-scenario, the element *i* of the feature vector is $t_{i+1,PenDown} - t_{i,PenDown}$. Classical metrics can be applied to those vectors. It has to be noted that $t_{i,PenUp}$ do not necessarily correspond to the logical order of the letters (since a key may be released at various points in time), also $t_{i,PenUp} < t_{i+1,PenDown}$ is not guaranteed.

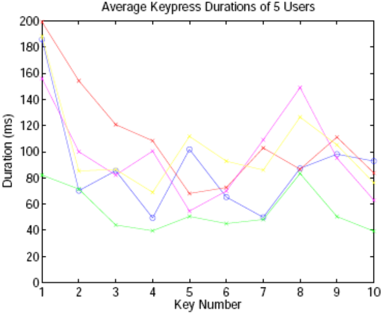


For monitoring, latency is collected for a set of letter-pairs and compared among users.

# Keystroke Dynamics: Pressure Duration

The element *i* of the feature vector is $t_{i,PenUp} - t_{i,PenDown}$ – this feature is often used in a fusion setting with latency information. Similar considerations wrt. Login and monitoring do apply.





# Keystroke Dynamics: Relative Keyevent Order

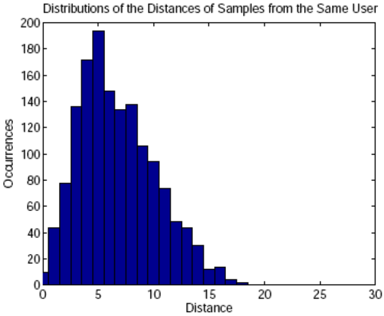
The relative order in which users press and release keys (i.e. keyevents) can vary greatly from user to user, especially while typing words or phrases in which each user has a more established typing pattern.

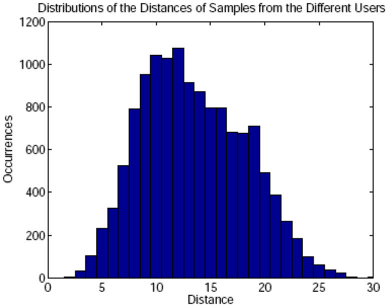
Given two samples a and b, the distance between sample a and sample b equals the number of key events that are swapped between the two samples (e.g. user types “h-i” by pressing h, releasing h, pressing i, and releasing i, while the second user presses h, presses i, releases h, releases i, the distance between the two samples is 1).

To find the distance between two trials, we first order the key events of each trial by time. Second, we find the sum of the absolute value of the difference in the positions for each key event in the two trials. Since every swap will cause two key events to be out of position, divide this sum by two to find the distance between the trials. Two trials with distance equal to one would feature very similar key orderings. A distance equal to the total number of letters typed would indicate a very different typing pattern.

# Keystroke Dynamics: Relative Keyevent Order Example

Resulting genuine and impostor score distributions do not at all exhibit a clear separation.

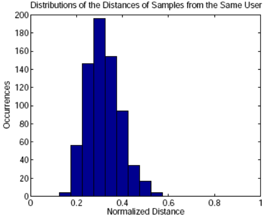


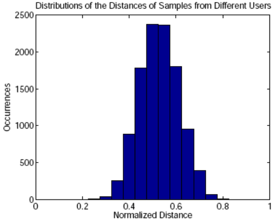


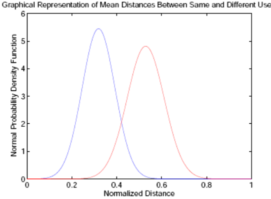
For this approach it turns out to be of advantage to use user-specific decision thresholds.

# Keystroke Dynamics: Relative Typing Speed

While intra-personal variability is very high wrt. overall typing speed, relative speed of various keyevents might be more reproducible (relative keystroke speed). To compute the distance between two typing samples $S$ and $S'$, we represent each sample as a vector of key pairs, filter out all key pairs that are not shared between the samples (to handle backspaces and deleted characters), and then sort the remaining pairs in each sample based on their latencies (the press time of the second key minus the release time of the first key).



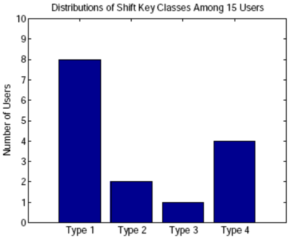




Letting $S[i]$ denote the location of key pair i in the sorted sample S, we compute the distance between $S$ and $S'$ as: $\sum_i |S[i] - S'[i]|$

# Keystroke Dynamics: Shift Key Usage

For each key, either the right or left Shift key may be employed. This can be used for habituated users as an additional feature. 4 classes of shift key users can be hypthesized: strictly right or left shift users, opposite-shift users, and chaotic shift users.



- Reality: Strict left and right shift key users do exist.
- Typ 3: both sides and fixed for all keys, but not opposite; Typ 4: only for a majority of keys shift usage is fixed.

# Keystroke Dynamics Demos

- `https://password.keytrac.net/en/tryout`
- `http://fingerprint.tappy.pw/`
- `http://www.keystroke-dynamics.com/`

# Keystroke Dynamics: Privacy

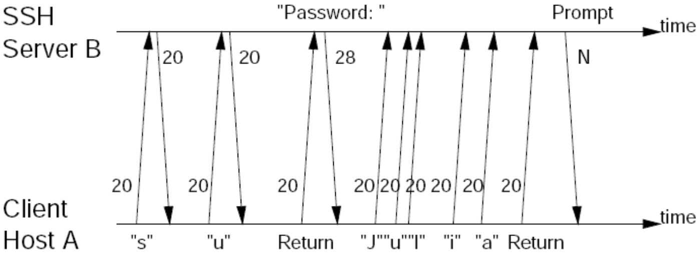
Fundamental problem: in case of availability of a typing profile, the content of a message can be reconstructed based on the data of typing behaviour only. This can be used for attacks, either by using the prifile of a specific user (to actually reconstruct messages) or by using typical average user behaviour to reduce the number of possible key-combinations in a brute force attack (accuracy is lower in this case of course).

Example SSH: In interactive mode individual keystrokes are transmitted in separate IP-packets immediately after the key is pressed. Actually the initial login to a remote site using SSH does not leak timing information to the network, because the initial login sends the whole password in one packet. The timing information is leaked when an established SSH connection is used, for example, to change to super-user account and writing the super-users password.

Now how can we exploit this for an attack ?

# Keystroke Dynamics: Attack against SSH I

First, the IP-packets and their timings when writing passwords have to be recognized from the network.



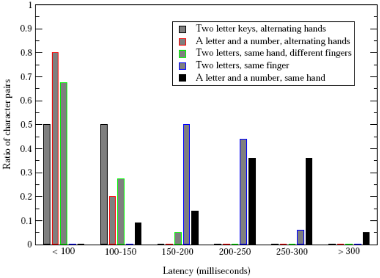
All the normal keystrokes sent to the SSH-server generate a returning packet because the character is echoed to the screen, but when writing a password characters are not echoed and consequently packets are send only to one direction, from the client to the server.

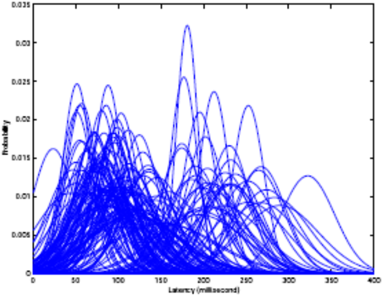
# Keystroke Dynamics: Attack against SSH II

Countermeasures: sending dummy data when the PWD is entered (to harden the detection of the PWD sequence) or adding additional delay when sending the packets (which would have to be so long to probably impact on user experience).

For conducting such an attack, there has to be information on what kind of latencies are to be expected between different keys. In an experiment 142 key pairs are considered and grouped into several categories based on how they are typed (alternating hands, etc.). This information already reduces search space in a brute force attack.

# Keystroke Dynamics: Attack against SSH III



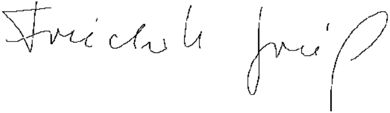


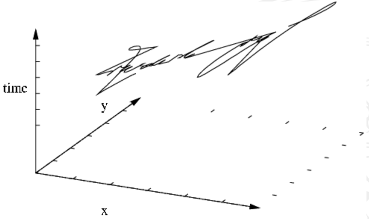
A Gaussian model was derived for all key pairs, and using this information a brute force attack against the SSH su PWD could be reduced to 2.7% of the PWD space (modelling was done using a Hidden Markov Model: character pairs are the hidden states while the latencies are the observed output; the Viterbi algorithm is used to compute the most likely sequence of states from given latencies – see Online Signatures !).

# On-Line Signatures

Off-Line Signaturen: The biometric trait is the result of signing, usually a digital image. Image processing techniques can be used to determine correspondences (see respective parts of these lecture notes).

On-Line Signatures: The biometric trait is the temporal dynamics of signing, usually a time-dependent function. Techniques from time-series analysis are applied to compare two samples.





# Pros & Cons of Signatures in Biometrics

Pros: As signing a document is still a widely used and accepted means of authentication in the analog domain, signing can be seen as a habituated action. Also, more and more tablets, PDAs etc. with touch sensitive screens are available, thus, required sensors for on-line signatures are available – collectability is given (for off-line signatures this is true in any case due to availability of scanners).
Cons: Signatures exhibit a high intra-personal variability, especially due to diseases occuring in ageing like Parkinson. This makes it specifically difficult to recognise professional forgeries, which are usually easier to generate compared to other modalities (which is a strong argument for on-line signatures). Also, the argument of habituated execution is only valid for analog media (and thus for off-line signatures).

# Techniques closely related

- Text dependent vs. text independent: E.g. handwriting recognition relies on a character-based recognition but is much more difficult since a direct comparison of time-series and visual data is not possible. On the other hand, the production of forgeries is much more difficult (e.g. if in case of authentication an arbitrary text needs to be written).
- Character recognition: OCR – does not depend on the text and the semantics is important (so it is more difficult), however, no forgeries are to be expected. Temporal dynamics are of no relevance and if present re-sampling is applied to correct it.
- Signature recognition is implicitely a verification since the name is given – can however only be expoited by pre-processing using OCR.
- Related variants are drawing of symbols or writing of PWDs (habituation does not hold true and we have a two-factor authentication).

# On-line vs. Off-line Signatures I

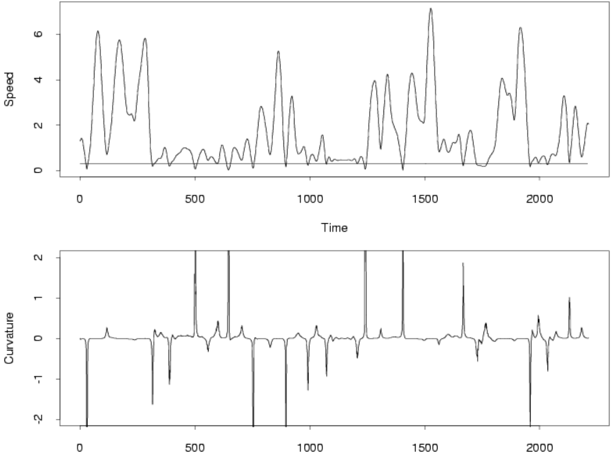
On-line signatures capture the temporal dynamics of the signature. Wrt. forgeries, the more accurate the visual appearance of the signature is forged, more less similar is the dynamics to an actual signature. The exception is of course if the attacker learns the signature dynamics, but this causes fairly high effort.

This slightly questions the high success of on-line signatures wrt. detecting forgeries - as soon as the dynamics is subject to forging, results do drastically change!

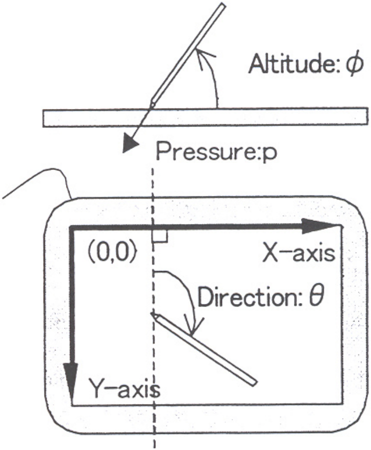
Many techniques for on-line signature recognition can be implemented with off-line features as well, since they do not really rely on the temporal dynamics exclusively. For example, in case we use x-coordinate, y-coordinate, curvatures etc. (local geometric properties) as features, this can also be facilitated by an ordering wrt. length of the signature curve (e.g. pixel count) instead of temporal information (this corresponds to an re-sampling of the data). Thus, such techniques are not intrinsically on-line and features can be generated from the off-line signature.

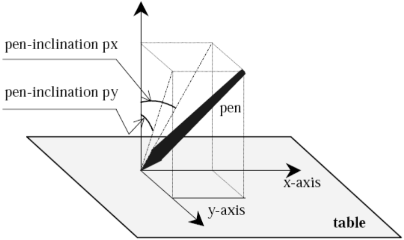
# On-line vs. Off-line Signatures II

In many cases, dynamical on-line features are highly correlated with shape-based (off-line) features, like for speed and curvature as shown in the example (which questions the requirement for on-line data recording if such features are used).



# On-line Signature Feature Recording





In addition to the (off-line) signature image, we can exploit the following recordings: $x(t)$, $y(t)$, $p(t)$, $\phi(t)$, und $\Theta(t)$.

These features can be recorded by sensors in tablets or suited pens. Example: WACOM ArtPad 2 pro Serial.

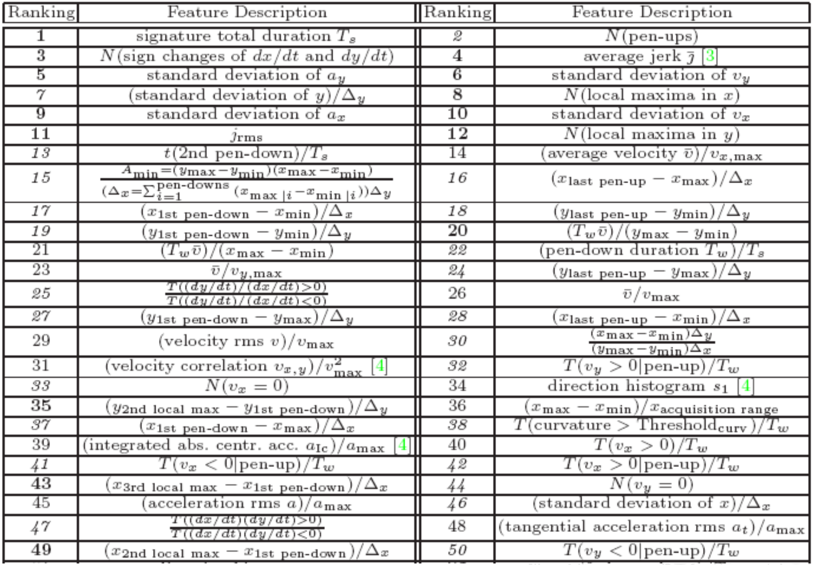
# Global vs. local On-line Features

Global features are computed from the entire signature, while local features represent time-dependent properties at a given time-position. Local features are used more often, since here time-series analysis technqiues may be applied.

Examples for global on-line features

- Overall signature duration, pen-down ratio (pen-down or stroke time divided by overll duration), numner of strokes
- Based on speed and acceleration: $v_x = \frac{dx}{dt}$, $v_y = \frac{dy}{dt}$, $a_x = \frac{dv_x}{dt}$, $a_y = \frac{dv_y}{dt}$. Velocity $v$ is defined as $v = (v_x^2 + v_y^2)^{1/2}$. From these quantities, average speed and acceleration, corresponding variances and histograms are used as well as correlation between $v_x$ and $v_y$.
- Histograms of $\phi(t)$, $\Theta(t)$ and $p(t)$.
- Due to correlations among shape-based and dynamical features as discussed, it is important to select combinations without these correlations, in case both types are used.

# Global On-line Features



| Ranking | Feature Description | Ranking | Feature Description |
|---|---|---|---|
| 1 | signature total duration $T_s$ | 2 | $N$(pen-ups) |
| 3 | $N$(sign changes of $dx/dt$ and $dy/dt$) | 4 | average jerk $\bar{j}$ [3] |
| 5 | standard deviation of $a_y$ | 6 | standard deviation of $v_y$ |
| 7 | (standard deviation of $y)/\Delta_y$ | 8 | $N$(local maxima in $x$) |
| 9 | standard deviation of $a_x$ | 10 | standard deviation of $v_x$ |
| 11 | $j_{\mathrm{rms}}$ | 12 | $N$(local maxima in $y$) |
| 13 | $t$(2nd pen-down)$/T_s$ | 14 | (average velocity $\bar{v})/v_{x,\max}$ |
| 15 | $\frac{A_{\min}=(y_{\max}-y_{\min})(x_{\max}-x_{\min})}{(\Delta_x=\sum_{i=1}^{\text{pen-downs}}(x_{\max\,\mid i}-x_{\min\,\mid i}))\Delta_y}$ | 16 | $(x_{\text{last pen-up}} - x_{\max})/\Delta_x$ |
| 17 | $(x_{\text{1st pen-down}} - x_{\min})/\Delta_x$ | 18 | $(y_{\text{last pen-up}} - y_{\min})/\Delta_y$ |
| 19 | $(y_{\text{1st pen-down}} - y_{\min})/\Delta_y$ | 20 | $(T_w\bar{v})/(y_{\max} - y_{\min})$ |
| 21 | $(T_w\bar{v})/(x_{\max} - x_{\min})$ | 22 | (pen-down duration $T_w)/T_s$ |
| 23 | $\bar{v}/v_{y,\max}$ | 24 | $(y_{\text{last pen-up}} - y_{\max})/\Delta_y$ |
| 25 | $\frac{T((dy/dt)/(dx/dt)>0)}{T((dy/dt)/(dx/dt)<0)}$ | 26 | $\bar{v}/v_{\max}$ |
| 27 | $(y_{\text{1st pen-down}} - y_{\max})/\Delta_y$ | 28 | $(x_{\text{last pen-up}} - x_{\min})/\Delta_x$ |
| 29 | (velocity rms $v)/v_{\max}$ | 30 | $\frac{(x_{\max}-x_{\min})\Delta_y}{(y_{\max}-y_{\min})\Delta_x}$ |
| 31 | (velocity correlation $v_{x,y})/v_{\max}^2$ [4] | 32 | $T(v_y > 0 \mid \text{pen-up})/T_w$ |
| 33 | $N(v_x = 0)$ | 34 | direction histogram $s_1$ [4] |
| 35 | $(y_{\text{2nd local max}} - y_{\text{1st pen-down}})/\Delta_y$ | 36 | $(x_{\max} - x_{\min})/x_{\text{acquisition range}}$ |
| 37 | $(x_{\text{1st pen-down}} - x_{\max})/\Delta_x$ | 38 | $T(\text{curvature} > \text{Threshold}_{\text{curv}})/T_w$ |
| 39 | (integrated abs. centr. acc. $a_{Ic})/a_{\max}$ [4] | 40 | $T(v_x > 0)/T_w$ |
| 41 | $T(v_x < 0 \mid \text{pen-up})/T_w$ | 42 | $T(v_x > 0 \mid \text{pen-up})/T_w$ |
| 43 | $(x_{\text{3rd local max}} - x_{\text{1st pen-down}})/\Delta_x$ | 44 | $N(v_y = 0)$ |
| 45 | (acceleration rms $a)/a_{\max}$ | 46 | (standard deviation of $x)/\Delta_x$ |
| 47 | $\frac{T((dx/dt)(dy/dt)>0)}{T((dx/dt)(dy/dt)<0)}$ | 48 | (tangential acceleration rms $a_t)/a_{\max}$ |
| 49 | $(x_{\text{2nd local max}} - x_{\text{1st pen-down}})/\Delta_x$ | 50 | $T(v_y < 0 \mid \text{pen-up})/T_w$ |

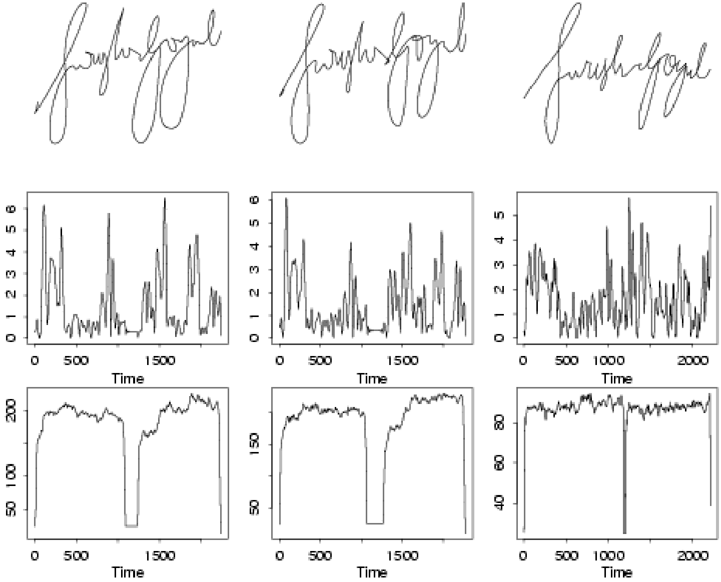
# Local On-line Features

In this class, time-dependent functions are classically considered: $x(t)$, $y(t)$, $p(t)$, $\phi(t)$, $\Theta(t)$, $v_x(t)$, $v_y(t)$, $a_x(t)$, $a_y(t)$.
Further features include:

- Path tangent angle $\Phi(t) = tan^{-1}(\frac{v_y(t)}{v_x(t)})$ or the angle $\alpha(t)$ between the connection line of two points on the signature at time $t$ and $t + 1$ and the x-axis. The concept is closely related to curvature, but is considered in time-dependent manner.
- Pen movement $V(t)$ as three-dimensional vector with $V_x(t) = sin\Theta(t)cos\phi(t)$, $V_y(t) = -cos\Theta(t)cos\phi(t)$, and $V_z(t) = sin\phi(t)$.

In order to facilitate a sensible comparison of these time-series, preprocessing these data is required: **Resampling, smoothing, opimal alignment - dynamic time warp DTW**.

# Example: Local On-line Features Speed and pressure



# Local On-line Features: Preprocessing

Classical procedures include position normalisation as well as size normalisation. Since these are also required for off-line features, some remarks follow in the corresponding chapter.

In order to assess shape features (off-line features), dynamical data need to undergo uniform re-sampling to mitigate temporal effects, while this is not desired for on-line features.

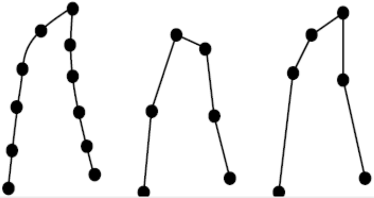
Smoothing helps to decrease or remove sampling noise, too much smoothing may also be counterproductive (see jitter !)

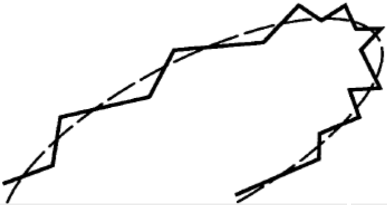
Strokes: some techniques connect strokes to a closed curve, in some techniques this is only done conditionally, to differentiate virtual pen-ups (there is only not enough pressure) from real ones (e.g.: AB and CD are the end points of two strokes which are connected only in case the direction of vector BC is in-between the directions of AB and CD.

# Local On-Line Features: Resampling & Jitter

Resampling can be required in case the sampling rate is too high (computational effort) or it is required to have a specified number of sampling points in a comparison. When doing that, it needs to be ensured that no important ("critical points") or perceptionally relevant points are removed. This can be avoided by enforcing $\left|\frac{y(t_i)-y(t_{i-1})}{x(t_i)-x(t_{i-1})}\right| \leq T$, otherwise a different sampling strategy is chosen.

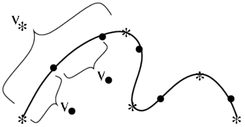
Smoothing is usually meant to remove sampling or quantisation noise, while it also impacts "jitter" which indicates a potential forgery. Jitter results from the attacker's attempt to follow the signature curve at small scale while applying multiple small direction adjustments. Signatures must be checked for jitter to identify eventual forgeries as early as possible.





# Critical Points and Signature Segmentation

Signatures exhibit a number of specifically important points which are important for the appearance and the dynamics of the signatures, so-called critical points (special points wrt. appearence) or extremal points (local maxima in a time-dependent function, e.g. curvature). These points are often used for recognition by considering properties of the signature curve between two of these points (e.g. speed) or the size / position / property of the neighbourhodd of these points. Determination of these points delivers a signature segmentation (simplest segmentation: stroke-based), where one feature per segment provides a short and (hopefully) descriptive feature vector.



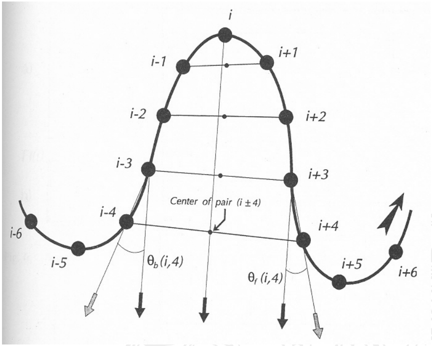
$*$ = critical point $v_*$ = speed between two critical points

● = sampling point $v_\bullet$ = speed between two sampling points

# Finding Critical Points I

For each point on the signature curve $i$ we determine, if its nieghbouring points $i+-n$ (point pairs) are in its zone of influence.

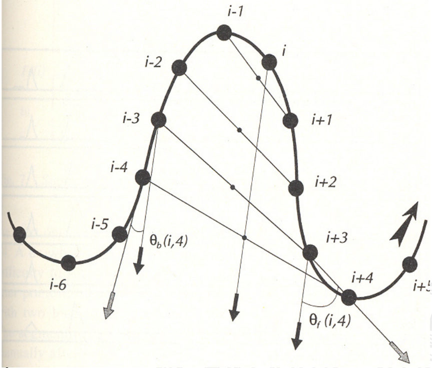
For this purpose, we determine the angles $\theta_f(i,n)$ and $\theta_b(i,n)$ as follows: a line g is established connecting $i-n$ and $i+n$ as well as a connecting line between $i$ and the and the midpoint of g. h is shifted parallel into points $i-n+1$ und $i+n-1$.



$\theta_b(i,n)$ is the angle between the line parallel to h through the point $i-n+1$ and the line connecting points $i-n$ and $i-n+1$, $\theta_f(i,n)$ is the angle between the line parallel to h though the point $i+n-1$ and the connecting line between $i+n$ and $i+n-1$.

# Finding Critical Points II

The more these two angles approach 90 degrees, the less significant is *i* with respect to its neighbours. This property is used to determine if the point pair $i + -n$ is in the zone of influence of *i*: Both angles need to be smaller than a threshold, which is between 0 and 90 degrees, die impact of the points is measured as:

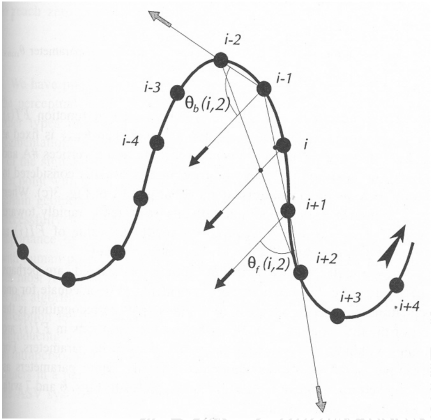


$$IMP(i,n) = cos(\theta_b(i,n)) * cos(\theta_f(i,n))$$

In case the angles are small, the importance of *i* is large, the multiplication indicates that both angles need to be small to result in a significant critical point.

# Finding Critical Points III

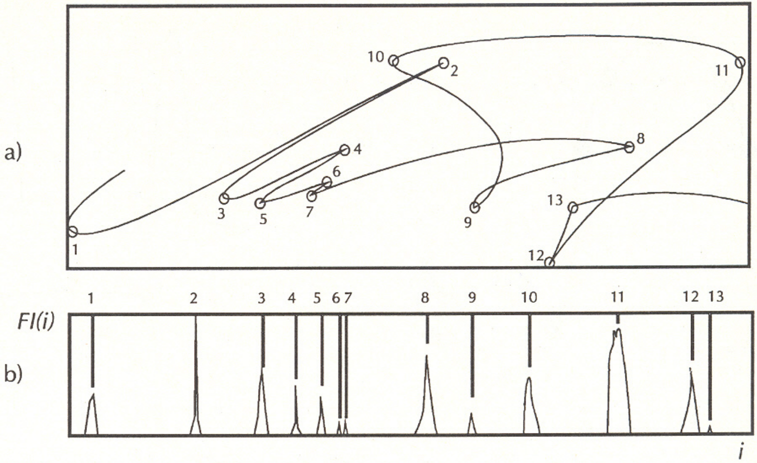
In order to include the contributions of all points in the zone of influence of *i*, we compute $IMP(i,n)$ for all $N$ point pairs in the neighbourhood of *i*, the angles $\theta_f(i,n)$ and $\theta_b(i,n)$ of which are smaller than the threshold mentioned above:



$$FI(i) = \sum_{i=1}^{N} IMP(i,n)$$

For finding all critical points, $FI(i)$ is computed for all points *i* and the local maxima of this function is the set of critical points.
Specific attention needs to be paid towards broad peaks, since the direct neighbours of *i* reduce the impact of the point, which needs to be considered in the computation.

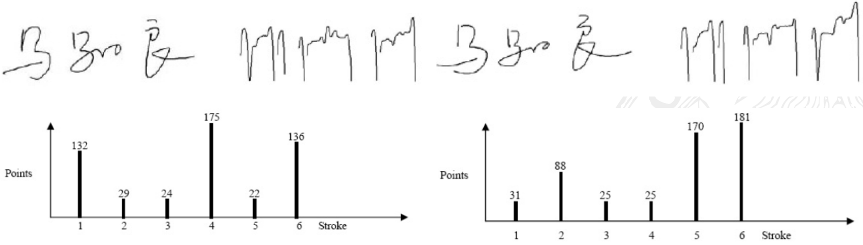
# Finding Critical Points: Example



# Matching Signature Segments

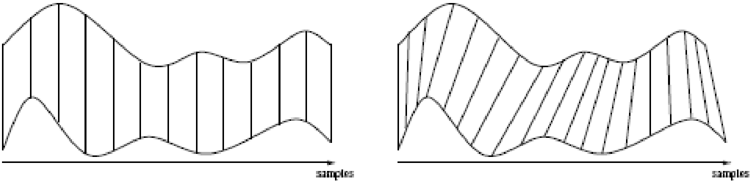
A weakness of segment-based recognition is poor recognition accuracy in case of incorrect segmentation. A possible countermeasure is to recombine segments in case of poor matching results, where one typically starts aligning the longest segment.

In the example, stroke 1 of signature 1 corresponds to strokes 1 & 2 of the second signature, while strokes 5 & 6 of signature 1 correspond to stroke 6 in signature 2.



# Dynamic Time Warp (DTW)

As seen before, segments or also entire signatures never consist of the same number of sampling points in reality. This is caused by the intra-personal variability which is common in dynamic traits. The data usually are not uniformly distorted but highly non-linear and locally varying. This leads to high errors in matching and a corresponding high FNMR.



(a) naive alignment after resampling, (b) alignment with DTW.

DTW alignes two signatures to be matched by using non-linear warping in a best possible manner.

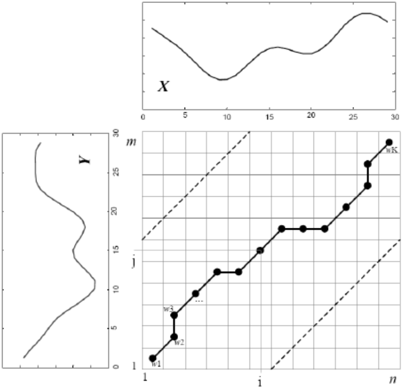
# DTW: Basic idea

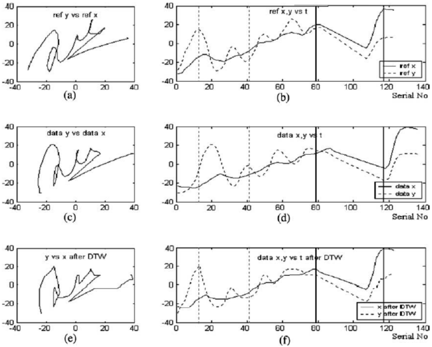
Two time series $X = x_1, x_2, \ldots, x_n$ and $Y = y_1, y_2, \ldots, y_m$ are given. For an alignment, an $n \times m$ matrix $d$ is constructed with entries $d(i,j) = (x_i - y_i)^2$ (which is a typical but not necessary choice). A warping path $W = w_1, w_2, \ldots, w_k$ is a connected set of matrix elements which defines a mapping between $X$ and $Y$. The $l$-the element of $W$ is defined as $w_l = (i,j)_l$. Depending on the application context, there are various restrictions possible wrt. the course of $W$.

- $w_1 = (1,1)$ and $w_k = (m,n)$: Start- and endpoint of the time series are aligned, the path needs to terminate at the opisite corners of $d$.
- Continuity: let $w_k = (a,b)$. For $w_{k+1} = (c,d)$ holds $c - a \leq 1$ and $d - b \leq 1$ (the path only moves to direclty adjacent or diagnonal cells in $d$).
- Monotonicity: let $w_k = (a,b)$. For $w_{k+1} = (c,d)$ holds $c - a \geq 0$ and $d - b \geq 0$.

There is an exponetially increasing number of paths $W$, however, the interest is in the ones minimising the distance between X and Y.

# Visualisation of Basic idea



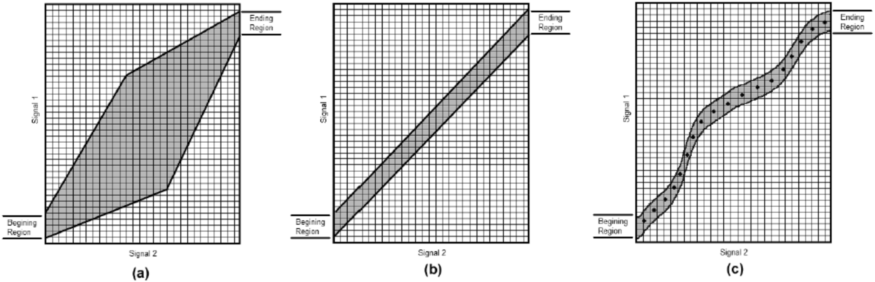


Compute an example: $X = \{2, 4, 8, 13, 9, 5, 8, 12, 15, 18\}$ und $Y = \{2, 3, 5, 9, 12, 8, 4, 9, 16\}$

# DTW: Dynamic Programming

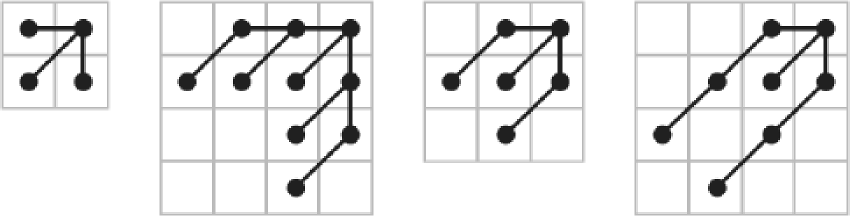
To identify the minimal distance warping path a matrix is defined as $D(i,j) = d(i,j) + min(D(i-1,j), D(i,j-1), D(i-1,j-1))$. We initialise with $D(1,1) = d(1,1)$. To find the optimal path, we need to store in each matrix element which has been the lowest cost path so far. The optimal path is then found vial backtracking.

In order to limit the required computational effort and to limit the time series distortion introduced, it has been sugggested to e.g. constrain the path to certain regions (around the main diagonal), to limit the distance a single step may take etc.



# DTW: Extensions

In the definition of $D(i,j)$ many other options are thinkable (the suggested standard variant is shown left-most), their respective sensibility depends on the application context.

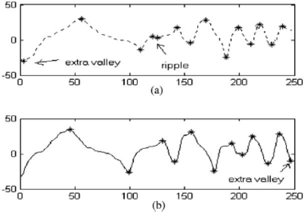


Also, more complicated cost functions are thinkable, e.g. by penalising specifically steep or flat path segments or by rewarding short paths.

There is one significant problem of DTW in the context with signatures: Also forgeries are aligned which may lead to an increased rate of flase positives caused by DTW.

# Extreme Point (EP) DTW

Besides an improved sensitivity against forgeries, also decreasing computational cost is an aim of this approach. The basic idea is to only apply DTW to extremal points of time series. It is required that “too local” extrema are not considered (i.e. a certain minimal distance is required in both coordinate directions). For a correct alignment, EPs are always considered pairs of maxima and minima – problems occur at the start and end of the time series and when “ripples” are present.



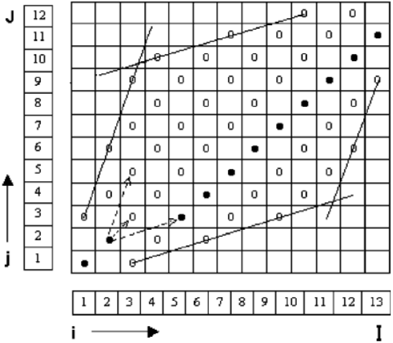
(a)

(b)

The procedure is simuilar to classical DTW but the matrix only contains EPs and specific local path shapes are defined to allow for skipping ripples (which is penalised in cost computations).

# EP-DTW Example

Circles in the matrix (in the allowed path area) depict potential maxima - maxima and minima - minima pairs. Global costs are computed as defined for $D(i,j)$. Black circles represent the minimal-cost path (passing the ripple-pair below the main-diagonal).



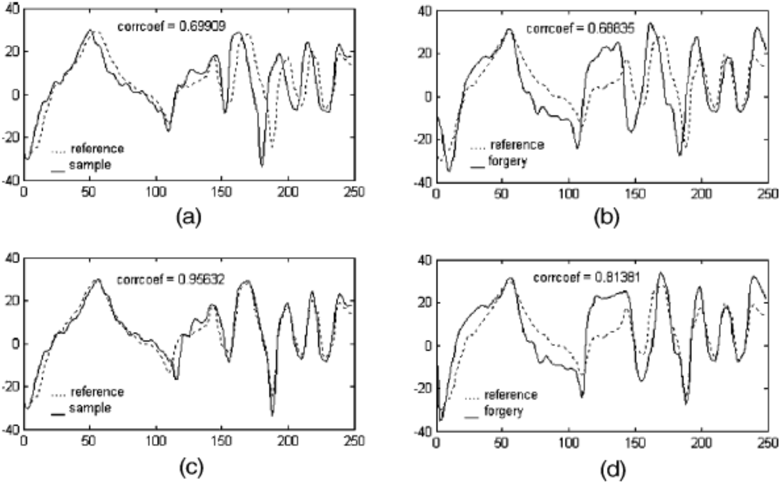
# EP-DTW Segment Warping

After a correct alignment of the EPs, the intermediate segments are stretched / compressed: let $(X_n, Y_n)$ and $(X_{n+1}, Y_{n+1})$ denote EPs of the reference time series which are warped to the EPs of the sample $(x_n, y_n)$ and $(x_{n+1}, y_{n+1})$. By applying DTW we get: $x'_n = X_n$ and $x'_{n+1} = X_{n+1}$. For a value in-between EPs we compute

$$x'_j = X_n + (x_j - x_n)\frac{X_{n+1} - X_n}{x_{n+1} - x_n}$$

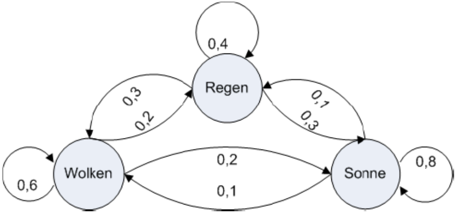
Contrasting to classical DTW the original shape of the curve is less detoriated.

# EP-DTW Segment Warping



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

# Discrete Markov Models I



When observing a time-discrete system, we are often interested in the state of a system at the time of observation. For example, we consider the weather to be the system to be observed and we are interested in knowing tomorrows weather. In this example we have three states, which follow each other with a certain probability.

Definition: A discrete markov model is a system with $n$ states $\omega(i)$. At a discrete time $t$ the system changes its state from $\omega(i)$ to $\omega(j)$ with a certain transition-probability $a_{ij} = P(\omega_{t+1}(j)|\omega_t(i))$. These transition-probabilities are summarised in the state-matrix (transition matrix) $A = (a_{ij})$ and we have $\sum_{j=1}^{n} a_{ij} = 1$ for all $i$.

# Discrete Markov Models II

Given this system, we are able to compute the probability of observing a certain state. For doing this, we need to know the state-probabilities at system initialisation. This is described as $\Pi = (\pi_1, \ldots, \pi_n)$ with $\pi_i = P(\omega_1 = \omega_1(i))$. A discrete Markov model is completely determined by the tupel $(A, \Pi)$. The "memoryless-ness" of the system is of central importance, i.e. a state only depends on the immediately preceeding state.

Example: What is the probability of SSSRRSWS in case the sun was shining on the first day ? $P(SSSRRSWS|Modell) = ??$

$P(S)P(S|S)P(S|S)P(R|S)P(R|R)P(S|R)P(W|S)P(S|W)$ =

$\pi_s a_{ss} a_{ss} a_{sr} a_{rr} a_{rs} a_{sw} a_{ws}$ = 1 0.8 0.8 0.1 0.4 0.3 0.1 0.2 = something.

# Hidden Markov Models

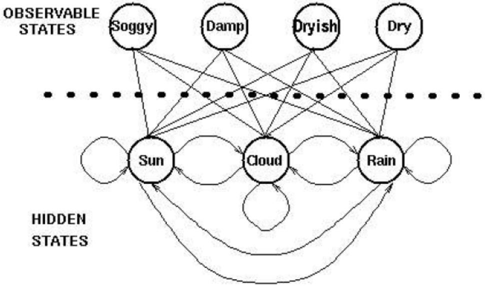
Often it is not possible to directly observe a system behaviour but only its impact on the environment or surroundings. For example, one might try to determine the weather by only observing the humidity of a piece of wood outside. The weather itself cannot be observed ("hiddenstates" – e.g. for someone inside a house not capable to go out side and the piece of wood is brought into the house) but only its impact on the wood.

Definition: A Hidden Markov Model consists of $n$ states $\omega(i)$ which cannot be directly observed. Each of these states emits one of $1 \leq k \leq m$ observable symbols (state) $v_t(k)$ at time $t$, i.e. the sequence $V^T = \{v_1(k), \dots, v_T(k)\}$.

As before we have $a_{ij} = P(\omega_{t+1}(j)|\omega_t(i))$ as the transition probabilities among the hidden states. The probability to emit a specific symbol $v_t(k)$ at time $t$ in case the system is at state $\omega_t(j)$ is defined as $b_{jk} = b_j(v_t(k)) = P(v_t(k)|\omega_t(j))$. These probabilities are not time-dependent due to the memory-less-ness of the system and are collected in the confusion matrix $B = b_{jk}$; $\sum_{k=1}^{m} b_{jk} = 1$ for all $j$.

# Hidden Markov Models II

In our example the hidden states are sun, clouds, rain, while the observations are sogggy, damp, dryish, and dry with their corresponding transition- (A) and confusion (B) matrices.

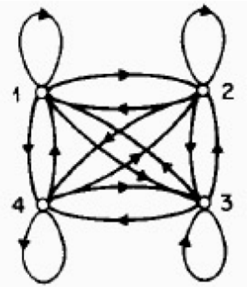


$$A = \begin{pmatrix} 0,5 & 0,25 & 0,25 \\ 0,375 & 0,125 & 0,375 \\ 0,125 & 0,625 & 0,375 \end{pmatrix}$$

$$B = \begin{pmatrix} 0,6 & 0,2 & 0,15 & 0,05 \\ 0,25 & 0,25 & 0,25 & 0,25 \\ 0,05 & 0,1 & 0,35 & 0,5 \end{pmatrix}$$

For example, this means that in the state sun the piece of wood is dry with P = 0.6. A HMM is characterized by the triple $(A, B, \Pi)$.

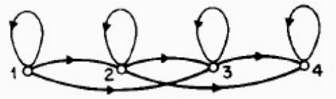
# Hidden Markov Models: Topologies



Ergodic HMM: $a_{ij} \neq 0, \forall(i,j)$ – all states can be reached within one step from everywhere.

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{pmatrix}$$

Left-Right HMM: bandmatrix $\neq 0$; additional constraints: no backleading transitions ($a_{ij} = 0$ for $j < i$) and no jumps of more than X states ($a_{ij} = 0$ for $j > i + X$).



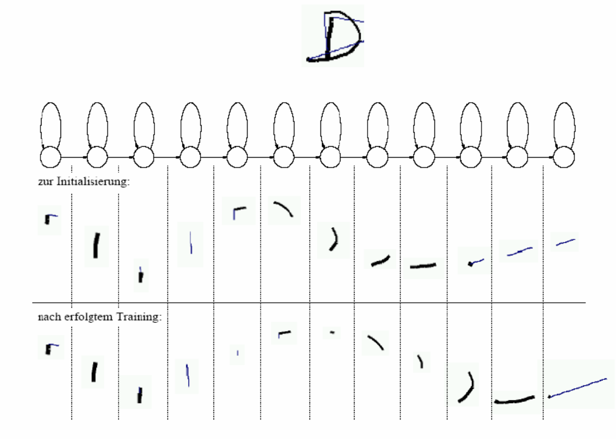
$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} & 0 \\ 0 & a_{22} & a_{23} & a_{24} \\ 0 & 0 & a_{33} & a_{34} \\ 0 & 0 & 0 & a_{44} \end{pmatrix}$$

# Three central problems in HMM

1. Evaluation: Given a HMM $= (A, B, \Pi)$: Find the probability that a sequence of visible states $V^T$ was generated by that model.
2. Decoding: Given a HMM $= (A, B, \Pi)$: Find the most probable sequence of hidden states that led to a sequence of visible states $V^T$.
3. Learning: Given the number of visible and hidden states and some sequences of training visible states: Determine the optimal parameters $a_{ij}$ and $b_{jk}$.

Biometric Interpretation: visible states correspond to computed features of biometric traits (which are the hidden states), single states represent segments of a signature or phonetic units of utterances. For a given signature or spoken-passphrase a HMM is trained, i.e. learning is conducted during enrollment. Evaluation assesses observed features wrt. a learned HMM in the context of verification, while decoding determines the most probable sequence of hidden states and thus can be used in identification.

# Signatures and HMM: Example



For signature modelling, usually about 6 states are used, however, this can be optimised in training. Left-Right topology with optional state skips is the classical solution.

# HMM: Evaluation

For computing the probability of a given HMM to emit a sequence of visible states $V^T$ one could consider the probability of all potential state sequences while emitting the sequence of visible states. Adding up all these probabilites produces the overall probability $P(V^T)$, where $T$ is the length of the observed sequence and $n$ the number of hidden states.

$$P(V^T) = \sum P(V^T|\omega^T)P(\omega^T)$$

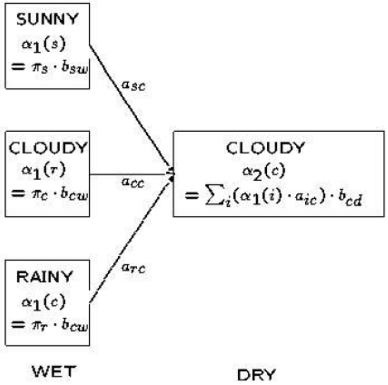
where the sum is extended over all potential hidden state sequences $\omega^T = \omega_1, \ldots, \omega_T$.

$$P(V^T) = \sum^{n^T} \prod_{t=1}^{T} P(v_t|\omega_t)P(\omega_t|\omega_{t-1})$$

There are $n^T$ state sequences of length $T$. For practical applications, this is too expensive to be computed (complexity $N^T T$).

# HMM: Forward Algorithm

Solves the evaluation problem recursively. We define the probability $\alpha_t(j)$ as the probability that the HMM is at state $j$ and has produced the first $t$ elements of $V^T$. For initialisation $t = 1$ we have $\alpha_1(j) = \pi_j b_j(v_1(k))$ (probability to emit the first visible state at hidden state $j$).



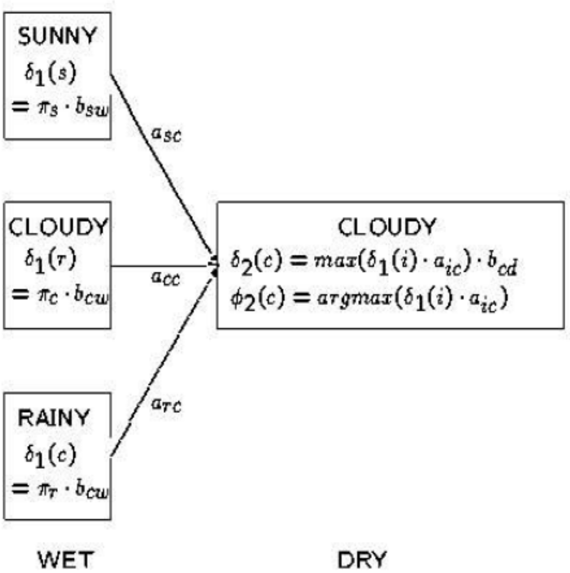
In general, $\alpha_t(j) = \sum_{i=1}^{n} [\alpha_{t-1}(i) a_{ij}] b_j(v_t(k))$. This expression is computed for all states $j$ and all times $t$ (which is $nT$ overall, each $\alpha$ requires $n$ operations, thus leading to $n^2 T$ operations). So the total probability is achieved by computing $P(V^T) = \sum_{i=1}^{n} \alpha_T(i)$.

# HMM Decoding - Viterbi Algorithm

Also for decoding one could compute the probabilities for all sequences of hidden states emitting the sequence of observed states and finally picking the one with the highest probability, but as seen, this is too expensive. The solution is to apply the dynamic programming priciple with maximising the probability, i.e. the identification of the most probable state at each time $t$. For this purpose we define the Viterbi variable $\delta_t(i)$ to be the largest probability to have $V^t$ emitted in case the HMM is in state $i$ at time $t$.

For initialisation $t = 1$ we have $\delta_1(j) = \pi_j b_j(v_1(k))$. In general, $\delta_t(j) = max_{1 \leq i \leq n}[\delta_{t-1}(i)a_{ij}]b_j(v_t(k))$. This means that from state to state, we identify the path with the largest probability until we arrive in $j$. Since the sequence of states is sought, the stes need to be stored in a path $\phi(j)$: $\phi_t(j) = argmax_{1 \leq i \leq n}[\delta_{t-1}(i)a_{ij}]$. Having reached the final state, the maximal value is traced along the path to $t = 1$.

# HMM Decoding - Viterbi Algorithm



# HMM Learning

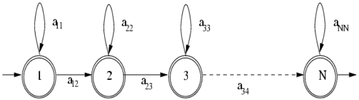
A classical approach to determine the HMM parameters is the “Baum-Welch Recursion”. A suited initial model is determined ($(A, B, \Pi)$ either randomly or using pre-knowledge). For this model we compute the generation probability using the Forward algorithm. In iterative manner $A$ and $B$ are adjusted using forward and backward probabilities (the latter using the Backward algorithm), leading to an increase of the generation probabilities.

Problems with HMM:

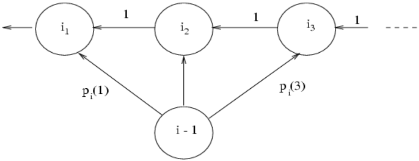
- A large amount of training data is required (leading to expensive enrollment)
- Optimal number of states cannot be determined using the Baum-Welch Recursion
- The assumption of memory-less-ness can be doubted in many applications

Nevertheless we see many applications in biometrics, in particular in signature verification and in speaker recognition (using text-dependent techniques).

# HMM and Time Steps



In the classical model with constant time steps resting in state $i$ is modelled by $a_{ii}$ meaning that the state does not change for some time steps. A more natural solution is to use HMM with variable time steps which are denoted as HSMM (S for semi) – they can be also represented by schemes using $a_{ii} = 0$.



A HSMM can be approximated by a HMM with a larger number of states. The simplest case is to partition a HSMM state $i$ into several sub-states $i_1, i_2, \ldots$ with constant time steps and transition probabilities $a_{i-1,i_k} = p_{i-1}(k)$ and $a_{i_j,i_{j+1}} = 1$.

# Biorhythms: Pulse / ECG

The fundamental idea is obvious, since the heart beat signal is relatively easy to acquire and a high inter-personal variability is to be expected due to the uniqueness of the organ and the electric conduction system. Moreover, much knowledge is available from medicine.

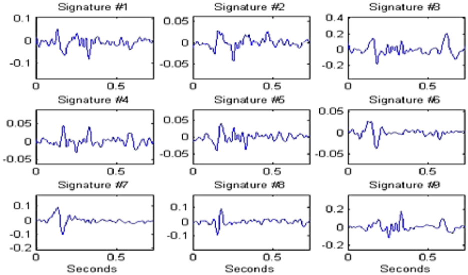
Measurements exceeding to take pulse only are quite involved (ECG) and there a large extent of pathologic and non-pathologic variability in heartbeat causing a potentially high intra-personal variability in case the wrong trait is chosen.

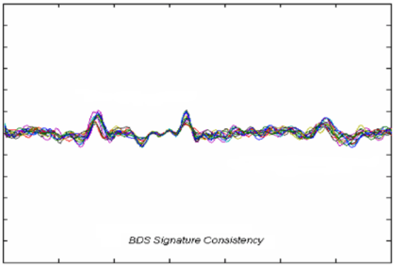
Examples: Short-term changes in heart rhythm (tachycards, extra systols (more than 60 on average per day), ventricular fibrillation, atrial fibrillation) as well as long term changes caused by diseases (e.g. endocarditis, myocardial infarctiona), stress-induced effects, etc.

Which ECG components might correspond to these requirements is unclear.

# Heartbeat: IDESIA

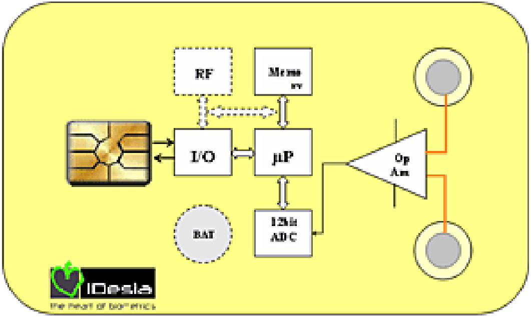
IDESIA, an Israeli startup, has been acquired by Intel in 2012. They have been offering a complete solution based on heart-beat, but it has never been revealed what they actually measure. Statistical vlaidations rely on 160 persons and cannot be verified. The figure shows curves of 9 test persons and an overlay of 16 measurements of a single test person. Intra-personal variability is very unclear, they claim to deliver excellent FMR, FNMR and EER.





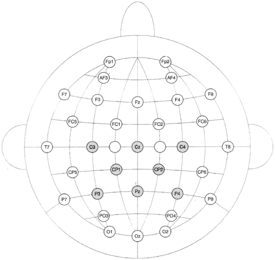
# IDESIA Products

IDESIA offered complete sensors to plug into PCs (signal is acquired with a single finger per hand lasting 3-4 seconds), sensor kits for integration into other devicesm corresponding smartcards plus sonsors to cobine with fingerprint readers (for “fusion”).



# Biorhythms: EEG

Compared to ECG, the EEG signal is by far more complex. A major problem is sensing: It is difficult to acquire signals with high quality since in the classical non-invasive scenario the sensors are separated from the signal emittors by the sculp. skin, and hair. In severla studies, electrode P4 has been identified as promising (which records $\alpha$-rhythm). The first approach to employ EEG data is similar to the ECG case by trying to use an individual fundamental brain wave pattern. EEG variability caused by conscious and unconscious effects is a problem in this setting.



In a study, 8 electrodes (F7+8, C3+4, FC1+2, P3+4)(F7+8, C3+4, FC1+2, P3+4) are used to acquire EEG epochs lasting 8.5 sec from subjects in rest-state – epochs impacted by muscle or heart activity are not used, so that finally 8 epochs could be used per person. In this setup. 40+ persons could be differentiated.

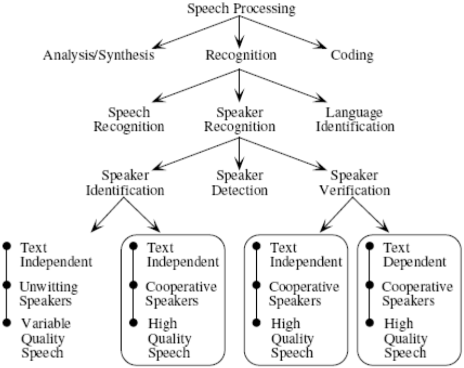
# Biorhythms: EEG / BCI Approach

A second approach is inspired by brain-computer-interface technologies. In BCI subjects are trained to steer EEG signals by visualising activity such that binary information can be read out. The aim is to enable paralised patients to communicate by navigating through (binary) menu controls. In biometrics, the idea is to conduct a specific mental activity – this can be seen as an advanced combination between knowledge-based and biometric authentication: visualising a secret password and reading out the corresponding EEG signal (not yet possible).

Reality: Visualising of left and right hand activity and words with identical first letter. Sessions enduring 4 minutes, 15 seconds a mental task using 8 electrodes. Laplacian filtering of the raw data, 3 segments lasting 0.5 seconds undergo an FFT and results are averaged. Based on three subjects, results similar to face recogniiton and speaker recognition have been achieved. Scalablity is unclear.

# Speech Processing and Speech as Biometric Trait

Speaking is habituated and a large scale distributed sensor infrastructure (telephone, VoIP) is available – ideal candidate for biometric application. It is important to distinguish between text-independent (free speech) and text dependent (user-specific PWD or system-provided one) systems.



# Speech Processing and Speech as Biometric Trait (contd.)

For biometric usage, text-independent techniques are clearly more difficult to implement, however, speaker-specifics like using certain words very often, pauses, frequently used set phrases can be exploited besides the acoustic information. Since there is no temporal ordering, such techniques employ Gaussian Mixture Models (GMMs) instead of HMMs (we do not go into details here).
The more techniques in speech recognition get improved and the more speech and speaker recognition do converge, the less improtant is to distinguish text-dependent from text-independent techniques.

For biometrics, text-dependent speaker recognition is currently the method of choice.

# Text-dependent Speaker Recognition: Feature Extraction

1. Recognition of speech activity: Segmentation of the audio signal into segments with and without recorded speech.
2. Feature extraction: It is well established that a speech signal is significantly determined by the physiological conditions of the speaker. The signal is partitioned into 20 ms temporal frames (using smooth Hamming window functions) with 10 ms overlap. Two different strategies can be identified: LPC & MPCC
3. Canal compensation: Different types of recording devices and transmission means cause different audio signal properties. Recognition should work independent of these contextual parameters, thus features need to be post-processed.

In any case, the result of feature extraction is a feature time series, which needs to be compared to a reference time series. Due to the conceptual simplicity to on-line signature recognition, similar techniques are used for comparision (e.g. HMMs, DTW, ...).

# Speaker Recognition Feature Extraction: LPC I

LPC (linear prediction coefficients) is based on a linear model of speech generation. The physiological apparatus for speech generation consists of 4 modules: glottis (producing a stream of impulses in case of voiced tone and white noise in case of voiceless tone), oral cavity / vocal tract, nasopharingal zone, and the lips. Each of these components may be represented by a specific filter, e.g. a low-pass filter for the glottis, an AR (auto-regressive) filter for the ral cavity, an ARMA (auto-regressive moving average) filter for the nasopharingal zone, and a MA filter for the lips. Overall, the entire system is represented by an ARMA filter.

Characterisation of a speech signal is done by determination of specific filter coefficients for a given utterance. In most techniques the ARMA model is simplified to a AR filter – in case of LPC analysis, filter parameters of an AR filter are computed for a frame of the audio signal. In each frame optimal parameters are computed which are used to produce the feature vector.

# Speaker Recognition Feature Extraction: LPC II

Given a speech signal segment $S = (s(0), s(1), s(2), \ldots, s(N))$, with $s(n)$ the observation at time $n$, $p$ the order of the predictor and $a_k$ the predictor coefficients. Furthermore, $u_n$ is the current input (physiological speech incitation signal) and $G$ the gain factor (air volume leading to different audio volume).

$$s(n) = \sum_{k=1}^{p} a_k s(n-k) + Gu(n)$$

In actual recognition applications, $u_n$ and $G$ are not knows and are therefore ignored. Thus, the LP approximation $\hat{s}(n)$ is given by

$$\hat{s}(n) = \sum_{k=1}^{p} a_k s(n-k) \text{ with } e_n = s(n) - \hat{s}(n)$$

and only consists of current and past sample values.

# Speaker Recognition Feature Extraction: LPC III

The aim is to determine $a_k$ to minimise MSE E over the entire frame:

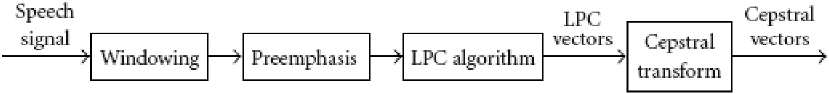
$$E = \sum_{i=1}^{N} e_i^2 = \sum_{i=1}^{N} [s(n) - \sum_{k=1}^{p} a_k s(n-k)]^2$$

Uisng the classical criterium for minimisation, i.e. $\frac{\delta E}{\delta a_k} = 0 \ \forall k = 1, 2, \ldots, p$ leads to the following condition after some algebraic computations:

$$\sum_{k=1}^{p} a_k \sum_{n} s(n-k)s(n-i) = -\sum_{n} s(n)s(n-i) \quad \forall i = 1, 2, \ldots, p$$

This expression leads to techniques that employ autocorrelation and co-variance to determine the sequence of $a_k$.

# Speaker Recognition Feature Extraction: LPC IV



- “Pre-emphasis” is applied to strengthen high frequencies which are surpressed in speech generation: $s_p(n) = s(n) - as(n-1)$ with $a$ from the interval $[0.95, 0.98]$. This preprocessing makes sense only under specific conditions and needs to be tested empirically.
- Computed vectors $a_k$ are used to generate finally used feature vectors, e.g. Linear Predictive Cepstral Coefficients (LPCC, with $c_1 = a_1$ and $K \leq p$ the number of desired coefficients):

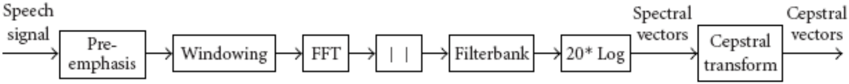
$$c_n = a_n + \sum_{k=1}^{n-1} (1 - k/n) a_k c_{n-k} \ , \quad n = 1, \ldots, K$$

# Speaker Recognition Feature Extraction: LPC V

- LPCC are often reduced by their mean (cepstral mean substraction CMS – the contribution of background noise is substracted) and variance is normalised to 1 (“reduction”).
- Dynamical information is also used, i.e. in how far the generated vectors change over time, i.e. delta cepstra, which are computed by approximating first and second derivate.

# Speaker Recognition Feature Extraction: MFCC I

Mel Frequency Cepstral Coeffcicients (MFCC) are the classical result of a short term Fourier Transfor (STFT) signal analysis with psychoacoustic modelling.

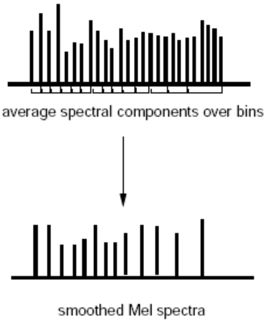


Following a STFT, spectral components undergo non-uniform quantisation follwoing a psychoacoustic model. The “Mel” frequency range is used, which emphasises low frequencies (see next slide) applies a logarithm to parts of the spectrum. Subsequently, feature vectors are generated in two classical ways:

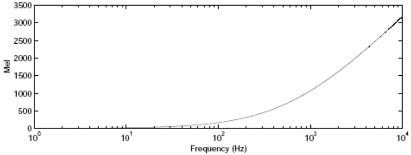
# Speaker Recognition Feature Extraction: MFCC II

1. Due to their high correlation, Mel-frequency vectors undergo a DCT, where only DC and low frequency AC coefficients are kept (this reduces the original 256 spectral comppents to 40 Mel-spectral values and finally to about 13 cepstral features per frame.
2. Mel-frequency vectors undergo an inverse FFT which results in a set of real cepstral coefficients (RCC).

# MFCC – Mel Frequency Space

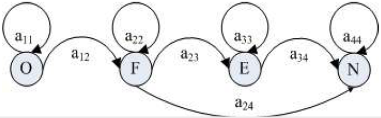


The Mel scala is based on a map between physically measurable frequency and the analogue psycho-acoustic unit, i.e. pitch; Pitch (measured in phone or sone) is perceived in a non-linear relation to frequency and is also dependent on signal intensity. For signal intensity – physically SPL sound pressure level – measured in dB the analogue psycho-acounstic measure is loadness. The map between frequency and pitch is linear below 1 kHz and logarithmically above that value.



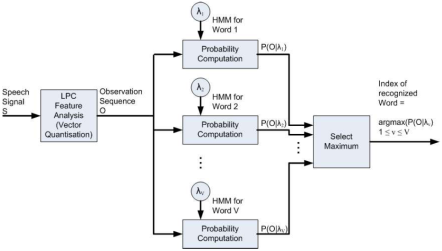
# Matching in Speaker Recognition

1. Template Models
   - DTW: obvious.
   - Vector Quantisation (VQ): Using classical methods in vector quantisation, enrollment data is used to generate a codebook approximating enrollment data with lowest possible error. Matching score is given by the minimal distance between a frame to verify and a codebook entry.
   - Nearest Neighbour (NN): All distances among enrollment frames and frames to be verified are computed. (e.g. by using DTW). The NN distance of a frame is the accumulated distance to its k NN. Finally, all NN distances are averaged giving the score.
2. Stochastic Models (HMM): Hidden states correspond to phonems of a word, the emitted sequence corresponds to the feature vectors of the corresponding frames, and in most cases left-right topology is used.



# Speaker Recognition: HMM

After enrollment, k emitted sequences are available, which are used train a HMM for each speaker using Baum-Welch recursion (either using the same word for each speaker or a PWD – a speaker-specific word). In verification / identification, the forward / Viterbi algorithm is used to determine the probability of the observed sequence being uttered by the claimed speaker or which speaker has the highest probability to utter the observed sequence.



In case an identical word is used for all speakers instead of an individual PWD, Word1 has to be replaced by Speaker1 in the graphics. When employing PWDs, speaker recognition is mixed with speech recognition.

# Uncommon Biometrics: Odour

While odour is suited for biometric applications (as demonstrated efficiently by well-trained dogs), specific problems arise:

- Intra-personal Variability: Body odour changes significantly caused by certain physiological and pathological processes in the body: E.g. physiologic – stress, hormonal changes, specific food; pathologic – diabetis, diseases of the gastointestinal tract, dental diseases, wound infection etc.
- Sensor problem: In the human "sensor" we have about 10000 sensors the signals of whch are processed by 10 million sensor neurons. This signal processing system analyses odours like coffee which consists of about 670 different chemicals. There is a discrepancy to eNoses, which provide 12-30 sensors for different substances and corresponding simple processing routines.

Still there is interest in the biometric community: ILi Systemas SL and the Pentagon. More realistic applications include search for victims after spillage accidents, in medical diagnostics, environmental surveillance (controlling emmissions), food industry, fragrance .......

# Outline

1. Formalia
2. Introduction & Motivation
3. Foundations of Biometrics
   - Introduction and Basic Terminology
   - Biometric Traits & their Properties
   - Biometric Systems System Model
   - Assessment of Biometric Systems
   - Security and Privacy in Biometric Systems
4. Non-visual Biometrics
   - Keystroke Dynamics
   - On-Line Signatures
   - Biorhythms
   - Speaker Recognition
   - Odour
5. Visual Biometrics
   - Off-line Signatures
   - Iris Recognition
   - Fingerprint Recognition

# Off-line Signatures: Basics

Advantages: No specific sensor technology is required, however, specific minmal scanner quality is necessary. Signing can safely be assumed to be habituated for off-line signatures.
Disadvantages: Temporal dynamics of on-line signatures is lost and with it an important aspect of forgery protection capability. However, due to the increase of intra-personal variability the importance of temporal dynamics is not as high as it is often assumed to be.

As already pointed out in the section on on-line signatures, many features praised to be intrinsically “on-line” can be generated from off-line data by replacing time by a time-independent parameter like arc length of the signature. Of course, this is not possible for pressure, pen inclination variants, speed, acceleration, total time, etc.

# Off-line Signatures: Normalisation

Normalisation is important to compare differently sized and positioned signatures. Depending on the type of features used, entirely different normalisation strategies are used. Optimally, the sensor already provides measures to support normalisation (e.g. given line, writing sensitive field of limited size etc.).

- Data given by analogy to time series $X(t)$
  - Normalisation by DTW (where of course T is not really correct) – DTW can be used for optimal alignment and if requiredm two signatures with an identical number of samples can be generated by interpolation.
  - Normalisation to obtain unified arc length is simpler, but eventually ignores problems caused by start- or end-artifacts.
- Data given as planar curve: Coordinates $(x_i, y_i)$
  - Fourier methods
  - Spatial domain methods

# Off-line Signatures: Fourier Normalisation for planar Curves I

Let be given a planar curve

$$\vec{z} = (z_1, z_2, \ldots, z_N) = ((x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)) \ .$$

The Fourier transform of this curve is given by

$$\vec{Z} = F\vec{z}$$

with $F$ being a Fourier matrix with entries $F_{jk} = \omega^{jk}$ and $\omega = e^{2\pi i/N}$. The sequence of Fourier coefficients can be used als an alternative representation (which removes correlation and leads to a more compact representation, similar to image processing): $\vec{Z} = Z_0, Z_1, \ldots, Z_{N-1}$. In this representation a normalisation can be applied which relies on a normalisation of the Fourier coefficients.

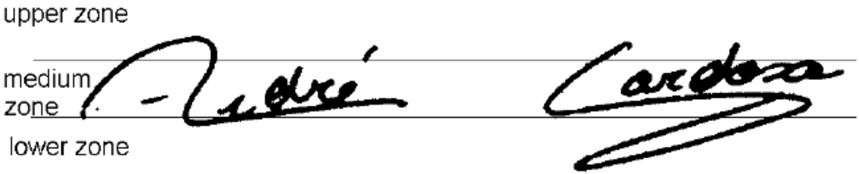
# Off-line Signatures: Fourier Normalisation for planar Curves II

1. As a first stage, we set $Z_0 = 0$ which corresponds to a translation of the coordinate system into the controid of the curve: $\vec{z} - Z_0 = \vec{z} - 1/N\sum_{k=1}^{N} z_k$. This procedure corresponds to a the substraction of the mean grey scale in image processing.
2. In the second stage, the next Fourier coefficient $Z_1$ is normalised to 1 – this is achieved by dividing all $Z_i$ by $Z_1$ (which is equivalent to a division of $\vec{z} - 1/N\sum_{k=1}^{N} z_k$ by $Z_1$). The geometric interpretation of this procedure is a scaling and rotation of the entire signature.

# Position Normalisation or planar Curves

For position normalisation, signatures can be aligned following their center of gravity followed by determining the optimal rotation by pattern matching.

Another variant uses graphological terminology and determines the central parts and the upper and power parts of the signatures. The separating line between central and lower parts can be used as x-axis, while the point with smallest x-coordinate determines the position of the y-axis.

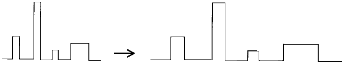


# Scaling Normalisation of planar Curves

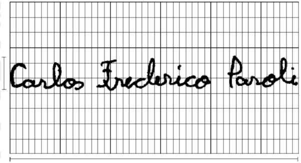
For scaling normalisation two different strategies are used, depending on the assumptions which variations are to be expected for an authentic signature.

The first approach normalises the aspect ratio, i.e. the relation between length and width of a circumscripted rectangle (in fact the ratio between the sum of all vertical displacements and the sum of all horizontal displacements is computed an transformed to a unified value).

The argument is that a signature can be made significantly longer without changing its height (the question is if not an important signature property is destroyed by this approach – te shape of a circumscripted rectangle is also used as a global feature !).



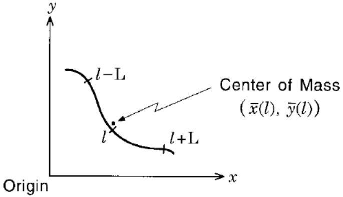
The second approach only normalises the length, where it is simply possible to count the number of empty boxes as shown in the figure.



# Off-line Signatures: “Time-series” Features I

Features described in the following are also often used in the context of on-line techniques, here regarded as features of a sequences parametrised using arc length $l$ of the signature.

Let $x(l)$ and $y(l)$ be the coordinates parametrised using $l$ and $g(\lambda)$ a Gaussian weighting window of width $+-L$.

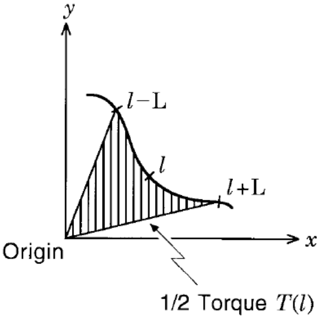


The coordinates of the center of gravity $(X(l), Y(l))$ are given as ($Y(l)$ by analogy):

$$X(l) = \int_{-L}^{L} g(\lambda)x(l+\lambda)d\lambda$$

The sequence of center of gravity coordinates is significantly robuster as the original coordinates.

# Off-line Signatures: “Time-series” Features II

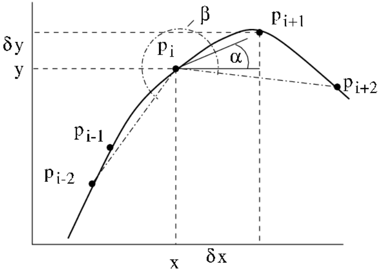


The **torsion or turning force** is given by:

$$T(l) = \int_{-L}^{L} g(\lambda)(y(l+\lambda)dx(l+\lambda) - x(l+\lambda)dy(l+\lambda))d\lambda$$

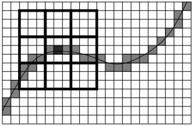
with $dx(l+\lambda)$ is the displacement in x-direction at position $(l+\lambda)$ when $\lambda$ is changed. There is a physical meaning – positive $T(l)$ is counterclockwise rotation !

# Off-line Signatures: “Time-series” Features III

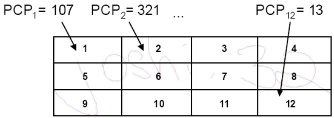


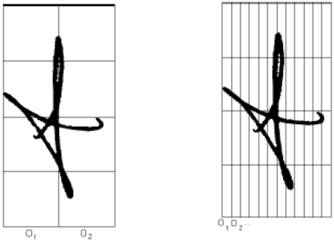
Typical further features are the curvatures $\beta(i)$ interpreted as angle between the line $p_{i-2}p_i$ and $p_ip_{i+2}$ as well as the angle $\alpha(i)$ (between the x-axis and the line $p_ip_{i+1}$) and the difference angle $\delta\alpha(i) = \alpha(i) - \alpha(i-1)$.

Also, the sequence of tangent angles on $p_i$ is used – a variant is the application of the DFT to a 10 element vector of this sequence. Another time series feature is the “sliding Bitmap Window” which considers a 9 × 9 pixel window at each point of the signature which is subdivided into 9 3 × 3 pixel blocks in which the number of set pixels is counted. This produces a 9-element vector in each signature point.



# Off-line Signatures: Pixel Feature



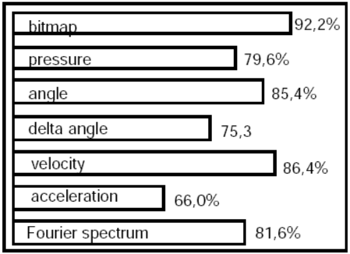


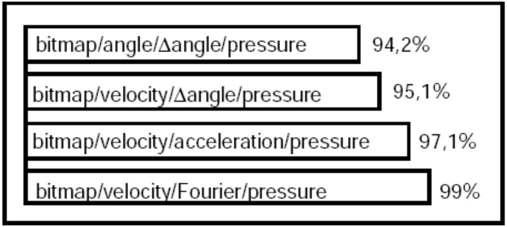
The “Bitmap” is one of the simplest features and does not require a parametrerised curve representation: The signature is circumscribed by a minimal rectangle which is scales and partitioned into a fixed number of squares or rectangles. In these boxes the number of set pixels is counted, which provides the Pixel Count Parameter (PCP).

Especially in areas below and above the central are it turns out that a more fine grained partitioning is of advantage. To limit the overall box count, multiresolution techniques can be applied.

# Off-line vs. On-line Signatures

The plot shows the number of correctly accepted authentic original signatures plus correctly rejected forgeries.





Interestingly, off-line features are competitive at least, really high accuracy is achieved by fusion techniques.

# Signature Verification Contest SVC

SVC was a biannual contest under normed condition and has been transformed into a contineous contest in the last years. Trainings data is provided, submitted algorithms are evaluated on a standard test data set resulting in a set of performance figures.

Two tasks are investigated: time-dependent coordinate data (more of off-line type) and a second one which additionally has pressure and pen orientation data (on- as well as off-line). Both datasets contain a significant number of forgeries, however, for privacy reasons not the real signatures of the subjects have been used (although this was trained it is not really habituated execution which is a design weakness).

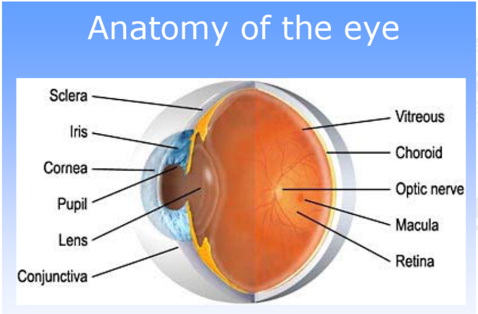
There are significant differences among the submitted techniques and there is no significant advantage of task 2 over task 1 which questions the usefulness of intriniscally on-line features. These results have been explained by the "non-habituated" signatures.

# Signature Products

- `http://www.cybersign.com` On-line and Off-line Features, in a mixture with digital signatures
- `http://www.signplus.com/en` pure signature verification (e.g. for Login – SignSecure) but also focussed to a combination with digital signatures (SignDoc)
- `http://www.bio-pen.com/` On-line signature vericikation, sensorics integrated in pen
- `http://www.penflow.com/` On-line and Off-line Features, in a mixture with digital signatures

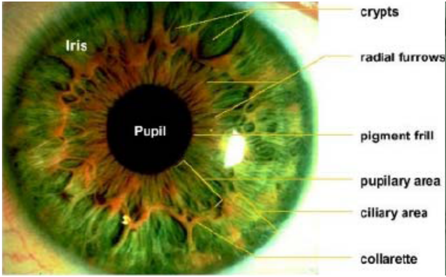
# Iris Recognition: Anatomy

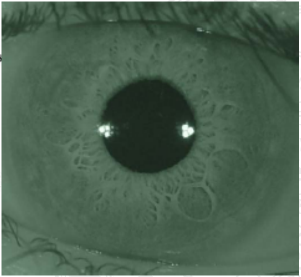
The iris is located within the human body (i.e. protected by the cornea from environmetal influences) and is visible quite well despite of this fact. It is assumed that the iris pattern remains stable after the first year of life, although recent work suggests ageing effects not only due to pupil dilation changes. Muscles used for pupil contraction change area and shape of the iris and it exhibits a pulsating effect in the direction of this muscle contraction (hippus) which can be used for liveness detection (besides the pupils' reaction to light exposure).



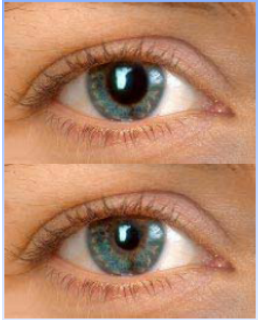
# Iris Recognition: Imaging Modality

Iris texture is highly detailed, colour information can be used for a first pre-classification. However, for dark irises, the texture cannot be read out due to nun-existant contrast, thus near-infrared (NIR) imaging with corresponding illumination is used which exhibits the texture pattern in great detail and good contrast (still there is work on iris recognition in the visible range, especially using mobiles, with questionable applicability). Another adantage of NIR is that illumination is hardly perceivable (red glow is perceived), applicability for wide range recognition is not well understood.

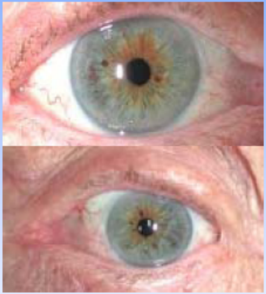




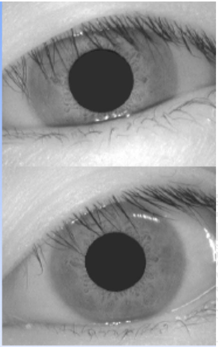
# Iris Recognition: Intra-personal Variability



Pupil Dilation
(lighting changes)

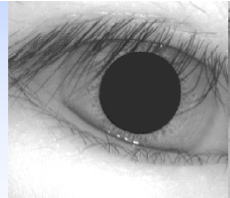


Inconsistent Iris Size
(distance from the camera)

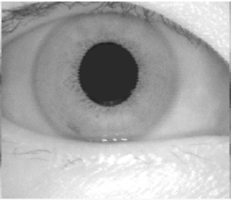


Eye Rotation
(head tilt)

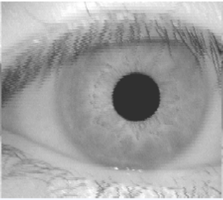
# Iris Regognition: Imaging & Medical Problems



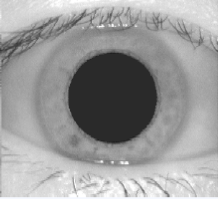
Occlusion (eyelids/eyelashes)



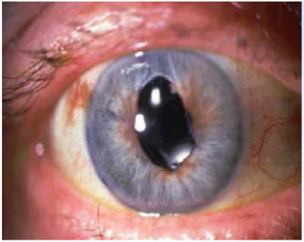
Defocus



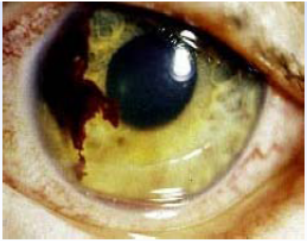
Motion blurred



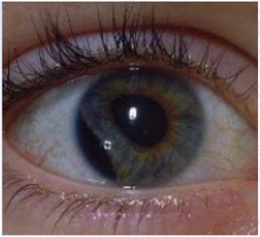
Large pupil



cataract surgery



hyphaema (blood clot)



iridodialysis

# Iris Recognition: Workflow I

Most iris recognition schemes are similar wrt. items 1) - 4), major differences are seen in item 5).

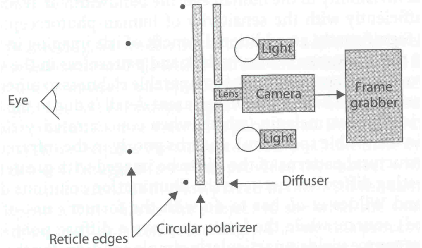
1) Acquisition: Commercial systems use NIR illumination, best results are seen in case of imaging with user cooperation in terms of optimal positioning and a small camera distance, altough recent systems ("iris on the move") capture data from moving subjects in a tunnel-like system, which requires iris tracking and subsequent capturing with a tele-lens.
2) Iris Localisation & Segmentation: In the eye greyscale image, the iris is detected and segmented usually based on boundary edge-chains.
3) The resulting annunus (almost circular ring) is difficult to be processed further also due to different radii – using a transform into polar coordinates, a rectangular data structure is generated.

# Iris Recognition: Workflow II

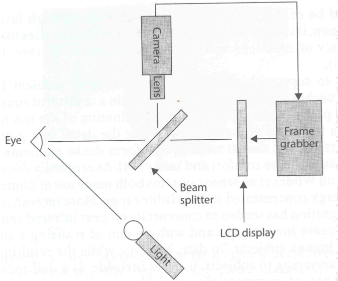
4) Image Enhancement: The resulting texture data exhibits poor contrast which needs to be improved by e.g. CLAHE (local histogramm equalisation) or local mean substraction.
5) Feature Extraction: This is the major point of difference among different techniques, however, a common property of the most accurate schemes is to use locality-preserving features (wavelet- or Gabor-based schemes), since the position of the various iris texture features is a highly distinctive property.
6) Matching: According to different features, also matching procedures do differ. The most common approach (which is most desirable due to its speed) is to compare binary feature vectors by computing Hamming distance.
7) Decision: Classically threshold-based decision is used.

# Iris Recognition: Cooperative Acquisition

Localisation in the first scheme (Wildes) is achieved by two displaced squares of different size which need to be superimposed by the used – having achieved this, the eye is (i) at the correct position and (ii) in focus (which is important due to the small depth of field when using tele-lenses).

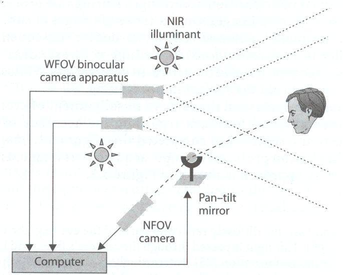


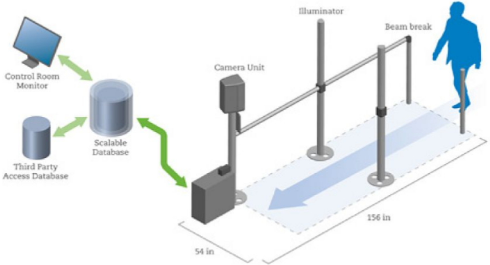
Im the second system (Daugman) positioning is achieved by pictorial feedback: the eye position relative to the required coordinates is shown on a screen in realtime, such that the user is able to adjust the position accordingly.



# Iris Recognition: Acquisition from a Distance

In same schemes, active vision is employed, e.g. using a wide-angle camera pair for eye detection and a steerable tele lens system for the iris acquisition. Sarnoff is the leader in commercial systems, having developed the “iris on the move” system (claimed 30 persons / minute).





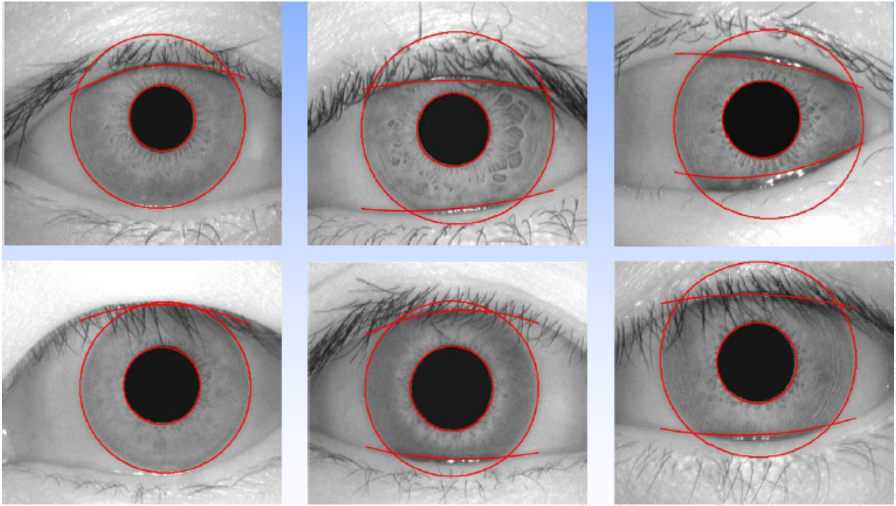
# Iris Segmentation

Iris segmentation is a crucial stage for an accurate recognition system, especially in unconstrained acquisition conditions. The iris is curtailed by several objects and may be partially occluded: the (black) pupil causes the inner boundary, while the (white) sclera constitutes the outer one (obviously with very different properties). Still, contrast is available which makes edge-based segmentation a promising approach, due to the differrent properties in contrast and sharpness adapted edge detection gives the best results.

When using a parameterised model, it turns out that boundary curves are neither concentric nor circular (especially for off-angle imagery) and should be modeled at least using ellipsoid curves.

Advanced techniques use explicit lid-modelling by parabolic curves and eye-lash modeling using especially the different edge orientation is compared to lid and iris-boundary edges.

# Iris Segmentation Examples



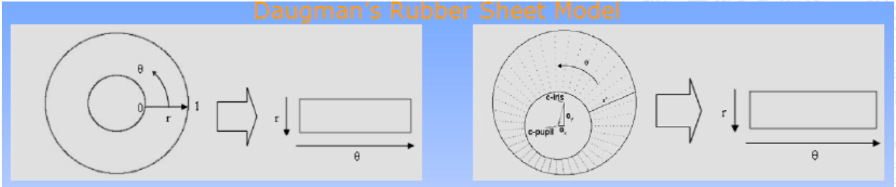
# Iris Recognition: Rubbersheet Transform, i.e. “Unwrapping” or “Unrolling”

The range of radial values between pupil border and sclera is mapped onto coordinate $r$ in the interval $[0, 1]$, the position on the circular arc is mapped to $\Theta$. Let $(x_i, y_i)$ be the original coordinates of iris pixels, $(x_0, y_0)$ the center of the pupil and $r_0$ the radius of the pupil, $M$ is the target distance between pupil and sclera border in pixels and $L$ this actual distance in the image.

$$r_i = \frac{M}{L}([(x_i - x_0)^2 + (y_i - y_0)^2]^{1/2} - r_0)$$

$$\Theta_i = arcsin\left(\frac{y_i - y_0}{x_i - x_0}\right) \quad \text{for} \quad y_i \geq y_0 \quad \text{otherwise plus Pi.}$$

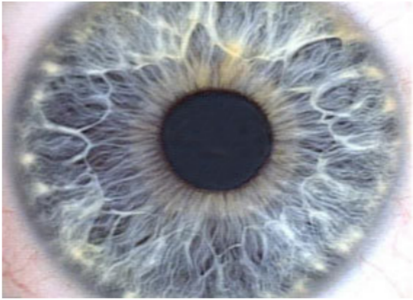
Daugman’s Rubber Sheet Model



Centers of iris and pupil coincide

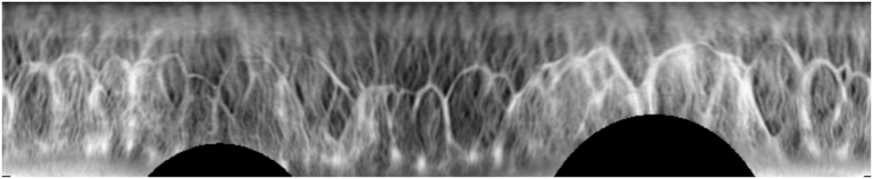
Centers of iris and pupil do not coincide

# Iris Recognition: The Rubbersheet Transform Example



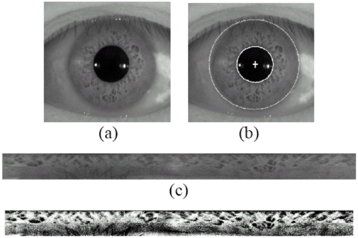
Infra-red illuminated image of iris

Iris image is ‘unwrapped’



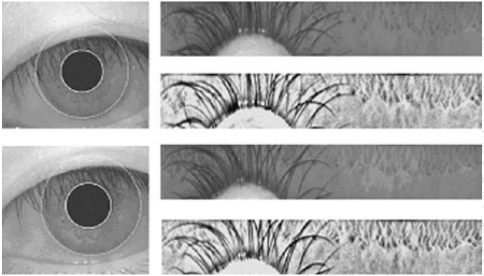
# Iris Recognition: Unwrapping in Practice

In case of large iris parts are occluded, recognition accuracy may suffer. In the left image, the ideal case is depicted, while the right image illustrates strong artifacts caused by lid and eye lash occlusions. These effects can be mitigated by user cooperation.

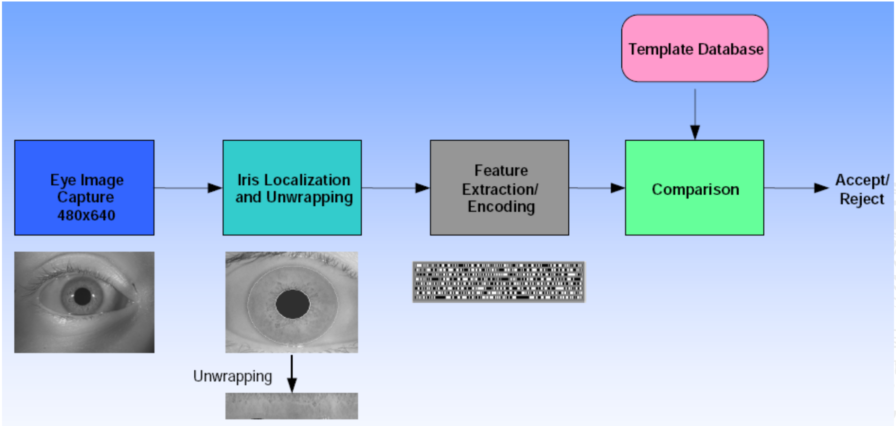

(a) (b)

(c)

(d)



# Iris Recognition: The Daugman Algorithm


Template Database
Eye Image Capture 480x640
Iris Localization and Unwrapping
Feature Extraction/ Encoding
Comparison
Accept/ Reject
Unwrapping

# The Daugman Algorithm: Gabor Features I

In the original scheme, 2D Gabor functions are employed;

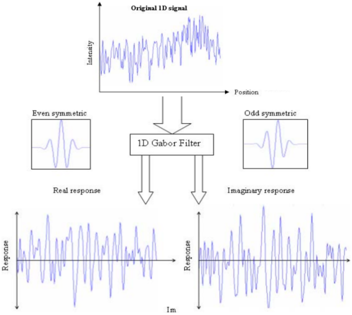
$$\Psi(x,y) = e^{-\pi[(x-x_0)^2/\alpha^2+(y-y_0)^2/\beta^2]}e^{-2\pi i[u_0(x-x_0)-v_0(y-y_0)]}$$

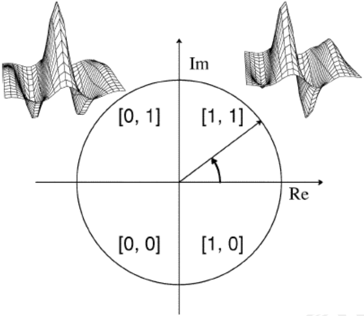
with $(x_0, y_0)$ die position of the function, $(\alpha, \beta)$ its length and width, $(u_0, v_0)$ are modulation parameters which provide frequency $\omega = (u_0^2 + v_0^2)^{1/2}$ as well as orientation $\Theta = arctan(v_0/u_0)$ in polar coordinates.
As Gabor functions are complex valued, we can use the real and imaginary part of their convolution $*$ with image $I(x,y)$ as image feature wrt. local amplitude and phase. The corresponding phase angle is

$$\phi(x,y) = tan^{-1}\frac{Im\{\Psi(x,y) * I(x,y)\}}{Re\{\Psi(x,y) * I(x,y)\}} \quad .$$

Quantising this phase angle to two bits results in the so-called “Iris Code”.

# The Daugman Algorithm: Gabor Features II



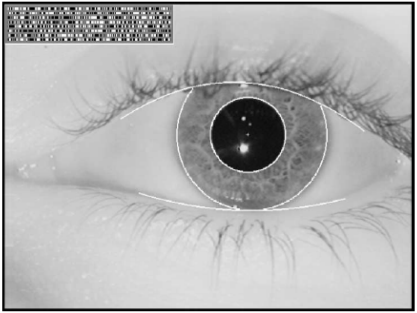


The convolution $* = G(r_0, \Theta_0, \alpha, \beta, \omega) = G$ applied to the iris texture $I(r, \Theta)$ is defined as (note that Gobor functions change wrt. their frequency but not wrt. their orientation):

$$G = \int_r \int_\Theta e^{-i\omega(\Theta_0 - \Theta)} e^{-(r_0 - r)^2/\alpha^2} e^{-(\Theta_0 - \Theta)^2/\beta^2} I(r, \Theta) r d\Theta dr \quad .$$

# The Daugman Algorithm: Gabor Features III

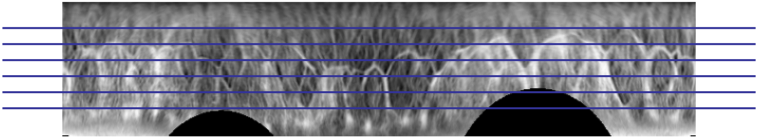
In case $Re\{G\} \geq 0$, the first Bit is set to 1 (= else), in case $Im\{G\} \geq 0$ the second bit is set to 1. Using this approach, 2048 phase bits (i.e. 256 bytes) are computed (8 scales and corresponding frequencies, 2 bits per position, at 128 positions $(r, \Theta)$.

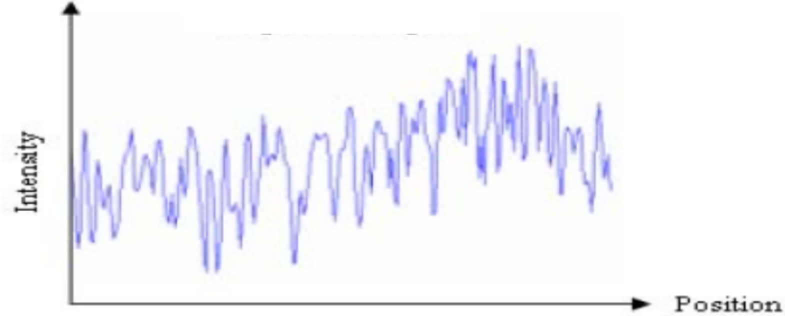


Eventually, additional 2048 masking bits are computed to indicate if the corresponding part of the Iris Code must not be considered in matching due to low quality or occlusion. For matching to codes, the Hamming distance is computed between two Iris Codes (counting the number of non-identical bits).

Head tilt is compensated by a circular shift of the Iris Codes and taking the minimum Hamming distance. Note that due to the usage of sign changes in the phase, the position of zero-crossings in $\Theta$-direction is coded implicitely.

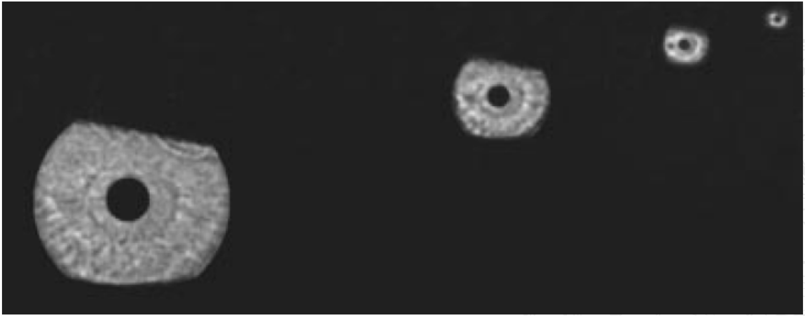
# Iris Recognition: The 1D Masek Implementation





# Iris Recognition: The Wildes Algorithm

Developed in parallel to the Daugman algorithm, however, much less suited for commercial deployment. Iris texture is represented as a Laplacian pyramid, i.e. recursive application of a Gauss function with downsampling in each stage. The lowest resolution image as well as three differential images are stored (compare hierarchical progressive JPEG for the prediction – interpolation scheme).



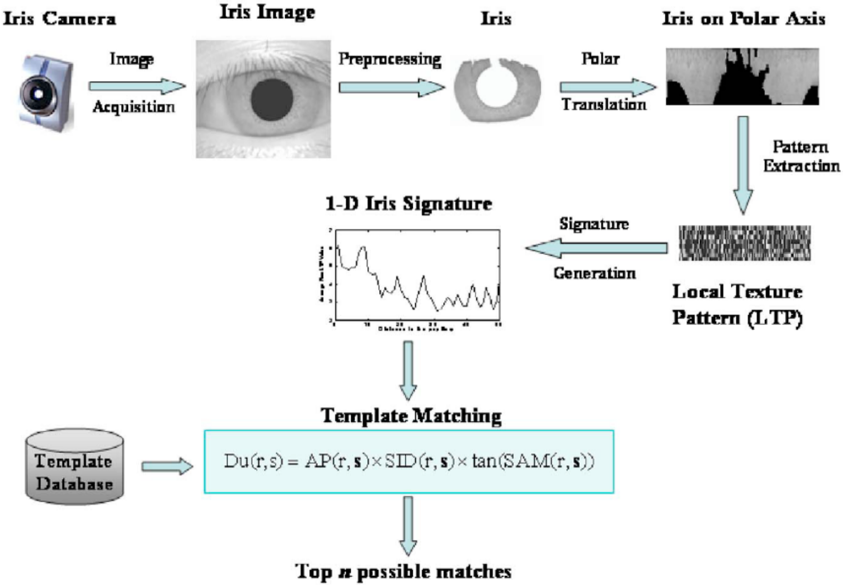
# The Wildes Algorithm: Features and Matching

Alignment is achieved following the pupil center and rotation is compensated during matching. For the matching itself, the normalised correlation between two sub-images $p_1(i,j)$ and $p_2(i,j)$ is computed, where $\mu_1$ and $\sigma_1$ are mean and variance of $p_1$, $n, m$ are image dimensions:

$$\frac{\sum_{i=1}^{n}\sum_{j=1}^{m}(p_1(i,j)-\mu_1)(p_2(i,j)-\mu_2)}{nm\sigma_1\sigma_2}$$

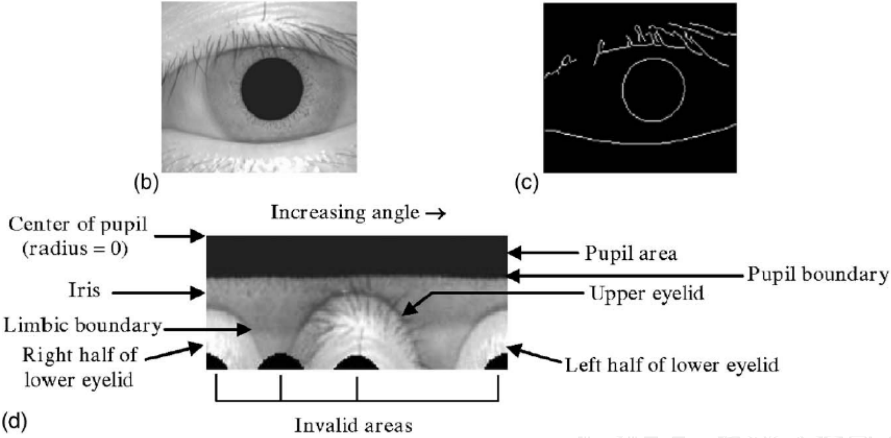
The implementation computes correlation on 8 × 8 pixel blocks in each of the 4 “frequency bands”. In each band the median of all blocks is used, which leads to 4 matching values which need to be combined. Overall, the algorithm is by far less efficient as the Daugman scheme (how to weight the four bands ?) which is one reason for the adoption of the Daugman scheme by Iridian (note that patent issues blocked other commercial deployments for years due to the wide scope of the patent).

# Iris Recogniton: The algorithm of Du



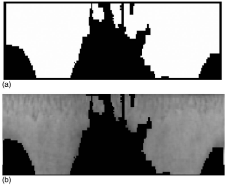
$$Du(r,s) = AP(r,s) \times SID(r,s) \times \tan(SAM(r,s))$$

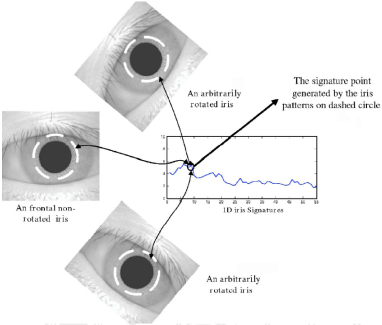
# The Algorithm of Du: Preprocessing



# The Algorithm of Du: Features

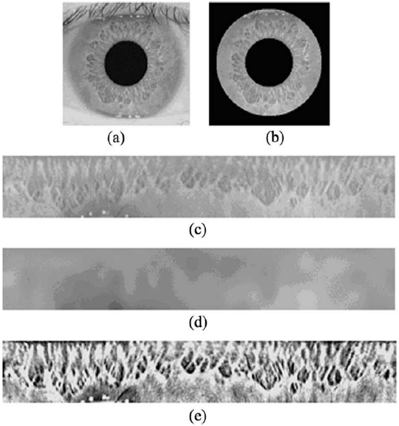
In preprocessing, the mean of a local window is substracted from the pixel values in that window which only leaves oscillations around the mean (which are more prominent close to the pupil). Subsequenty, all values in a line of the unwrapped iris texture (which roughly corresponds to all pixel data on a concentric ring) are summed up (in case more than 66% of the pixels are iris texture and are not occluded). These vales, differentiated wrt. their distance from the pupil center constitute a 1D signature. Note: This signature is entirely rotation-invariant !




An arbitrarily rotated iris
The signature point generated by the iris patterns on dashed circle
An frontal non-rotated iris
1D iris Signatures
An arbitrarily rotated iris

# Iris Recognition: Algorithms of Ma and Boles – Preprocessing

Preprocessing is similar using unwrapping as well as substraction of the local mean. Ma additionally employs histogram equalisation in a 32 × 32 pixels window. Similar to the Masek implementation, M neighbouring lines in the unwrapped texture image are averaged on a pixel basis, resulting in a 1D signal per annulus. 78% of the data (annuli closer to the pupil) are used to create 10 1D signatures. Rotation invariance is achieved by translation of the iris texture during matching.

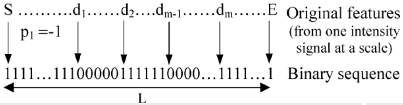


# Algorithm of Ma: Features

Both algorithms are based on multiscale representations using a redundant (non-downsampled) wavelet transform – two scales are selected and in particular, significant variations in the data are emphasised.

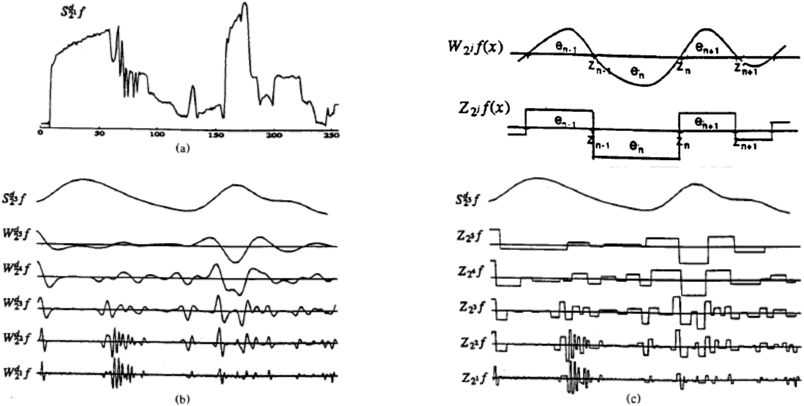
Ma consider the local extremal values of two detail signals and consider neighbouring pairs of extremal points (exhibiting sufficient amplitude difference to guarantee significant extremal property) – one local maximum and one local minimum. The positions of these extremal points are stored, for both considered scales and all 1D signatures. This is the provisorial feature vector.

These vectors are converted into binary ones subsequently (to be able to apply Hamming-distance based matching): At each position indicated in the feature vector, a 0 is changed to 1 (and vice versa), the start value is defines by the type of feature vector; thus, the binary vector has the double size of the original signal (2 scales).



# Algorithm of Boles: Features

“Wavelet Zero Crossings” has been developed to represent wavelet coefficients more efficiently but still offer good quality reconstruction. Positions of zero crossings in the detail signals are stored together with a constant value between those positions (the value is chosen to maintain the integral between two zero crossings).



# The Algorithm of Boles: Matching

Let the Zero-crossing representation of a 1D signature $f$ at scale $j$ be given as $Z_jf$. $Z_jf$ can be represented uniquely by a set of ordered complex numbers the imaginary part of which is the position of the zero crossings while their real parts represent the magnitude of $Z_jf$ betwenn adjacent zero corssings. To be able to apply a distance measure directly, the number of zero crossings at the corresponding scale needs to be identical – it is suggested to only use this approach in case two neighbouring scales are found which fulfil this condition. Alternatively, it is suggested to directly compute the distance between the $Z_jf$ of the 1D signaturees which need to be matched (which is much more expensive in terms of computation) or to postprocess the zero-crossing representation to result in an equal number of zero-crossings (elimination of “incorrect” crossings). Rotation invariance is again achieved by shifting 1D signatures against each other.

# Iris Recognition: The Algorithm of Zhu

Gabor and DWT filtering (Daub4 filter) are applied to the normalised iris texture, the resulting subbands are represented by their means and standard deviations – methods of classical texture classification are employed. The employed Gabor filter of opposite symmetry $h_e(x, y)$ and $h_o(x, y)$ are given as

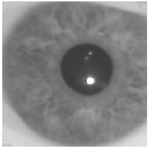
$$h_e(x, y) = g(x, y) cos[2\pi\omega(xcos\Theta + ysin\Theta)]$$

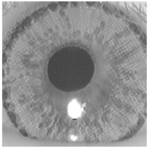
with $g(x, y)$ a 2D Gauss function and $h_o(x, y)$ with *sin*[] instead of *cos*[].
24 different combinations of $\omega$ und $\Theta$ are used (resulting in 48 features) while in theDWT case 13 subbands are used (26 features). Rotation and scale invariance is achieved by statistical texture description, the actual iris pattern is lost and the accuracy is therefore clearly lower.

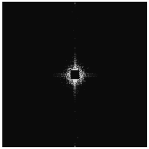
# Iris Recognition: Spoofing Detection

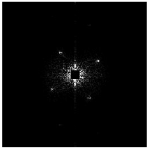
An obvious approach to conduct spoofing attacks is to use fake contact lenses with printed iris patterns. Liveness detection can be used by measuring changes in pupil dilation when illumination conditions are changed (which is not a time efficient way to do that of course – lowering illumination would cause the pupil to grow much larger which cannot be seen/detected due to the lens covering most pupil parts).

Other possibilities include measuring the hippus (righ resolution in both spatial and temporal domain required) or detecting the edge of the lenses by sensitive edge detection techniques. The illustration shows that the amplitude of the DFT exhibits priniting artifacts, but this example has an exaggerated poor quality which makes this detection easy.


Natural iris


Fake iris printed on a contact lens


2D Fourier spectrum of natural iris


2D Fourier spectrum of fake iris

# Iris Recognition: Products

Daugman technology was commercialised by Iridian (`http://www.iridiantech.com/`) for years, after the patent expired some years ago, many companies in the business now offer iris recognition technology. Currently Sarnoff and Morpho are among the largers deployers of iris Technology.

One of the largest installations is the UAE immigration watchlist (IrisGuard Technologie (`http://www.irisguard.com/`) installed at all border stations, where a daily number of up to 10000 incoming people are matched against a 1 million entries watchlist.

# Iris Recognition: Frequent Flyer Programs

Has been experimentally installed in many airports, e.g. Shipol (NL), Narita (Japan) and Frankfurt (the latter system was installed by Byometric Systems Inc. from Ainring with automated adaption to passenger height).

# Iris Recogntion: Further Products

Airport security areas, Securimetrics (`http://www.securimetrics.com/`) produces portable devices for enrollment and verification (partially multimodal), application in military and refugee / migration control, also in the Indian UID-Aadhaar project (`http://www.uid-aadhaar.com`) an iris scanner is used.

# Fingerprint Recognition: Basics

Fingerprints are to most frequently employed biometric modality, the modality investigated most thoroughly and with the longest history and also the modality with highest turnover (about 50% of all turnover involving biometrics is gained with FP systems). FPs are formed in the 7th month of pregnancy and exhibit some degree of similarity in case of close relatives.

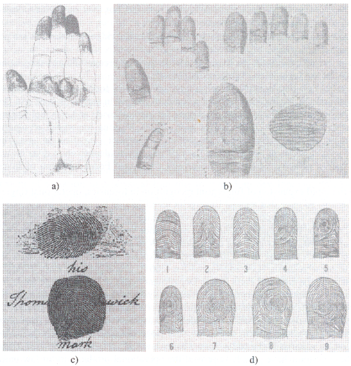
Sensors are in the lower price segment due to mass production (e.g. FP-mouse, FP-stick) and we see many applications in forensics and governmental fields (border control – e-Passport, social security), but a fast growing purely commercial sector (access control – FP readers at front doors, PDA, ATM, Mobile).

Advantage is high acceptance (compared to iris) and the availability of multiple fingers (supporting several systems and eventual compromise). User cooperation and overt systems are required for data capturing (compared to face). Disadvantage is the emotional connection to forensics, injuries or skin diseases, spoofing techniques. Ageing robustness (i.e. permanence) is currently being investigated.

# Fingerprint Recognition: Pre-history

Fingerprint patterns can be found on archeological specimens, as here e.g. 5000 and 2000 B.C. – however it is unclear if the importance for individuality was known.

# Fingerprint Recognition: History



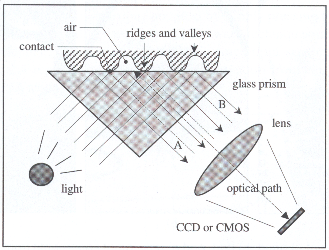
In 1684 the first scientific manuscript about FP property published (a). Another publication provided detailled anatomical FP descriptions in 1788 (b).

In 1809, a FP was used the first time as trademark by Thomas Bewick (c). 1823 the first procedure was suggested to classify FPs based on fundamental ridge structure into 9 classes (d). In 1888, the term “minutiae” has been introduced by F. Galton who also defined several FP features, while in 1899 the Henry-System for FP classification has been defined (which is still in use). The endpoint of historical development can be seen as the introduction of the FBI FP database with 810000 FP, which comprises more than 300 million FPs nowadays, requiring automation.

# Fingerprint Sensing: Off-line Acquisition

Even though FP sensors have been developed over 30 years ago, still an ink-based procedure is partially used in forensics: The finger is covered with ink and pressed against cardboard. This procdure allows to capture the FP form one side of the nail to the other providing more information content compared to “flat” FP sensors. However, imprint quality depends on a uniform ink application and the finger condition (sweat, grease). In forensics, it is custom to take ten-prints, constrasting to commercial applications. Captured prints are digitised by scanners or cameras to be used in an AFIS (automated fingerprint identification system). Thus, current forensic dataset often contain off-line as well as on-line FPs, which complicates automated matching. In forensics, “latent” FPs are specific – in case a finger touches an object, a veil of moisture and grease is applied to the object representig the structure of the ridges. Several techniques have been develop to improve the quality of FP for acquisition and processing (e.g. powder).

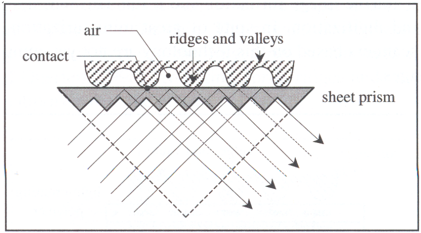
# Fingerprint Sensing: Optical Sensors I



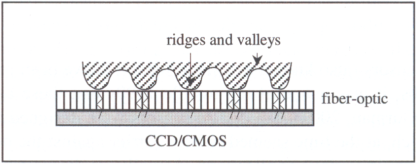
FTIR (Frustrated Total Internal Reflection) is the oldest but still most used technique. The finger touches a glass prism – ridges touch the glass while valleys keep a distance. The left side of the prism is illuminated (LEDs), light is absorbed at the ridges but reflected at the valleys – in this way valleys are represented bright and valleys dark. The reflected light is focused to a CCD sensor by a lens. The resulting distortion (A and B are not equally long !) is corrected optically or numerically. It is difficult to miniaturise this sensor type, e.g. for small devices like mobiles or PDAs.

# Fingerprint Sensing: Optical Sensors II

FTIR with a composed prism ("sheet prism") reduce the sensor size while trading off image quality.

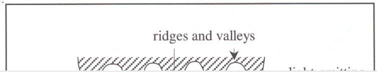


Prism and lense can be replaced by a fiberglass plate ("optical fiber") with directly attached CCD sensor. The finger residual light is transfered by the fiber and directyl acquired (without additional illumination). The high sensor area causes high production costs.



# Fingerpritn Recognition: Optical Sensors III

Electro-optical sensors consist of two layers: A polymer which emits light in case of applied voltage, the strength of which depends on the potential present at one side. Since ridges touch the polymer and valleys do not, the potential is different causing a different extent of emitted light. The second layer is an array consisting of photo diodesm which convert the emitted light into a digital image. Can be built in very small scale, but the quality is below the FTIR one.

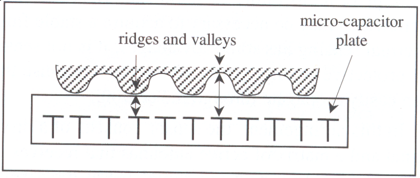


Direct photographic capturing has been suffering from low contrast, further, it has been shown to be sensitive to illumination and spoofing attacks and finger alignment has to be provied.

Recently, Safran Morpho has come up with the contactless “Finger on-the-fly” system.

# Fingerprint Recognition: Silicon Sensors I

Also denoted as “solid-state sensors”. Available since the mid 90ies; all types consist of an array of pixels, where each pixel is a small sensor – the user directly touches the silicon surface. There are several variants depending on how phyiscal information in converted into electrical signals.



Capacitive Sensors consist of micro-condensators embedded in a chip, the second condensator plate is the finger itself. When touching the sensor small electric charge is generated between finger and silicon plate, the size of which depends on the distance to the condensator plate (ridges vs. valleys). The protection layer of the surface if of high importance (but must not be too thick) as well as is resistance against electro-static discharge. These sensors cannot be fooled by a FP image of course.

# Fingerprint Recognition: Silicon Sensors II

Thermal: Sensor consist of pyro-electrical material whch generates voltage depending on temperature differences. Ridges touch the sensor, thus their temperature is measured which the valley areas are not measured. The image corresponds to the temperature of the ridge areas and the ambient valley temperature. These sensors require heating in case the surrounding temperature almost equals that of a finger and the images disappear quickly due to the fast equalisation of

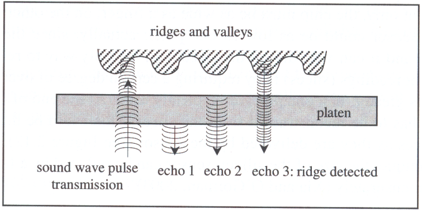
Robstness against electrical discharges and less sensitivity against thicker protection layers are advantages of thermal sensors.

Piezo-electric: Sesnors are sensitiv to pressure and generate asignal according to the extent of pressure which is higher in ridge areas as compared to valley areas. Current material is not sensitive enough, protection layers make troubles and the result is a binary image only.

Electric field: The sensor generates an electrical field which is influenced by the structure of skin sur-

# Fingerprint Recognition: Ultrasound Sensors

In ultrasound acoustic signals are emitted which are reflected by ridge and valley structures. The reflected signals are captured by measurement units and the different timings in receiving them is used to compute the traveled distance. Signals reflected by ridges have a shorter way obviously.



The quality of generated images is very good, furthermore the images are not impacted by sleeves, dirt etc. since lower parts of the derma are captured. The size and price of ultrasound systems is problematic is is the rather long capturing time (a few seconds) – resulting in few actual deployments. Similar properties are found in recent optical coherence tomography (OCT) sensors.

# FP-Sensing: Touch

Pressing / positioning the finger on the surface of the sensor has certain disadvantages (although no user training is required):

- The sensor gets dirty quickly, which lowers user-acceptance and reduces capturing quality.
- A latent FP is left on the sensor which could be fraudulently captured.
- A finger might be placed on the sensor in highly rotated manner which might cause problems for some matching technqiues.
- The sensor costs are relatively high due to its area and with the size the number of faulty chips increases.

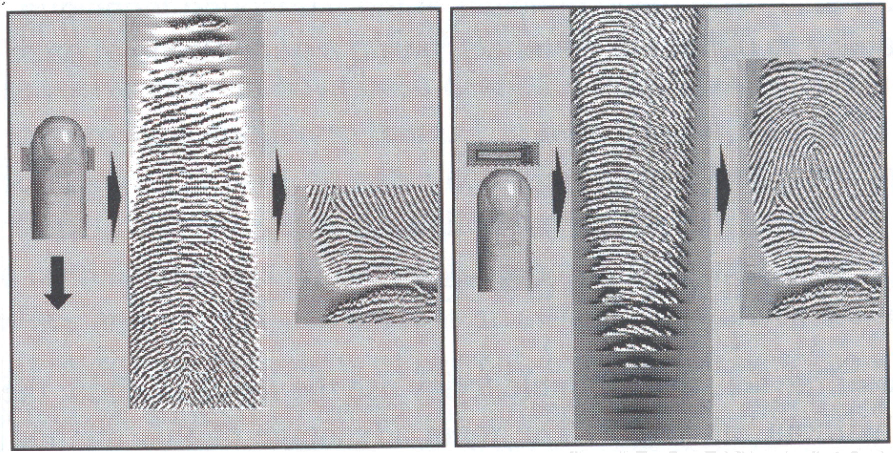
# FP-Sensing: Sweep

Due to the disadvantages of touch-based sensor, there are FP sensors, where finger are swept over the sensor surface. Due to the verttical movement, the width of the sensor is limited to finger width, while height may be fairly low. For combining captured stripes in robust manner, the height need to span across several pixels (to have some overlap). This concept has been originally developed for thermal sensors due to the fast temperature equalisation when touching the sensor.

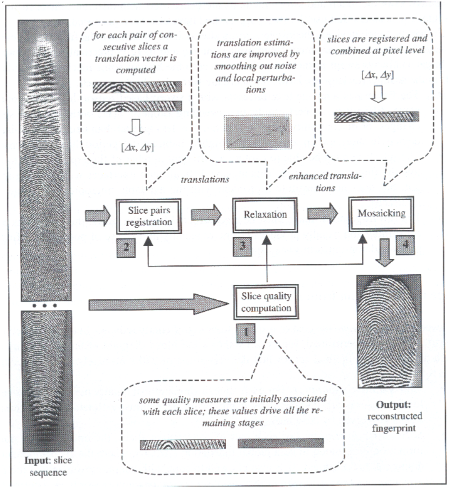
Advantages include lower costs due to reduced sensor area, better cleanliness due to contineous cleaning (the sweep itself) and no rotational positioning problem. Disadvantages include:

- Users require training for uniform sweep and correct speed
- Sensor needs to be fast to follow the sweeping speed
- The FP image needs to be reconstructed from the captured slices which can be demanding on low-power hardware and may introduce errors.

# FP-Sensing: Sweep Visualisation

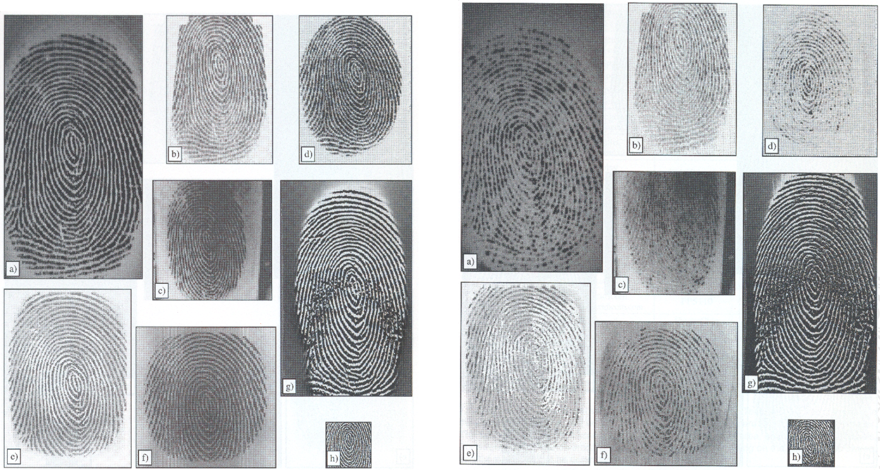


# FP-Sensing: Image reconstruction from FP-Slices


for each pair of consecutive slices a translation vector is computed
[Δx, Δy]
translation estimations are improved by smoothing out noise and local perturbations
slices are registered and combined at pixel level
[Δx, Δy]
translations
enhanced translations
Slice pairs registration
2
Relaxation
3
Mosaicking
4
Slice quality computation
1
some quality measures are initially associated with each slice; these values drive all the remaining stages
Input: slice sequence
Output: reconstructed fingerprint

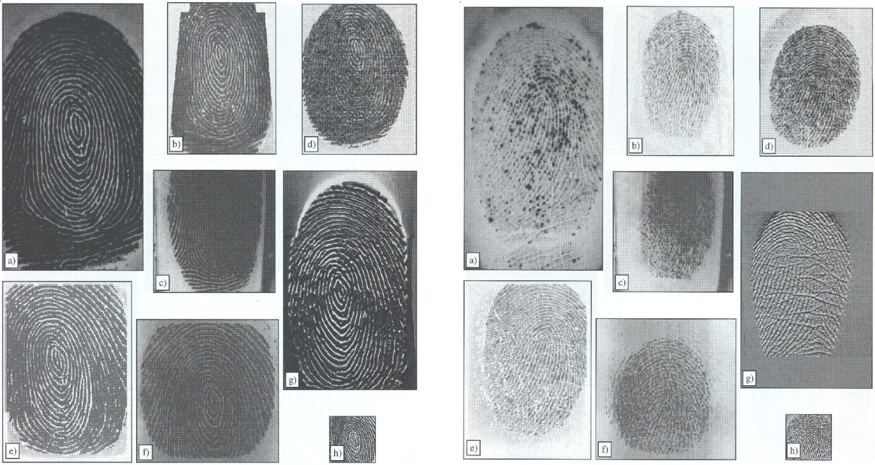
Slices typically have 280 x 30 pixels. After quality determination of each slice registration is done by determination of the global translation vector between to adjacent slices (usually using full search also allowing horizontal displacement with small extent. Finger movement is assumed to be contineous which is used to correct for statistical outlyers. Finally, the entire FP image is generated by computing a weighted sum of the pixels in the single slices.

# FP-Sensing: Visual Examples – normal vs. dry



a) FTIR b) FTIR c) Sheet prism d) Electro-optical e) Capacitive f) Capczitive g) Thermal (sweep) h) Electric field

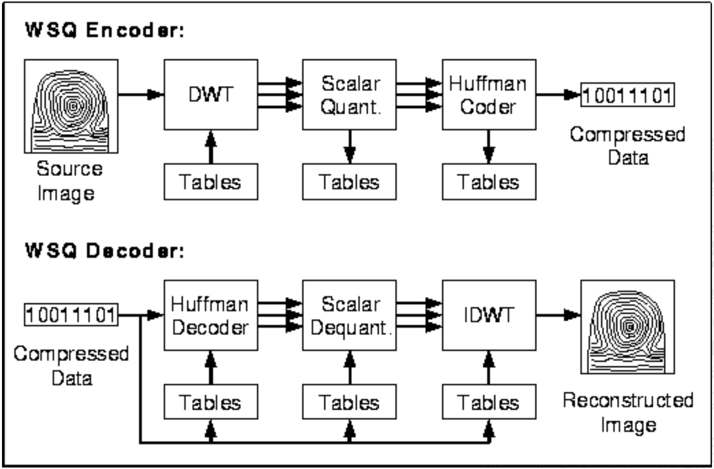
# FP-Sensing: Visual Examples – humid vs. low quality



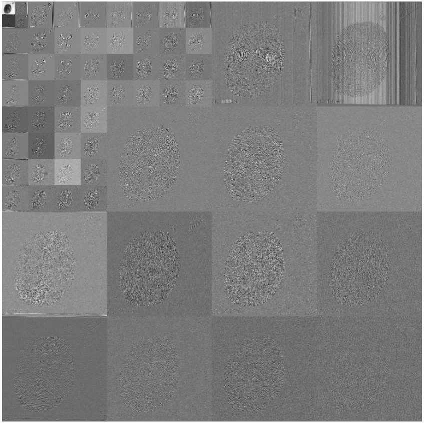
a) FTIR b) FTIR c) Sheet prism d) Electro-optic e) Capacitive f) Capacitive g) Thermal (sweep) h) Electric field

# FP - Kompression: FBI WSQ Standard I

ISO/IEC 19794-4 standard on Biometric Data Interchange Formats defined in its previous version three ways how to store FP image data in lossy manner: JPEG, JPEG2000 and **W**avelet **S**calar **Q**uantization (WSQ).



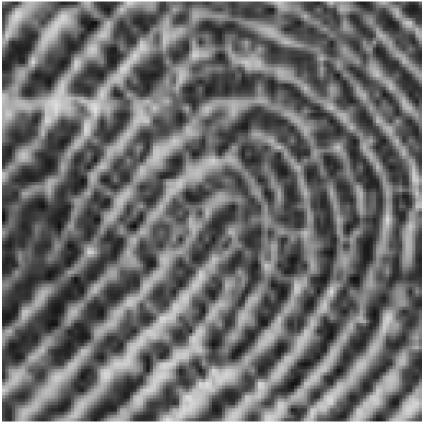
# FP - Kompression: FBI WSQ Standard II

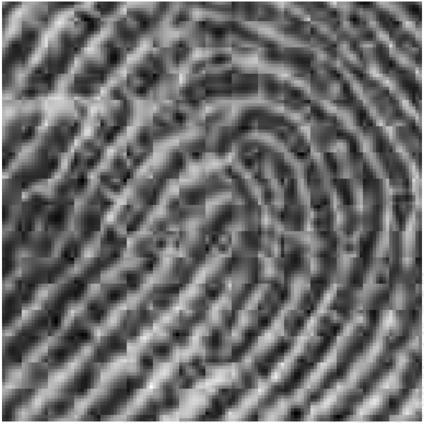


WSQ is a first generation wavelet-based image coder, i.e. no inter-subband correlations are exploited. The used wavelet packet subband structure deviates significantly from the pyramidal one (higher freuqncy resolution in mid-range frequencies) and was chosen according to the higher energy present in middle frequencies. This is caused by the ridge-valley structure which is present in the mid frequencies.

Similar subband structures have been used for FP compression in 2nd generation codecs ans well as in JPEG 2000 Part 2.

# FP-Compression: WSQ cs. JPEG (Ratio 19)





In this context is of interest, in how far quality measurements and visual impression have influence on AFIS matching (there is hardly literature on that).

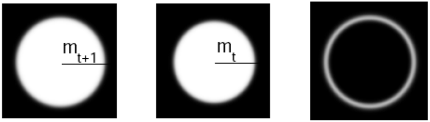
# FP Quality

Recent investigations emphasize the importance of FP image quality related to accuracy of AFIS. Automatised quality determination is important when acquiring a FP (if capturing needs to be repeated), when assessing the impact of enhancement procedures and then choosing which FP recognition system will be employed (the best methods at high quality do not need to be the best ones at low quality). Constrasting to general images there are specific FP quality measures (like considering the energy of specifically important WSQ subbands), since general measures like smoothness perform worse in general (however, we have shown the surprisingly good performance of generic image quality measures). Also when assessing compressed FPs these measures can be employed, in this case correlation to image quality mesaures is of interest !

NFIQ 1.0 and 2.0 are the “official” NIST quality measures, the first being based on the number of high quality minutiae, the second being a fusion of several quality concepts.

# Global FP Quality I

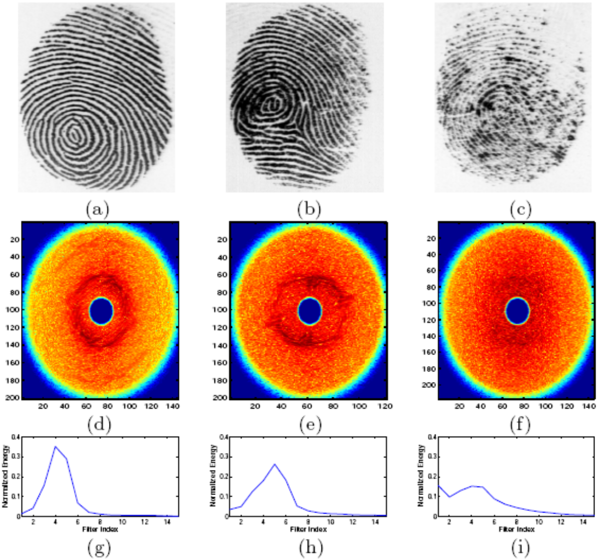
The idea is to analyse the DFT frequency spectrum of a FP. A high quality fingerprint should exhibit high energy in the dominating ridge frequency. In order to investigate this, repeated band-pass filterings are done, and the corresponding energy content recorded. Bandpass filtering can be implemented by computing the difference of two Butterworth low pass filtering masks in frequency space.



Energy concentration is determined by computing entropy (which is minimal in case of uniform distribution and grows in case of peaky distributions) – its deviation from the minmal value, normalised to $[0, 1]$ is computed. FPs on the subsequent page have quality $Q = 1.0,\ \ 0.6,\ \ 0.3$.

Criticism: Since we do not look for an arbitrary peak in the frequency spectrum, a peak in the important range could be modelled better.

# Global FP Quality II


(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

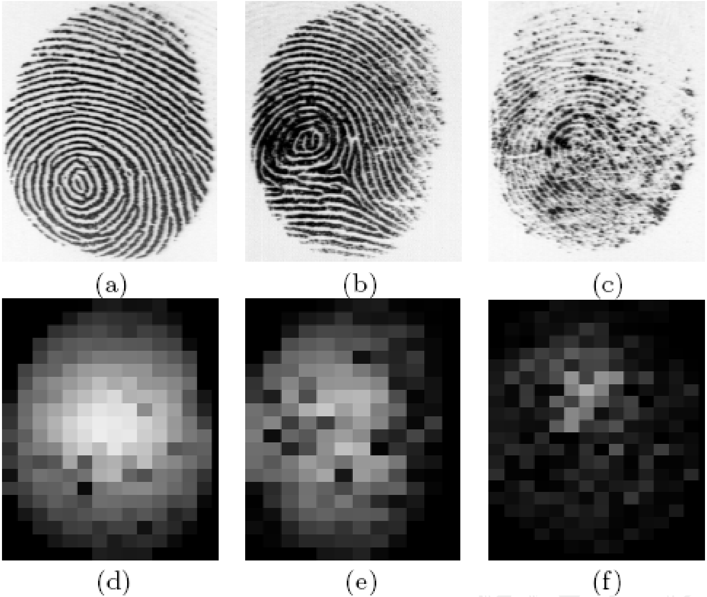
# Local FP Quality I

The image is partitioned into $b \times b$ pixel blocks. For each block, $g_s = (g_s^x, g_s^y)$ is the gradient at position $s$. The $2 \times 2$ covariance matrix of the gradients of all $b^2$ positions per block is given by

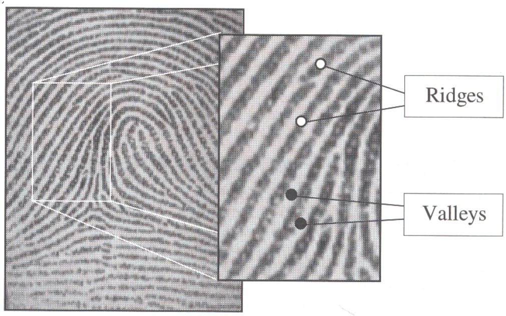
$$J_{i,j} = \frac{1}{b^2} \sum_{s \in B} g_s g_s^T$$

This symmetrical matrix has eigenvalues
$\lambda_1 = \frac{1}{2}(trace(J) + (trace^2(J) - 4det(J))^{\frac{1}{2}}$ and
$\lambda_2 = \frac{1}{2}(trace(J) - (trace^2(J) - 4det(J))^{\frac{1}{2}}$, where $trace(J) = J_{1,1} + J_{1,2}$ and $det(J) = J_{1,1}J_{2,2} + J_{1,2}^2$, with $\lambda_1 \geq \lambda_2$. A normalised coherence measure $0 \leq k \leq 1$ is defined as $k = \frac{(\lambda_1 - \lambda_2)^2}{(\lambda_1 + \lambda_2)^2}$ which expresses the distinctness of the ridge-valley orientation in each block (in case of good quality, $\lambda_1 \geq\geq \lambda_2$ and $k$ is close to 1).
The overall quality measure is computed by generating a weighted sum over all blocks, giving more weight to blocks close to the center.

# Local FP Quality II



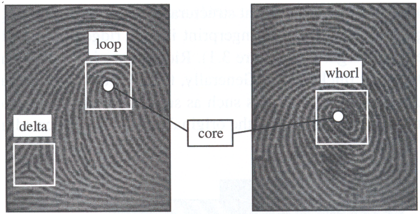
# FP Features: Ridges and Valleys



Ridge lines are usually dark and valleys bright, ridges having a typical width of 100 – 300 micrometers. Superficial injuries like cuts or light burning do not modify the pattern since it is duplicated by re-growing skin. Ridges and valleys are parallel in most situations, they sometimes part or finish.

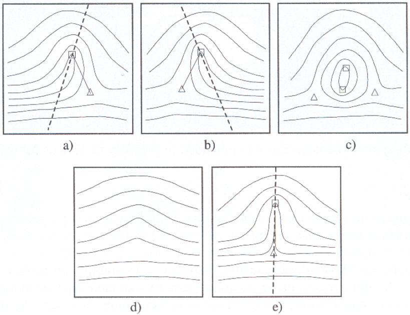
# FP Features: Singular Points

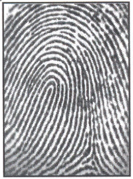
Considering ridges and valleys on a global FP scale, there are several areas in which ridges have a certain appearance (high curvature, frequent endings etc.) – these singular regions can be classified into three types: loop, delta, and whorl. The “core” is the north-mostern singularity following the Henry classification.



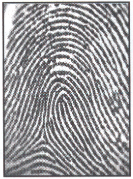
If there is no singularity with defined center, the core is defined to be the point with maximal ridge curvature. The core is of high importance in case it is used for alignment (compare pupil for iris and optical disc for retina). Different types of singularities are used to classify FPs (e.g. to speed up search by restricting it to a single class).

# FP Features: Henry-Galton Classification

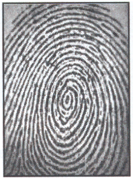




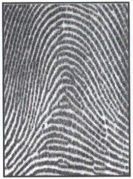
Left loop



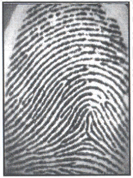
Right loop



Whorl



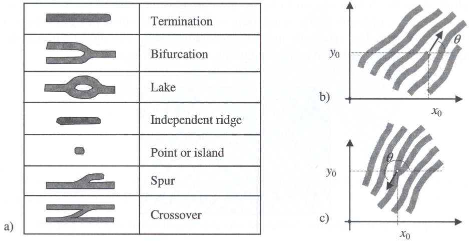
Arch



Tented Arch

# FP Features: Minutiae

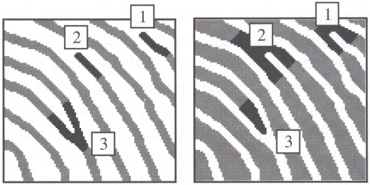
Minutia means small detail – in the FP context this refers to non-contineous ridge parts: Endings, bifurcations, trifurcations and undefined tye (ANSI typification), FBI only uses endings and bifurcations. Galton discovered, that minutiae do not change over lifetime.

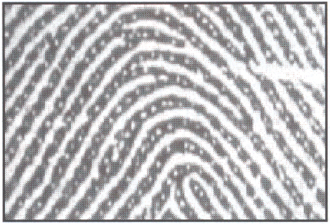


In addition to using minutiae coordinates, often the angle between ridge-tangent and the horizontal axis is used, in case of bifurcations also the corresponding angle between ending valley and the axis is employed.

# FP Features: Minutiae and Sweat Pores

In practice, depending on the pressure on the sensor an ending might be a bifurcation and vice versa. Additionally, there is the temrination / bifurcation duality: considering the original image and its negative version, endings in the original correspond to bifurcations in the negative image and vice versa.





In case FPs are acquired with high resolution, it is possible to identify sweat pores on the ridges (60 – 250 micro meters); although their position, number and shape represent a useful feature, this is hardly used due to the required high resolution and high quality of FPs.

# FP Features: Local Ridge Direction I

The angle $\theta_{x,y}$ at position $(x, y)$ in the FP image is usually defined as the angle between ridge tangent and horizontal axis, it is determined in $[0, 180]$ since there is no “flow direction”. Ridge orientation is not determined for each pixel but only at points on a less dense grid. The “orientation image” displays an average ridge orientation per image block, in some cases a degree of reliability is computed (compare local quality).

The gradient $\Delta(x_i, y_i)$ in $(x_i, y_i)$ is a two-dimensional vector with components $\Delta_x$ and $\Delta_y$. The angle $\theta_{x_i,y_j}$ is orthogonal to the gradient ($\theta_{x_i,y_j} = arctan\frac{\Delta_y}{\Delta_x}$).

This technique has problems at 90 degrees (caused by the denominator in the fraction) and when computing average angles: The average of 5 and 175 degrees is not 90 but 0 degrees and the average of 0 and 90 degrees might be 45 or 135 degrees.

# FP Features: Local Ridge Direction I

Based on the idea to double all involved angles, the following expression is suggested to compute $\theta_{x_i,y_j}$ in a 17 × 17 window:

$$\theta_{x_i,y_j} = 90 + 1/2 arctan\left(\frac{2G_{xy}}{G_{xx} - G_{yy}}\right)$$

with

$$G_{xy} = \sum_{h=-8}^{8} \sum_{k=-8}^{8} \Delta_x(x_i + h, y_j + k)\Delta_y(x_i + h, y_j + k)$$

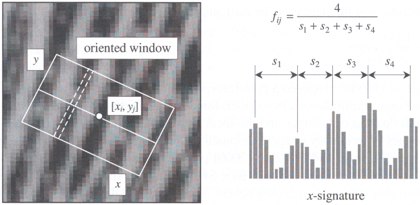
$$G_{xx} = \sum_{h=-8}^{8} \sum_{k=-8}^{8} \Delta_x(x_i + h, y_j + k)^2$$

$$G_{yy} = \sum_{h=-8}^{8} \sum_{k=-8}^{8} \Delta_y(x_i + h, y_j + k)^2$$

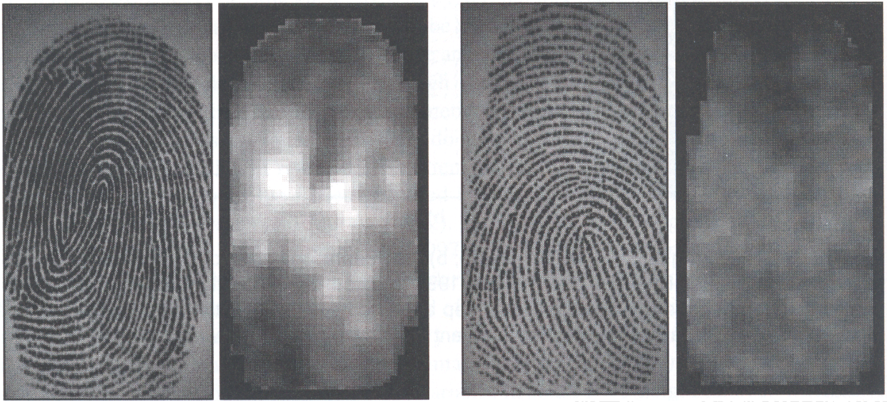
# FP Features: Local Ridge Frequency I

The local ridge frequency $f_{x,y}$ in $(x, y)$ in the inverse of the number of ridges in unit length along a segment centered in $(x, y)$ orthogonal to local ridge direction. By analogy to the orientation image a frequency image is defined, in which average ridge frequency is shown only at discrete grid points. This frequency varies among different fingers and among different areas of a single finger. $f_{x,y}$ is computed as:

1. a $32 \times 16$ window in $(x, y)$ is rotated parallel to the ridge orientation
2. the x-signature of the grey values is obtained by summing uo the values in each column of the window
3. $f_{x,y}$ is the inverse of the average distance between two local maxima of the signature

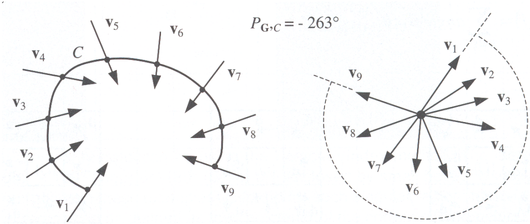


# FP Features: Local Ridge Frequency II



# FP Features: Detection of Singularities and Core I

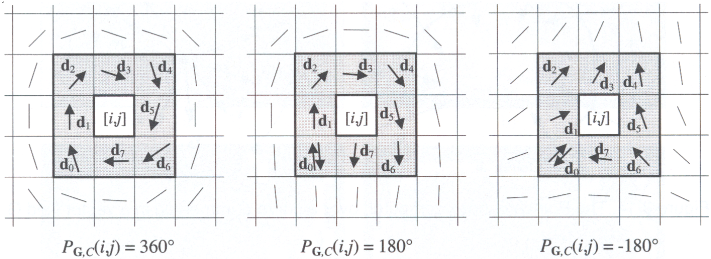
The most elegant technique uses the “Poincare Index”. In case G is a vector field and C a curve in this field, the Poincare index $P_{G,C}$ is defined as the overall rotation of the vectors of G along the curce C:



For FPs, G is the orientation image and C is a closed curve of elements of G such that $(x, y)$ is an inner point. $P_{G,C}(x, y)$ is computed by summing up all orientation differences between adjacent elements of C (here an oriented direction is required which is chosen randomly for the first vector). $P_{G,C}(x, y)$ only attains values of 0, +-180 and +-360 degrees for closed curves (for FP: $P_{G,C}(x, y) = 0$ no singularity, $= 360$ whorl, $= 180$ loop, $= -180$ delta.

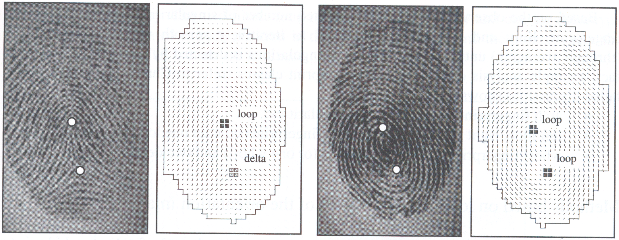
# FP Features: Detection of Singularities and Core II

The illustration shows three parts of orientation images. C is an ordered sequence of 8-neighbours $d_k$ of $(i, j)$. The orientation of the $d_k$ is defined such that $d_0$ points upwards; $d_k$ is oriented that the absolute value of the angles between $d_k$ and $d_{k+1}$ is $\leq$ 90 degrees.

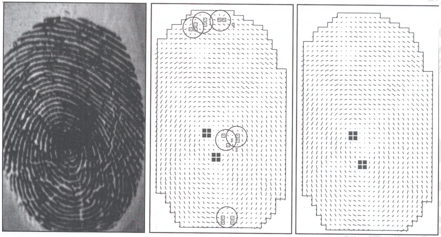


$$P_{G,C}(i,j) = \sum_{k=0}^{7} angel(d_k, d_{(k+1)mod8})$$

# FP Features: Detection of Singularities and Core III

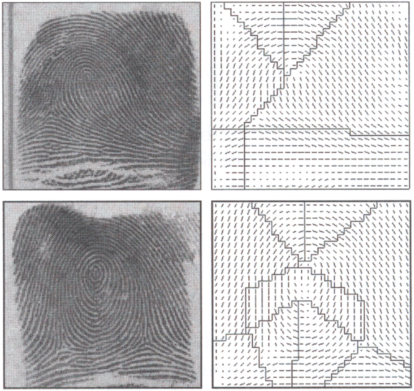


Smoothing the orientation image prevents detection of false singularities !



# FP Features: Detection of Singularities and Core IV

- Singularities are areas where the orientation image exhibits irregularities, i.e. high curvature, quickly changing orientation
- When partitioning the orientation image into homogeneous regions the intersection of border lines correspond to signularities
- Local templates can be used to find the core (“sextet technqiue” of the FBI)
- Focal point: Intersection of lines orthogonal to ridge orientation



# FP Enhancement

Classicial techniques like contrast enhancement, histogram equalisation, Wiener filerting, and normalisation are the first stages in FP enhancement. However, these techniques cannot resolve the problem of partially disconneted ridges.
Classical normalisation, given $I(x, y)$, $m$ and $v$ as well as the desired mean $m_0$ and variance $v_0$ is

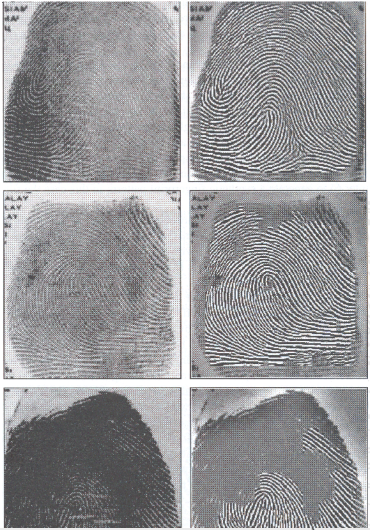
$$I'(x, y) = m_0 + ((I(x, y) - m)^2 v_0/v)^{1/2} \text{ in case } I(x, y) > m$$

and $m_0 - \dots$ otherwise. This operation does not affect the fundamental ridge and valley structure.
The most used technique is context-based filtering, where filter characteristics change with local context. For FPs, this local context is given by ridge orientation and frequency. The sinuidal pattern of the ridge and valley change is determined by these parameters and changes smoothly across the FP area. A filter optimised correspondingly is able to remove noise and other distortions while emphasizing the underlying structure.

# FP Enhancement usinf Gabor Filters

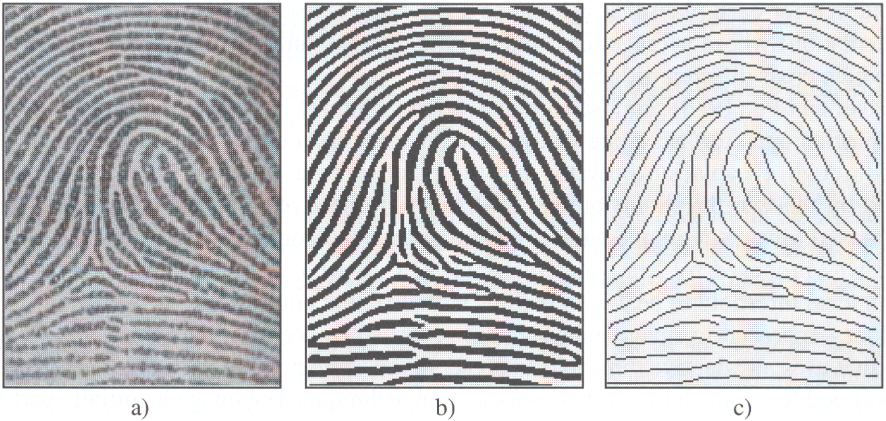
A set of Gabor filters is pre-computed corresponding to frequency and orientation as gien in frequency and orientation images. Each pixel is convolved which corresponds most closely to the orientation and frequency as pre-computed at this position (computing the optimal filter for each position is computationally too expensive).

Similar techniques have been developed using directional DFT bandpass filtering, however, this results in artifacts do to worse localisation.



# Detecting Minutiae

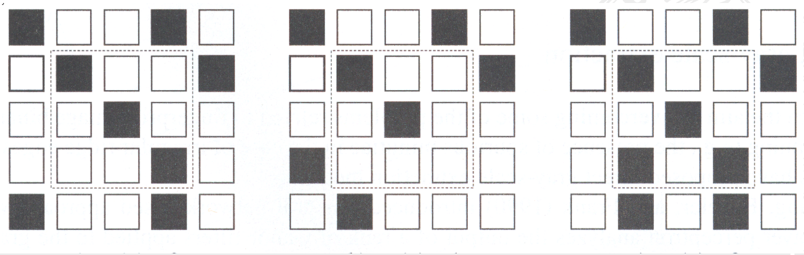
Classical strategy is FP enhancement followed by binarisation. Subsequently, thinning is applied resulting in 1-pixel wide ridges. The resulting image is searched for minutiae patterns using specific templates.



# Binarization and Minutiae Detection

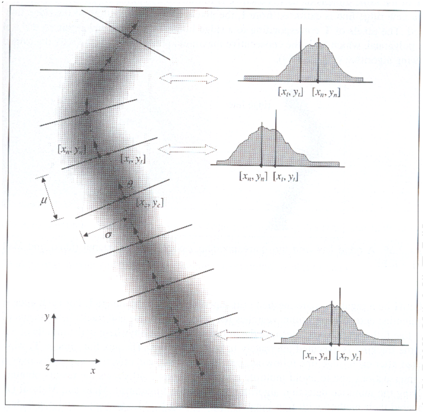
Simplest techniques used include local and global thresholds. Improvements are achieved by using x-signatures (local averaging) and using peaks and neighbouring pixels as forground. Further techniques apply ridge following and eliminate gaps, also morphological operators using structuring elements designed to match the ridge pattern are suggested.

After having computed the binary skeleton, each minutia pixel can be identified by a simple scan: a minutia pixel has a crossing number unequal to 2 (crossing number is the number of 8-neighbours in the binary image).



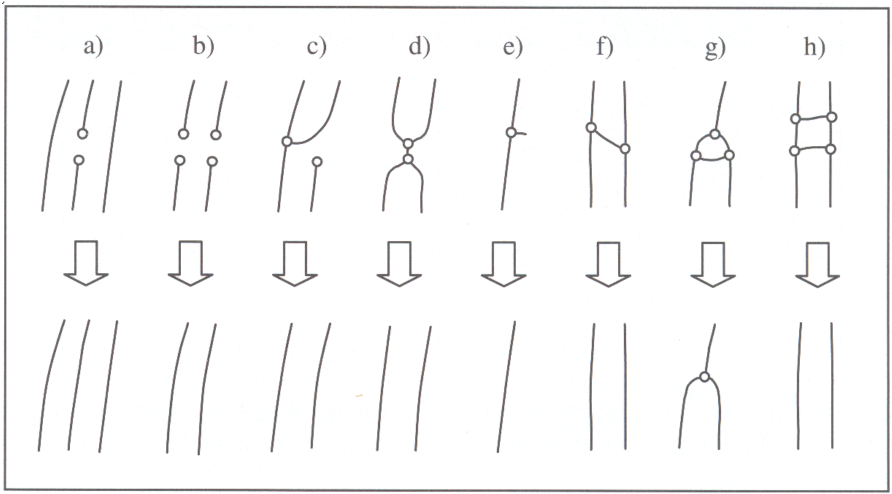
# Detecting Minutiae in Grey-scale Images

To avoid artifacts caused by binarisation and thinning, techniques directly applied to greyscale images are also proposed: local maxima of a segment positioned orthogonal to ridge orientation are computed and defined to be ridge center. In a second image, these helper-ridges are depicted with a constant additional seam and used to detect minutiae in this second image.



# Minutiae Filtering

To obtain good matching results, false positive minutiae need to be deleted. Below some examples are shown:



# FP Features: Ridge Count in Forensics

In Forensics, operators employ the number of ridges between specific points, usually using singularities as reference points. One possible technique is to compute the x-signature and determine the number of local maxima.

One of the biggest problems is the reliable determination of reference points – if this fails, the entire procedure is useless of course.

# Basics of FP Matching

We have given a gallery template T (from enrollment) and a probe image I. FP matching is complicated by high intra-personal variability caused by

- translation and rotation during acqusition
- non-linear distortions: Sensing maps a 3D stracture to the 2D FP image - caused by skin elasticity distortion arise in case of movement non-orthogonal to the sensor surface
- pressure and skin condition (moisture, grease, dirt)
- sensor noise
- errors in feature extraction

# Types of FP Matching

1. Correlation-based matching: Two FPs are superimposed and the correlation of corresponding pixels is computed for various translations and orientations.
2. Minutiae-based matching: The origin of this approach is in forensic manual FP comparison procedures, currently, most systems use this approach. Fundamental idea is to determine the alignment between T and I resulting in the maximal number of corresponding minutiae pairs.
3. Ridge-feature based matching: Used in low quality FPs where it is difficult to extract minutiae. Features like local ridge orientation and frequency, texture information etc. can be extracted more reliably under such conditions, however, these are usually less discriminative.

This distinction shows the importance of FP quality determination !

# Correlation-based FP Matching

$CC(T, I) = T^T I$ is the cross correlation between T and I – similarity between two FPs is computed by

$$S(T, I) = max_{\delta x, \delta y, \Theta} CC(T, I^{(\delta x, \delta y, \Theta)})$$

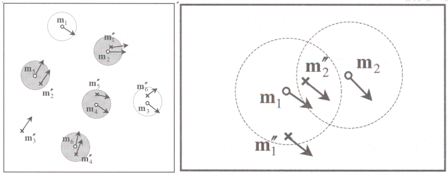
where $\Theta$ is a rotation of the images (around their center, center of gravity) and $(\delta x, \delta y)$ is a translation.
A direct application is hardly successful:

- Non-linear distortions lead to significant errors
- Skin condition and pressure cause a change in luminance, contrast and ridge thickness. This needs to be compensated during computation (normalised cross correlation, zero-mean cross correlation).
- Computational effort is significant.

Possible solutions include local correlation computed in windows (centered at minutiae or singularities) and complexity can be bounded by applying multiresolution techniques and computing correlation in the DFT domain (applying the convolution theorem).

# Minutiae-based FP Matching I

Matching to sets of minutiae is often called “Point Pattern Matching” – two minutiae are considered as matching in case their position distance and their orientation distance are smaller than set thresholds (and in case the type is a fit in case this is used). FPs needs to be aligned to find a mximal number of matching pairs: Translation and Rotation are usually permitted. Classically, the number of matching pairs is maximised, even though the error in matching pairs might be smaller in other configurations, however, also distances can be taken into account. The number of possible solutions is exponential in the number of minutiae.



# Minutiae-based FP Matching II

- Relaxation: The correspondance of minutiae pairs is evanuated iteratively considering neighbouring pairs (compare edge relaxation)
- Hough transform: An accumulation array is used, containing alignment parameters of FPs. A double loop over all minutiae in T and I is conducted, determining distance in x- and y-direction for all angles and scales and quantising them. The array is incremented at in case of matching minutiae at the corresponding position, the maximum in the 4D array shows the optimal alignment.
- Pre-alignement
  - Absolute: Translation according to core point position
  - Relative: Determination of a “prinicpal” minutiae pair, which is characterised by connecting lines, the angle and length of which are used. Alternatively alignement can be done according to singularities, orientation image correaltion, correlation of ridge features.

# Minutiae-based FP Matching: Local Methods

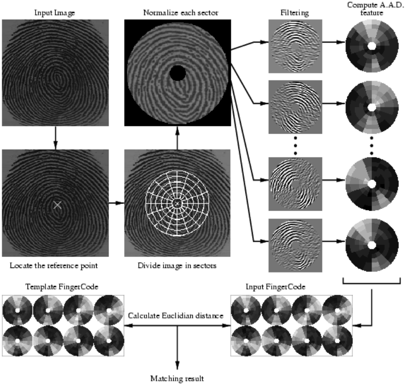
The idea is to employ local minutiae structures which are less sensitive against global transformations. Computational effort is lower, but the same is true for distinctiveness. A possible application uses local matching for alignment followed by a global FP matching stage.

Variants:

- Feature vector contains the number of minutiae of different types in an area.
- Relative orientation to the orientation of a central minutia, distance and ridge count to surrounding minutiae is also recorded.
- Graph notation: A star connected to a minutia consists of the nodes (minutiae in the neighbourhood) and the edges (connecting lies to those minutiae). Each star in T is matched against each star in I using different rotation angles.

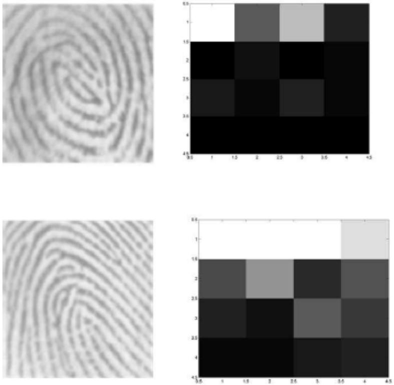
# FP Matching: Ridge Feature Methods - Gabor

The idea is to develop a FingerCode – after identification of a reference point (core - critical stage !) a normalised circle segment patter is superimposed over the FP (e.g. 80 segments). These segments are normalised and filtered using Gabor filters with frequency corresponding to the ridge frequency and 8 orientations. For each of these 640 segments the difference to the sector mean is used as feature. matching is done by computing Euclidian distance between feature vectors. Rotation invariance is achieved by repetitive matching of rotated FingerCodes.



# FP Matching: Ridge Feature Methods - DWT & WPT

Texture descriptos can be used where the FP is decomposed into a defined subband structure (DWT based or optimised wrt. matching). Subbands are partitioned into blocks, for which a feature based on coefficient magnitude is computed. In the feature vector, subbands with higher importance can be given more weight in the computation of the Euclidian distance. Again, a FP registration is required, robustness against a small extent of rotation comes for free.



# Outline

1. Formalia
2. Introduction & Motivation
3. Foundations of Biometrics
   - Introduction and Basic Terminology
   - Biometric Traits & their Properties
   - Biometric Systems System Model
   - Assessment of Biometric Systems
   - Security and Privacy in Biometric Systems
4. Non-visual Biometrics
   - Keystroke Dynamics
   - On-Line Signatures
   - Biorhythms
   - Speaker Recognition
   - Odour
5. Visual Biometrics
   - Off-line Signatures
   - Iris Recognition
   - Fingerprint Recognition

# Literature I