
Index

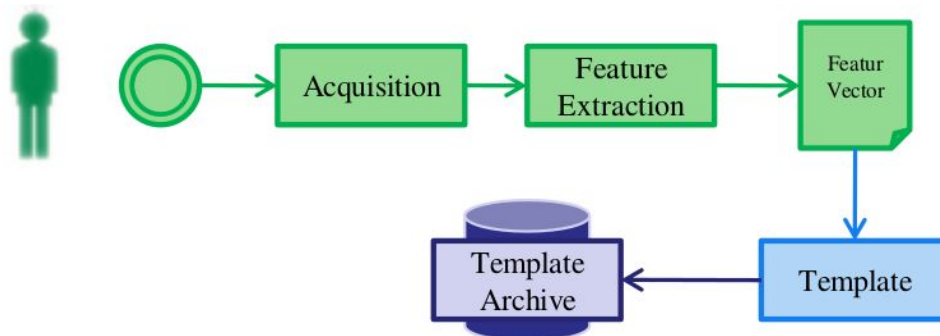
Introduction to Biometric Systems (Lesson 1)	1
Performance Evaluation (Lesson 2)	3
Possible errors on verification	3
More on performance evaluation (Lesson 2bis)	5
Response Reliability (Lesson 3)	6
Face recognition:introduction and face localization (Lesson 4)	7
Comparisons	8
Problems	8
Popular Databases	8
Structure of a face recognizer	9
Face Localization	9
More about face localization (Lesson 5)	10
Algorithm A (by Hsu, Mottaleb and Jain, 2002)	10
Algorithm B (Viola-Jones, 2004)	12
AdaBoost	12
Evaluation	14
Face Recognition 2-D (Lesson 6)	14
Image reduction	15
PRINCIPAL COMPONENT ANALYSIS (PCA)	15
LINEAR DISCRIMINANT ANALYSIS (LDA)	16
Features extraction	18
Gabor Filter	19
Local Binary Pattern (LBP)	20
Classification of recognition system	21
Face Recognition 3-D (Lesson 7)	22
3D model	23
3D face recognition	24
Face Recognition: evaluation (Lesson 8)	26
Face spoofing and anti-spoofing (Lesson 8 bis)	28
Ear recognition (Lesson 9)	30
Iris recognition (Lesson 10)	31
Fingerprints recognition (Lesson 11)	34
Failures of biometric system (Lesson 11bis)	37
Multibiometric system (Lesson 12)	37
Some research prototypes (Lesson 13)	39
FOVEA: video Frame Organizer Via identity Extraction and Analysis	39
Entropy of a gallery of templates (Lesson 14)	40

Introduction to Biometric Systems (Lesson 1)

Probe: each template which is submitted for recognition.

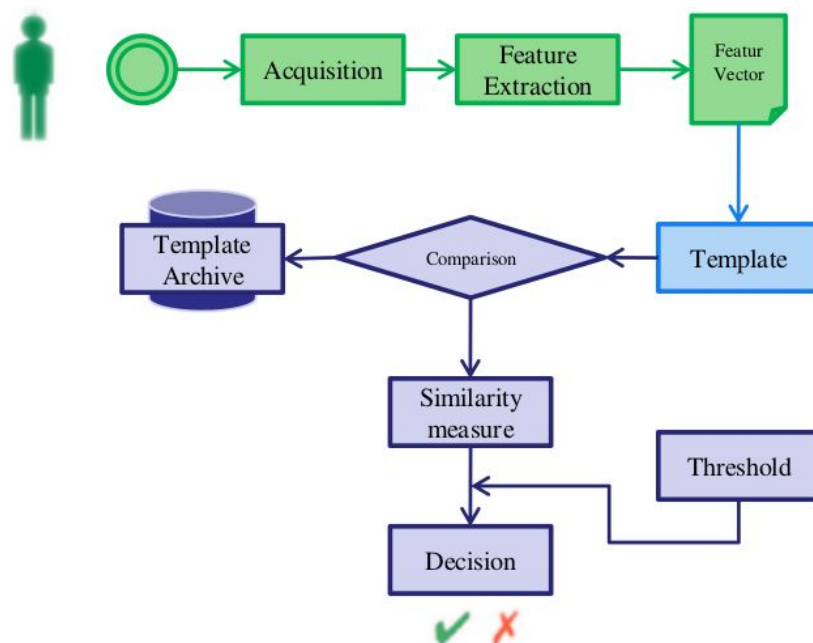
Gallery: the set of templates pertaining to enrolled subjects.

Enrollment: capture and processing of user biometric data for use by system in subsequent authentication operations (gallery).



Enrollment procedure

Recognition: Capture and processing of user biometric data in order to render an authentication decision based on the outcome of a matching process of the stored to current template. (Note: verification is always a 1:1 task, while identification is always 1:N)



Recognition Procedure

Type of environments:

- **Controlled:** capture settings can be controlled, distortions mostly avoided (e.g., for face, pose, illumination, and expression), defective templates can be rejected, and capture repeated (for instance entrance authorization in a company)
- **Uncontrolled/undercontrolled:** capture settings cannot be controlled, template can present various levels of distortion, defective templates can be rejected, but capture cannot be repeated (for instance video surveillance)

Types of recognition tasks:

- **Verification** the user claims an identity, possibly by presenting an ID card or other additional stuff. The system performs a 1:1 matching to verify the claimed identity.
- **Identification** the user does not make an identity claim; the system has to determine whether the biometrical trait of the subject matches one of the biometrical trait in the gallery. We have two variants here:
 - *Open-set identification:* the user may not be enrolled in the system
 - *Closed-set identification:* probes always have a correspondent match in the database.

Requirements for a (strong) biometrical trait:

- **Universality:** The trait must be owned by any person (except for rare exceptions)
- **Uniqueness:** Any pair of people should be different according to the biometric trait
- **Permanence:** The biometric trait should not change in time
- **Collectability:** The biometric trait should be measurable by some sensor
- **Acceptability:** Involved people should not have any objection to allowing collection/measurement of the trait

If permanence is not satisfied then we talk about weak biometric traits.

Performance Evaluation (Lesson 2)

Problems:

- **Wide Intra-class variation:** the biometric trait of the same subject can "look" different from the one stored in the gallery.
- **Small inter-class variation:** two different subjects may have biometric traits that look very similar.
- **Noisy or distorted acquisitions**
- **Non-universality**

Possible errors on verification

A subject is accepted if the similarity (or score) achieved from matching with the gallery template(s) corresponding to the claimed identity is greater than or equal to the acceptance threshold (or, if the distance with such gallery template(s) is less than or equal to the acceptance threshold). Otherwise it is rejected.

We can identify 4 possible cases:

- The **claimed identity** is **true** and the **subject** is **accepted** (Genuine Acceptance GA, also indicated as Genuine Match - GM)
- The **claimed identity** is **true** but the **subject** is **rejected** (False Rejection - FR, also indicated as False Non Match - FNM, or type I error)
- An **impostor subject** is **rejected** (Genuine Reject - GR, also indicated as Genuine Non Match - GNM)
- An **impostor subject** is **accepted** (False Acceptance - FA, also indicated as False Match - FM, or type II error)

A simple count of errors is not suitable.

We thus now define a few common measures for verification:

- **False Acceptance Rate** (FAR): Percentage of identification instances in which false acceptance occurs. For example, if the FAR is 0.1 percent, it means that on the average, one out of every 1000 impostors attempting to breach the system will be successful. Stated another way, it means that the probability of an unauthorized person being identified as an authorized person is 0.1 percent.

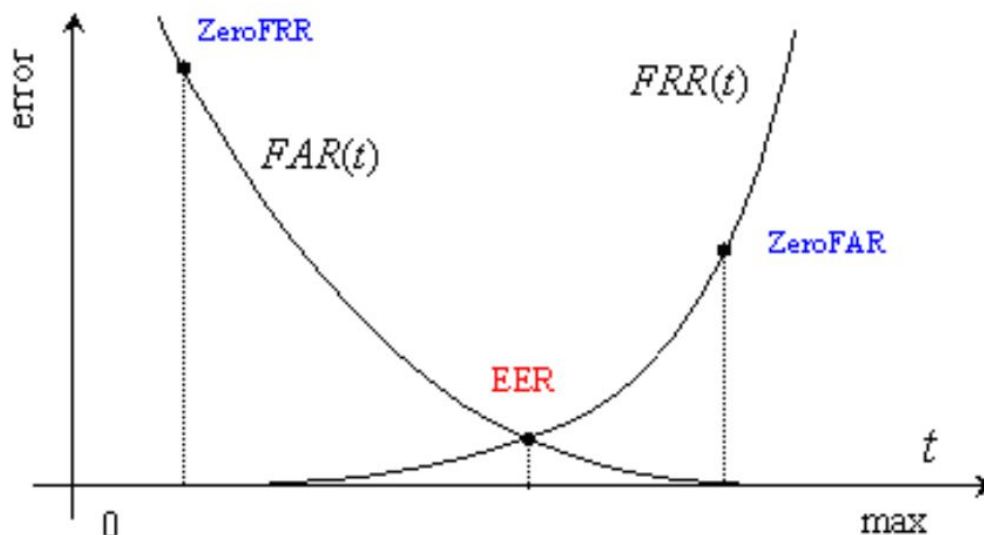
$$FAR = FP / (FP + TN)$$

- **False Rejection Rate** (FRR): Percentage of identification instances in which false rejection occurs. This can be expressed as a probability. For example, if the FRR is 0.05 percent, it means that on the average, one out of every 2000 authorized persons attempting to access the system will not be recognized by that system.

$$FRR = FN / (TP + FN)$$

Both the measures we listed before are dependant on the adopted threshold: the higher it is more False Rejections, the lower more False Acceptance.

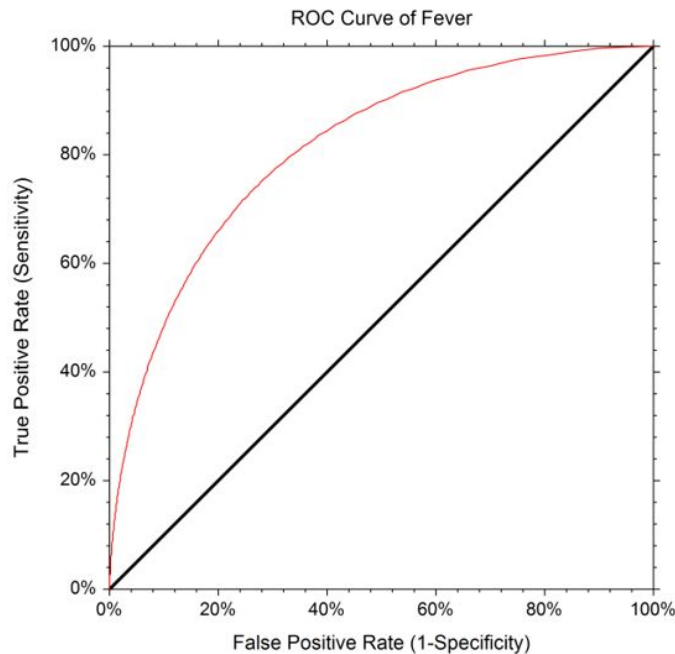
We can thus collect these measures varying each time the threshold. The point in which FAR and FRR intersects is the **Equal Error Rate** (ERR). We can also identify two other points in the plot: the **ZeroFRR** which is the point on the FAR plot where FRR = 0, and **ZeroFAR** which is the point on the FRR plot where FAR = 0.



FAR and FRR plots, with the three points of interest

Finally it is worth showing other two interesting measures:

Receiver Operating Characteristic (ROC): Plot of FP rates against TP rates.



Detection Error Trade-Off (DET): False Accepts vs False Rejects.

In the open set identification (or closed set) the biometric system determines if the individual's biometric signature matches a biometric signature of someone in the gallery. We can define the following concepts:

- **CORRECT DETECT AND IDENTIFY RATE:** a correct result occurs when the individual in the probe image is also in the database and the correct individual has the highest similarity score. If we run many trials with probes belonging to the subjects in the database (set P_g), we will know how often the system will return a correct result.
- **FALSE ALARM RATE:** if we run many trials with probes belonging to subjects not in the database (set P_n), we will know how often the system will return an incorrect alarm.

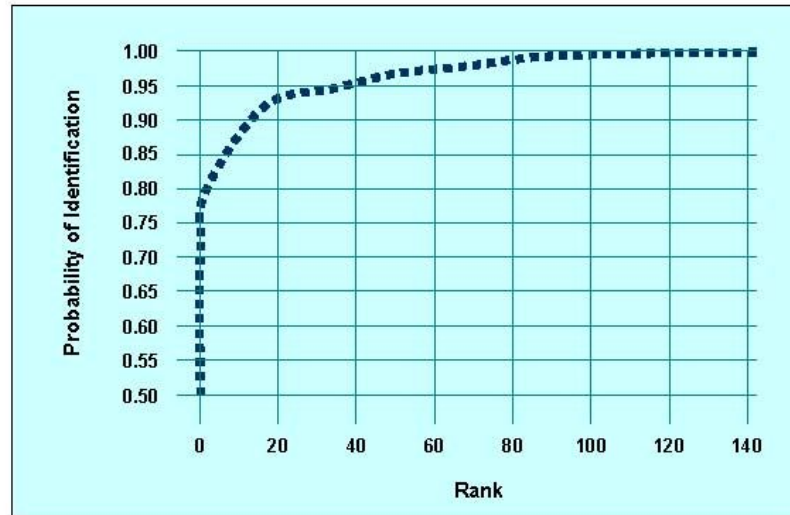
In a more formal way:

- Detection and identification rate at rank k (DIR): the probability of correct identification at rank k (the correct subject is returned at position k). This is the rate between the number of individuals correctly recognized at rank k and the number of probes belonging to individuals in P_g
- False Reject Rate (FRR): the probability of false reject expressed as $1 - \text{DIR}$
- False Acceptance Rate (FAR): the probability of false acceptance/alarm. This is the rate between the number of impostor recognized by error and the total number of impostors in P_n

- Equal Error Rate (ERR): the point where the two probability errors are equal, i.e. $FRR = FAR$

Finally for identification we can plot the **CMS** that is the probability of correctly identifying a subject within the first k entries, for k that varies:

Cumulative Match Characteristic



More on performance evaluation (Lesson 2bis)

What do we do when we have a very limited number of templates to test our system? Suppose we have three subjects A, B, C, and three templates for each (for a total of 9 templates, $A_1, A_2, A_3, B_1, B_2, B_3, C_1, C_2, C_3$). We can build the following table:

	A_1	A_2	A_3	B_1	B_2	B_3	C_1	C_2	C_3
A_1	—	X	X	X	X	X	X	X	X
A_2	X	—	X	X	X	X	X	X	X
A_3	X	X	—	X	X	X	X	X	X
B_1	X	X	X	—	X	X	X	X	X
B_2	X	X	X	X	—	X	X	X	X
B_3	X	X	X	X	X	—	X	X	X
C_1	X	X	X	X	X	X	—	X	X
C_2	X	X	X	X	X	X	X	—	X
C_3	X	X	X	X	X	X	X	X	—

Where each **X** contains the similarity/distance of the templates of its row and column. Once we have computed this table we can easily compute statistics (obviously we do not compute the distance/similarity for the same couple of templates). We treat each row as a probe and each column as a template. So in this case each row acts as a **genuine user 2 times** and as an **impostor 6 times**. With this in mind we can easily compute all the statistics we need.

Response Reliability (Lesson 3)

Features extracted from a sample of a biometric trait, which are labeled with the individual's identity, represent its **template** (Note that a template should not allow the reconstruction of its original sample).

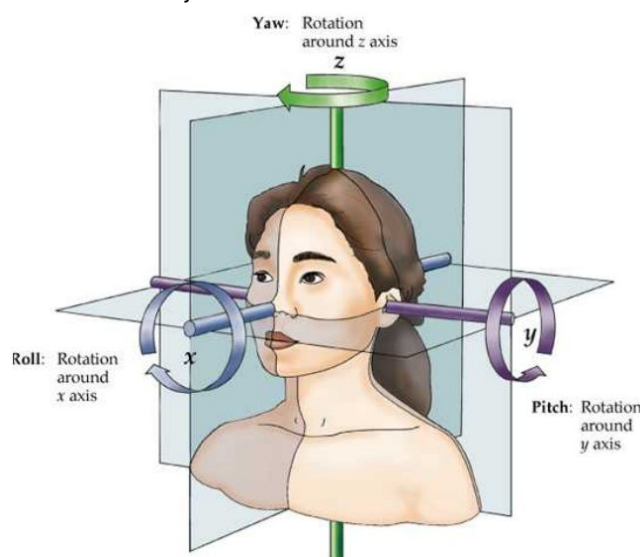
During the life of the recognition system much more data becomes available (by people that uses the system). We can use such data to augment the existing gallery to tackle problems such as **template aging** (for example in face recognition a man can enroll without beard and later starts growing it).

(I don't even try to remember the doddington zoo)

The definition of a **reliability measure** for each single response from a system provides further information to be used in setting up an operation policy (e.g., is f the identification is not reliable enough and if possible, repeat capture), but also to merge results from different systems (ensemble of biometric systems).

For what regards face recognition we can define the quality of an image basing on three values:

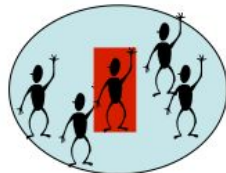
- **Yaw:** rotation along the z axis
- **Roll:** Rotation around the x axis
- **Pitch:** Rotation around the y axis



We can define a quality measure of a face image based on these three values and use it to discard instances that are too noisy. A **good quality measure** is one that allows to get better performance without discarding too much data.

System Response Reliability measures the ability of an identification system to separate genuine subjects from impostors on a single probe basis.

Basically it measures how much is "crowded" the cloud around the returned subject: the less crowded it is the more reliable is the response.



Cloud around the returned subject
"less crowded" =
More reliable response



Cloud around the returned subject
"more crowded" =
Less reliable response

Face recognition: introduction and face localization (Lesson 4)

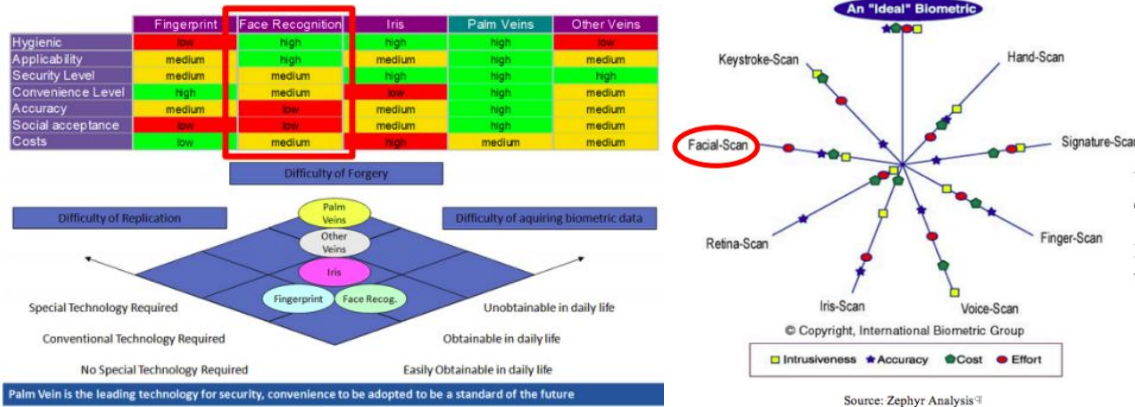
The most important factors influencing the validity of a biometrics are Accuracy/Reliability and Acceptability. Face has a high acceptability, the user may be unaware of image capture, but accuracy is still to be improved.

Comparisons

Biometrics	Univer- sality	Unique- ness	Perma- nence	Collect- ability	Perform- ance	Accept- ability	Circum- vention
Face	H	L	M	H	L	H	L
Fingerprint	M	H	H	M	H	M	H
Hand Geometry	M	M	M	H	M	M	M
Keystroke Dynamics	L	L	L	M	L	M	M
Hand vein	M	M	M	M	M	M	H
Iris	H	H	H	M	H	L	H
Retina	H	H	M	L	H	L	H
Signature	L	L	L	H	L	H	L
Voice	M	L	L	M	L	H	L
Facial Thermogram	H	H	L	H	M	H	H
DNA	H	H	H	L	H	L	L

H=High, M=Medium, L=Low

Biometric Comparison Chart						
Sr. No	Characteristics	Fingerprint	Iris	Face	Palm Print	Voice Recognition
1	Speed	Medium/Low	Medium	Medium	Medium	Medium
2	FTE Rate	Medium/Low	Low	Low	Low/Medium	Medium
3	Standards	High	Medium	High	High	Low
4	Uniqueness	High	High	Medium	High	Medium
5	Maturity	High	Medium	Medium	Medium	Low
6	Durability	High	High	Medium	High	Low
7	Invasiveness	High	Medium	Low	High	Low
8	Overtness	High	Medium	High	Low	Low
9	Range	Low	Low	Medium	Low	High
10	Template Size	Medium (250-1,000 bytes) (per finger)	Medium (688 bytes)	High (84-2,000 bytes)	Medium (250-1,000 bytes)	High (1,500-3,000 bytes)
11	Age Range	High	High	High	High	Medium
12	Universality	High	Medium	High	High	High
13	Stability	High	High	Medium	High	Low
14	Skill	Medium	Medium	Low	High	Low
15	Accuracy	Medium-High	High	Medium	High	Low
16	Hygienic Level	Low	High	High	Low	High
17	Performance	High	High	Low	High	Low
18	Cost	Low	High	Low	High	Low



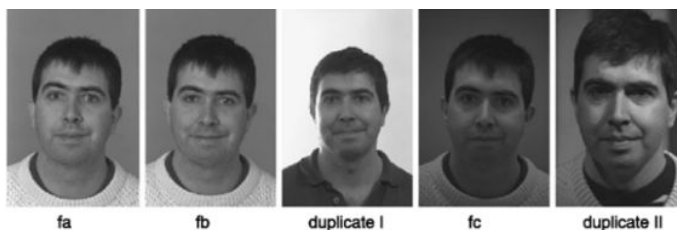
Problems

1. **Intra-person variations:** a same subject in different poses could same different.
2. **Inter-personal similarities:** two different subjects could same the same because of similarities.

Popular Databases

PIE variation is the acronym for *Pose, Illumination, Expression* variation (We can also consider the A-*PIE* variance that adds aging as a type of variation). The difference between the **controlled** variation and the **uncontrolled** is that in the first one we know which is the distortion.

1. **FERET (1996):** this database contains **pose variation** (the subject is captured under different orientation, each denoted by a pair of letters in the image name), **illumination variation** (not for all subjects and no denoted in the image name), **time variation** (subjects are captured in different time in different sessions). The number of instances in the database is **14051**, such instances are divided into categories. For each category is provided a list with image names. A set of files contains the position of eyes and mouth for each image. An example of FERET's images captured from the same subject is the following:



2. **AR-Faces (1998):** this database collects different facial expressions, illumination conditions and occlusions for the same subject (126 subjects). For each person two session (that is two different days for person for collecting its images).
3. **CMU-PIE (2000):** this database contains about **608** color pictures per subject (resolution 640*486 pixels). The number of subjects is **68** (so we have **41344** images). Each person is under 13 different poses, 43 different illumination conditions, and 4 different expressions.

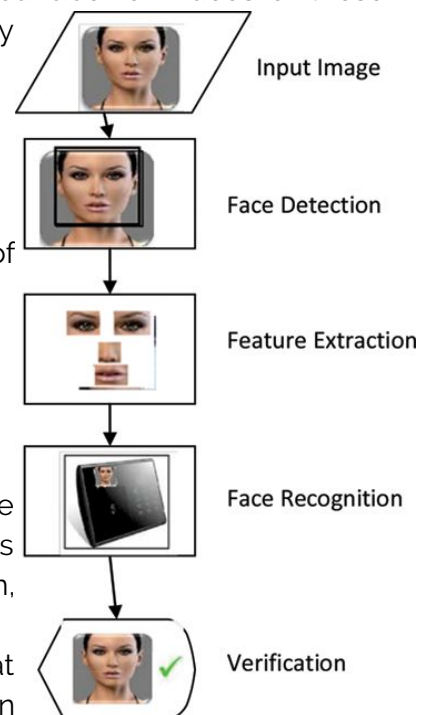
4. **CASIA 3D FACE V1 (2004)**: this database contains **4624** scans of **123** people using the non-contact 3D digitizer. An example of scans for a given subject is the following:



5. **LFW (Labeled Faces in the Wild 2007)**: this database contains more than **13000** images of faces collected from the web. Each face has been labeled with the name of the person pictured. 1680 of the people pictured have two or more distinct photos in the data set. The only constraint on these faces is that they were detected by the Viola-Jones face detector.
6. **YouTube FACES (2011)**: this database contains **32425** videos of **1595** different people. In average, for each subject 2.15 videos are available. The shortest clip duration is 48 frames, the longest clip is 6,070 frames, and the average length of a video clip is 181.3 frames. The creation started by using the 5,749 names of subjects included in the LFW dataset to search YouTube for videos of these same individuals. The top six results for each query were downloaded.

Structure of a face recognizer

1. Face capture and possible image enhancement
2. Localization (cropping of one or more region of interest *ROIs*) + normalization
3. Feature extraction
4. Template construction (biometric key)



Face Localization

The aim is to detect the presence of one or more faces inside a video or a single image and to locate their position. It is necessary to be independent with respect to position, orientation, scale, expression, illumination, background.

"According to Adam Harvey, the key part of the face that computers can read is the "nose bridge," or the area between the eyes. If we obscure that, we have a good chance of tricking computers into thinking we don't have a face. Another technique is to create an "anti-face," which just means inverting face's color scheme.

Approaches for face localization are:

- **FEATURE BASED TECHNIQUES**: use of knowledge about the expected feature appearance, which is characterized by a set of features at different levels. Examples include:
 - 1) Pixel properties: detecting edges and skin color (figures that resemble a face contour with a color similar to the skin tone and that have a something red in it, e.g. the mouth then it is probably a face)
 - 2) Face Geometry: Constellation of feature searching (for example if an object has two eyes a mouth and a nose than it is a face)

- 3) Template matching: try to match objects in an image with a model that has all the features of a face such as eyes, eyebrows, mouth, nose and face contour.
- **IMAGE-BASED TECHNIQUES:** learn how to recognize an image according to a number of examples using the image as a whole (for example through neural network (as in our approach of Basic CNN or support vector machine).

More about face localization (Lesson 5)

Algorithm A (by Hsu, Mottaleb and Jain, 2002)

Based on several phases:

- Illumination compensation
- Color space transformation
- Localization based on skin model
- Localization of main features (eyes, mouth ecc...)

I. FACE CANDIDATES DETECTION:

A. Illumination compensation

Why: the **skin-tone depends** on the scene overall **illumination**.

For the **normalization** of the color appearance is used the **reference white**. Luma represents the brightness in an image (the "black-and-white" or achromatic portion of the image). Luma is the weighted sum of gamma-compressed R' G' B' components of a color video—the prime symbols ($'$) denote gamma compression: $Y' = 0.212R' + 0.715G' + 0.0722B'$.

B. Color space transformation

Why: RGB is not a perceptually uniform color space => colors that are close in RGB are not close to how we view colors. Better color spaces are YUV (Luminance and Chrominance components) or HSV (Hue Saturation Value).

C. Localization based on skin model

TWO METHODS ARE POSSIBLE:

1. VARIANCE-BASED SEGMENTATION:

In computer vision and image processing, **Otsu's** method is used to automatically perform clustering-based image **thresholding**, or, the **reduction of an image to a binary image** (only two colors). The algorithm assumes that the image contains two classes of pixels following bi-modal histogram (**foreground** pixels and **background** pixels), it then calculates the **optimum threshold separating the two classes** so that their combined spread (intra-class variance) is minimal.

2. CONNECTED COMPONENTS:

The detected skin tone pixels are iteratively segmented using local color variance. Connected components are grouped according to spatial closeness and similar color.

Up to here we have generated a set of face candidates...the following part of the algorithm discards those that do not have enough features.

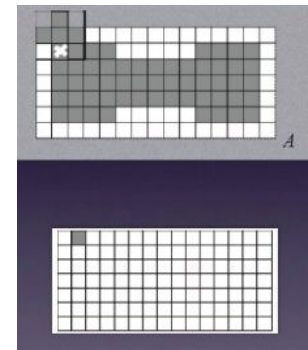
II. FACE CANDIDATES VERIFICATION THROUGH FACIAL FEATURES DETECTION:

A. Localization of eyes

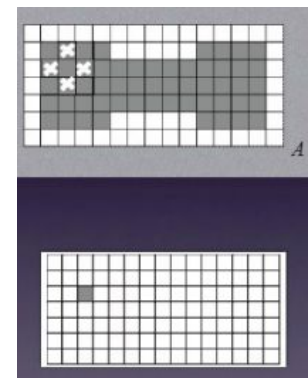
Two different eye map:

1. CHROMINANCE MAP (*EyeMapC*) -> A map of the eyes is created by noticing that eye regions have a high value on one chrominance component (Cb) and low value on the other (Cr).
2. LUMINANCE MAP (*EyeMapL*) -> Eyes usually contains both light and dark zone. To emphasize eye regions a dilation and an erosion is then performed on the constructed map.

Given an image I the algorithm performs $EyeMapC(I)$ and $EyeMapL(I)$. The resulting images are then combined through AND (giving I_{AND}). At this point the dilation operator is performed on I_{AND} . The dilation operator takes in input I_{AND} and a structuring element (*kernel*), it returns as output the set of all points s.t. the intersection between I_{AND} and kernel is not empty.



The erosion operator is like the above operator, except for the output: set of all points s.t. the intersection between I_{AND} and kernel is a subset of I_{AND} .

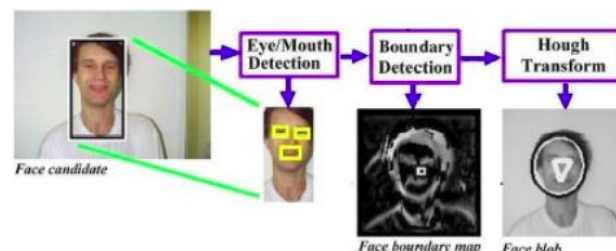


B. Localization of mouth

Again it is created a map that enhance the mouth region (basically the red color is enhanced), and like before that region is emphasized using dilation.

C. Localization of face contour

Face contour is simply performed using edge detection techniques.



Algorithm B (Viola-Jones, 2004)¹

This is an image based algorithm for face detection. It requires a classifier that is initially trained using some **instances containing face** (positive examples) and some other

¹ http://bias.csr.unibo.it/franco/SB/DispensePDF/8_Volto_localizzazione.pdf

instances non containing face but which may cause an error (negative examples). Training is designed to extract several features from the examples and to select the most discriminating one. **Misses** (a present object is not detected) or **false alarms** (an object is detected but it is not present) can be decreased by retraining adding new suited examples (positive or negative). Therefore, the face classifier's classes are **face/non face**. Training is slow but detection is very fast.

The main contributions are 3:

1. Extraction and evaluation of **Haar-like feature**
2. Classification through **boosting**
3. **Multiscale detection**

Slide a search window of varying size over the image from the top-left corner to the right bottom corner. The features inside this window are extracted and the window is classified as face/non-face. The localization of faces is done by analyzing consecutive sub- windows (overlapping) of the image as input and evaluating for each of them if it belongs to the class of faces.

In the training phase *AdaBoost* algorithm is used.

AdaBoost

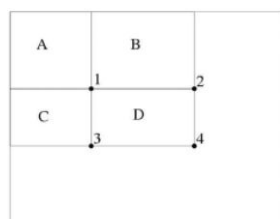
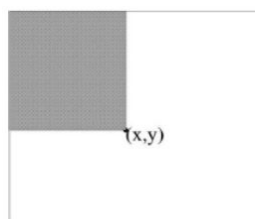
MEMENTO: Learning Procedure

We want to build a non linear classifier $H_M(x)$ (complex classifier) as linear combination of M more simply classifiers (weak classifiers).

Haar-like features

Each time the search window is moved, a set of *haar kernel* are passed over the image.

Haar Kernels are simple (rectangular) features: 2d vectors with black and white regions. Obviously at test time it is infeasible to try all possible haar features (for a search window of 24x24 pixels that are 160.000+ possible filters!). So we select the best ones using adaboost. The value of a haar kernel is the sum of pixels in the white regions is subtracted by the sum of pixels in the black region. Such a value is computed through **integral image**. The integral image, $II(x,y)$, is the sum of the values of pixels above and on the left of (x, y) (i.e (x', y') s.t. $x' \leq x, y' \leq y$):



A, D white; B,C black:
 $II(4)+II(1)-II(2)-II(3)$

Example:

- <https://vimeo.com/12774628>

Weak classifiers

The simplest weak classifier is a **decision tree** with only one node. Suppose we built $M-1$ weak classifier, $h_m(x)$ with $m=1 \dots M-1$, we want to build $h_M(x)$. The classifier compares the value of a **feature** z_k with a **threshold** τ_k , and it assigns the values $+1$ -1 as follows:

$$h_M(x) = +1 \text{ if } z_k > \tau_k$$

= -1 otherwise

The two parameters, z_k and τ_k , can be fixed with respect to the minimum error of the classifier.

Strong classifier

Once we have the sequence of weak classifiers, we can combine them to obtain the strong classifier H_M :

0. (Input)

- (1) Pattern di training $\mathcal{Z} = \{(x_1, y_1), \dots, (x_N, y_N)\}$, dove $N=a+b$;
 a pattern hanno etichetta $y_i = +1$
 b pattern hanno etichetta $y_i = -1$;
- (2) Il numero M di classificatori da combinare.

1. (Inizializzazione)

- $$w_i^{(0)} = \frac{1}{2a} \text{ per i pattern con etichetta } y_i = +1$$
- $$w_i^{(0)} = \frac{1}{2b} \text{ per i pattern con etichetta } y_i = -1$$

2. (Costruzione del classificatore)

Per $m = 1, \dots, M$:

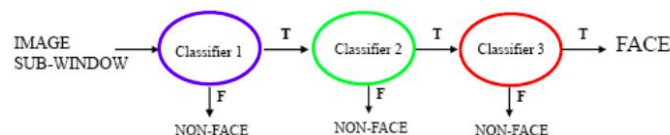
- (1) Scegli il classificatore debole ottimale h_m che minimizzi l'errore pesato
- (2) Scegli α_m
- (3) Aggiorna i pesi $w_i^{(m)} \leftarrow w_i^{(m-1)} \exp[-y_i \alpha_m h_m(x_i)]$ e normalizzali ($\sum_i w_i^{(m)} = 1$).

3. (Output)

- (1) Classificatore $H_m(x)$
- (2) Classificazione dei pattern di training $\hat{y} = \text{sign}[H_m(x)]$

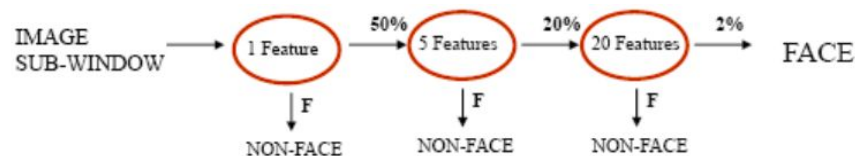
Cascade of strong classifiers

Even though a single strong classifier eliminates a great number of subwindows which do not contain face, it is not so efficient. For this purpose, one solution is the use of a cascade of strong classifiers, from the simplest to the most complex.



We start with simple classifiers which reject many of the negative sub-windows while detecting almost all positive sub-windows. Positive results from the first classifier triggers the evaluation of a second (more complex) classifier, and so on. A negative outcome at any point leads to the immediate rejection of the sub-window.

HOW TO TRAIN THE CASCADE



1. Adjust weak learner threshold to minimize false negatives (as opposed to total classification error).
2. Each classifier trained on false positives of previous stages.

3. A single-feature classifier achieves 100% detection rate and about 50% false positive rate.
4. A five-feature classifier achieves 100% detection rate and 40% false positive rate (20% cumulative).
5. A 20-feature classifier achieve 100% detection rate with 10% false positive rate (2% cumulative).

Evaluation

For the evaluation of the localization we use 3 measures:

1. **False Positive:** that is the percentage of windows classified as faces that do not contain any face.
2. **Not Localized Faces:** that is the percentage of faces that have not been identified.
3. **C-Error:** that is the localization error (i.e. the Euclidean distance between the real center of the face and the one estimated by the system, normalized with respect to the sum of axes of the ellipse containing the face).

Face Recognition 2-D (Lesson 6)

An image I can be represented as a point in a multi-dimensional (feature) space:

- **AS A TWO-DIMENSIONAL FUNCTION:** function $I(x, y)$ defined in the *Cartesian Space* $\rightarrow RGB$ (i.e. assign a value between 0 and 255 to all points (x, y) in the plane). Such values can be stored either in a **matrix $w \times h$** or in a **one-dimensional vector $n=w \times h$** .

The downside of image representation is its dimension:

- **CURSE OF DIMENSIONALITY:** refers to various phenomena that arise when analyzing and organizing data in high-dimensional spaces (often with hundreds or thousands of dimensions) that do not occur in low-dimensional settings such as the three-dimensional physical space of everyday experience. When the dimensionality increases, the volume of the space increases so fast that the available data become **sparse**. Since to classify data we detect areas where objects form groups with similar properties, in high dimensional data, all objects appear to be sparse and somehow dissimilar, and this prevents data organization from being efficient. In Machine Learning the problem of curse-dimensionality reduces the predictive power. Indeed since the image is represented by a feature vector composed by an high number of feature (each of which has a number of possible values) , an enormous amount of training data is required.

Therefore the solution is to decrease the space of the image representation.

Image reduction

PRINCIPAL COMPONENT ANALYSIS (PCA)

Is used to reduce a set of (possibly) correlated variables into a set of variables (called principal components) that are uncorrelated (no redundancy). The number of principal components is obviously less than the number of original variables.

Algorithm:

1. $TS = \{x_i \text{ in } R^n \mid i = 1...m\}$ is the training set
2. Compute the mean vector u (sum of all the points in TS) divided by $|TS|$
3. Denote as d_i as the difference of x_i with the mean vector
4. Compute the covariance matrix C as the sum for each i of d_i with itself, everything divided by $|TS|$ (the size of C is $n \times n$)
5. The new k -dimensional space is defined by the projection matrix U whose columns are the k eigenvectors of C corresponding to the k highest eigenvalues of C (Eigenvalues represent variances along eigenvectors). **Probabilmente c'è un modo matematico per calcolare ste cose...sulle slide non ci sta**

EigenFaces is basically just a fancy name for PCA applied to images. Following the algorithm above we have a set of images of the same size (in order for the algo to have good results we also expect them to be aligned and centered). First of all we need to represent the image matrices as vectors then we compute the mean image and the covariance matrix, we then identify the eigenvectors (that will be called eigenfaces). We finally build the projection matrix made only of the eigenvector with highest eigenvalues.

Now...how to perform recognition?

Once eigenfaces are created a new face image f can be transformed into its eigenface components vector with the operation:

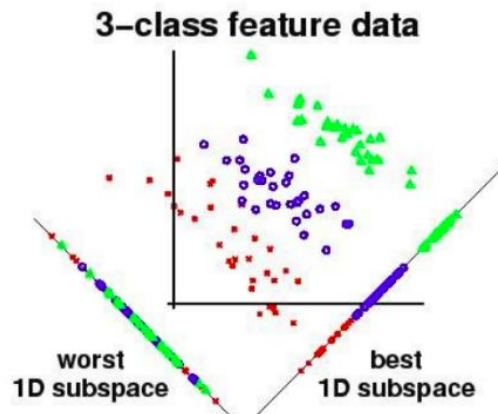
$$PCA_f = U^T (f - u)$$

We can use this vector for face recognition by finding the smallest euclidean distance between f and the training faces principal components.

(<http://blog.manfredas.com/eigenfaces-tutorial/>)

LINEAR DISCRIMINANT ANALYSIS (LDA)

The objective of LDA is to perform dimensionality reduction while preserving as much of the class discriminatory information as possible. Hence, what we want is to reduce the space so that the separation between the classes is maximizing. This is a linear supervised dimensionality reduction technique.

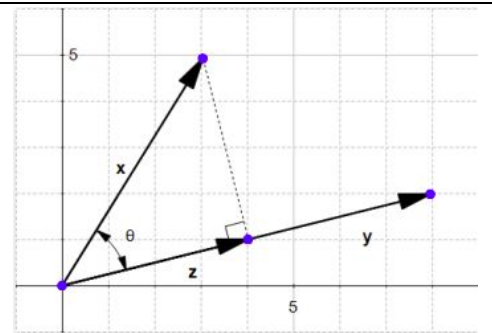


1. $TS = \{x_i \text{ in } \mathbb{R}^n \mid i=1 \dots m\}$
2. s is the number of classes so TS is partitioned as $PTS = \{P_1, \dots, P_s\}$ where samples in class P_i have class i and $|P_i| = m_i$.
3. We seek to obtain a scalar y by projecting the sample x onto a line s.t.: $y = w^T x$

In order to find a good projection vector, we need to define a measure of separation between the projections.

Remember Projection:

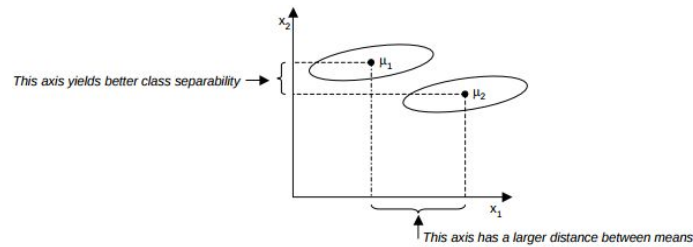
Given two vector \mathbf{x} and \mathbf{y} , we want to find the orthogonal projection of \mathbf{x} into \mathbf{y} (that is \mathbf{z}). Let \mathbf{u} be the unit vector of \mathbf{y} . Then \mathbf{z} will be: $\mathbf{z} = (\mathbf{u}^T \mathbf{x}) \mathbf{u}$.
 \mathbf{z} is in the same direction of \mathbf{y} and $\|\mathbf{z}\| = \|\mathbf{x}\| \cos(\theta)$



Let us suppose that in TS there are only two class (hence $s=2$). In such a case $PTS = \{P_1, P_2\}$ and $|P_1| = m_1$, $|P_2| = m_2$.

- The mean of each class in the two spaces is: $\mu_i = \frac{1}{m_i} \sum_{j=1}^{m_i} x_j$ $\tilde{\mu}_i = \frac{1}{m_i} \sum_{j=1}^{m_i} y_j = \frac{1}{m_i} \sum_{j=1}^{m_i} w^T x_j = w^T \mu_i$
- We could choose the distance between the projected means as our objective function (to maximize): $J(w) = |\tilde{\mu}_1 - \tilde{\mu}_2| = |w^T(\mu_1 - \mu_2)|$

However, the distance between projected means is not a very good measure since it does not take into account the standard deviation within the classes (scatter):



We want to find the line on which the difference between the means of each class is the highest. In the example above, we have only two class P_1 and P_2 , so we have only two mean. We projecting our mean on the two axis. Since $J(x_1) > J(x_2)$, we choose x_1 (reduction from two to one dimension).

SCATTER MATRICES

Better results are obtained by **maximizing the ratio of between-class variance to within-class variance** starting from scattering matrices.

- **Within-class scatter matrix S_w :** indicates how the vectors of each class are scattered with respect to its center (that is, given the samples having the same class i , we see the distance of those from the mean).
- **Between-class scatter matrix S_b :** indicates how the centers of the classes are scattered (that is, given a dataset with only two class, we compute the mean of each single class, hence we see the distance of such means from the means of the means).

1. $TS = \{x_i \text{ in } \mathbb{R}^n \mid i=1 \dots m\}$
2. s is the number of classes so TS is partitioned as $PTS = \{P_1, \dots, P_s\}$ where samples in class P_i have class i and $|P_i| = m_i$.
3. For each class P_i compute the mean vector (centroid) and the mean of means (centroid of centroids):

$$\mu_i = \frac{1}{m_i} \sum_{j=1}^{m_i} x_j \quad \mu_{TS} = \frac{1}{m} \sum_{i=1}^s m_i \mu_i$$

4. Compute the covariance matrix for the vector P_i :

$$C_i = \frac{1}{m_i} \sum_{j=1}^{m_i} (x_j - \mu_i)(x_j - \mu_i)^T$$

5. Compute S_w (same for $\text{proj}(S_w)$):

$$S_w = \sum_{i=1}^s m_i C_i$$

6. Compute S_b (same for $\text{proj}(S_b)$):

$$S_b = \sum_{i=1}^s m_i (\mu_i - \mu_{TS})(\mu_i - \mu_{TS})^T$$

7. Compute the scatter of projected class P_i ($\text{proj}(s_i)$):

$$\tilde{s}_i^2 = \sum_{y \in P_i} (y - \tilde{\mu}_i)^2$$

Remember Covariance Matrix:

A **covariance matrix** is a matrix whose element in the i, j position is the **covariance** between the i^{th} and j^{th} elements of a random vector. Covariance is a measure of the joint variability of two random variables. If the variables tend to show similar behavior, the covariance is positive. In the opposite case, when the variables tend to show opposite behavior, the covariance is negative.

For instance let $X=(x_1, \dots, x_n)$ be a random vector and x_i , with $i=1 \dots n$, be random variables each with a finite variance. Then the covariance matrix Σ is the matrix whose (i, j) entry is the covariance:

$$\Sigma = \begin{bmatrix} E[(X_1 - \mu_1)(X_1 - \mu_1)] & E[(X_1 - \mu_1)(X_2 - \mu_2)] & \cdots & E[(X_1 - \mu_1)(X_n - \mu_n)] \\ E[(X_2 - \mu_2)(X_1 - \mu_1)] & E[(X_2 - \mu_2)(X_2 - \mu_2)] & \cdots & E[(X_2 - \mu_2)(X_n - \mu_n)] \\ \vdots & \vdots & \ddots & \vdots \\ E[(X_n - \mu_n)(X_1 - \mu_1)] & E[(X_n - \mu_n)(X_2 - \mu_2)] & \cdots & E[(X_n - \mu_n)(X_n - \mu_n)] \end{bmatrix}$$

FISHER'S SOLUTION

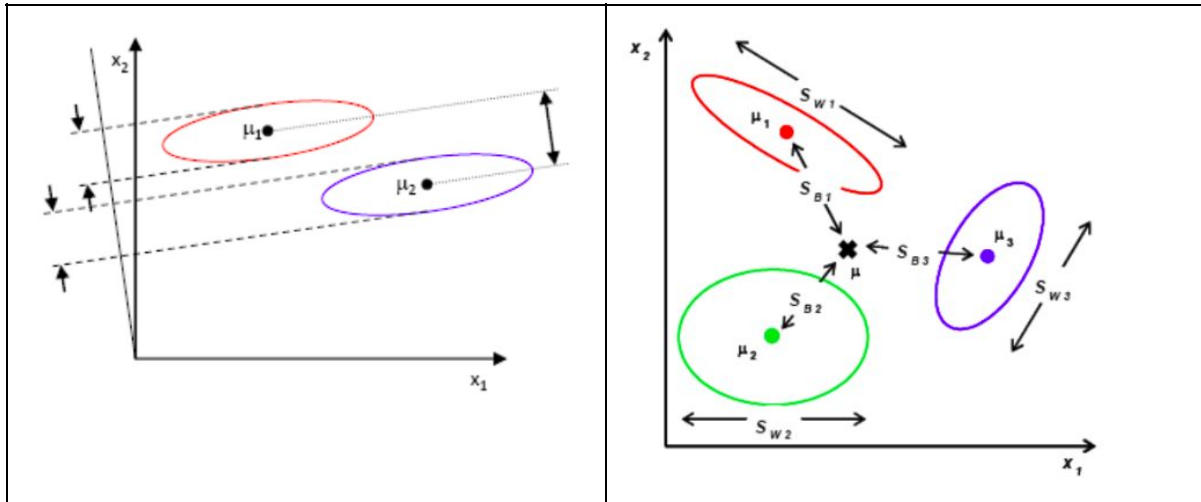
1. N is the number of images in the database, c is the number of people in the database. Each face is represented by a vector. (e.g 8 faces, 2 faces per person hence we have 4 people i.e, $N=8$, $c=4$).
2. Computing the average of all faces (average face) with mean vector (that is the mean between all the 8 vectors).
3. Computing the average face of each people with mean vector (that is computes 4 means each of them computed between the two vectors of the same person)
4. Subtracting average of each person from the training faces (that is if x_1 and x_2 are the vectors representing the faces of a same person, compute $x'_1 = x_1 - \text{mean}_1$ and $x'_2 = x_2 - \text{mean}_1$).
5. Compute scatter matrix S for each class
6. Compute S_w as the sum of the scatter matrices ($S_w = S_1 + S_2 + S_3 + S_4$)
7. Compute S_b

$$J(w) = \frac{w^T S_b w}{w^T S_w w}$$

8. Maximize:

http://www.cs.haifa.ac.il/~rita/visual_recog_course/talks/Eigenfaces%20vs%20Fisherfaces.pdf

TWO CLASSES	MORE CLASSES
-------------	--------------



What we have seen with PDA and LDA are methods for the reduction of the dimension of the image representation. In the next section we will see how to extract features.

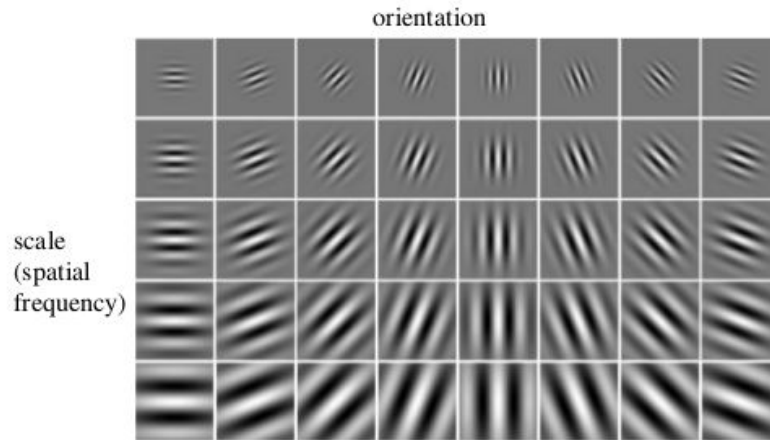
Features extraction

BUBBLES (Experiment): Two main tasks. The first one is judging whether a face is expressive or not (EXNEX) the second one is determining the gender of the face (GENDER). Two observers for each task. One is the human observer and the other is the ideal observer. Ideal observer considering the classes in the task (male vs. female, and neutral vs. expressive), will capture all the regions of the image that have highest local variance. This ideal considers the stimuli as images and it might not necessarily be sensitive to the regions that humans find most useful (i.e. the diagnostic regions), but rather to the information that is mostly available in the data set for the task at hand.

WAVELET TRANSFORMS: Fourier Transform provides frequency info about a signal, that is, it tell us how much of each frequency exists in the signal. Wavelet transform is capable of providing the time and frequency information simultaneously, hence giving a time-frequency representation of the signal. a "mother wavelet" is shifted in time-domain, and the process is repeated with different scales (inversely proportional to frequencies). Wavelet algorithms process data at different scales or resolutions. If we look at a signal with a large "window", we would notice gross features. Similarly, if we look at a signal with a small "window", we would notice small features. The result in wavelet analysis is to see "both the forest and the trees".

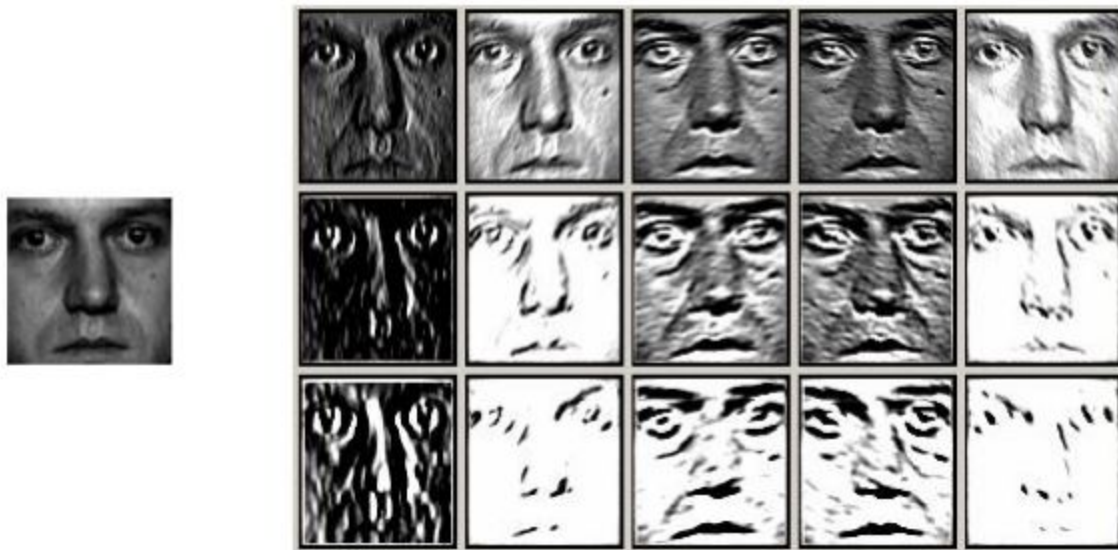
Gabor Filter

Gabor filters are compatible with this expression. A **Gabor filter** is a linear filter used for edge detection. Frequency and orientation representations of Gabor filters are similar to those of the human visual system, and they have been found to be particularly appropriate for texture representation and discrimination. In the spatial domain, a 2D Gabor filter is a Gaussian kernel function modulated by a sinusoidal plane wave. A feature vector is obtained by convolution of an image with a Gabor filter bank (different orientations and different scales).



Example of gabor filters at different scales and orientations.

2D Gabor functions enhance edge contours. This corresponds to enhancing eye, mouth, and nose edges, and also moles, dimples, scars, etc. If the convolution is performed on all the pixel of the image and we take the overall result, dimensionality is high (size of the image). In alternative it can be performed on a regular grid of points, or on salient face regions only. Example of applications of various Gabor filters to an image:



Local Binary Pattern (LBP)

Local binary pattern (LBP) is a popular texture descriptor for feature representation. Inspired by its success in texture classification, a novel was proposed for face recognition. The basic idea is as follows:

- for each pixel (\mathbf{p}), its 8-neighborhood pixels are thresholded into 1 or 0 by comparing them with the center pixel (\mathbf{p}_c). Then the binary sequence of the 8-neighborhoods is transferred into a

example	thresholded	weights																											
<table border="1"> <tr><td>6</td><td>5</td><td>2</td></tr> <tr><td>7</td><td>6</td><td>1</td></tr> <tr><td>9</td><td>8</td><td>7</td></tr> </table>	6	5	2	7	6	1	9	8	7	<table border="1"> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </table>	1	0	0	1	1	0	1	1	1	<table border="1"> <tr><td>1</td><td>2</td><td>4</td></tr> <tr><td>128</td><td>16</td><td>8</td></tr> <tr><td>64</td><td>32</td><td>16</td></tr> </table>	1	2	4	128	16	8	64	32	16
6	5	2																											
7	6	1																											
9	8	7																											
1	0	0																											
1	1	0																											
1	1	1																											
1	2	4																											
128	16	8																											
64	32	16																											

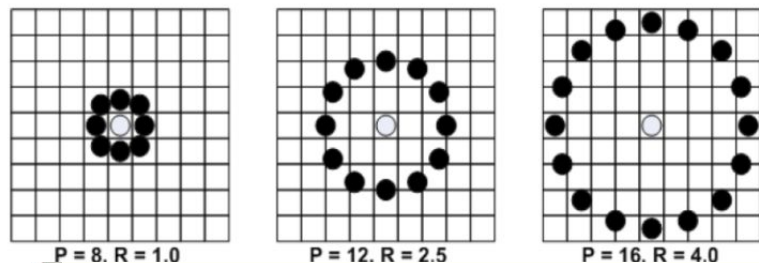
Pattern = 11110001

LBP = $1 + 16 + 32 + 64 + 128 = 241$

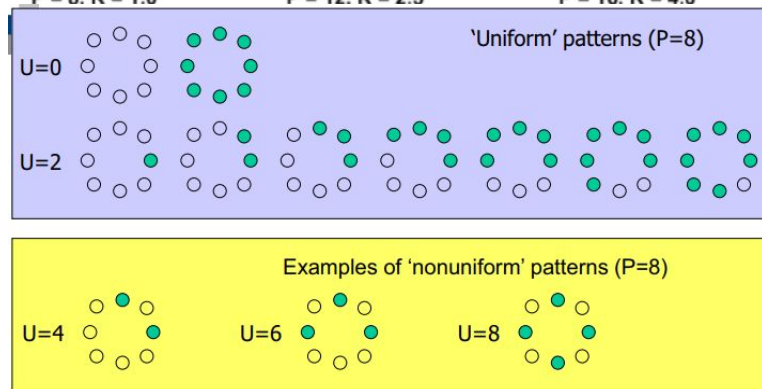
$C = (6+7+8+9+7)/5 - (5+2+1)/3 = 4.7$

decimal number (bit pattern states with upper left corner moving clockwise around center pixel, in the example to the right we start from the pixel indicated by the red arrow), and the histogram with 256 bins of the processed image is used as image feature. Furthermore, a kind of contrast measure referred to the central pixel can be computed, subtracting the average value of neighbors with a higher or equal value from the average value of neighbors with a lower value.

To capture the dominant features, LBP has two parameters (P, R), **P is the number of points** and **R is the interval radius**.



A pattern is called **uniform** when, if considered in a circular fashion, it contains at most two transitions 0-1 or 1-0. For example the patterns 10000011, 11110000, 00000000 are uniform. Uniform patterns are useful to save memory, since they are only $P \times (P-1) + 2$ over a total of P^2 .



HOW TO COMPUTE HISTOGRAMS:

1. The image is partitioned into subwindows through a grid of $k \times k$ elements.
2. For each sub-window a histogram is constructed in which each bin is associated with a specific pattern (the number of bins depend of the LBP type).
3. The final feature vector is obtained by concatenating the histograms calculated for all subwindows.

Classification of recognition system

1. **GLOBAL (HOLISTIC) APPEARANCE METHODS:** these methods are based on the whole appearance of the face. **PROS:** we consider all regions of the face, hence we do not destroy any information in the images. Several of these algorithms have been modified to compensate for PIE variations, and dimensions. **CONS:** considering all regions (so all pixels) of the face is also negative. Indeed, such algorithms are computationally expensive and require a high degree of correlation between test and training set. Examples of this type system are:

a. Eigenface

PROS: identification phase is fast. If eigenvectors are preserved it is possible to reconstruct the initial information. **CONS:** Training phase is slow. If a significant number of new subjects is added it is necessary to

retrain the system. The system is highly sensible to illumination and pose variations, to occlusions.

b. Neural Network

PROS: reduces the ambiguity among subjects belonging to similar classes- Robust to occlusions. **CONS:** more images for training phase and:

- **Overfitting:** when a model is excessively complex, such as having too many parameters relative to the number of observations.
- **Overtraining:** when the system "memorizes" patterns and thus loses the ability to generalize.
- **Database size:** when the number of subjects increases, the NN becomes inefficient.

2. LOCAL OR FEATURE- BASED METHODS: these methods are based on relevant points or on local characteristics of single zones, possibly anatomically significant (e.g. LBP). **PROS:** these methods are robust with respect to variations in the input image. Compact representation of the faces images and high speed matching. **CONS:** arbitrary decision about which features are important. If the feature set lacks discrimination ability, no amount of subsequent processing can compensate for that intrinsic deficiency. Example of this type system is:

- a. System based on Graphs:** each face is associated with a graph, hence matching two faces means matching two graphs. The graph is build localizing a set of reference points on the face and then connecting them with weighted edges. **PROS:** robust with respect to pose variations, illumination variations and they do not require retraining the system. **CONS:** training is slow and also testing (graph matching is NP-hard).


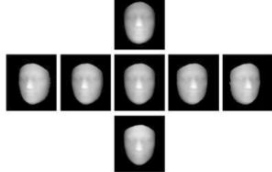

3. SENSORY INPUT BASED: Example of this type system is:

- a. Thermogram:** the face image is acquired through a thermal sensor. The sensor detects temperature variations from face skin. The face is segmented and indexed. **PROS:** robust with respect to time variations, illumination variations and they are efficient both indoor and outdoor. **CONS:** require expensive capture devices. Such devices are too sensitive to subjects movements and provide a quite low resolution. Moreover, they are affected by emotional state of the subject. A glass between the subject and the capture device makes the capture operation quite ineffective.

Face Recognition 3-D (Lesson 7)

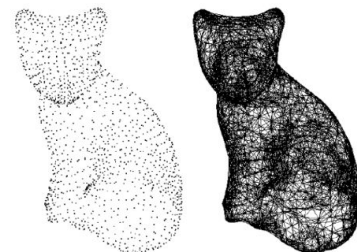
Let us remember the possible factors affecting face recognition performance. Those going under the name of *A-PIE* are pose, illumination, expression and ageing. Some others malicious factors that can compromise recognition are makeup, plastic surgery, glasses and scarfs. If 2D recognition is affected by all these factors, 3D is influenced only by expression, ageing, glasses and scarfs. So 3D appear more robust than 2D. There exists also another type of recognition in the middle between 2D and 3D, this is called 2.5D.

	2D	2.5D	3D
--	----	------	----

REPRESENTATION	2D grid where the value of each pixel is given by the intensity of the illumination reflexed in that point (value expressed through a gray value or RGB). Reflectance properties.	2D grid where the value of each pixel is given by the distance between that point and the light source (value expressed through a gray value or RGB). Shape properties.	Face patch are expressed with polygon. The structure is made by points and polygons connected in the 3D space. Shape properties.
IMAGE EXAMPLE			

The following are methods for the acquisition of 3D images:

- 1. Stereoscopic camera:** is a type of camera with two or more lenses with a separate image sensor or film frame for each lens. This allows the camera to simulate human binocular vision, and therefore gives it the ability to capture three-dimensional images. The face is captured by one or more of this devices having different points of view. For each image captured by the different cameras, a set of same feature is searched. Low cost, medium accuracy, low robustness to illumination, real time capture.
- 2. Structured light scanner:** is the process of projecting a known pattern (often grids or horizontal bars) on to a face. The way that these deform when striking surfaces allows vision systems to calculate the depth and surface information of the objects in the scene, as used in structured light 3D scanners. The process must be repeated from different point of view. Medium-high cost, medium-high accuracy, medium-high robustness to illumination, 3/8 seconds per scan, not dangerous for eyes.
- 3. Laser scanner:** as before, but this time a single laser beam is projected along the face. Medium-high cost, high accuracy, high robustness to illumination, 6/30 seconds per scan, dangerous for eyes.
- 4. Integrating several 2.5D face scans captured from different views:** 2.5D face scan can be acquired in the following way:
 - a. For each 2.5D image a cloud of 3D points is generated
 - b. A mesh of adjacent triangles is derived from that points

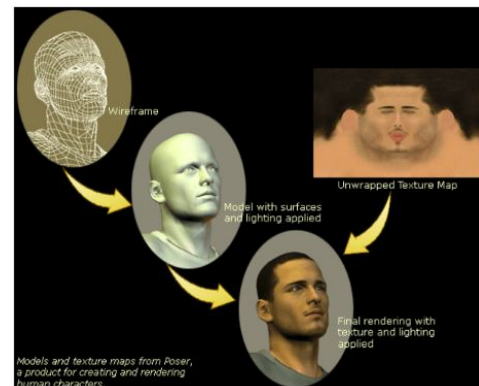


3D model

A 3D image is composed by a mesh of polygons (geometrical structure). Such polygons indicate the approximation of the figure (smaller polygons better approximation). Each polygon (as always) is formed by coplanar points connected by segments. Such

segments are the sides of the polygon. Each vertex in the mesh is adjacent to two side at least. The orientation of the 3D surface is determined by the normals to vertices or to polygons. The normal to a polygon is a vector perpendicular to the plane on which the polygon lies. The second step for building a 3D image (after the creation of the mesh) is the assignment of colors to vertex/polygon (perceptive structure). The color of polygon is determined by the color of its vertices. We can assign to it a uniform color (that is derived from the sum of the colors of its vertices) or a varying color (depending from the color of the vertices and from the distances of each point of the polygon from the different vertices). The process called texture mapping is used to compute position and orientation of a texture on a surface.

How we can pass from:



That is, how can we pass from 2D model to 3D one. There are two methods:

1. **SHAPE FROM SHADING:** compute the three-dimensional shape of a surface (face) from the brightness of that surface. The surface is assumed to be a Lambertian one. A Lambertian surface is an ideal surface which reflects brightness in the same way in all directions (i.e. the apparent brightness of such a surface to an observer is the same regardless of the observer's angle of view). To solve this issue two approaches:
 - a. Using an **ensemble of shapes of 3D faces + PCA** (to derive dimensionally reduced representation for shapes in the same class) + **principal components** that provides an excellent low-dimensional parametrization of head shape that maintains facial detail and identity person. This can recover an accurate 3D surface of the head/face of any person from a single 2D image of the face.
 - b. Avoiding the representation of input faces as combinations (of hundreds) of stored 3D models, using only the input image as a guide to "mold" a single reference model to reach a reconstruction of the sought 3D shape.
2. **MORPHABLE MODELS:** the procedure starts from a generic 3D model called morphable model. Such model is obtained from a set of 3D faces. Shape and texture of the generic model are manipulated to adapt arriving to a 3D model of the specific subject. This procedure allows to synthesize face expressions approximating the possible expressions of a specific subject.

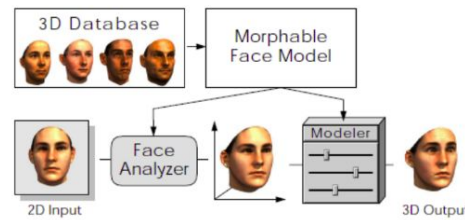


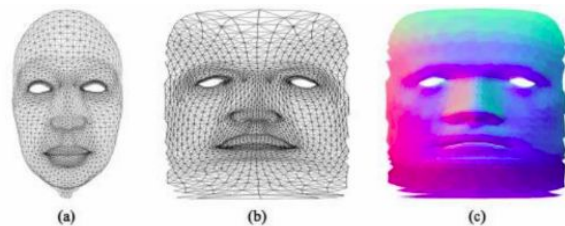
Figure 1: Derived from a dataset of prototypical 3D scans of faces, the morphable face model contributes to two main steps in face manipulation: (1) deriving a 3D face model from a novel image, and (2) modifying shape and texture in a natural way.

3D face recognition

In order to do 3D recognition of the face we must extract features. Features are about the local and global curvature of the face. Three types: *Crest Lines* (selecting the area within the 3D face with greatest curvature), *Local Curvatures* (representing the local curvature with color) and *Local Features* (segmenting the face into regions of interest). Also alignment is an important step before doing recognition. Alignment can be done in two ways: finding a finite number of characteristic points into the face and then performing rotation, translation and scaling minimizing the distance between the corresponding points; using the *Iterative Closest Point* (ICP) algorithm. It is used when an initial estimate of the relative pose is known. Given two 3D surfaces ICP finds an initial match between the two surfaces, then it computes the distance between the two surfaces by the least squares method and it does the transformation which minimizes such distance. At the end it performs the transformation and it reiterates the procedure until the distance is less than a threshold.

There are several types of 3D face recognition:

1. **RECOGNITION WITH NORMAL MAPS:** Three steps: a) acquisition and generation of the face, b) conversion from 2D to 3D, c) generation of the normal map. The normal map is a regular RGB image where the RGB components corresponds respectively to x, y, z of the surface normal (hence this normal is a vector of the form (x, y, z) and the RGB of the polygon will be (x, y, z)). As we can see, using normal



maps we can a 2-dimensional representation of 3-dimensional information. The curvature of a model is represented by the set of surface normals. Given two normal maps we can compute the difference map. Each pixel of the difference map represents the angular distance between two normal maps in that point.

2. **RECOGNITION VIA MORPHABLE MODELS (FaceGen modeller):** FaceGen modeller is a tool to generate 3D model of the face of an individual, the tool can simulates the dynamics of the face. The process with which the 3D face is created is that described previously with morphable models using features extracted directly from the photos to modify the base model. Synthetic facial expressions can be used to obtain a good approximation or real facial expressions that an individual may have during the process of acquisition, and

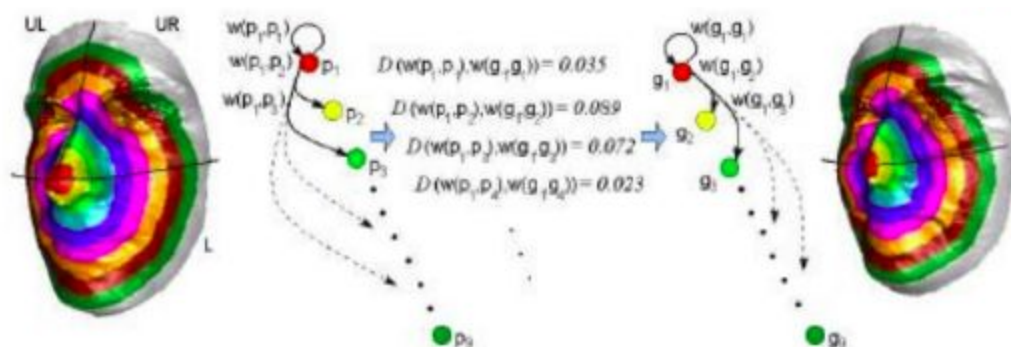
can be used in some cases to add samples to the gallery or to reproduce probe expression on the fly but also to simulating aging.

3. **RECOGNITION USING iso-Geodesic Stripes:** 3D faces are more sensitive to face expressions than 2D faces, indeed the points on the surface can exhibit large variations (especially those that lies on concave regions). For this purpose the geodesic distance of points on curves (convex regions) is considered, Geodesic distance is the shortest path between two points in a curved space. Such distances are slightly affected by the changes of the facial expressions. This approach provides the exploration of the whole 3D surface. The relevant information is encoded into a compact representation in the form of a graph and face recognition is reduced to matching the graphs.

How the graph is computed: the 3D face image is divided into **iso-geodesic facial stripes** of equal width and increasing distance form the nose tip. Each of these stripes **represents a node** in the graph. Experiments show that use of 9 stripes of width 1cm yields good invariance to facial expressions of the same individual and discrimination between different individuals. To account for the larger deformation of the mouth area with respect to the nose area, each stripe is partitioned into three parts lower (L), upper-left (UL) and upper-right (UR) with respect to the coordinates of the nose tip.

Each node is then connected with the others nodes by an edge. Edges between pairs of nodes are labeled with **3D Weighted Walkthroughs (3DWWs)**. This captures the mutual spatial displacement between all the pairs of points of the corresponding stripes and show smooth changes of their values as the positions of face points change.

How the matching is performed: the similarity between two 3D faces reduces to matching their corresponding graphs. Actually, since there is an unambiguous ordering of stripes from the nose tip to the border of the face model, the graph matching problem is reduced to the **computation of a distance, D , between multidimensional vectors**, by comparing homologous stripe pairs (see figure below).



Pros: Geodesic distances is a good measure because the distances between two facial points keep sufficiently stable under expression changes, indeed the large majority of the points of each stripe still remain within the same stripe, even under facial expression changes. Moreover, only few parts of the face are affected by deformations due to expressions.

Face Recognition: evaluation (Lesson 8)

Factors influencing the performance in the face recognition task:

1. **ILLUMINATION:** the feature vector associated to a subject in different illumination conditions may result closer to that of a different subject with a similar illumination, than to that of the same subject with a different illumination. Systems can address this problem by extrapolating the 3D face image.
2. **POSE:** in many (most) applications the pose of the subject during testing can be different from that during enrollment. Systems can address this problem by collecting a set of images of the subject in different pose (multi-view system), or can correct the pose (pose correction system).
3. **OCCLUSIONS:** there are several type of occlusions hiding partially the face. For instance hair, sunglasses, scarfs, make-up, noise, acquisition error. In many (most) applications the pose of the subject during testing can be different from that during enrollment.
4. **TIME AND AGE VARIATIONS:** Time variations (even only one week) between two images of a same subject, may reduce the performance of a recognition system. Systems robust to this problem are thermogram.

For evaluating a system some protocol have been devised. Examples of such protocols are the following:

1. **FERET (Facial Recognition Technology) PROTOCOL:** the goal of the FERET program was to develop automatic face recognition capabilities. The program consisted of three major elements: sponsoring research, collecting the FERET database, performing the FERET evaluations. The goal of the sponsored research was to develop face recognition algorithms. The FERET database was collected to support the sponsored research and the FERET evaluations. The FERET evaluations were performed to measure progress in algorithm development and identify future research directions. Before FERET was created, only a few of these algorithms used a common database, let alone meet the desirable goal of being evaluated on a standard testing protocol that included separate training and testing sets. As a consequence, there was no method to make informed comparisons among various algorithms. The FERET database made it possible for researchers to develop algorithms on a common database and to report results in the literature using this database.
 - a. **August 1994:** It was designed to measure performance on algorithms that could automatically locate, normalize, and identify faces from a database. The test consisted of three subtests, each with a different gallery and probe set. The first subtest examined the ability of algorithms to recognize faces from a gallery of 316 individuals. The second subtest was the false-alarm test, which measured how well an algorithm rejects faces not in the gallery. The third subtest baselined the effects of pose changes on performance.
 - b. **March 1995:** The goal was to measure progress since the initial FERET evaluation, and to evaluate these algorithms on larger galleries (817 individuals). In this evaluation probe sets contained duplicate images: a

duplicate image was defined as an image of a person whose corresponding gallery image was taken on a different date.

- c. **September 1996:** The new evaluation protocol required algorithms to match a set of 3323 images against a set of 3816 images (algorithms had to perform approximately 12.6 million matches). The new protocol design allowed the determination of performance scores for multiple galleries and probe sets, and perform a more detailed performance analysis.
2. **FRVT (Face Recognition Vendor Test):** it was a series of large scale independent evaluations for face recognition systems.
 - a. **FRVT 2000:** demonstrated that another difficult task for 2D facial recognition systems is to recognize faces that are not represented in frontal pose. Most systems provide good 2D facial recognition performance when the image is frontal. If the pose of the face changes (both horizontally and/or vertically) the performance decrease.
 - b. **FRVT 2002:** demonstrated that the performance of the 2D algorithms are drastically reduced when using images of the same subject yet taken with very different illuminations (indoor & outdoor images even if acquired on the same day).
 - c. **FRVT 2006:** measured performance with sequestered data (data not previously seen by the researchers or developers). A standard dataset and test methodology was employed so that all participants were evenly evaluated. The government provided both the test data and the test environment to participants. The test environment was called the Biometric Experimentation Environment (BEE). The BEE was the FRVT 2006 infrastructure. It allowed the experimenter to focus on the experiment by simplifying test data management, experiment configuration, and the processing of results. Present challenges follow the BEE philosophy.
3. **FRGC (Face Recognition Grand Challenge):** is a challenge between different groups of researcher. The challenge is designed to achieve an increase in performance of 2D and 3D recognition algorithms, FRGC aimed at reliably achieving a verification rate of 98%. The dataset is composed by images captured with indoor controlled condition, indoor/outdoor uncontrolled conditions and 3D model. Experiments:

Matching for Exp1 and Exp4

The image (test) of a subject is matched against with the images (gallery) of all subjects in the database, and a similarity measure is computed. This is repeated for each test subject, to obtain a similarity matrix.

Matching for Exp3

Texture and shape of two 3D models are matched

- Exp 1: Controlled indoor still versus indoor still
- Exp 2: Indoor multi-still versus indoor multi-still
- Exp 3: 3D versus 3D
 - 3t, texture channel only
 - 3s, shape channel only
- Exp 4: Controlled indoor still versus uncontrolled still

- Exp5: 3D versus Controlled single still.
- Exp6: 3D versus Uncontrolled single still.

separately, and then the single similarities are fused into a single value. The 3D model (test) of a subject is compared with the models (gallery) of all subjects in the database to obtain a similarity measure. This is repeated for each subject to obtain a similarity matrix.

Matching for Exp2

Matching between two subjects is performed by comparing $N > 1$ images (test) with $M > 1$ images (gallery), and the obtained values are fused into a single one. Each subject in the Query set is matched against all subjects in the Target set.

Face spoofing and anti-spoofing (Lesson 8 bis)

Biometric spoofing is the act of fooling a biometric app by using a copy of performing an imitation of a biometric trait identifying a genuine subject. This is different from **camouflage** or **disguise** that is the attack performed by presenting an artifact biometric trait to the system pretending not to be oneself.

For **face spoofing** we have several types of attack: **2d face spoofing** (i.e. print attack consisting in show to the system a photo or a video), **3D mask attack** or **plastic surgery**.

To account for the photo print attack one can exploit a **liveness detection system** => try to detect whether what we have in front is a still image or a real face. The main difference between live face and photo is the depth. A real face is a three dimensional object, a photograph can be considered as a two dimensional planar structure. We have several methods:

- detect head movement (weak against photo motion and photo bending);
- fusing face-voice to verify movement of the lips with what the subject is saying;
- eye-blinking analysis;
- face prints contain printing quality defects that can be well detected using texture features. Performing micro-texture analysis noticing that face obviously reflects light differently from a photo (LBP show differences);
- take a picture of the image submitted by the subject => a photo of a photo has lower resolution than the photo of a face;
- estimate how the background change when the head moves => background should not move along with the face (does not work with 2D mask)
- image distortion analysis (??)

For what regards video spoof attacks:

- Detect moirè patterns (the bands that are shown when you take a photo on a display)

For what regards 3d masks attack one can try to exploit differences between genuine and impostors with lights reflectance properties (e.g. skin reflects light differently from a mask made of silicon). Also a mask have a different texture than skin.

Another way is to measure the bpm of the person by detecting small changes in color under the skin epidermis, caused by variations in volume and oxygen saturation of the blood in the vessels, due to heart beats.

Requiring motion at a random time is sufficient to avoid an attack through a pre-recorded video (e.g., replay-attacks). Challenge-response may be spoofed by video, if the system would always ask a basic and always the same head motion, e.g. turn your head from left to right. This latter attack can be successfully addressed by requiring a specific motion type at random times, but this asks for a 3D model to track such motion and distinguish it from an appropriately presented photo.

One final way for video/photo anti-spoofing is to verify three dimensionality of the face through geometric invariants. Geometric Invariants are shape descriptors, that are not affected by object pose and scale, by perspective projection and intrinsic parameters of the camera. They are expressed as Ratios of distances/measures or as a combination of 3D/2D coordinates of the points of the object. In this method such definition is used in the following way: Given a configuration of points on an object, which are known as not coplanar, a geometric invariant which would instead require coplanarity is computed from them, on more consecutive images. If the pose of the subject in front of the capture device changes, but the computed cross ratio stays constant, the points from which it is computed must be coplanar. This would not be possible assuming a 3D face, and therefore the object is not 3D. By applying this argument to face recognition, we can distinguish a real face in front of a capture device from a picture.

FATCHA: ask the user to perform an action such as "Show this part of the face".

Ear recognition (Lesson 9)

Ear is a static and passive biometric. Ear has some advantages compared with face. It has less details, hence requires lower resolution; it is more uniform in color distribution and it is less sensitive to expression variation. Anyway differently from face, its 3D is subject to illumination and pose variation.

Human ear has sufficient variability to distinguish two different subjects (4 features are sufficient). It is quite equal during the time (the ear growth proportional from birth to the first 4 months, and the lobe is constant between 8 and 70 years, after this it further elongates due to tissue relaxation).

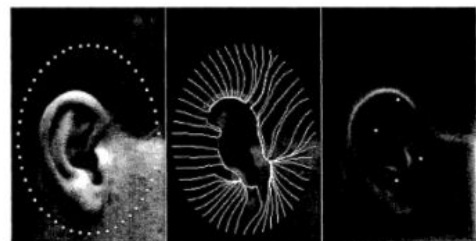


Ear localization:

1. Localization of points of interest: it is used a neural network that must be trained with a large amount of images. On each image a set of point of interest are detected. Such points determine reference system on the image.
2. General object detection: localization of the object within an image. In such case AdaBoost is used.
3. Geometric 3D methods: training is performed offline. Starting from a 3D model of the face profile, the points of maximum curvature are identified. A binary image is created and the region corresponding to the ear is manually extracted. Extracted regions are fused together to make up a reference model (template). Testing is performed online. The binary image is computed for the new model, the points of maximum and minimum curvature are identified and the regions corresponding to the template are searched for.

Ear recognition:

1. 2D geometric (global) approaches: the whole ear is considered starting from a 2D image and extracting relevant features from the region of interest (ROI) identified during localization.
 - a. Iannarelli system: identification of crus of helix (punto zero). This is the origin of the measurement system. Starting from this points, 12 different measurements are performed (if the point zero identification is incorrect, all wrong).
 - b. Voronoi diagrams: sensitive to pose and illumination. The surface of the ear is subdivided into regions (called Voronoi cells) by the Voronoi diagram. The matching process searches for subgraph isomorphism also considering possibly broken curves.
 - c. Force fields: a series of points are fixed along an ellipse around the ear, starting from these points the path from each point to the force field is followed. Field lines converge in points defined as sinks.
 - d. Jets: The presented approach introduces an "ear graph" whose vertices are labeled by the Gabor Jets at the body of the antihelix, superior antihelix crus, and inferior antihelix crus. These Jets are stored as ear graphs in the gallery, and PCA is used to obtain the eigen ear graph; an ear is detected using the similarity between sampled Jets and those reconstructed by the probe
 - e. Angle vectors
 - f. Alignment
2. 3D models: evaluate depth and curvature of relevant ear regions. Matching can be performed between the same region of two 3D models called patches.
3. Thermogram: ear image is captured by thermal camera. PROS: the ear is easily locatable, robustness to hair occlusion, different color facilitate segmentation. CONS: sensitive to movement, low resolution, high costs.



Face VS ear:

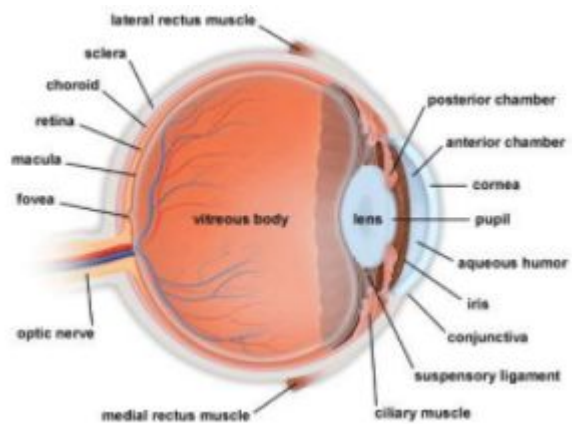
Tests were performed on a set of 294 subjects, with a total of 808 images. For each subject, one image for the face and one for the ear were provided at least.

Preprocessing entails in resizing each image at a resolution of 400×500 pixels. As for normalization, two reference points are provided, both for face and for ear, and photometric normalization is performed too. Regions not belonging to the ear are masked out.

Experiment #	Face/Ear compared		Expected Result	Result
1	Same day, different expression	Same day, opposite ear	Greater variation in expressions than ears; ears perform better	Face performs better
2	Different day, similar expression	Different day, same ear	Greater variation in expression across days; ears perform better	Face performs better
3	Different day, different expression	Different day, opposite ear	Greater variation in face expression than ear; ears perform better	Face performs better

Iris recognition (Lesson 10)

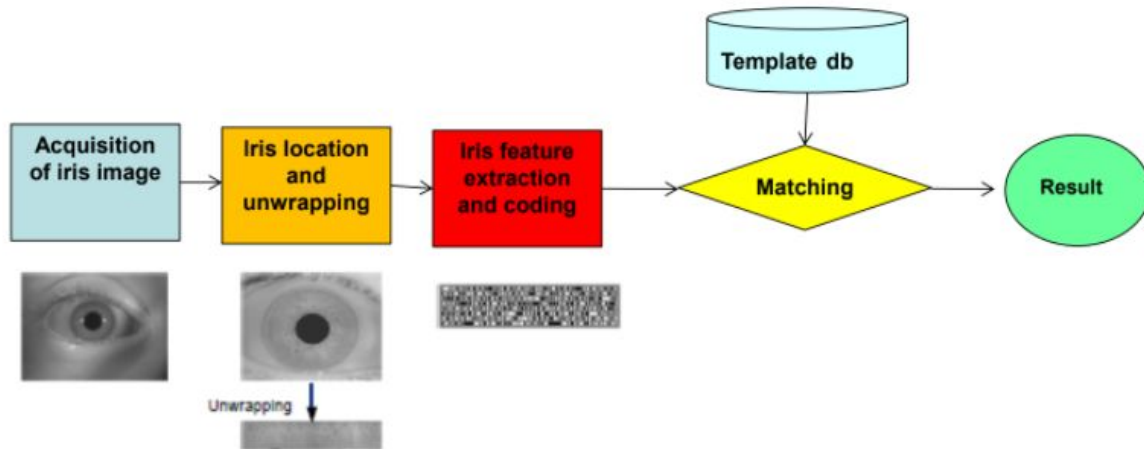
Iris is a muscle membrane situated behind the cornea, in front of the lens and perforated by the pupil. It is pigmented and it consists of a flat layer of muscle fibers which circularly surround the pupil. Iris has regular texture and irregular patterns, this factor provides a very high level of discrimination. Some of its advantages are the non-sensitive to time, its position. Some of iris disadvantages are the limited surface and the way in which the acquisition must be performed.



Capture modalities:

1. Visible light: melanin absorbs the light, layers of the iris visible, possibility of noisy. The noisy requires a good pre-processing/segmentation.
2. Infrared light: melanin reflects infrared light, texture more visible, complex devices. Also brown iris are more detailed.

Dougman system:



1. Iris location and unwrapping:

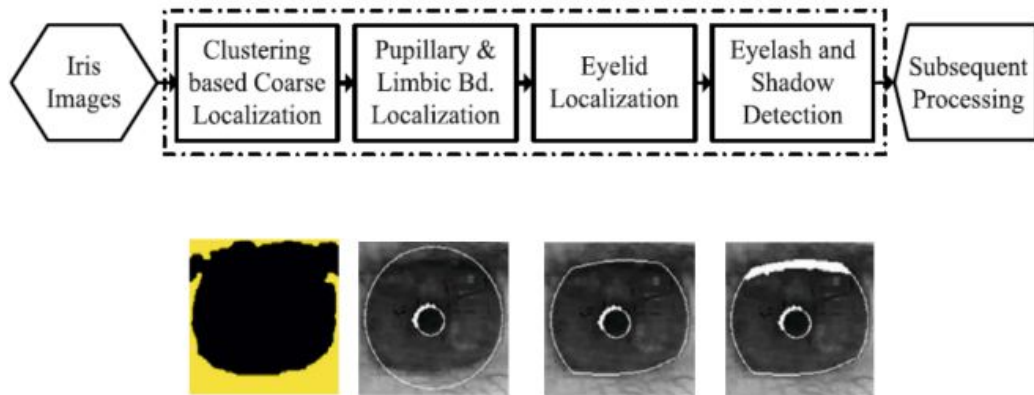
- IRIS LOCATION: this approach uses a kind of circular edge detector to localize both the pupil and the iris.
- EYELIDS LOCATION: as before but we detect also eyelids splines.
- IRIS SEGMENTATION: we obtain a mask so that only iris pixels are further processed.
- IRIS UNWRAPPING: through polar coordinates. Since the determination of the right centre for the polar coordinates is very important and since the pupil and the iris are not perfectly concentric and the size of the pupil can change, a normalization procedure is performed (Rubber Sheet Model).

2. Iris feature extraction and coding: Extraction is done by the use of gabor filters.

NICE (Noisy Iris Challenge Evaluation):

- Nice Dataset: UBIRIS database. The device for image capture was installed into a lounge under both natural and artificial condition. Latin caucasian 90%, black 8%, asian 2%. Two distinct acquisition sessions were performed each lasting two weeks and separated by an interval of one week. From the first to the second session, both the location and orientation of the acquisition device and artificial light sources were changed. Approximately 60% of the volunteers participated in both imaging sessions, whereas 40% participated exclusively in one or the other.
- Nice I: 97 participants from 22 countries. Evaluation of iris segmentation and noise detection techniques. Algorithms must run on Windows XP, Service Pack 2 or Fedora Core 6. No internet during the NICE I evaluation. Let Alg denotes the algorithm for iris segmentation of the free noisy region of the iris, which we want to evaluate. Then I is the set of data containing iris image, O is the set obtained from $Alg(I)$ where $i=1...|I|$, and C is the set of correct classified binary iris image provided by NICE I. Two measures of evaluation of Alg were used:
 - CLASSIFICATION ERROR RATE (E')
 - TYPE-I and TYPE-II ERROR RATE (E'')

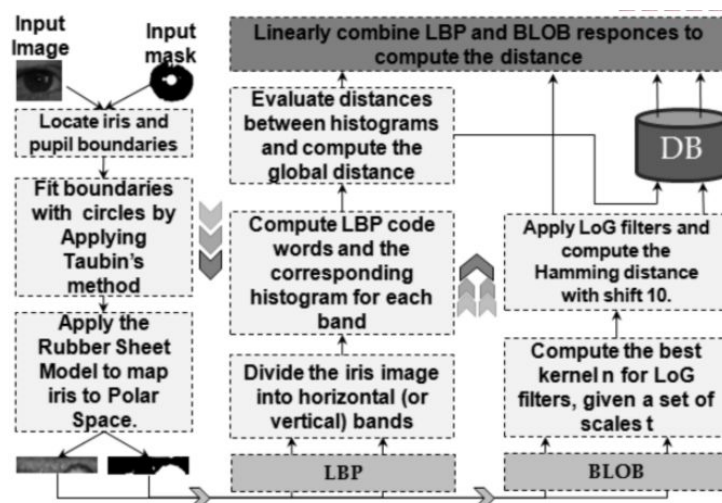
The winning algorithm was that proposed by CASIA:



Hence the algorithm has 4 phase:

- I. PRE-PROCESSING: this is necessary because there are some factors that negatively influence the iris detection (for example skin pores, eyelashes). For this purpose some filters are used (e.g. posterization filter, canny filter).
- II. PUPIL LOCATION: performed by circular detection. Even if the pupil is not a perfect circle, searching for a circle causes a lower error than obtaining a noise-conditioned ellipse. Many circle are found while searching for the pupil. For each candidate is computed the homogeneity value and the separability value. The final score assigned to each candidate is the sum of the previous two values. At last, the candidate having the highest score is selected.
- III. LINEARIZATION
- IV. LIMBUS LOCATION

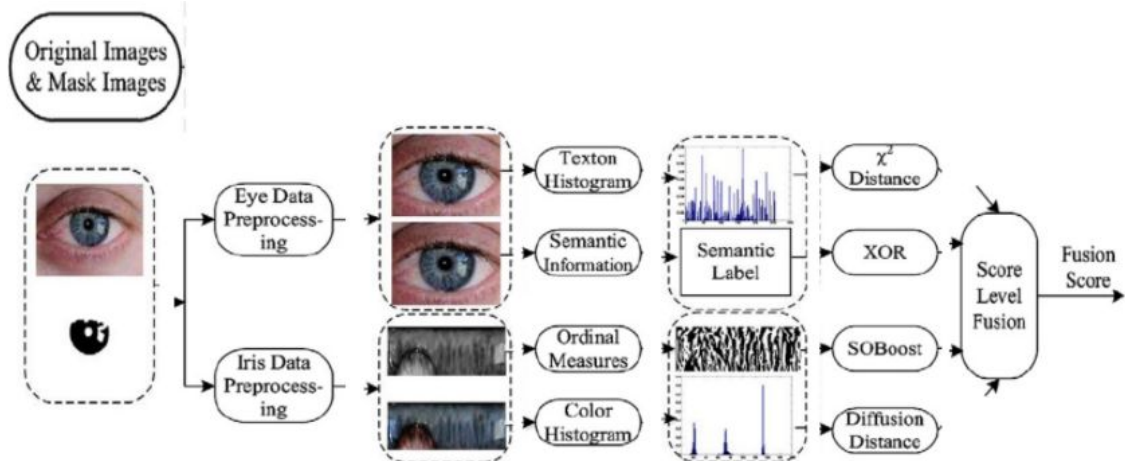
The following scheme represents the N-IRIS recognition system. In such a case LBP provides a division of the surface in horizontal or in vertical bands. Blobs are the iris features. LBP-BLOB fuses the two methods.



2. Nice II: evaluation of encoding and matching strategies for biometric signatures. 67 participants from over 32 countries. Algorithms must run on Windows XP, Service Pack 2 or Fedora Core 6. No internet during the NICE II evaluation. Let P denote the submitted application, I the set of iris image and M the set of the corresponding binary maps of each instance in I . Then P receives as input two iris (i.e. I', I'') with their binari maps (i.e. M', M''). The output will be a real positive value

indicating the dissimilarity between the segmented iris image in input. P is tested for each pairs in I . This gives a set of dissimilarity intra class values and dissimilarity inter class values (depending on whether the captured images are from the same or different irises). Hence a decidability value is computed as evaluation measures.

Once again the winning algorithm was that proposed by CASIA:



Fingerprints recognition (Lesson 11)

Fingerprints appear as a series of dark lines, that represents the high portion of friction ridge skin, while the space between such lines appears represents the shallow portion of friction ridge skin. Minutiae are major features of a fingerprint, using which comparisons of one print with another can be made. Examples of minutiae are in the right figure.



The formation of fingerprints is completed during the seventh month of the fetal development. Fingerprint shape is influenced by the amniotic fluid and the fetal position. The microdiversity of the environmental conditions in the immediate vicinity of each fingertip characterizes the formation of most minute details of the surface. In the case of identical twins, the minute details of the fingerprints are different, but most of the studies showed significant similarities in the classification of configurations relative to other generic attributes. The maximum difference between the fingerprints is to be found among individuals belonging to different races.

Regarding the acquisition of fingerprints we have two main methods. The first one is the off-line method (first the fingertips are passed on an ink pad and then the image is transferred pressing on a paper, second digitalization of the pressing image through optical scanner. There is also the possibility to capture the so called latent-fingerprint thank skin oil with special chemical reagents); the second one is live-scan method (direct contact of the fingertip with a special sensor).

The parameters of fingerprint digitalization are: the resolution, the area of acquisition, the depth, the contrast, the geometric distortion.

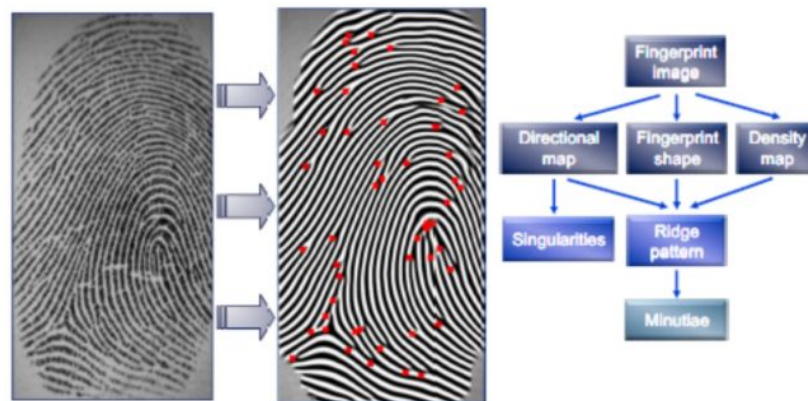
We have three types of scanner; optical scanner (PROS: inexpensive, robust to climate variation, good resolution. CONS: size, devices wear, residual prints from previous image can cause image degradation), capacitive scanner (PROS: better resolution of the fingerprint and smaller dimension of the sensor. CONS: duration), thermal scanner (PROS: impossible to deceive with artificial fingerprints. CONS: the image disappears quickly).

Factors for matching if two fingerprints belong to the same individual:

1. Configuration of the global pattern (common typology of the two compared fingerprints).
2. Qualitative accord (minutiae details identical).
3. Quantitative factor (minimum number of minutiae details in common).
4. Correspondence of minutiae details.

The methods for matching fingerprints can be summarized into 3 categories: matching based on correlation (the two fingerprints are overlapped to determine their similarities); matching based on ridge features (this is done when we have low quality of fingerprints causing problems for minutiae extraction, hence similarity compared through ridges); matching based on minutiae (minutiae are first extracted from the two fingerprints and stored as two sets of points in two dimensional space and then the methods search for the alignment between the two sets that maximizes the number of corresponding pairs of minutiae, and based on this measure the similarity between the fingerprints). Some problems regarding the extraction of fingerprints are: too much movement/distortion, non-linear distortion of the skin, variable pressure and skin conditions, errors in the extraction of the features.

The image below summarizes the steps involved in the feature extraction:



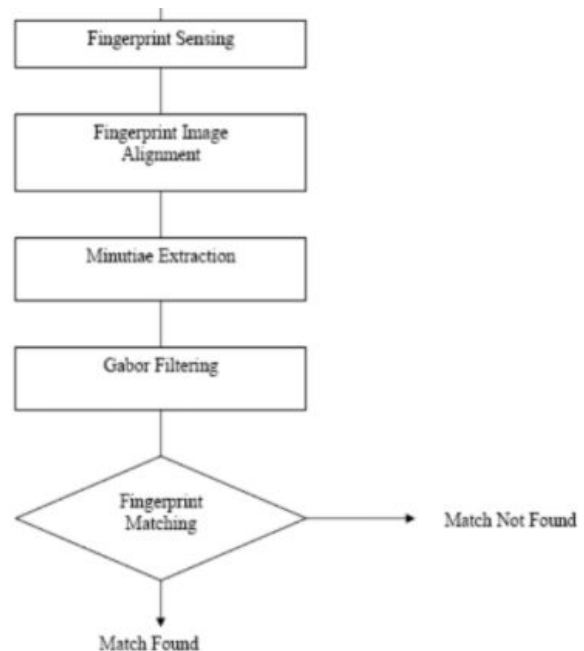
The first step after the acquisition of the fingerprint is its **segmentation**: the separation between the foreground (anisotropic) fingerprint from the background (isotropic).

After that, the **directional map** is computed. The directional map is a discrete matrix whose elements denote the orientation of the tangent to the ridge lines. In detail, each element $[i, j]$, in a grid superimposed to the fingerprint image indicates the average orientation of the tangent to ridge points in a neighborhood of the point. The **density map** is computed analogously. From the directional map **singularities** are extracted. Many approaches extract **minutiae** and perform matching based on them or in combination with them. Minutiae extraction entails: binarization (gray level image into binary image), thinning (reduces the thickness of the ridge lines to 1 pixels), location (locates pixels corresponding to the minutiae). Another important feature of the

fingerprints is the detection of the number of ridges between two significant points. This is called **ridge count**.

Another approach for fingerprint recognition is an hybrid one. This combines the representation of fingerprints based on minutiae with a representation based on Gabor filter that uses local texture information. It is summarized below.

1. Image alignment: The alignment phase starts with the extraction of minutiae from both the input and from the template to match. The two sets of minutiae are compared through an algorithm of point matching that preliminarily selects a pair of reference minutiae (one from each image), and then determines the number of matching of minutiae pairs using the remaining set of points. The reference pair that produces the maximum number of matching pairs, determines the best alignment. The regions of the background image of the fingertip input are masked out as unnecessary. At this point, the input images and the templates are normalized by building on them a grid that divides them into a series of non-overlapping windows of the same size and also normalizing the light intensity of the pixels within each window with reference to a constant mean and variance.
2. Feature extraction: in order to perform the feature extraction from the cells resulting from the tessellation a group of 8 Gabor filters is used, all with the same frequency, but with variable orientation. Such filtering produces 8 sorted images for each cell.
3. Matching: the comparison of the input image with the stored template is made by calculating the sum of the squared differences between corresponding characteristic vectors, after discarding the missing values. The similarity score is combined with that obtained with the comparison of minutiae, using the rule of the sum of the combinations. If the score of similarity is below a threshold, you can state that the input image has a corresponding template in memory and recognition is successful.



It is possible to reproduce fake fingerprints through gelatin, silicon, latex. The detection of the fake fingerprint can be done by determining if the source of the input signal is a living genuine biometric trait. One of the most common approaches for the vitality test are based on: pulse, temperature, pores, change of the color of the skin due to pressure, bloodstream and sweats. The potential difference between two specific points of the musculature of the finger can be used to distinguish it from a dead finger.

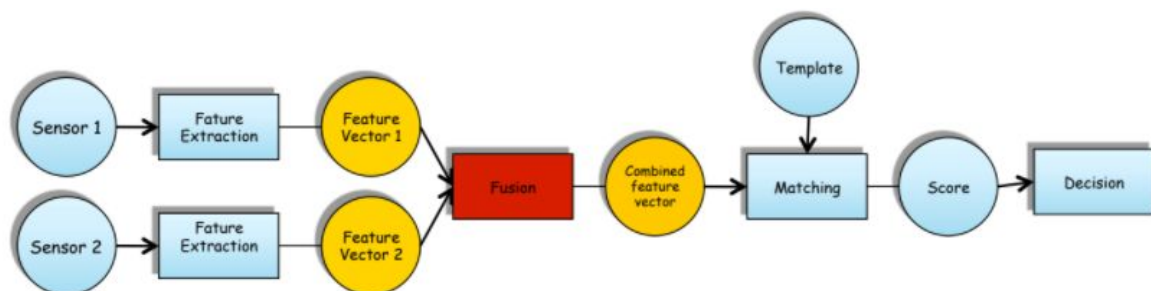
Failures of biometric system (Lesson 11bis)

1. **APPLE TOUCH ID:** Apple's iPhone 5s and later, and the iPad Air 2 and later, and the MacBook Pro 2016, all boast a Touch ID fingerprint sensor, which allows you to ditch your passcode in favour of your fingerprint. The first port of call in fixing an unresponsive fingerprint scanner is to clean the Touch ID sensor. This takes a high-resolution picture of the sub-epidermal layers of your skin to read your fingerprint and compare it to the fingerprint it has on file, so if there's dirt or grime on the Home button, the likelihood is your iPhone or iPad will have difficulty confirming that you really are who you say you are. You can clean the Home button with a lint-free cloth (like the kind you use to clean glasses or a tablet screen). Others problems can be: position of the fingertip on the home button, covers having cutouts for the home button quite tight, fingers not clean².
2. **APPLE FACE ID:** even if wired spends a lot of dollars for testing the robustness of FaceID, various events have occurred: it seems that FaceID is a little bit sensitive to kinship relation. Two brothers (not twins) have been able to unlock the same phone and the same things is happened with a child who unlocked his mother phone. Some videos have also shown that FaceID is subject to mask attack. Such a mask is printed in 3D, made with a mixture of crushed stone and polymers and with two-dimensional images of the eyes (crushed stone seems the secret!).
3. **FINGERPRINTS IN FORENSICS:** there are vary cases where the fingerprints are not used in a right way causing the sentence for innocents.

Multibiometric system (Lesson 12)

Instead of using one biometry, we use more than one. A multimodal system provides an effective solution, since the drawbacks of single systems can be counterbalanced thanks to the availability of more biometrics.

FEATURE LEVEL FUSION

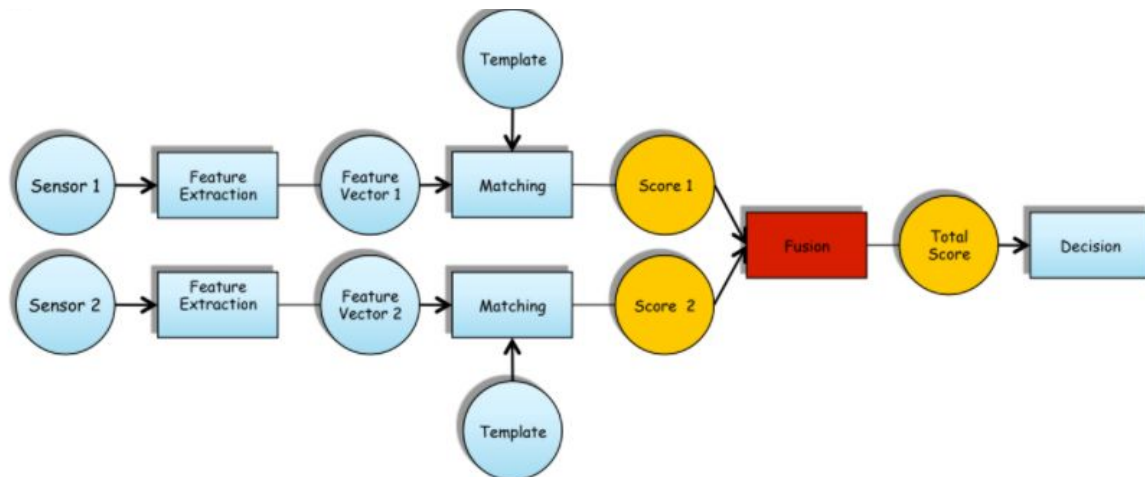


Features that were extracted with possibly different techniques can be fused to create a new feature vector to represent the individual. Problems are related to incompatible feature set, curse of dimensionality, complex matcher, noisy and redundant data. We have two types of fusion for obtaining the combined feature vector: serial (feature fusion is based on the simple linking of feature vectors. This approach requires the feature

² <https://www.macworld.co.uk/how-to/iphone/touch-id-fixes-3489429/>

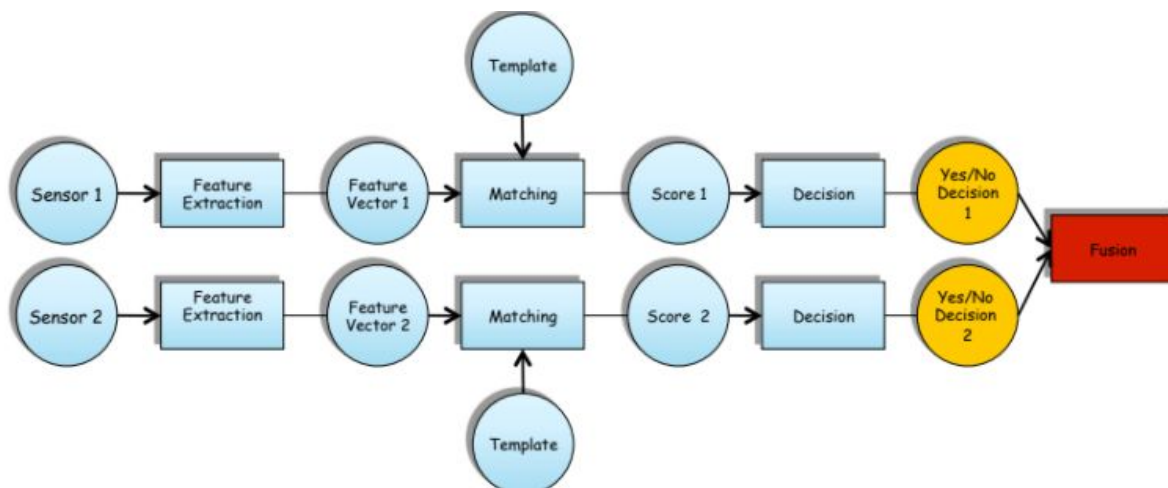
extraction phase and the feature normalization phase); parallel (the resultant vector is obtained through the parallel combination of the two feature vectors. This approach requires the vector normalization phase, the pre-processing of vectors and further feature processing); CCA (Canonical Correlation Analysis finds a pair of linear transformations, a and b , such to maximize the correlation coefficient between characteristics).

SCORE LEVEL FUSION



Different matching algorithms return a set of scores that are fused to generate a single final score. **Transformation-based** : the scores from different matchers are first normalized (transformed) in a common domain and then combined using **fusion rules** (that is each matcher is a classifier that votes for a class, the pattern is assigned to the most voted class. Reliability of multi-classifier is computed by averaging the single confidences). **Classifier-based**: the scores from different classifiers are considered as features and are included into a feature vector. A binary classifier is trained to discriminate between genuine and impostor score vectors (NN-Neural Networks, SVM – Support Vector Machine).

DECISION LEVEL FUSION



Each classifier outputs its decision (accept/reject for verification or identity for identification). The final decision is taken by combining the single decisions according to

a fusion rule. Different combination strategies are possible. The simplest ones imply a simple logical combination (serial combination AND, parallel combination OR). A further important fusion rule at decision level is Majority Voting.

Some research prototypes (Lesson 13)

PIFS & FARO????????

FACE - Face Analysis for Commercial Entities

1. Detection of face with Viola Jones
2. Eyes nose and mouth are detected (using the 68 point detection we also used in the project)
3. Face rotation is corrected by using eyes position
4. The face is divided symmetrically into left and right regions (by computing the line that passes through the center of the eyes, the tip of the nose and the center of the mouth)
5. The face region r with a better illumination is preserved while the other is discarded.
6. r is stretched so that predefined key points falls under predefined positions and finally the obtained image is mirrored to the other side in order to create a full face

FACE QUALITY MEASURES E GEOMETRIC INVARIANTS FOR FACE ANTISPOOFING LE ABBIAMO GIÀ FATTE.

FOVEA: video Frame Organizer Via identity Extraction and Analysis

Traces faces on a video. We have two kind of identities: **temporary** (belongs to a subject that is detected in **consecutive frames**) and **permanent** (identities that have been found in the whole video, they are distinct and to each may correspond a set of templates representing the face with different expressions) To trace faces in a video the system maintains a set V of faces currently examined (**temporary identities**) and updates it frame by frame basis. The system tracks the center of the nose for each face. For each new frame f it finds all the faces j inside f and computes the center of the nose (x_{new}^j, y_{new}^j) for each one and then computes the distance d with each x_i in V . If there is one identity k for which d is below some predefined threshold then j is mapped to i , otherwise a new temporary identity is created. If one temporary identity i is not found in a frame (but it was present in the previous one) then it is mapped onto a permanent identity: if a previous permanent identity i' for the subject already exists then i is mapped to i' .

HERO????????

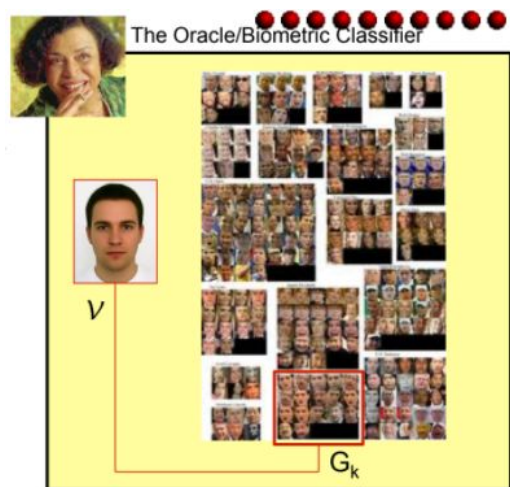
ISIS l'abbiamo già visto

Entropy of a gallery of templates (Lesson 14)

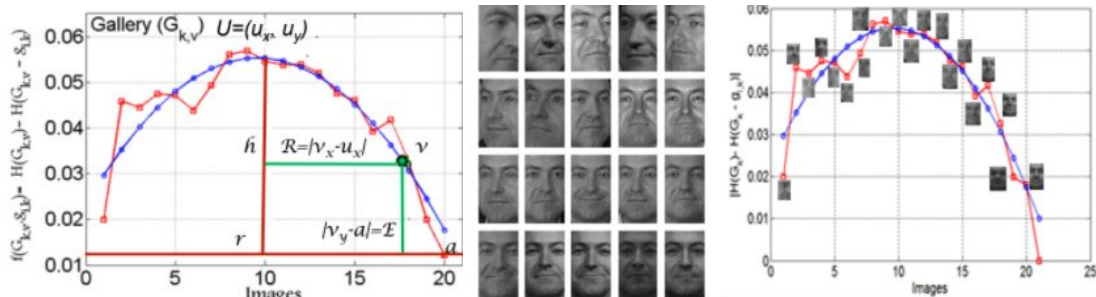
We will consider systems whose galleries contain more templates for each subject, and will explore the concept of representativeness of a biometric sample. This parameter varies from subject to subject, and is not a feature of the whole gallery. The gallery samples of certain subject might be too much similar, just because they are all of excellent quality, due to a systematic acquisitions in well-controlled conditions. In this situation, even a moderately different sample probe of the subject in input will cause an error, even by an accurate system, if the excellent conditions are not maintained. This seems to indicate that quality measures alone cannot guarantee good performances. Correct recognition in different situations may require a sufficient amount of variation in gallery templates. We call this (subject's) gallery feature representativeness, and investigate the role of mutual information/entropy in defining it.

1. **THE ORACLE:** let v be a template, we are interested in measuring the representativeness of G_k and how v alters it. G_k is the portion of the gallery of templates (such templates are denoted as $g_{i,k}$) corresponding to the identity of v . Assuming that an oracle has assigned v to the identity k , the score $s_{i,v}$ is: $s_{i,v} = p(v \approx g_{i,k})$. Clearly the sum of all $s_{i,v}$ is 1.

The **entropy of G_k with respect to a probe v** is $H(G_k | v)$. The **entropy for the gallery G_k** , $H(G_k)$, is computed by considering each gallery template $g_{i,k}$ in turn as a probe v .



2. **ENTROPY BASED ORDERING:** this procedure takes a gallery G_k as input, and starting from it computes a similarity matrix M_k and the value for $H(G_k)$. The matrix is computed by applying the similarity measure d to all pairs of templates in G_k . The matrix is $n \times n$, where n is the number of templates in G_k . After that, we compute the parabola as the simplest curve to approximate this behaviour with sufficient accuracy as follows: $f(G_{k,v}, g_{i,k}) = H(G_{k,v}) - H(G_{k,v} \setminus \{g_{i,k}\})$.

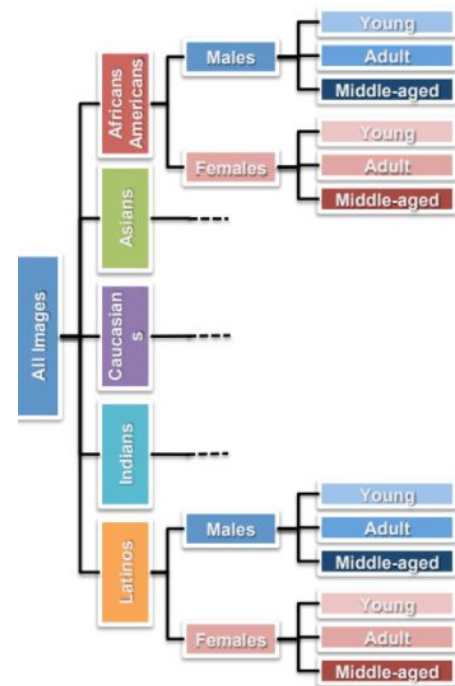


3. **WOLF DETECTION:** We experimentally observed that wolves present a much lower value for the area below the function $f(G_k, g_{i,k})$. It comes out that a threshold can be defined to well separate sheep from wolves.

4. **GALLERY UPDATING:** An important aspect in the process of identity handling is represented by the choice of when the gallery is to be restructured. In case of failure, the temporary set becomes the gallery of a new permanent identity; we can consider two different pruning strategies: Prune while merging (PWM) (the restructuring operation is delayed until merging; as soon as two sets of templates are fused, the whole resulting gallery will be restructured); Prune then merge (PTM) (as soon as no more face samples are found for a temporary identity, its collected set is pruned, before trying the merging).

5. **EGA DATASET:** *Ethnicity, gender and age*. The idea is to integrate into a single dataset face images from different databases, and organizing images according to individual features such as ethnicity, gender and age. The first version of the dataset merge six different datasets (CASIA, FEI, JAFFE, FERET, FRGC, INDIAN FACE DATABASE). EGA is organized as depicted to the right. Some observation:

- Some datasets contribute for only one ethnicity (africans-americans comes only from FERET).
- Asian ethnicity is the one taking from the largest number of datasets.
- FERET and FRGC provide a more or less significant contribution for most ethnicities.



Testing EGA is done through multiple classifiers. Each classifier is trained on a specific demographic function and any human intervention is avoided during normal system operations. So we have 5 classifiers (african-americans, asians, caucasians, indians, latinos). Two ways for testing: **A Priori Demographics Selection (APrDS)** (a system recognizes relevant demographic features, and each probe image is submitted to the corresponding classifier); **A Posteriori Demographics Selection (APoDS)** (the probe image is inputted to all the classifiers; to complete the recognition process, it is necessary to adopt a criterion for the selection of the global best answer).