# Unit 2

# **Biometrics**

# Topics

- *Biometric Sensors and Data Acquition*
- *Biometric Data Acquition and Database*
- *Biometric Pre-processing*
- ***Biometric Processing and Technologies***
- *Image Restoration & Segmentation*
- *Pattern Extraction & Classification*
- ***Fingerprint Recognition & Acquisition***
- *Fingerprint features, matching and synthesis*
- ***Face Recognition & Acqu****isition*
- *Face Detection, feature extraction and matching*
- ***Iris recognition & Acquisition***
- *Iris segmentation, normalization and matching*
- ***Ear Recognition and Detection***
- ***Hand Geometry Features and Palm print features***

# Session 1

- *Biometric Sensors and Data Acquition*
- *Biometric Data Acquition and Database*

# Biometric sensors

- A biometric sensor is a **transducer** that changes a biometric **traits** of a person into an **electrical signal**.
- Every biometric device requires at least **one type** of sensor
- The working of a sensor may be based on the measurement of **one or more physical quantities** like light intensity, capacitance, temperature, image, etc.

# Biometric sensors

# Types of Biometric Sensors

- Biometrics is broadly divided into **three categories** as follows
  - Biological/ Chemical biometrics
  - Morphological/Physiological biometrics
  - Behavioral biometrics

# Types of Biometric Sensors

- **Biological biometrics sensors** involve biological measurements at the **genetic or molecular** level, and **DNA sequencing** is a biological biometric system.
- It requires the **sampling** of DNA from blood or bodily fluids.
- This **cannot be used for security or authentication** systems but has other practical applications like DNA matching, genetic diseases, and microbiological studies.

# Types of Biometric Sensors

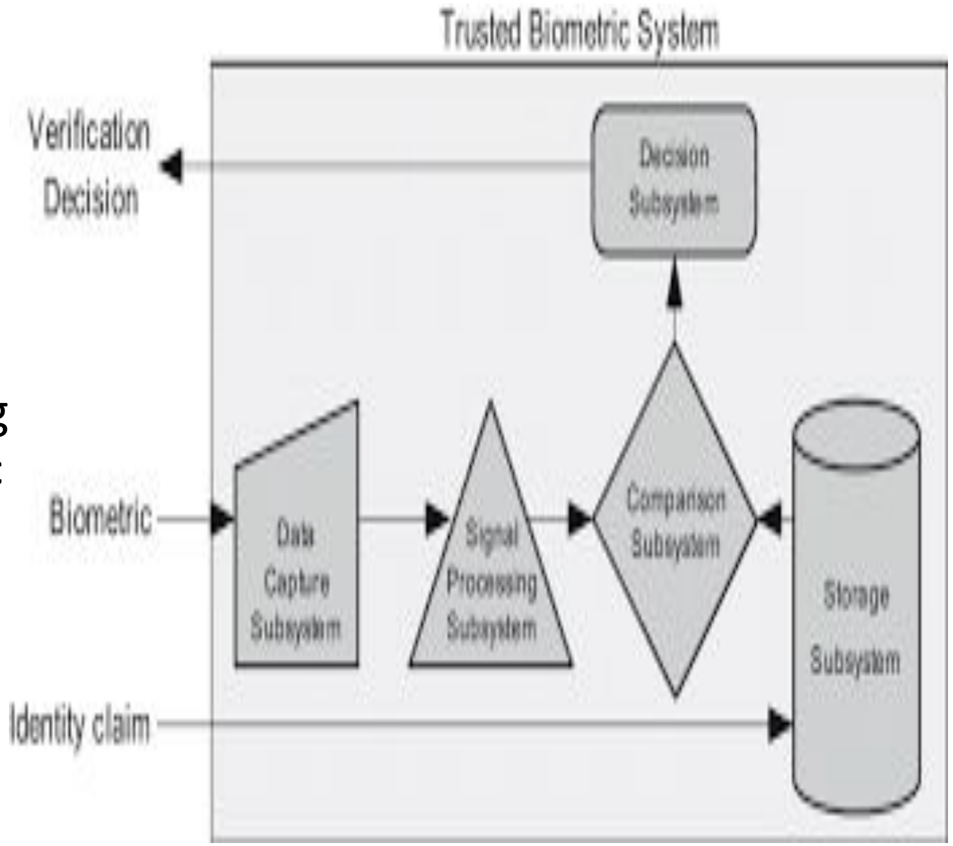
- **Morphological biometrics sensors** involve the measurement of **physical traits** and **body structures**.
- The **security** and **authentication systems** are usually based on one or more morphological biometrics.
- This includes fingerprint mapping, iris scan, face recognition, finger geometry recognition, vein recognition, hand geometry, ear recognition, and odour recognition.

# Types of Biometric Sensors

- **Behavioral biometrics sensors** involve the measurement of **behavioral identifiers** unique to a person.
- These systems are not common but are reserved for **special applications**.
- This includes signature recognition, voice biometrics, gait biometrics, keystroke recognition, and gesture recognition.

# Data Collection subsystem

- **Data Acquisition**
- Comprises **input device** or sensor that **reads** the biometric information from the user
- **Converts** biometric information into a suitable form for processing by the remainder of the biometric system
- **Examples**: video camera, fingerprint scanner, digital tablet, microphone, etc.



# Challenges in Data Collection

- The biometric **features** may change
- The **presentation** of the biometric feature at the sensor may change
- The **performance** of the sensor itself may change
- The **surrounding** environmental conditions may change

# Image Preprocessing

- Preprocessing involves steps to make **image** suitable for **feature extraction and matching**
- Most of the engineers spend a good amount of time in data pre-processing before building the model.
- The aim of pre-processing is an **improvement** of the image data that suppresses undesired distortions or enhances some image features relevant for further processing and analysis task.

# Topics

- *Biometric Sensors and Data Acquition*
- *Biometric Data Acquition and Database*
- ***Biometric Pre-processing***
- ***Biometric Processing and Technologies***
- *Image Restoration & Segmentation*
- *Pattern Extraction & Classification*
- *Fingerprint Recognition & Acquisition*
- *Fingerprint features, matching and synthesis*
- *Face Recognition & Acquisition*
- *Face Detection, feature extraction and matching*
- *Iris recognition & Acquisition*
- *Iris segmentation, normalization and matching*
- *Ear Recognition and Detection*
- *Hand Geometry Features and Palm print features*

# Image Preprocessing

The steps to be taken are :

- Read image
- Image Enhancement
  - Remove noise
  - Segmentation
  - Removing Specular Highlights
  - Region of Interest Detection and so on…
- Morphology(smoothing edges)

# Image Enhancement

- Image enhancement refers to the process of **highlighting** certain **information of an image**, as well as **removing** any **unnecessary information** according to specific needs.
- The **tools** used for image enhancement include many different kinds of **software** such as **filters**, **image editors** and **other tools** for changing various properties of an entire image or parts of an image.
- The principal objective of Image Enhancement is to **modify attributes** of an image to make it more suitable for a **given task** and a specific observer

# Image Enhancement

- The greatest difficulty in image enhancement is **quantifying** the **criterion** for enhancement and, therefore, a large number of image enhancement techniques are empirical and require interactive procedures to obtain satisfactory results.
- Image enhancement methods can be based on either **spatial** or **frequency domain** techniques


Spatial Domain
Frequency Domain
How to Enhance Images in
Spatial and Frequency Domain?

# Image Enhancement (Spatial Domain)

- Spatial domain techniques are performed to the image plane itself and they are based on direct manipulation of pixels in an image.
- The operation can be formulated as $g(x,y) = \mathbf{T}[f(x,y)]$, where $g$ is the output, $f$ is the input image and $\mathbf{T}$ is an operation on $f$ defined over some neighborhood of (x,y).
- According to the operations on the image pixels, it can be further divided into 2 categories: *Point operations* and *spatial operations* (including linear and non-linear operations).

# Image Enhancement (Frequency Domain)

## Frequency Domain

- These methods enhance an image $f(x,y)$ by convoluting the image with a linear, position invariant operator.
- The 2D convolution is performed in frequency domain with DFT.

Spatial domain: $g(x,y)=f(x,y)*h(x,y)$
Frequency domain: $G(w_1,w_2)=F(w_1,w_2)H(w_1,w_2)$

# Image Enhancement Techniques

## Image Enhancement by Point Processing

1. **Simple Intensity Transformation**
   a. Image Negatives
   b. Contrast Stretching
   c. Compression
   d. Gray-level Slicing
2. **Histogram Processing**
   a. Histogram Equalization
   b. Histogram Specification
   c. Local Enhancement

# Image Enhancement by Point Processing

These processing methods are based only on the **intensity** of single pixels.

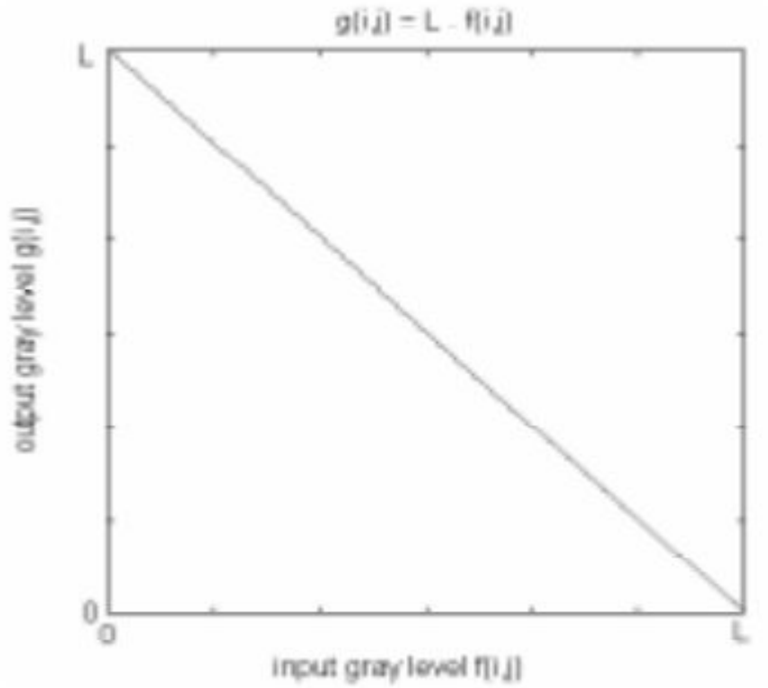
**1. Simple intensity transformation:**

**a. Image Negatives**

- Negatives of digital images are useful in numerous applications, such as displaying medical images and **photographing** a screen with **monochrome positive film** with the idea of using the resulting negatives as normal slides

# Image Enhancement by Point Processing

- Transform function T : g(x,y)=L-f(x,y), where L is the max. intensity.



# Image Enhancement by Point Processing

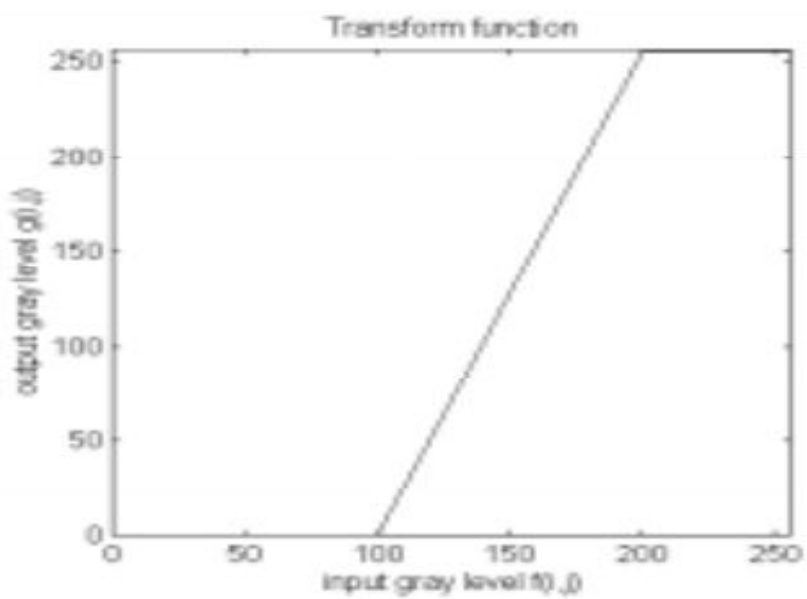
Original

Negative

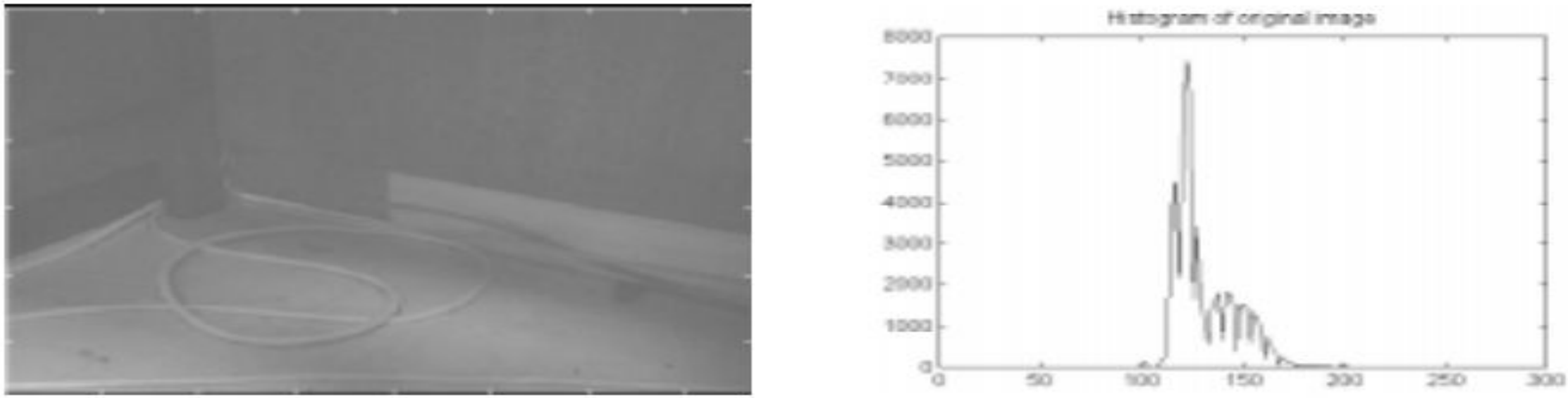
# Image Enhancement by Point Processing

(b). *Contrast stretching*

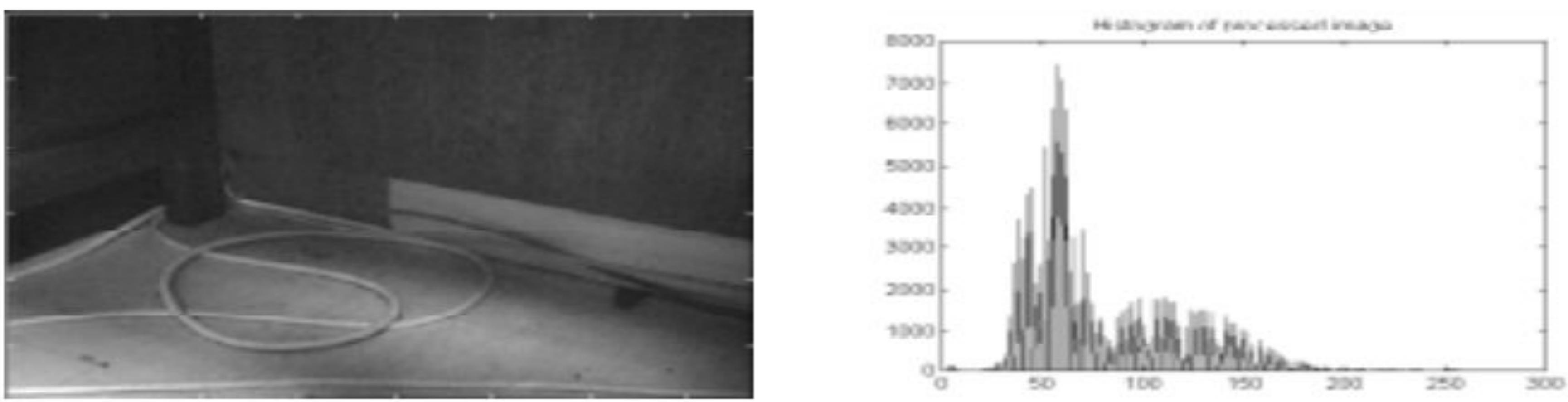
- Low-contrast images can result from poor illumination, lack of dynamic range in the image sensor, or even wrong setting of a lens aperture during image acquisition.
- The idea behind contrast stretching is to increase the dynamic range of the gray levels in the image being processed.



# Image Enhancement by Point Processing




Original




Processed image

# Image Enhancement by Point Processing

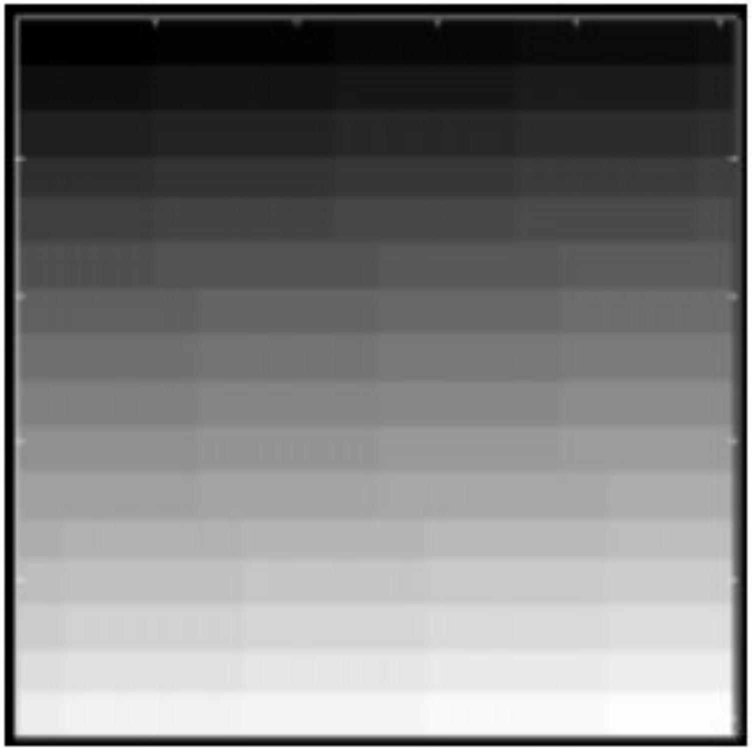
**C) Compression of dynamic range**

- Sometimes the dynamic range of a processed image **far exceeds** the capability of the display device, in which case only the brightest parts of the images are visible on the display screen.
- An effective way to compress the dynamic range of pixel values is to perform the following **intensity transformation** function:

$$s = c \log(1+|r|)$$

where c is a scaling constant, and the logarithm function performs the desired compression.

# Image Enhancement by Point Processing

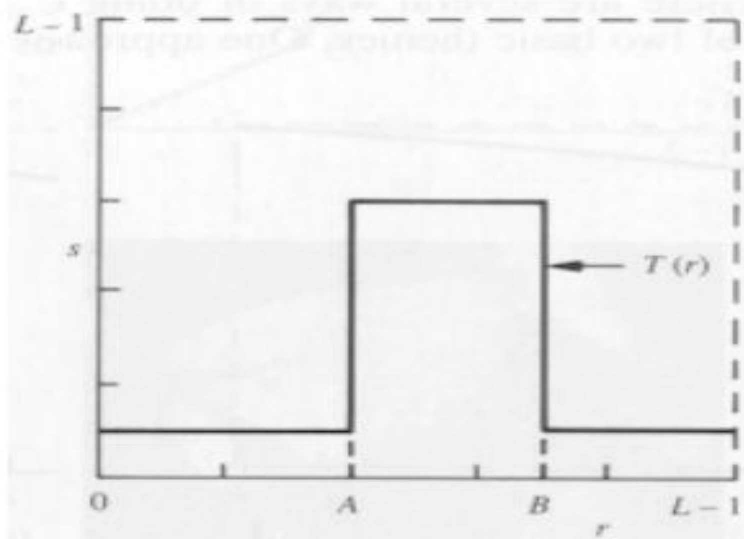


Original

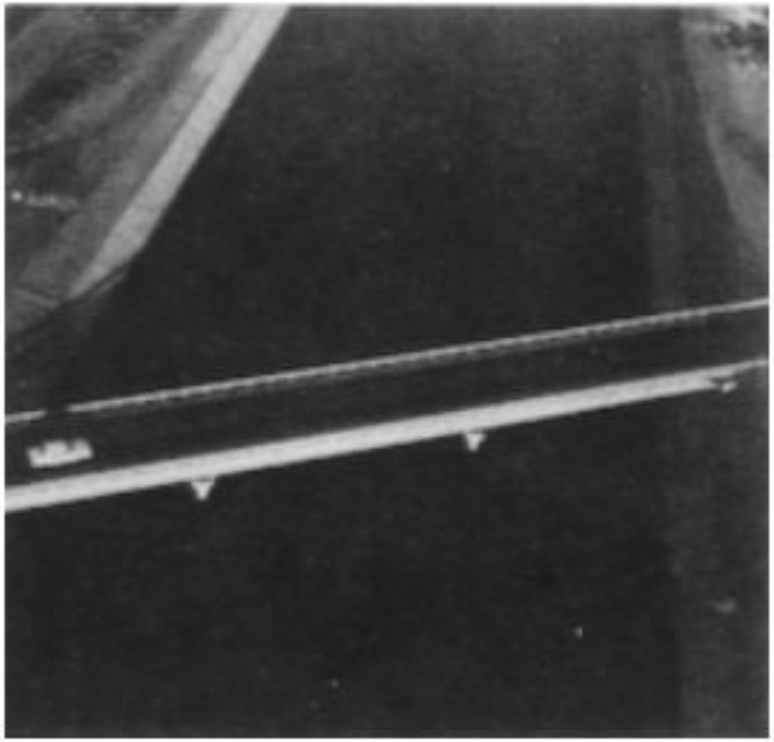
Processed output

## (d) **Gray-level slicing**

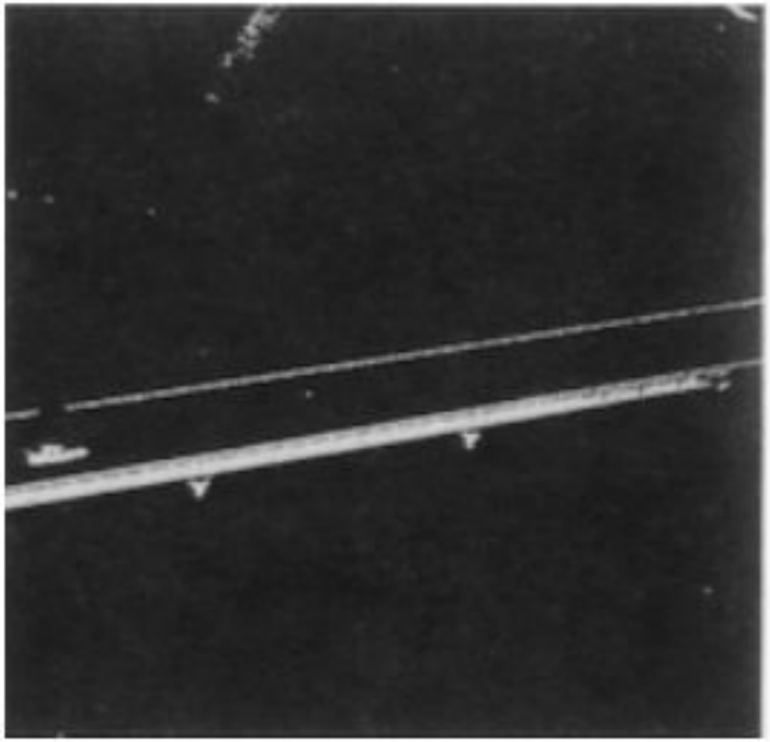
- Highlighting a **specific range** of gray levels in an image often is desired.
- Applications include enhancing features such as **masses of water** in satellite imagery and **enhancing flaws** in x-ray images
- Example: A transformation function that highlights a range [A,B] of intensities while diminishing all others to a constant.



# Image Enhancement by Point Processing



(b)



(c)

Fig 1. (a) Transfer function, (b) Original image, (c) Processing output.

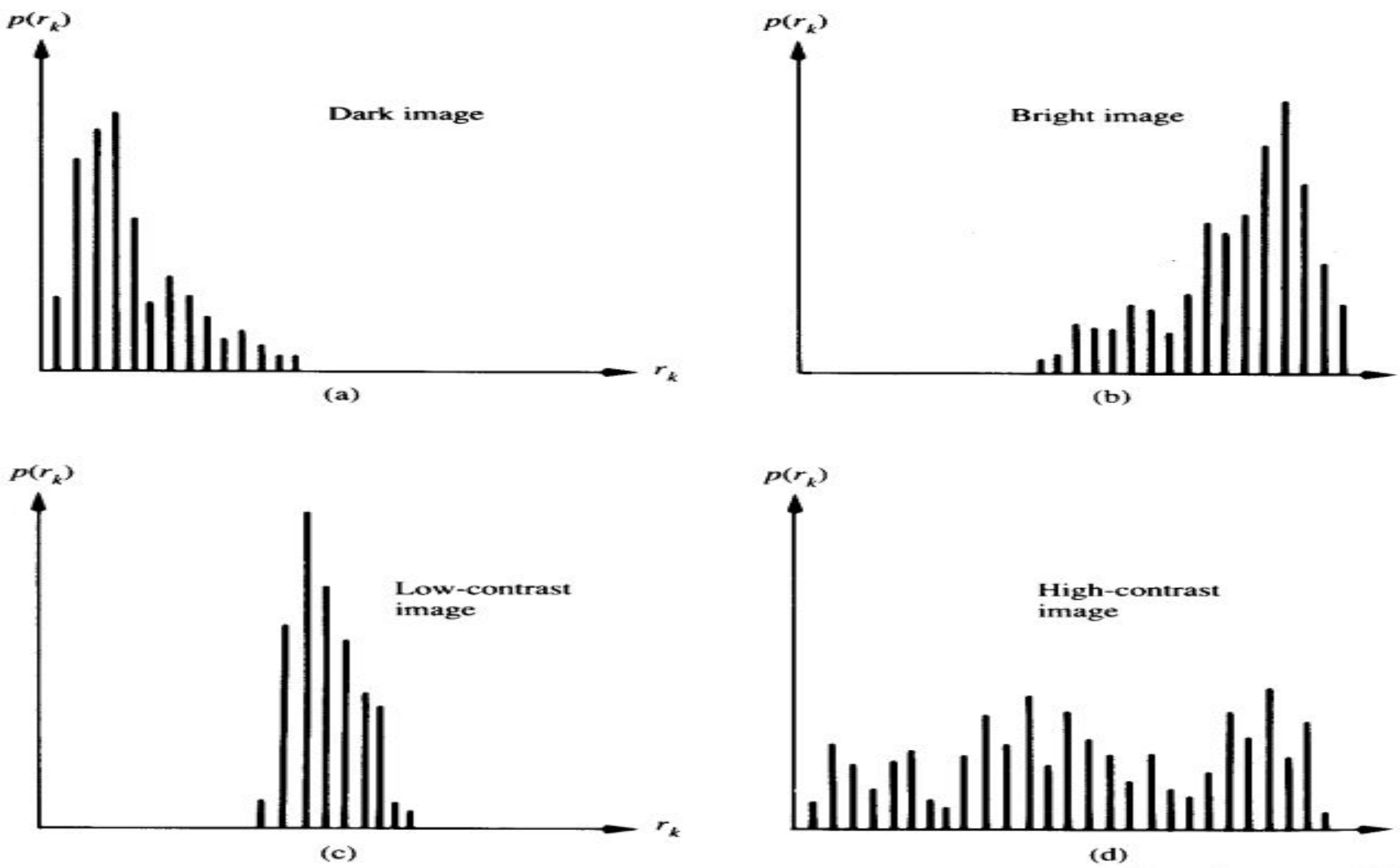
# Image Enhancement by Point Processing

## 2 Histogram processing:

- The histogram of a digital image with gray levels in the range [0,L-1] is a discrete function $p(r_k)=n_k/n$, where $r_k$ is the kth gray level, $n_k$ is the number of pixels in the image with that gray level, n is the total number of pixels in the image, and k=0,1..L-1.
- $P(r_k)$ gives an estimate of the probability of occurrence of gray level $r_k$.
- The shape of the histogram of an image gives us useful information about the possibility for contrast enhancement.
- A histogram of a narrow shape indicates little dynamic range and thus corresponds to an image having low contrast.

# Image Enhancement by Point Processing

## 2 Histogram processing:



# Image Enhancement by Point Processing

(a) *<u>Histogram equalization</u>*

- The objective is to map an input image to an output image such that its histogram. is uniform after the mapping.
- Let $r$ represent the gray levels in the image to be enhanced and $s$ is the enhanced output with a transformation of the form $s=\mathbf{T}(r)$.
- Assumption:
  1. $\mathbf{T}(r)$ is single-valued and monotonically increasing in the interval [0,1], which preserves the order from black to white in the gray scale.
  2. $0 \leq T(r) \leq 1$ for $0 \leq r \leq 1$, which guarantees the mapping is consistent with the allowed range of pixel values.

# Image Enhancement by Point Processing

- If $P_r(r)$ and $\mathbf{T}(r)$ are known and $\mathbf{T}^{-1}(s)$ satisfies condition (a), the pdf of the transformed gray levels is $P_s(s) = P_r(r)\left.\dfrac{dr}{ds}\right|_{r=T^{-1}(s)}$

- If $s = T(r) = \int_0^r P_r(w)dw$ for $0 \le r \le 1$, then we have $\dfrac{ds}{dr} = P_r(r)$ and hence $P_s(s) = 1$ for $0 \le s \le 1$.

- Using a transformation function equal to the cumulative distribution of $r$ produces an image whose gray levels have a uniform density, which implies an increase in the dynamic range of the pixels.

# Image Enhancement by Point Processing

- In order to be useful for digital image processing, eqns. should be formulated in discrete form:

$$P_r(r_k) = \frac{n_k}{n} \quad \text{and} \quad s_k = T(r_k) = \sum_{j=0}^{k} \frac{n_j}{n}, \text{ where } k=0,1\ldots L-1$$

- A plot of $P_r(r_k)$ versus $r_k$ is actually a histogram, and the technique used for obtaining a uniform histogram is known as histogram equalization or histogram linearization.

# Image Enhancement by Point Processing

- Example: Equalizing an image of 6 gray levels.

| Index $k$ | 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| Normalized Input level, $r_k/5$ | 0.0 | 0.2 | 0.4 | 0.6 | 0.8 | 1.0 |
| Freq. Count of $r_k$, $n_k$ | **4** | **7** | **2** | **1** | **0** | **1** |
| Probability $P(r_k) = n_k/n$ | 4/15 | 7/15 | 2/15 | 1/15 | 0/15 | 1/15 |
| $s_k = T(r_k) = \sum_{j=0}^{k} \frac{n_j}{n}$ | 4/15 = **0.27** | 11/15 = **0.73** | 13/15 = **0.87** | 14/15 = **0.93** | 14/15 = **0.93** | 15/15 = **1.00** |
| Quantized $s_k$ | 0.2 | 0.8 | 0.8 | 1.0 | 1.0 | 1.0 |

# Image Enhancement by Point Processing

| | freq |
|---|---|
| 0.0: | 4 |
| 0.2: | 7 |
| 0.4: | 2 |
| 0.6: | 1 |
| 0.8: | 0 |
| 1.0: | 1 |

| $r_k$ | T | $s_k$ | |
|---|---|---|---|
| 0.0 | $\longrightarrow$ | 0.2 | 4 |
| 0.2 | $\longrightarrow$ | 0.8 | 7 |
| 0.4 | $\longrightarrow$ | 0.8 | 2 |
| 0.6 | $\longrightarrow$ | 1.0 | 1 |
| 0.8 | $\longrightarrow$ | 1.0 | 0 |
| 1.0 | $\longrightarrow$ | 1.0 | 1 |

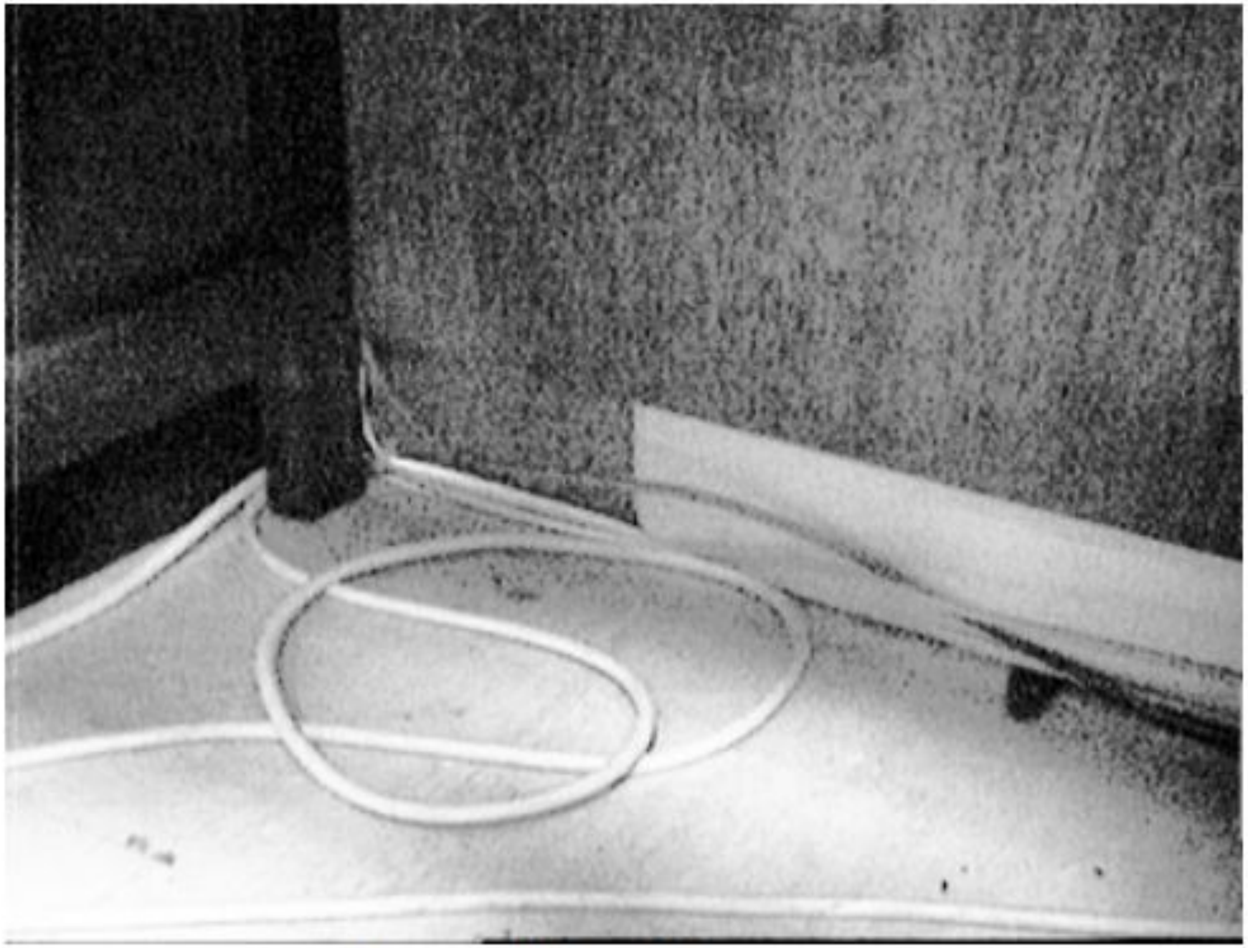
| | freq |
|---|---|
| 0.0: | 0 |
| 0.2: | 4 |
| 0.4: | 0 |
| 0.6: | 0 |
| 0.8: | 9 |
| 1.0: | 2 |

Fig 2. Example of histogram equalization

# Image Enhancement by Point Processing

Original image

# Image Enhancement by Point Processing



Equalized image

# Image Enhancement by Point Processing

(b) *Histogram specification*

- Histogram equalization only generates an approximation to a uniform histogram.
- Sometimes the ability to specify particular histogram shapes capable of highlighting certain gray-level ranges in an image is desirable.

# Image Enhancement by Point Processing

- Procedures:
  1. Determine the transformation $s_k = T(r_k)$ that can equalize the original image's histogram $p_r(r)$.
  2. Determine the transformation $s_k = G(b_k)$ that can equalize the desired image's histogram $p_b(b)$.
  3. Perform transformation $G^{-1}(T(r_k))$.

- The principal difficulty in applying the histogram specification method to image enhancement lies in being able to construct a meaningful histogram.

| | | | | |
|---|---|---|---|---|
| 0 | 2 | 1 | 3 | 4 |
| 1 | 3 | 4 | 3 | 3 |
| 0 | 1 | 3 | 1 | 4 |
| 3 | 1 | 4 | 2 | 0 |
| 0 | 4 | 2 | 4 | 4 |

**input: Image A**

| Pixel Value | Histogram | Equalized Histogram |
|---|---|---|
| 0 | 4 | 4 |
| 1 | 5 | 9 |
| 2 | 3 | 12 |
| 3 | 6 | 18 |
| 4 | 7 | 25 |

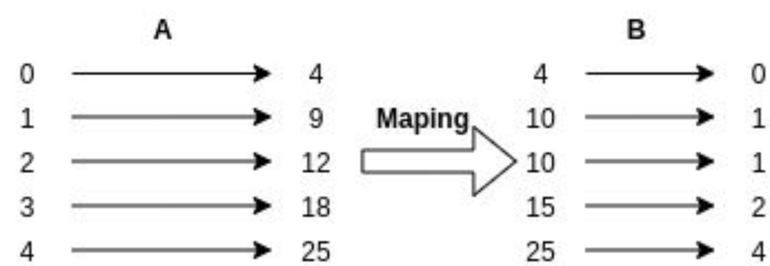
**Equalized Histogram of A**

| | | | | |
|---|---|---|---|---|
| 2 | 1 | 2 | 1 | 0 |
| 3 | 3 | 2 | 4 | 4 |
| 1 | 3 | 2 | 4 | 4 |
| 0 | 0 | 3 | 2 | 1 |
| 1 | 3 | 1 | 4 | 0 |

**Target: Image B**

| Pixel Value | Histogram | Equalized Histogram |
|---|---|---|
| 0 | 4 | 4 |
| 1 | 6 | 10 |
| 2 | 5 | 15 |
| 3 | 5 | 20 |
| 4 | 5 | 25 |

**Equalized Histogram of B**




| | | | | |
|---|---|---|---|---|
| 0 | 1 | 1 | 2 | 4 |
| 1 | 2 | 4 | 2 | 2 |
| 0 | 1 | 2 | 1 | 4 |
| 2 | 1 | 4 | 1 | 0 |
| 0 | 4 | 1 | 4 | 4 |

**Modified: Image A**

# Image Enhancement by Point Processing

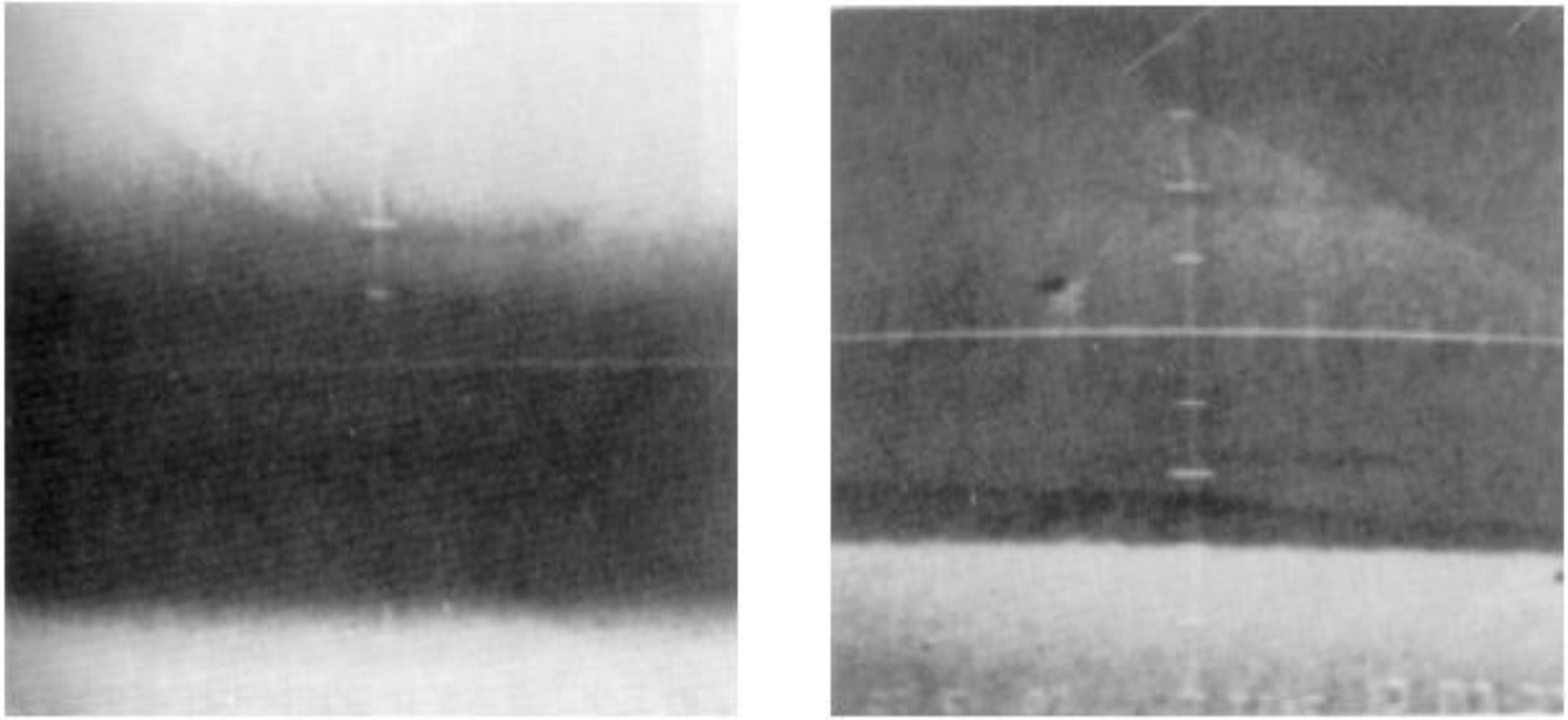
## (c) *Local enhancement*

- It is often necessary to enhance details over small areas.
- The number of pixels in these areas may have negligible influence on the computation of a global transformation, so the use of global histogram specification does not necessarily guarantee the desired local enhancement.

# Image Enhancement by Point Processing

- Procedures:
  1. Define a square or rectangular neighborhood and move the center of this window from pixel to pixel.
  2. Determine the histogram equalization or histogram specification transformation function with the histogram of the windowed image at each location.
  3. Map the gray level centers in the window with the transformation function.
  4. Move to an adjacent pixel location and the procedure is repeated.

# Image Enhancement by Point Processing



(a) (b)

Fig 3. Image before and after local enhancement.

# Enhancement in the frequency domain:

- We simply compute the Fourier transform of the image to be enhanced, multiply the result by a filter transfer function, and take the inverse transform to produce the enhanced image.

Spatial domain: $g(x,y)=f(x,y)*h(x,y)$

$$\Updownarrow$$

Frequency domain: $G(w_1,w_2)=F(w_1,w_2)H(w_1,w_2)$

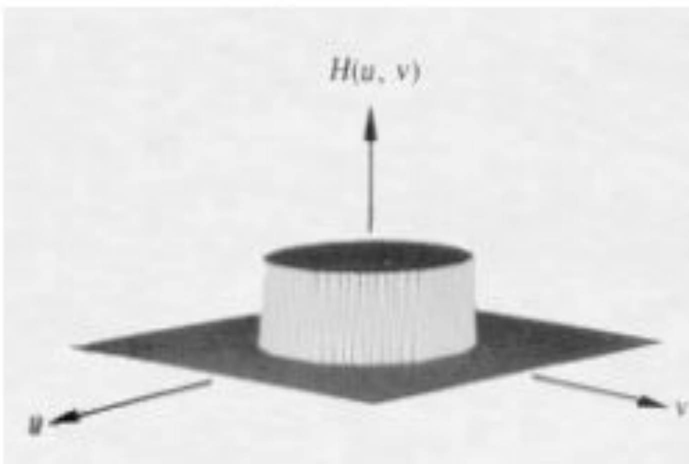
*Lowpass filtering*

- Edges and sharp transitions in the gray levels contribute to the high frequency content of its Fourier transform, so a lowpass filter smoothes an image.

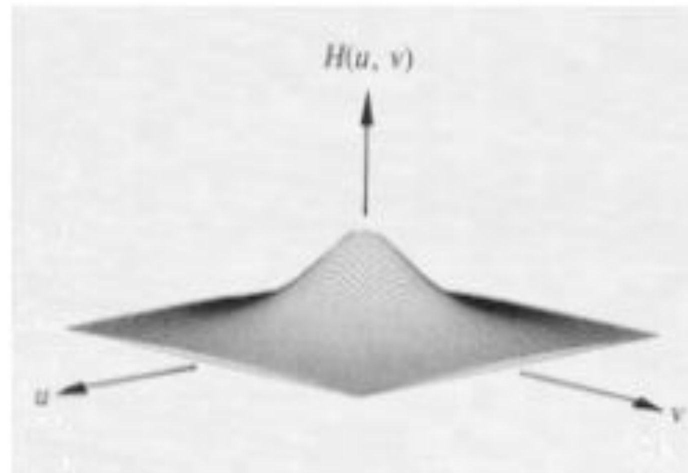
# Enhancement in the frequency domain:

- Formula of ideal LPF

$$H(u,v)=\begin{cases} 1 & \text{if } D(u,v)\le D_o \\ 0 & \text{else} \end{cases}$$




(a)




(b)

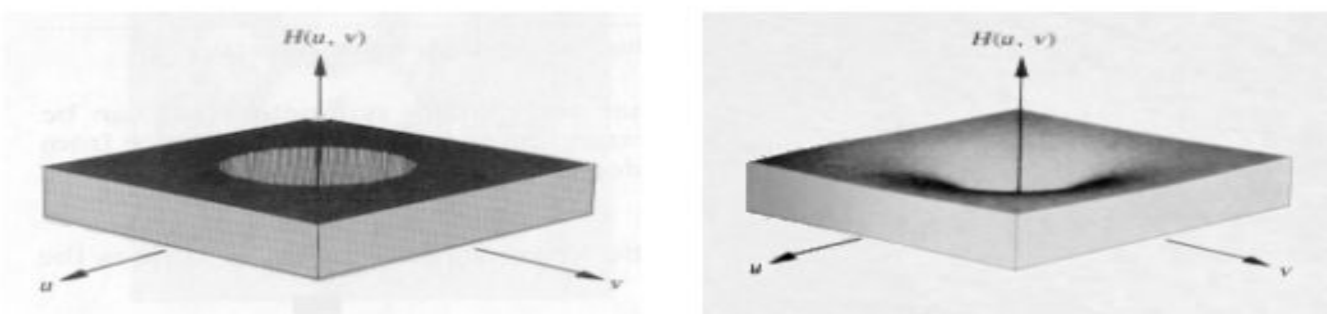
Fig 6. (a) Ideal LPF; (b) Butterworth LPF.

# Enhancement in the frequency domain:

*Highpass filtering*

- A highpass filter attenuates the low frequency components without disturbing the high frequency information in the Fourier transform domain can sharpen edges.
- Formula of ideal HPF function

$$H(u,v) = \begin{cases} 0 & \text{if } D(u,v) \le D_o \\ 1 & \text{else} \end{cases}$$

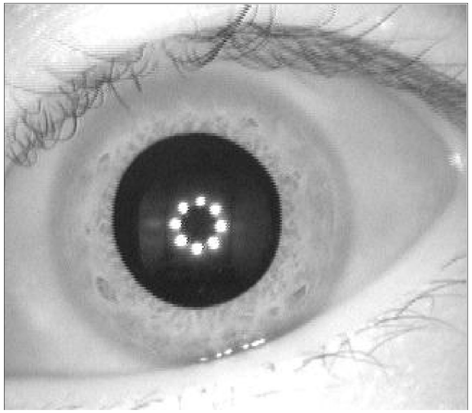



(a) (b)

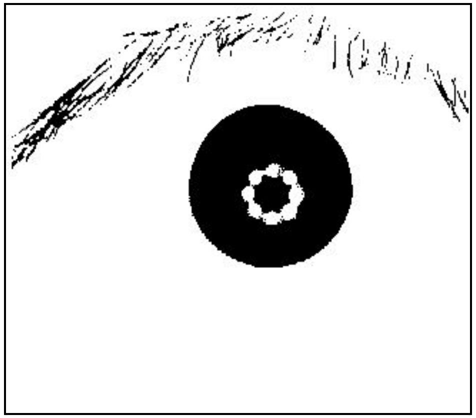
Fig 7. (a) Ideal HPF; (b) Butterworth HPF.

# Removing Specular Highlights

- Image contains specular highlights due to source of light
- Such highlights can be removed in 2 stages
  - **Adaptive Thresholding** - Adaptive thresholding is the method where the **threshold** value is calculated for **smaller regions** and therefore, there will be different threshold values for different regions.
  - **Hole Filling** - The hole-filling method is to **fill in** the **holes in the depth image** captured from the sensors used for rendering.



Input Image



Binarized Image with holes

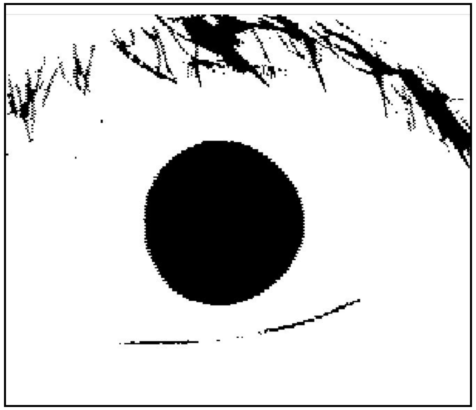


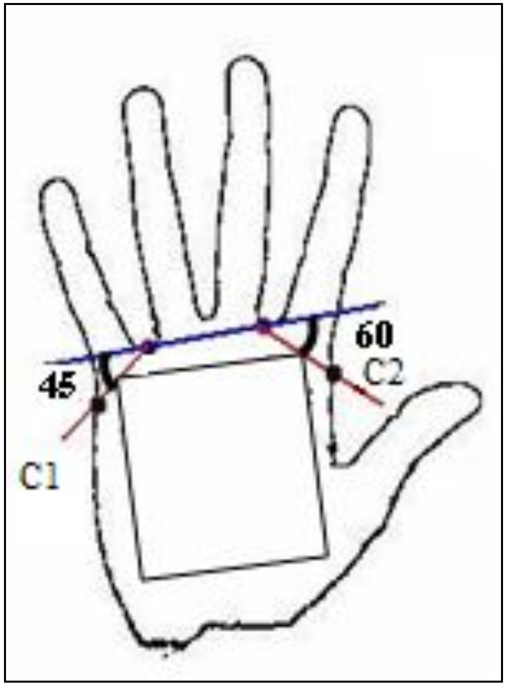
Image after hole filling

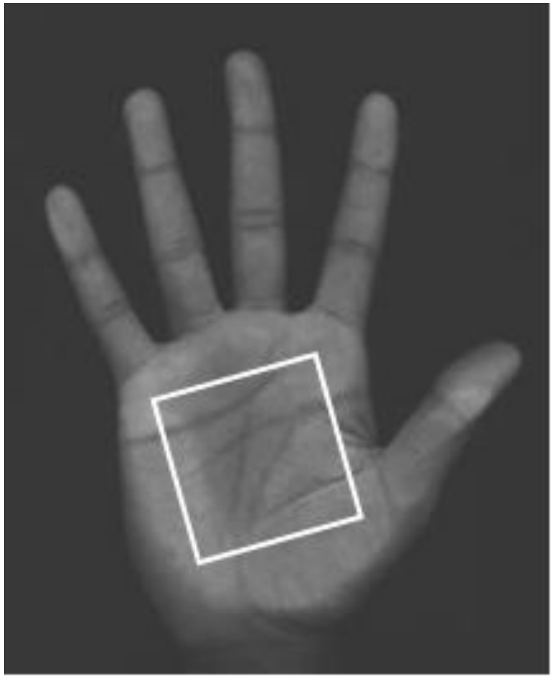
Removal of Specular Highlights from Iris

# Region of Interest Detection

- A **region of interest** (ROI) is a portion of an image that you **want to filter or operate** on in some way.
- You can represent an ROI as a **binary mask image**.
- In the mask image, pixels that belong to the ROI are set to 1 and pixels outside the ROI are set to 0.
- The Matlab toolbox offers several options to specify ROIs and create binary masks.
- It supports a set of objects that you can use to create ROIs of many shapes, such circles, ellipses, polygons, rectangles, and hand-drawn shapes.
- After you create the objects, you can modify their **shape, position, appearance, and behavior**.

# Palm print Extraction - ROI



Region of interest in Binary Image

Region of Interest in Original Image

# Topics

- *Biometric Sensors and Data Acquition*
- *Biometric Data Acquition and Database*
- *Biometric Pre-processing*
- *Biometric Processing and Technologies*
- ***Image Restoration & Segmentation***
- *Pattern Extraction & Classification*
- *Fingerprint Recognition & Acquisition*
- *Fingerprint features, matching and synthesis*
- *Face Recognition & Acquisition*
- *Face Detection, feature extraction and matching*
- *Iris recognition & Acquisition*
- *Iris segmentation, normalization and matching*
- *Ear Recognition and Detection*
- *Hand Geometry Features and Palm print features*

# Image Restoration

- **Image restoration** is the operation of taking a **corrupt/noisy** image and estimating the clean, **original image**.
- Corruption may come in many forms such as **motion blur, noise and camera mis-focus**.


# Image Restoration

- Image restoration is performed by **reversing** the process that blurred the image and such is performed by imaging a point source and use the point source image, which is called the **Point Spread Function** (PSF) to **restore** the image information lost to the blurring process

# Image degradations

## Degradations

- original

- optical blur

- motion blur

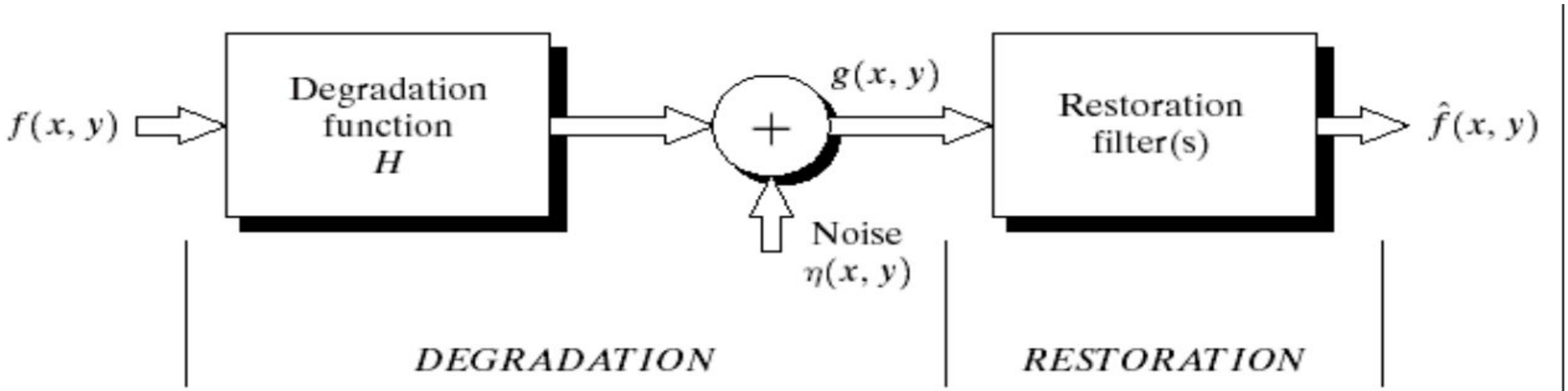
- spatial quantization (discrete pixels)

- additive intensity noise

# Image Restoration

- Image restoration is different from image enhancement
- Image enhancement is designed to emphasize features of the image that make the image more pleasing to the observer,
- Image restoration is to produce realistic data from a scientific point of view.
- The most straightforward and a conventional technique for image restoration is **deconvolution**, which is performed in the frequency domain

# Image Restoration model




$g(x,y)=f(x,y)*h(x,y)+\eta(x,y)$ – Spatial domain

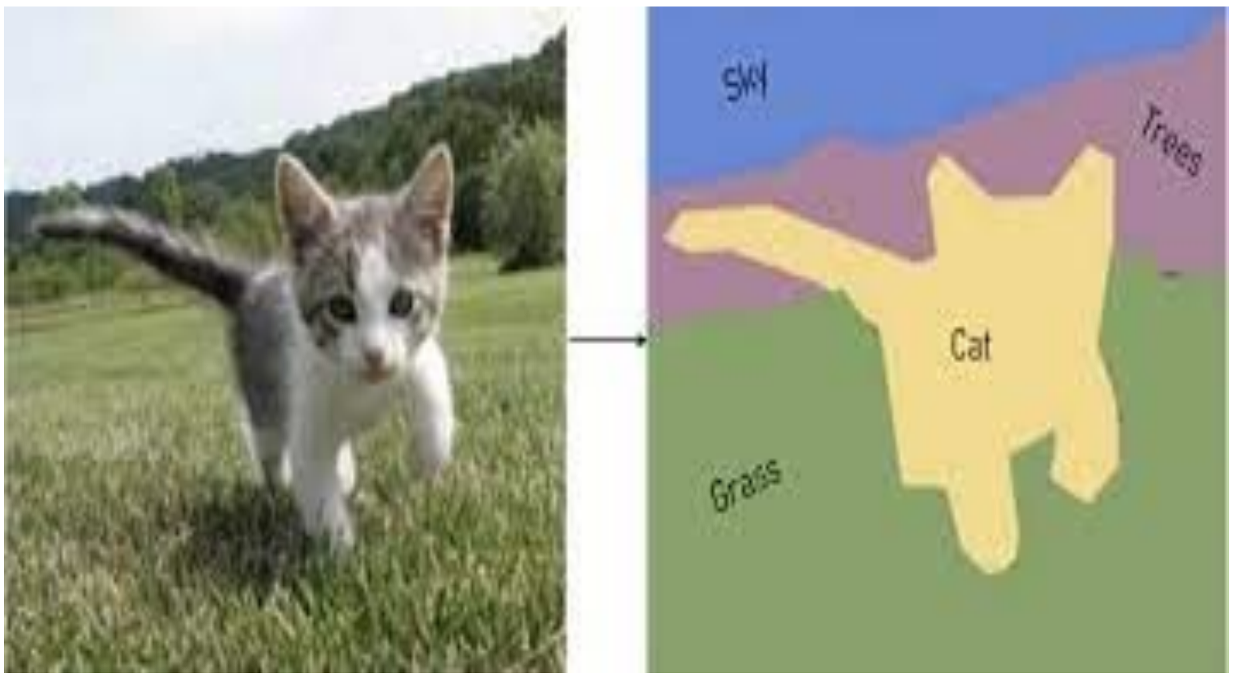
$G(u,v)=F(u,v)H(u,v)+N(u,v)$ – Frequency doma

- Where,
  - $f(x,y)$ - input image
  - $f^\wedge(x,y)$ - estimated original image
  - $g(x,y)$ - degraded image
  - $h(x,y)$ - degradation function
  - $\eta(x,y)$ - additive noise term

# Image Segmentation

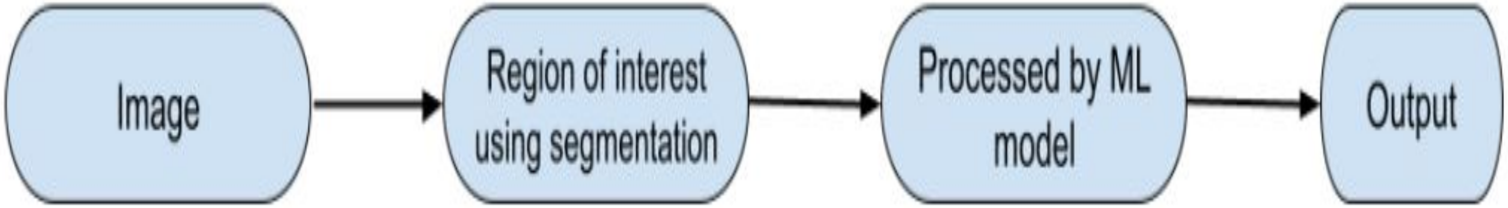
- Goal of Segmentation to **simplify** and/or **change** the **representation** of an image into something that is more **meaningful** and **easier** to **analyze**.
- Image segmentation is a method in which a digital image is **broken** down into various **subgroups** called **Image segments** which helps in **reducing** the **complexity** of the image to make further processing or analysis of the image simpler.
- Segmentation is **assigning labels to pixels**.
- All picture elements or pixels belonging to the **same category** have a **common label** assigned to them.

# Image Segmentation



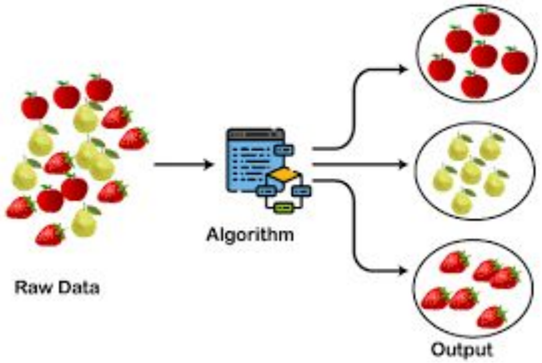
# Image Segmentation

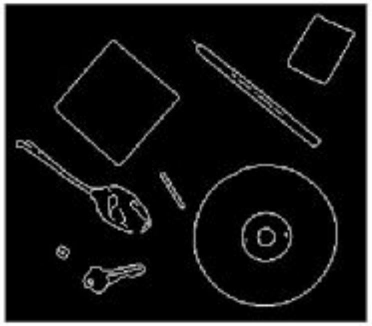
- For example: Let’s take a problem where the picture has to be provided as input for object detection.
- Rather than processing the whole image, the detector can be inputted with a **region selected** by a segmentation algorithm.
- This will prevent the detector from processing the whole image thereby reducing inference time.



# Approaches in Image Segmentation

- **Similarity approach**: This approach is based on detecting **similarity between image pixels** to form a segment, based on a **threshold.** ML algorithms like clustering are based on this type of approach to segment an image.
- **Discontinuity approach**: This approach relies on the **discontinuity of pixel intensity values** of the image. Line, Point, and Edge Detection techniques use this type of approach for obtaining intermediate segmentation results which can be later processed to obtain the final segmented image



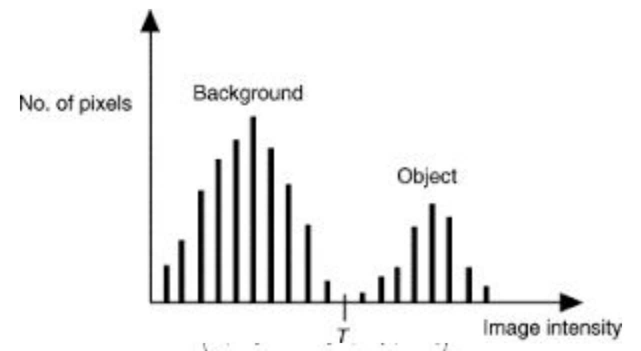



# Image Segmentation Techniques

1. Threshold Based Segmentation
2. Edge Based Segmentation
3. Region-Based Segmentation
4. Clustering Based Segmentation
5. Artificial Neural Network Based Segmentation

# Threshold based Segmentation

- Image thresholding segmentation is a simple form of image segmentation.
- It is a way to create a **binary** or **multi-color image** based on setting **a threshold** value on the **pixel intensity** of the original image
- In this thresholding process, we will consider the **intensity histogram** of all the pixels in the image.
- Then we will set a threshold to **divide** the image into sections.



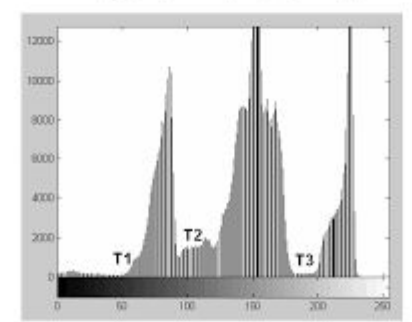



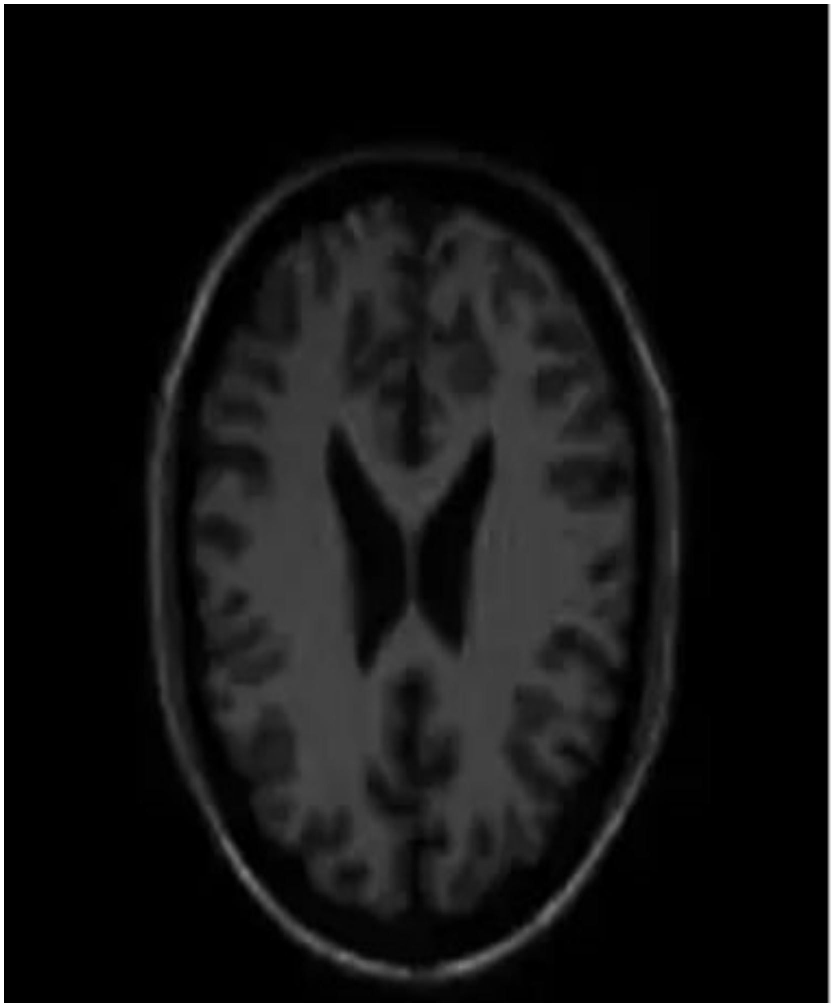

Figure 1. Threshold value multiples

III. PROPOSED METHOD

# Threshold based Segmentation

- For example, considering image pixels ranging from 0 to 255, we set a **threshold** of **60**.
- So all the pixels with values **less than or equal** to 60 will be provided with a value of 0(**black**) and all the pixels with a value **greater than** 60 will be provided with a value of 255(**white**).
- But this threshold has to be perfectly set to segment an image into an object and a background.

# Threshold Based Segmentation



# Edge Based Segmentation

- Edge-based segmentation relies on **edges** found in an image using various **edge detection** operators.
- These edges mark image locations of **discontinuity** in gray levels, color, texture, etc.
- When we move from one region to another, the gray level may change. So if we can find that discontinuity, we can find that edge
- We can use various edge detectors like Sobel edge operator, canny edge detector, Kirsch edge operator, Prewitt edge operator, Robert's edge operator, etc.

# Edge Based Segmentation

# Region-Based Segmentation

- Region-based segmentation involves dividing an image into **regions with similar characteristics**.
- The similarity between pixels can be in terms of intensity, color, etc.
- Each **region** is a group of pixels, which the algorithm locates via a **seed point**.
- Some **predefined rules** are set which have to be obeyed by a pixel in order to be classified into similar pixel regions. The preferred rule can be set as a **threshold**.
- Region-based segmentation methods are preferred over edge-based segmentation methods in case of a noisy image.

# Region-Based Segmentation

- Region-Based techniques are further classified into 2 types based on the approaches they follow.
  - Region growing method
  - Region splitting and merging method
- In Region growing method, we start with some pixel as the seed pixel and then check the adjacent pixels.
- If the adjacent pixels abide by the **predefined rules**, then that pixel is **added** to the region of the seed pixel and the following process continues till there is no similarity left.
- This method follows the bottom-up approach.

# Region-Based Segmentation

- In Region splitting, the whole image is first taken as a single region.
- If the region does not follow the predefined rules, then it is further divided into multiple regions (usually 4 quadrants) and then the predefined rules are carried out on those regions in order to decide whether to further subdivide or to classify that as a region.
- This process continues till there is no further division of regions required i.e every region follows the predefined rules.

# Region-Based Segmentation

- Usually, first region splitting is done on an image so as to split an image into maximum regions, and then these regions are merged in order to form a good segmented image of the original image.
- In case of Region splitting, the following condition can be checked in order to decide whether to subdivide a region or not.
- If the **absolute value of the difference** of the maximum and minimum pixel intensities in a region is less than or equal to a threshold value decided by the user, then the region does not require further splitting.

# Cluster-Based Segmentation

- Clustering algorithms are **unsupervised classification** algorithms that help identify **hidden** information in images.
- One of the most dominant clustering-based algorithms used for segmentation is **KMeans Clustering**.
- This type of clustering can be used to make segments in a **colored image**.
- The algorithm divides images into **clusters of pixels** with **similar characteristics**, separating data elements and grouping similar elements into clusters

# Cluster-Based Segmentation

$$|Z_{max} - Z_{min}| <= threshold$$

(Image by author)

$$Z_{max} \rightarrow Maximum\ pixel\ intensity\ value\ in\ a\ region.$$

(Image by author)

$$Z_{min} \rightarrow Minimum\ pixel\ intensity\ value\ in\ a\ region.$$

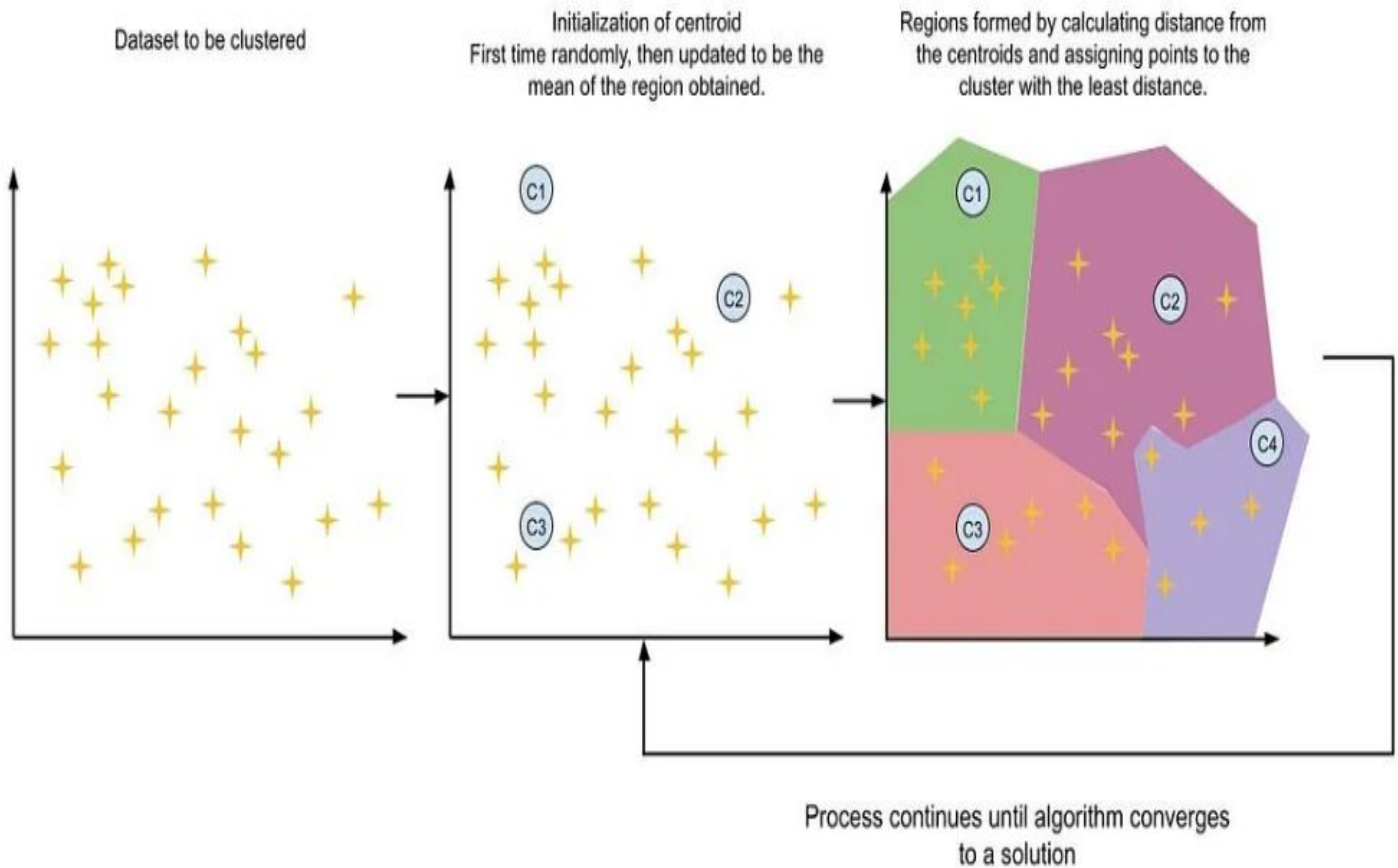
(Image by author)

# Cluster-Based Segmentation

**KMeans Clustering**

- Let's imagine a 2-dimensional dataset for better visualization.
- First, in the dataset, **centroids** (chosen by the user) are first randomly initialized.
- Then the **distance** of all the points to all the clusters is calculated and the point is **assigned** to **the cluster** with the **least distance**.
- Then centroids of all the clusters are **recalculated** by taking the mean of that cluster as the centroid.
- Then again data points are assigned to those clusters. And the process continues till the algorithm **converges** to a **good solution**.

# Cluster-Based Segmentation



# Pattern classification and Extraction

- **Pattern** is everything around in this digital world.
- A pattern can either be seen **physically** or it can be **observed mathematically** by applying algorithms.
- **Example:** The colors on the clothes, speech pattern, etc. In computer science, a pattern is represented using vector feature values.

# Pattern classification and Extraction

- **What is Pattern Recognition?**
  Pattern recognition can be defined as the **classification** of data based on **knowledge** already gained or on statistical information extracted from patterns and/or their representation.
- One of the important aspects of pattern recognition is its application potential.
- **Examples:** Speech recognition, speaker identification, multimedia document recognition (MDR), automatic medical diagnosis.

# Pattern classification and Extraction

- In a typical pattern recognition application, the raw data is **processed and converted** into a form that is amenable for a machine to use.
- Pattern recognition involves the **classification** and **cluster** of patterns.
- In **classification**, an appropriate **class label** is **assigned** to a pattern based on an **abstraction** that is generated using a set of **training patterns** or **domain knowledge**. Classification is used in supervised learning.
- Clustering generated a partition of the data which helps **decision making**, the specific decision-making activity of interest to us.
- **Clustering** is used in **unsupervised learning**.

# Pattern classification and Extraction

- **Pattern** is defined as **composite of features** that are characteristic of an individual.
- In **classification**, a pattern is a **pair of variables** {x,w} where **x** is a collection of observations or **features** (feature vector) and w is the concept behind the observation (**label**).
- The quality of a feature vector is related to its ability to discriminate examples from different classes
- Examples from the same class should have similar feature values and while examples from different classes having different feature values.

# Pattern classification and Extraction

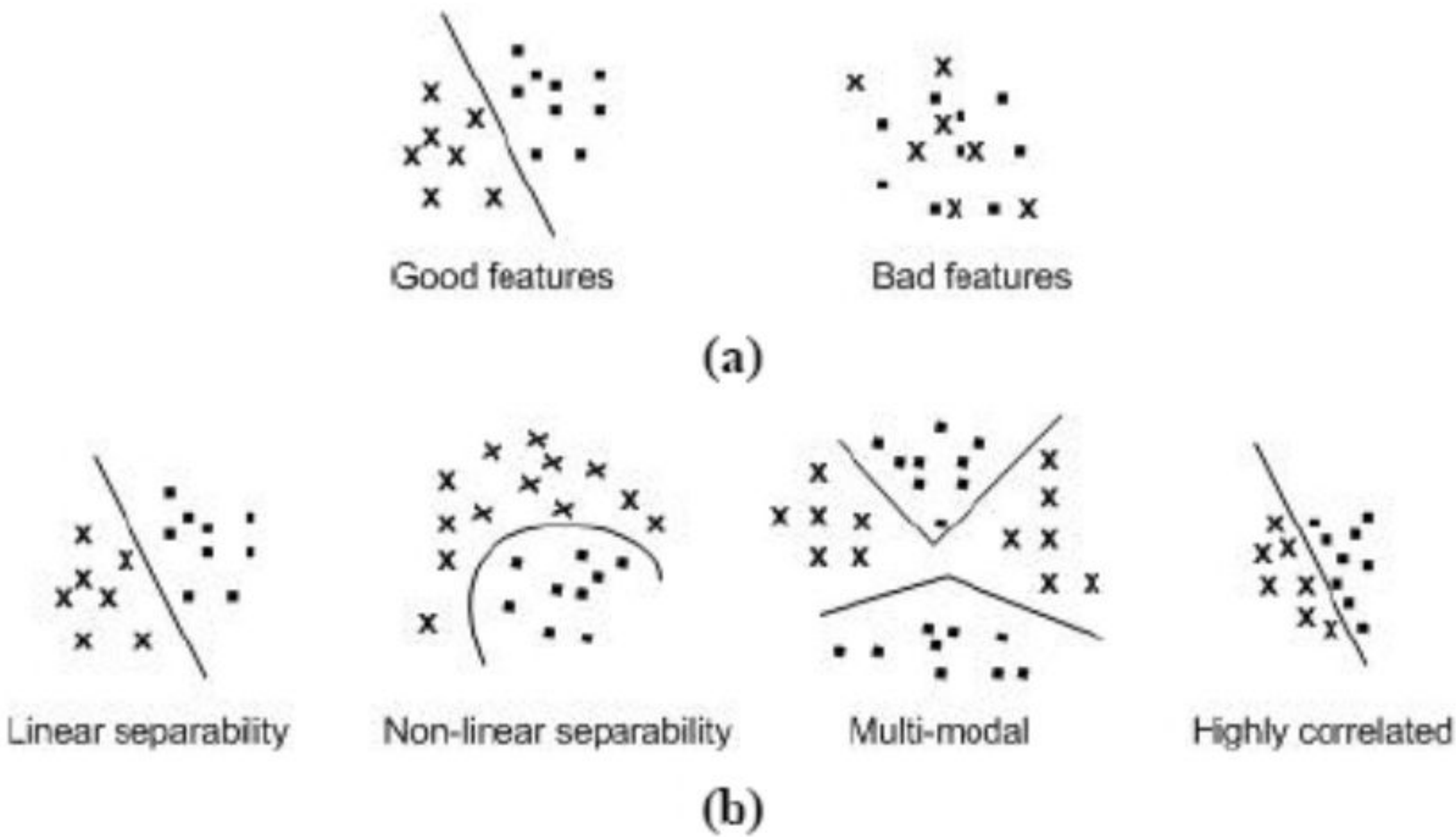
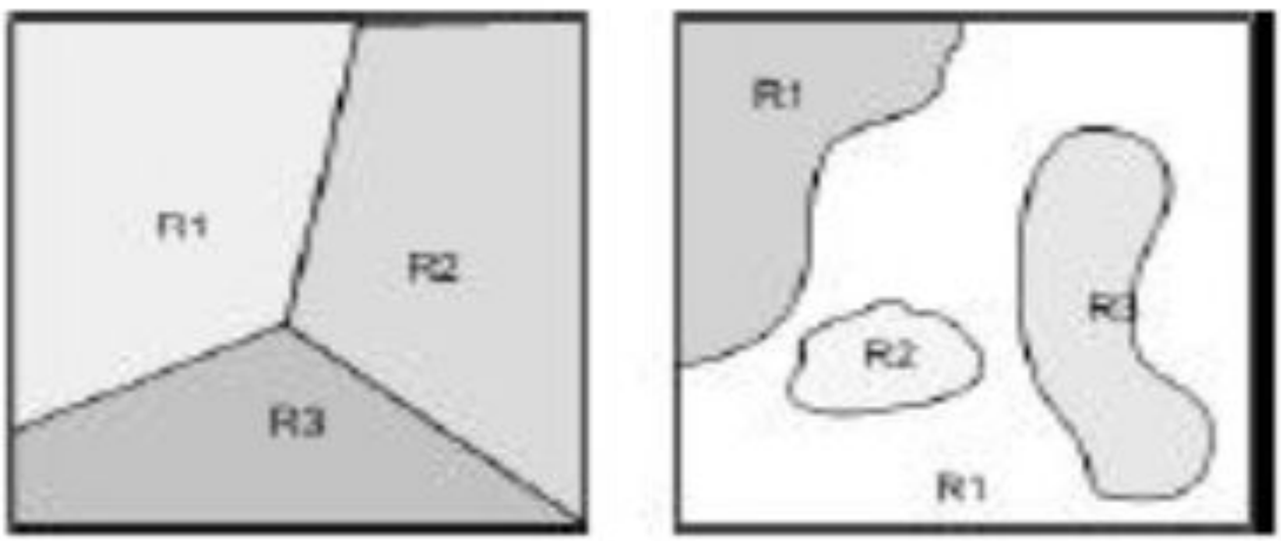



Fig: Characteristic (feature); a. the distinction between good and poor features, and b. feature properties

# Pattern classification and Extraction

- The goal of a classifier is to **partition** feature space into class-labeled decision regions.
- **Borders** between decision regions are called **decision boundaries**

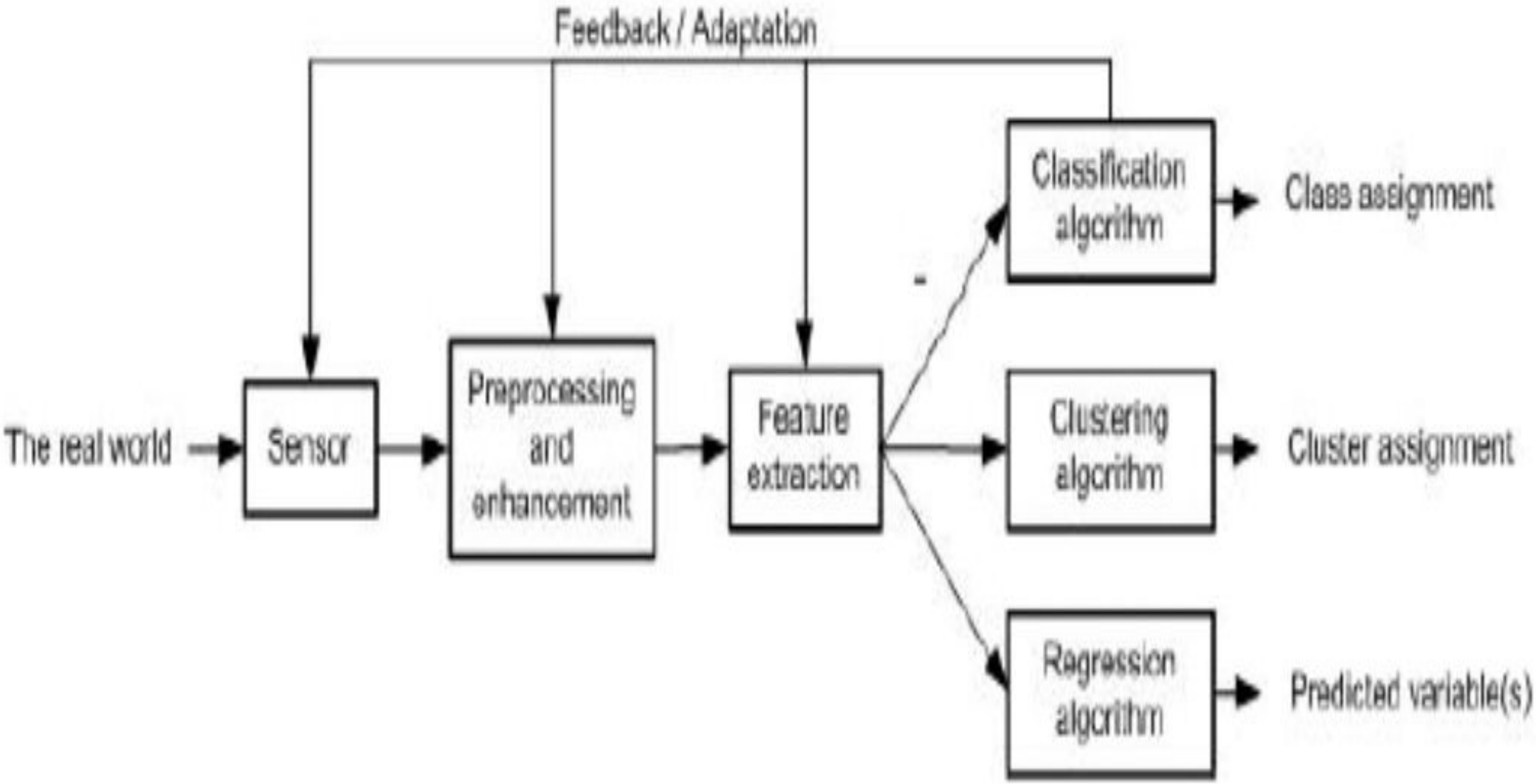



Classifier and decision boundaries

# Pattern classification and Extraction

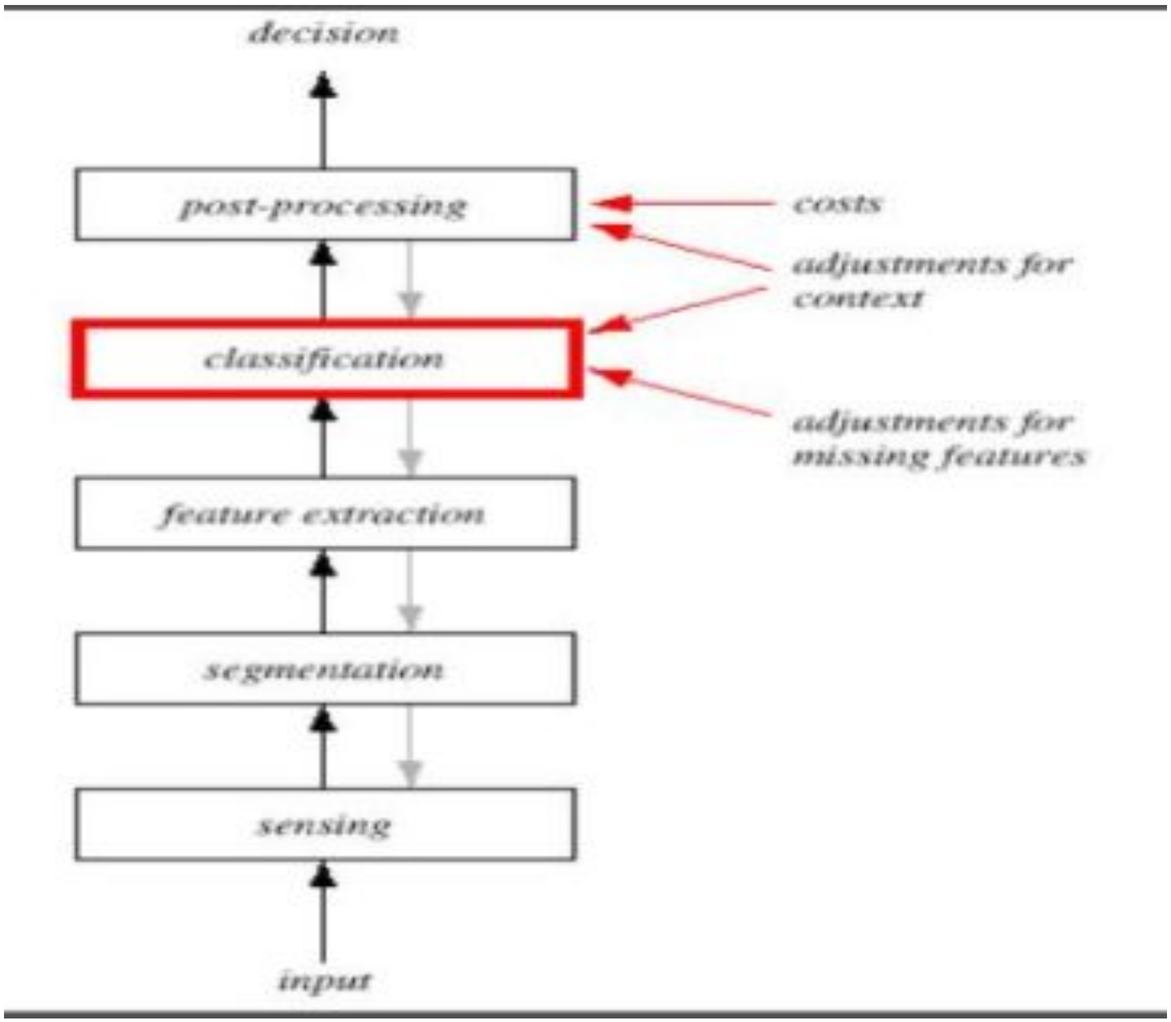
- If the characteristics or attributes of a class are **known**, individual objects might be **identified** as belonging or not belonging to that class.
- The objects are assigned to classes by observing **patterns** of **distinguishing characteristics** and comparing them to a **model** member of each class.
- A **pattern recognition system** contains a sensor, a preprocessing mechanism (segmentation), a feature extraction mechanism (manual or automated), a classification or description algorithm, and a set of examples (training set) already classified or described (post-processing)

# Pattern classification and Extraction




A pattern recognition system

# Steps in making decision



# Pattern classification and Extraction

**Sensing**

- The input to a pattern recognition system is often some kind of a transducer, such as a camera or a microphone array.
- The difficulty of the problem may well depend on the characteristics and limitations of the transducer- its bandwidth, resolution, sensitivity, distortion, signal-to-noise ratio, latency, etc.

# Pattern classification and Extraction

**Segmentation and Grouping**

- We assumed that each object can be isolated, separate from others and could easily be distinguished from the Others.
- In practice, the object would often be overlapping, and our system would have to determine where one ends and the next begins-the individual patterns have to be segmented.
- If we have already recognized the object then it would be easier to segment their images.
- **Segmentation** is one of the deepest problems in pattern recognition. Closely related to the problem of segmentation is the problem of recognizing or grouping together the various pans of a composite object.

# Pattern classification and Extraction

**Feature Extraction**

- The conceptual boundary between feature extraction and classification proper is somewhat arbitrary: An ideal feature extractor would yield a representation that makes the job of the classifier trivial; conversely, an omnipotent classifier would not need the help of a sophisticated feature extractor The distinction is forced upon us for practical rather than theoretical reasons.
- As with **segmentation**, the task of feature extraction is much more problem- a domain-dependent and thus requires knowledge of the domain

# Pattern classification and Extraction

**Classification**

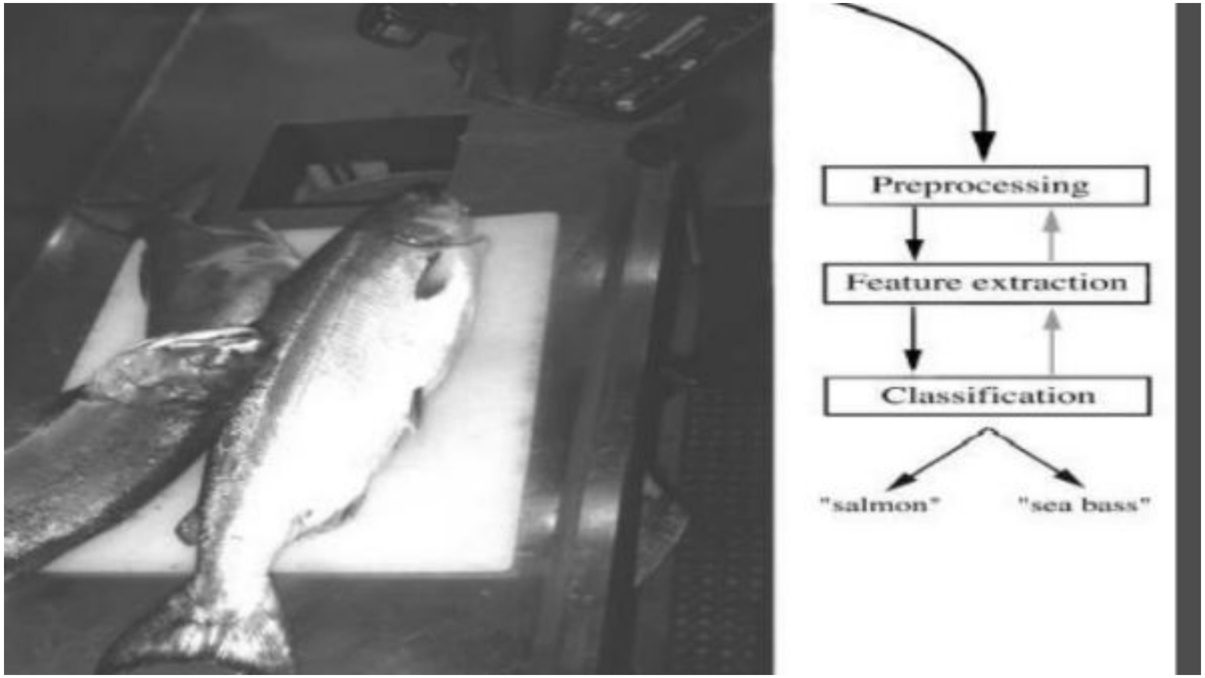
- The task of the classifier component proper of a full system is to use the feature vector provided by the feature extractor to assign the object to a category.
- Because perfect classification performance is often impossible, a more general task is to determine the probability for each of the possible categories.
- The abstraction provided by the feature-vector representation of the input data enables the development of a largely domain-independent theory of classification
- All non-trivial decision and pattern recognition problems involve noise in some form.

# Pattern classification and Extraction

**Post Processing**

- The post-processor uses the output of the classifier to decide on the recommended action.
- The simplest measure of classifier performance is the **classification error rate** -the percentage of new patterns that are assigned to the wrong category. Thus, it is common to seek **minimum-error-rate** classification.
- We could also do better if we used multiple classifiers, each classifier operating on different aspects of the input.

# example



Preprocessing
Feature extraction
Classification
"salmon"
"sea bass"

# Fingerprint

- *Fingerprint  Recognition and acquisition*
- *Fingerprint features, matching and synthesis*

# Fingerprint

- The fingerprint is a physical biometric aspect.
- It is used to identify a person’s identity due to its uniqueness where no two persons can share the same fingerprint.
- Besides, a fingerprint is unchangeable with time and can be easily recognized during the whole life of the individual.
- The fingerprint is an impression or model of ribs and valleys at the top of a person’s fingers.

# Fingerprint

- **Fingerprint recognition** is the automatic process of comparing saved fingerprint pattern with the input fingerprint to determine human characters.
- The fingerprint identification system is a cheap but solid mechanism at the same time. Moreover, it’s a simple way to identify humans speedily and accurately
- Many applications applied fingerprint recognition such as the military, judiciary, health, teaching, civic serving, mobiles and laptop log-in, and many more.

# Introduction

**Fingerprint**

- Interleaved ridges and valleys
- Ridge width: 100µm-300 µm
- Ridge-valley cycle: 500 µm

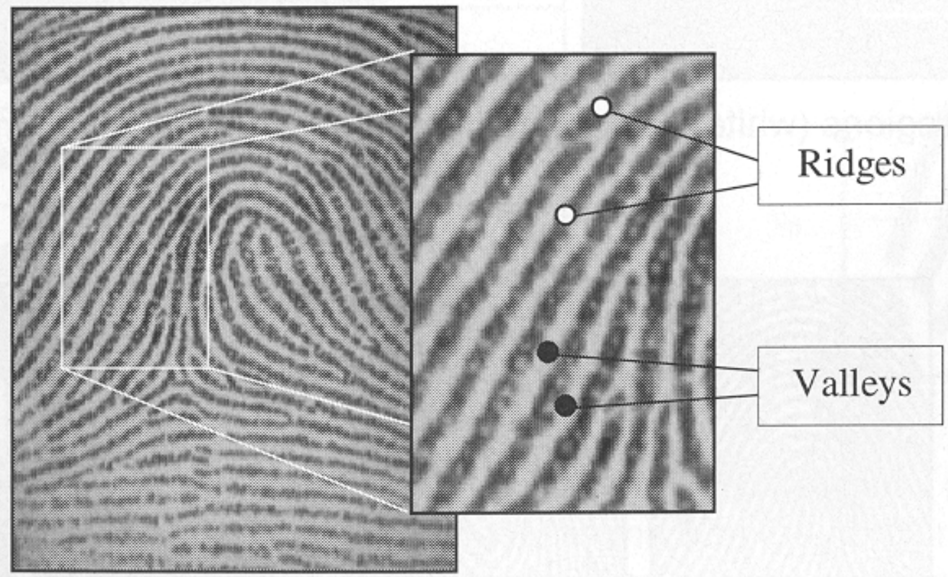
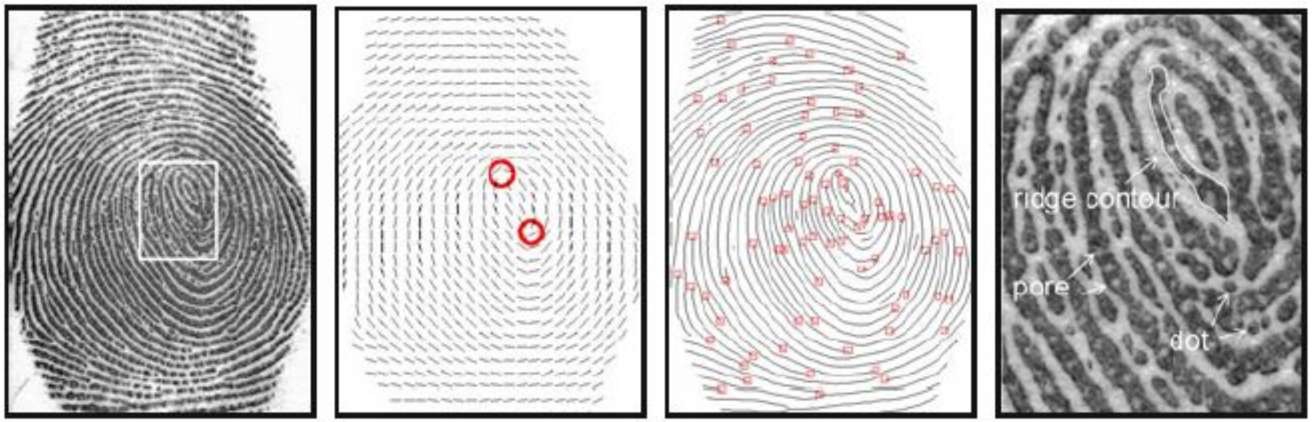



Figure 3.1. Ridges and valleys on a fingerprint image.

# Levels of features

- **Level 1** features(orientation field or ridge flow and singular points)
- **Level 2** feature(ridge skeleton)
- **Level 3** features(ridge contour,pore,dot)

# Three different levels of features




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

**Fig. 2.5** Features at three different levels in a fingerprint. (a) Grayscale image (NIST SD30, A067_11), (b) Level 1 feature (orientation field or ridge flow and singular points), (c) Level 2 feature (ridge skeleton), and (d) Level 3 features (ridge contour, pore, and dot).

# Three different levels of features

- At the global level a fingerprint shows a smoothed structure except in one or more regions containing distinctive characteristics called **singularities** which can be:
  - **delta** (represented with the symbol Δ);
  - **loop** (represented with the symbol ∩);
  - **whorl** (represented with the symbol O)

# Fingerprint acquisition

- The first step in fingerprint recognition is **image acquisition** - the process of capturing and digitizing the fingerprint of an individual for further processing.
- The primary reason for the popularity of fingerprint recognition is the availability of mature, convenient, and low-cost sensors that can rapidly acquire the fingerprint of an individual with minimum or no intervention from a human operator.
- These compact fingerprint sensors have also been embedded in many consumer devices such as laptops and mobile phones.

# Fingerprint acquisition

- **Finger Print Scanners**





# Fingerprint Acquisition: Sensing Techniques

- Digital images of the fingerprints can be acquired using **off-line** or **on-line** methods.

**<u>Off line Method</u>**:

- Off-line techniques generally do not produce the digital image directly from the fingertip. Rather, the fingerprint is first **transferred to a substrate** (e.g., paper) that is subsequently digitized.
- For example, an inked fingerprint image, the most common form of off-line capture, is acquired by first applying ink to the subject's fingertip and then rolling or pressing the finger on paper, thereby creating an impression of the fingerprint ridges on paper.
- This kind of fingerprint is often called **rolled fingerprint**.
- A very important kind of off-line fingerprint image is the **latent fingerprint**: a partial fingerprint image lifted from a crime scene by a forensic expert. Compared to a rolled fingerprint, the latent is most of the times of bad quality and hard to process

# Different fingerprint impressions

(a) (b) (c)

**Fig. 2.3** Three different fingerprint impressions of the same finger. (a) Rolled fingerprint, (b) plain fingerprint, and (c) latent fingerprint.

# On-line techniques

## Online:

- On-line techniques produce the digital image directly from a subject's fingertip via digital imaging technology that circumvents the need for obtaining an impression on a substrate. The resulting fingerprint image is referred to as a live-scan fingerprint.
- A typical fingerprint scanner comprises:
  - a sensor to read the ridge pattern on the finger surface;
  - an A/D (Analog to Digital) converter to convert the signal;
  - an interface module responsible for communicating with external devices.
- Almost of all existing sensors belong to one of the following families: optical, solid-state and ultrasound. These sensors are also called **touch sensors**.

# Different ways of acquiring

(a) (b) (c)

**Fig. 2.11** There are different ways to acquire and digitize the fingerprint of an individual. (a) A fingerprint can be first transferred to a paper substrate by manually placing an inked fingertip on paper, and then digitizing the resulting impression using a flatbed scanner, (b) a live-scan fingerprint can be directly imaged from a finger based on a number of advanced sensing technologies, (c) a latent fingerprint can be lifted from objects in a crime scene using chemical or electrical processes.

# Different sensors

**(a) Optical Frustrated Total Internal Reflection (FTIR)**

**(b) Capacitance**

**(c) Ultrasound Reflection**

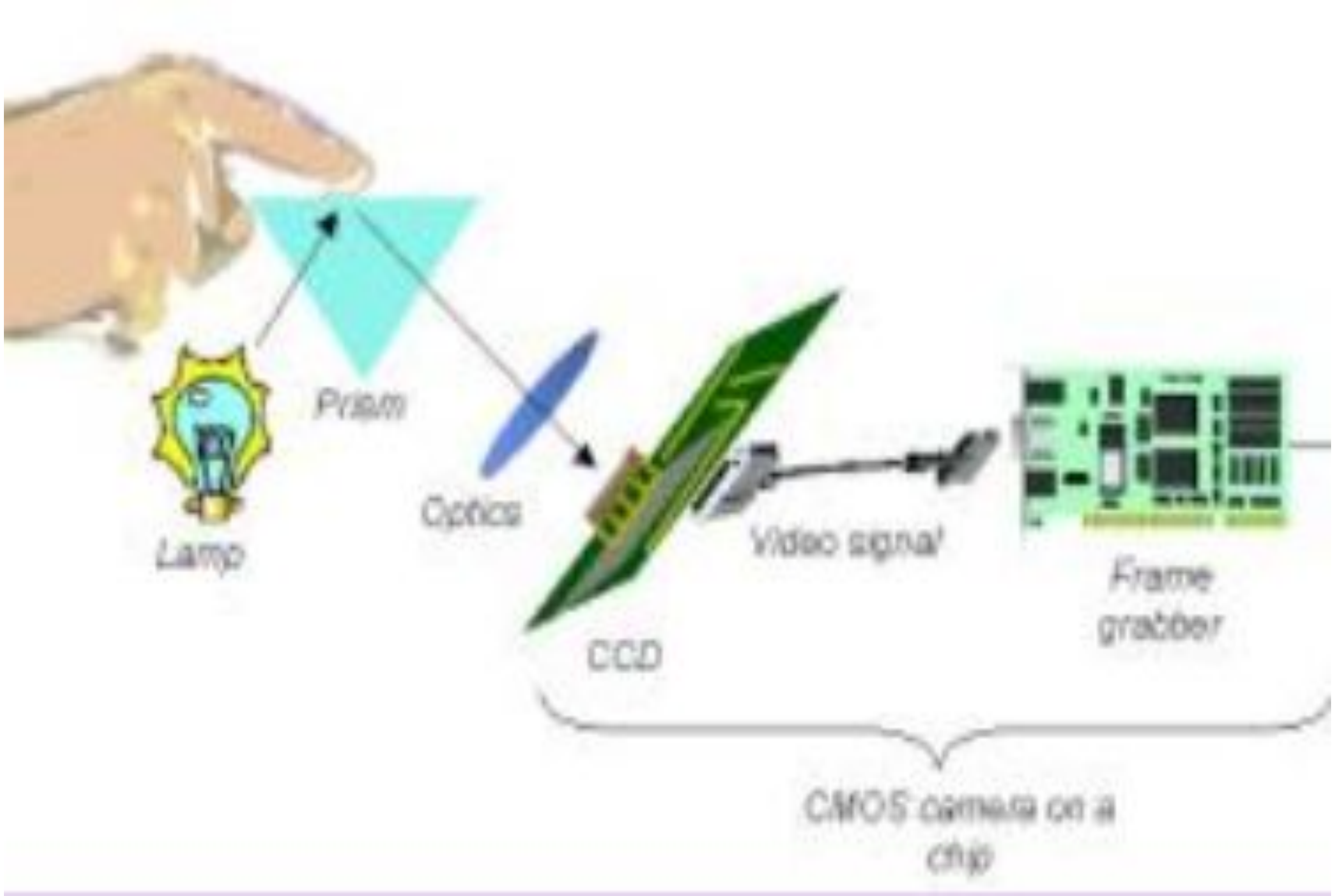
**(d) Piezoelectric Effect**

**(e) Temperature Differential**

# FTIR Based Fingerprint Sensing

- Optical frustrated total internal reflection
- This technique utilizes a glass platen LED light source and a CCD Camera for  constructing fingerprint images.
- When finger is placed on one side of a glass platen ( prism) only the ridges of the finger are in contact with the platen not the valleys.
- The light source illuminates the glass at a certain angle and the camera is placed such that it can capture the light reflected from the glass.

# FTIR Based Fingerprint Sensing



- The light incident on the ridges is randomly scattered ( results in dark image) while the light incident on the valleys suffers total internal reflection(results in bright image)
- Difficult to have this arrangement in compact form, since the focal length of small lenses can be very large.

# FTIR BASED SENSING

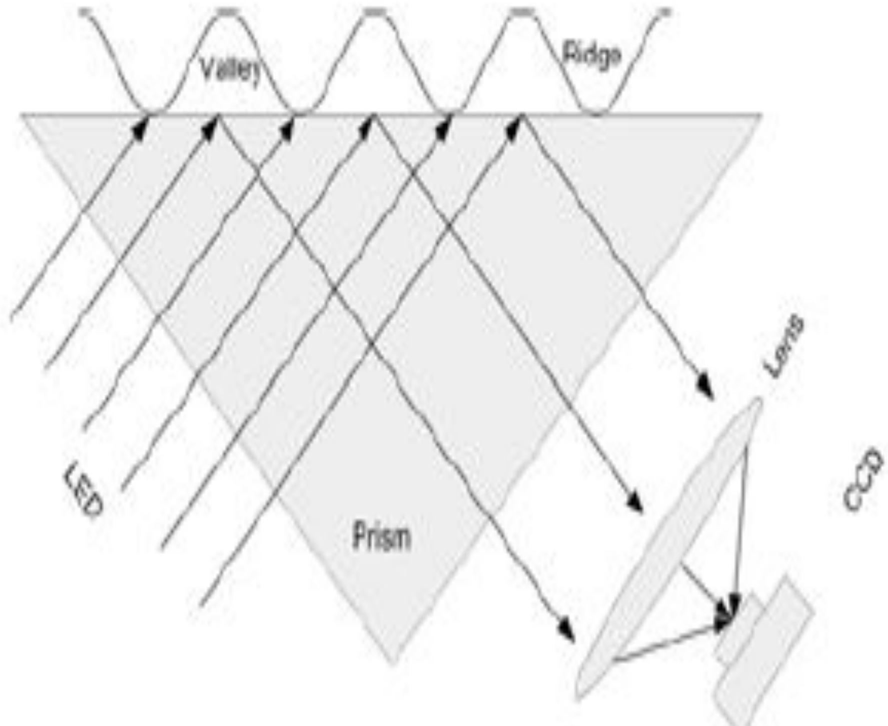
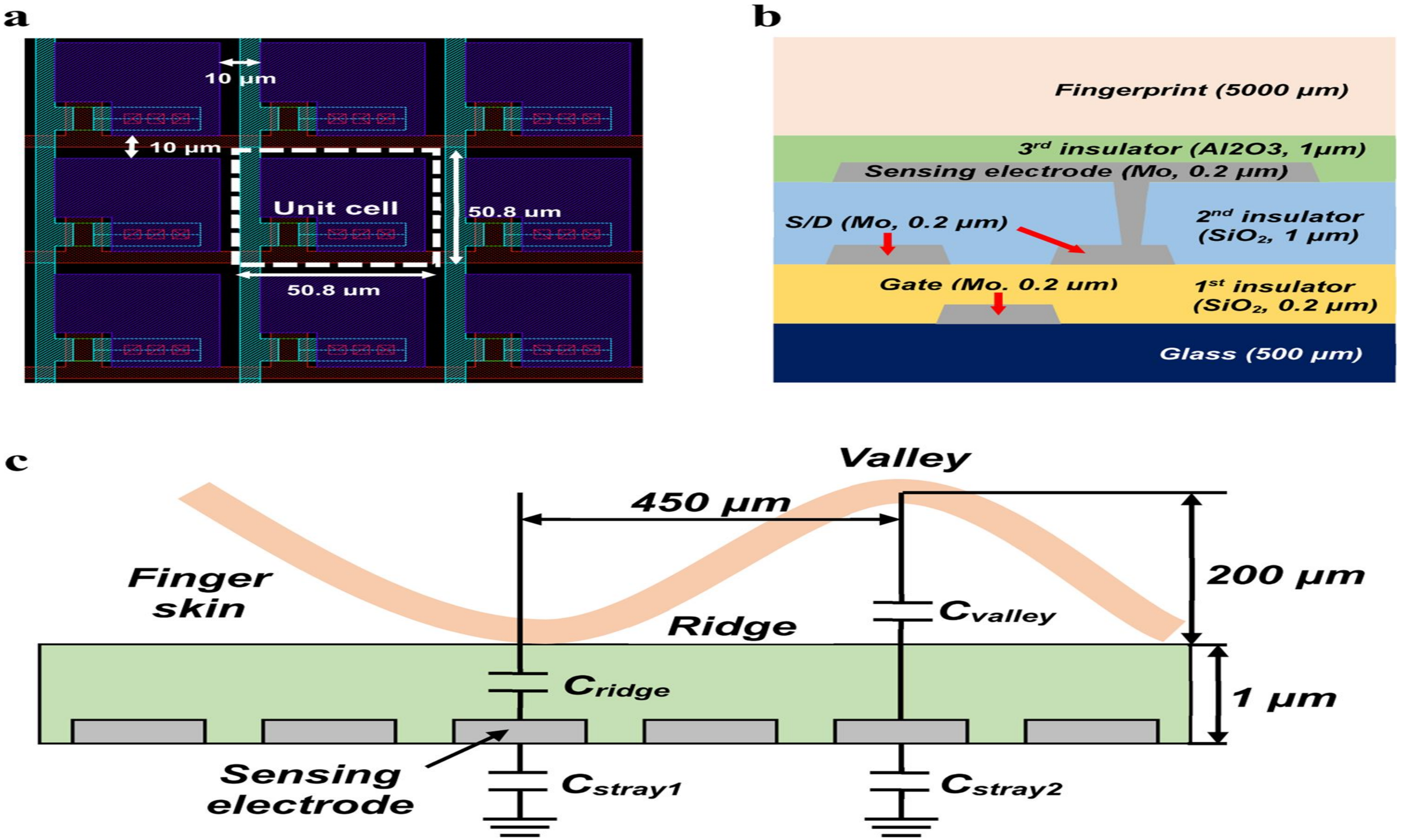



Fig. 2.12 FTIR-based fingerprint sensing.

# CAPCITANCE

- Capacitance based solid state live scan fingerprint sensors are more commonly used than the Optical FTIR sensors since they are very small in size and can be easily embedded into laptop ,mobile phones and computer peripherals.
- Essentially consists of an array of electrodes.
- There are tens of thousands of small capacitance plates(electrodes ) embedded in chip.
- Fingerprint skin acts as the other electrode, thereby forming a  miniature capacitor

# CAPCITANCE



a
10 µm
10 µm
Unit cell
50.8 um
50.8 um
b
Fingerprint (5000 µm)
3rd insulator (Al2O3, 1µm)
Sensing electrode (Mo, 0.2 µm)
S/D (Mo, 0.2 µm)
2nd insulator (SiO2, 1 µm)
Gate (Mo, 0.2 um)
1st insulator (SiO2, 0.2 µm)
Glass (500 µm)
c
Valley
450 µm
200 µm
Finger skin
Ridge
Cvalley
Cridge
1 µm
Sensing electrode
Cstray1
Cstray2

# Ultrasound reflection

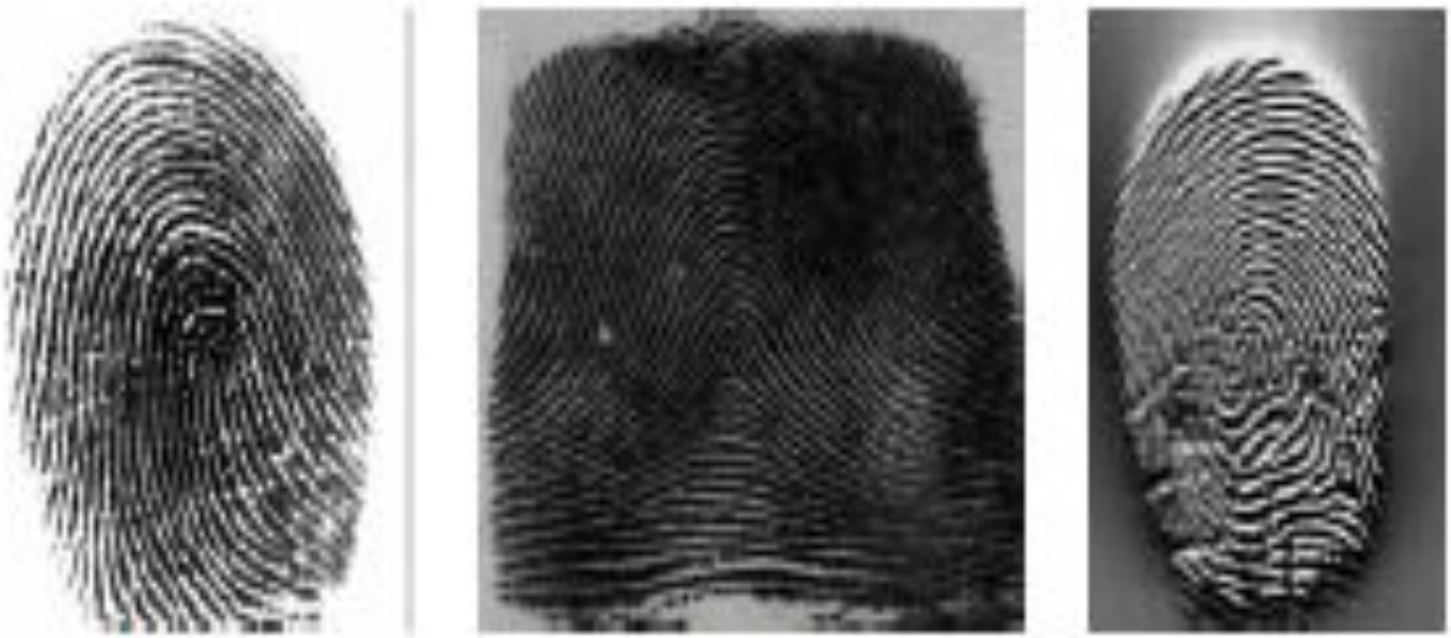
- It is based on sending acoustic signals toward the fingertip and capturing the echo signal.
- The sensors has two main components the sender that generates short acoustic pulses,and the receiver that detects the responses obtained when these pulses bounce off the fingerprint surface.
- Resilent to dirt and oil accumulations that may visually mar the fingerprint
- Expensive not suited for large scale production

# Temperature differential

- This mechanism is made up of pyro electric material that generates a current based on temperature differentials.
- Temperature differential is created when two surfaces are brought into contact.
- Fingerprint ridges,being in contact with the sensor surface, produce an different temperature differential than valleys that are away from the sensor surface.

# Fingerprint scanner operation methods

- **Plain fingerprint** is obtained by simply placing the finger on the surface of fingerprint sensor.
- A **rolled fingerprint** is obtained by rolling the finger from nail to nail on the surface of fingerprint sensor
- **Sweep fingerprint** is obtained by combining narrow fingerprint slices .( typically 3mm wide) while the user swipes his finger across the sensor
- Clarity of ridge pattern is another important determinant of quality.



(a) (b) (c)

**Fig. 2.13** Fingerprint scanner operation methods: (a) A plain fingerprint is obtained by simply placing the finger on the surface of fingerprint sensor; (b) a rolled fingerprint is obtained by rolling the finger from "nail to nail" on the surface of fingerprint sensor (this typically requires someone to hold the finger to assist in proper rolling); (c) a sweep fingerprint is obtained by combining narrow fingerprint slices (typically 3mm wide) while the user swipes his finger across the sensor.

# Feature extraction

- **Commercial** fingerprint recognition systems are mainly based on **Level 1** (ridge orientation and frequency) and **Level 2** features (ridges and minutiae).
- Generally, Level 1 features are first extracted and then Level 2 features are extracted with the guidance of Level 1 features.

# Feature extraction

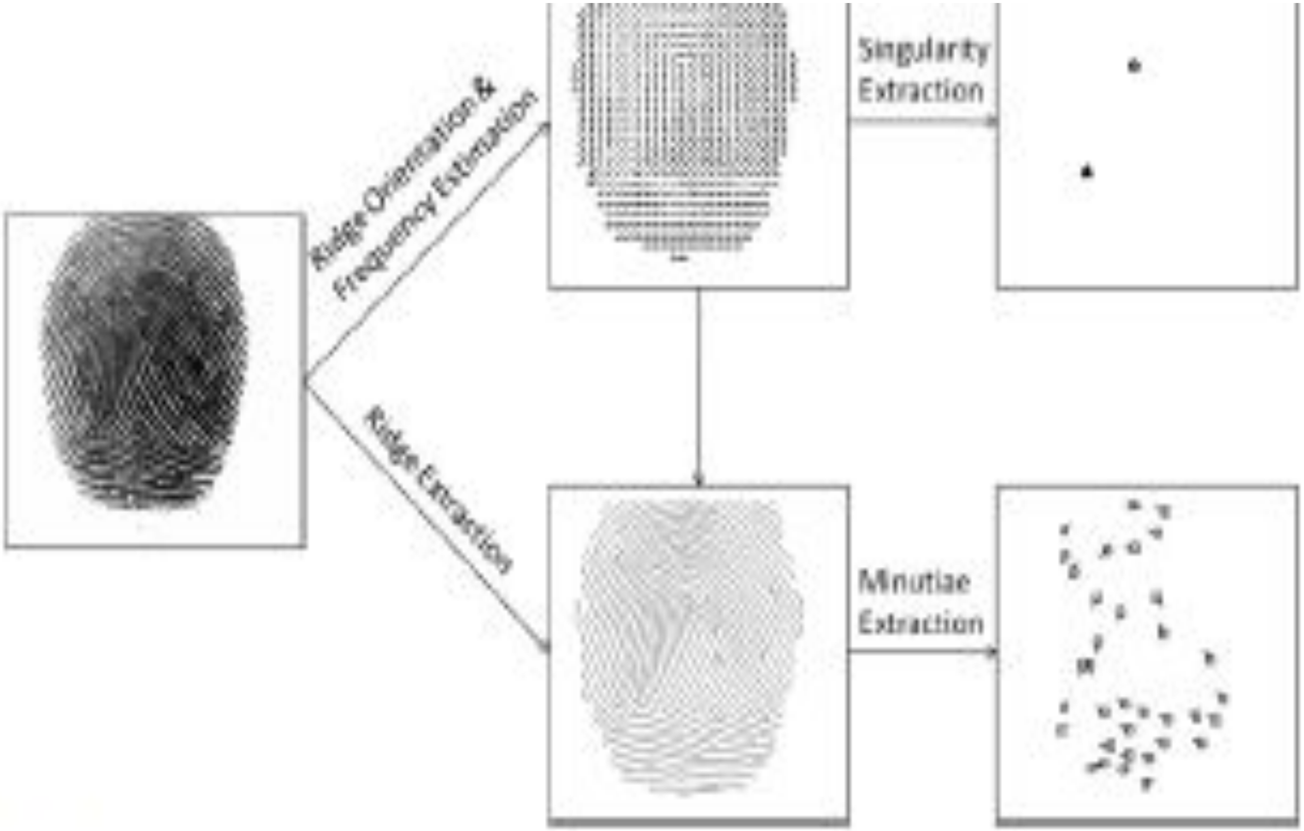



**Fig. 2.15** Schematic diagram for the extraction of level 1 and level 2 features from a fingerprint image.

# Steps

Typical **feature extraction** algorithm which includes four main steps, namely

(a) ridge orientation and frequency estimation
(b) ridge extraction
(c) singularity extraction
(d) Binarization and Thinning

# 1. Ridge orientation and frequency Estimation

Ridge pattern in a local area of a finger can be approximated by a cosine wave

$$w(x,y) = A\cos(2\pi f_0(x\cos\theta + y\sin\theta)), \tag{2.1}$$

where $A$, $f_0$, and $\theta$ denote the amplitude, frequency, and orientation of the cosine wave. Then, the 2D Fourier transform of the cosine wave is given by

# 2. Singularity Extraction

- Fingerprint singularity can be extracted from the orientation field using the well known ***Poincare* index method**.
- **Poincare index** refers to the **cumulative change of orientations** along a **closed path** in an orientation field.
- To accurately detect the location and type of singularity, Poincare index is generally evaluated using the eight neighbors of a pixel.

# Types of Finger Prints

Poincare index of a pixel corresponding to a singular point can take one of four possible values:

0 (**non-singular**),

1 (**loop**),

-1 (**delta**), and

2 (**whorl**).

The non-singular, loop, delta and whorl are types of finer prints

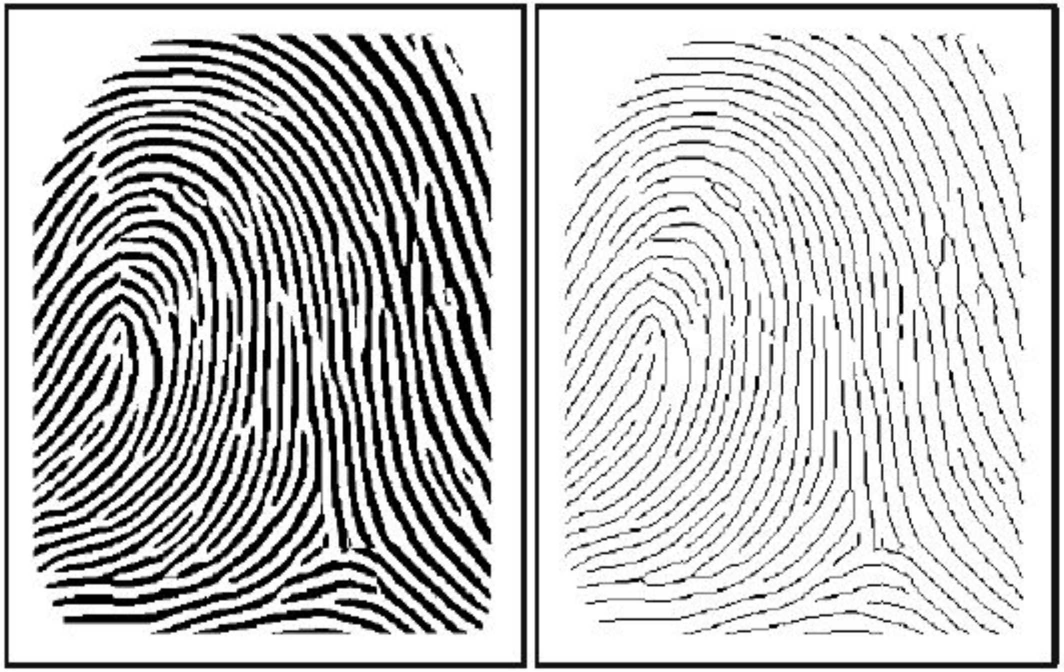
# 3. Ridge Extraction

- Since **minutiae** are **special points** on ridges, it is natural to first **extract ridges** and then detect the minutiae on ridges.
- Since **ridges** are **darke**r than **valleys**, a straightforward method to detect ridges is to **classify** any pixel as a ridge pixel if its **gray value** is **lower** than a **threshold** (for example, the **mean** of its local neighborhood).
- However, for most fingerprint images, this method does not work well for the following reasons: (a) **pores** on ridges are **brighter** than the surrounding pixels; (b) ridges can be **broken** due to cuts or creases; (c) adjacent ridges may appear to be **joined** due to moist skin or pressure.

# Binarization and Thinning

- The enhanced image can be converted to a **binary image** by using either a global threshold (for example, using the mean pixel value of the enhanced image) or thresholds that are **locally** computed.
- A morphological operation, called **thinning**, is used to **reduce** the ridge **width** to **one pixel**.
- Thinning is a common technique in image processing, which involves iteratively removing outer ridge pixels

# Binarization and thinning



(a) (b)

**Fig. 2.23** Binarization and thinning results of the fingerprint image in Figure 2.22. (a) Binarized ridge image and (b) thinned ridge image.

# Three steps of matching

1. Alignment: Determine the geometric transformation between the two minutiae sets so that they are in the same coordinate system.
2. Correspondence: Form pairs of corresponding minutiae.
3. Score generation: Compute the match score based on the corresponding minutiae points.

# Minutiae matching algorithm

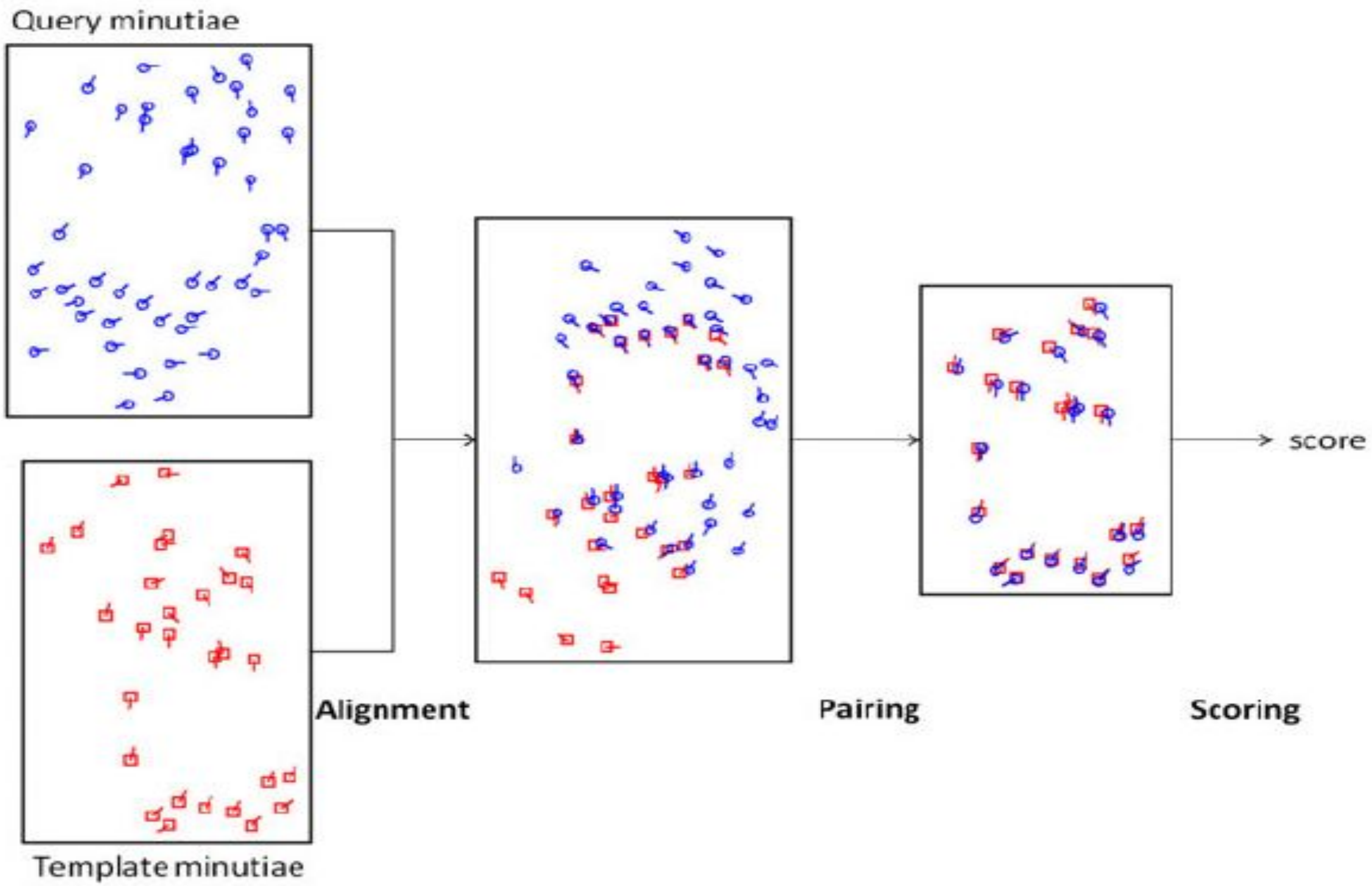



Fig. 2.27 Flowchart of a minutiae matching algorithm.

# Alignment

- Since **two impressions** of the same finger taken at **different instances** could **differ** due to different **placement** of the finger on the sensor, an **alignment process** is required to transform them to the **same coordinate** system.
- This process, also known as **registration**, transforms one image in such a way that it is geometrically aligned with the other.
- First, we need to specify a spatial transformation model

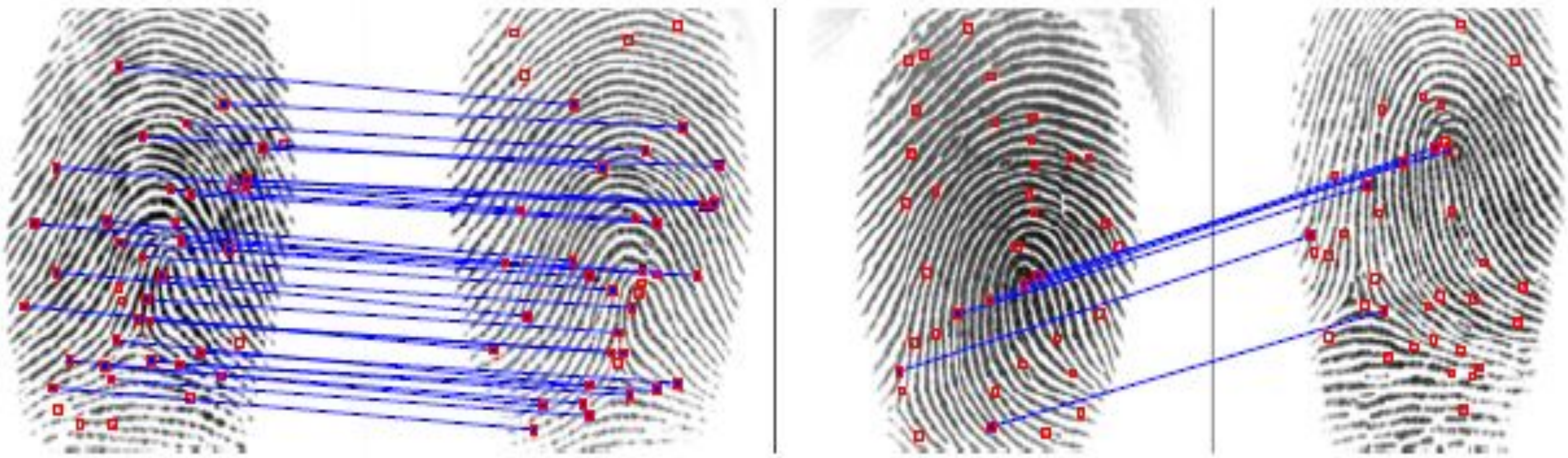
# PAIRING

- After the two minutiae sets are aligned, the **corresponding** minutiae are paired.
- A minutia *a* in the template (reference) minutiae set is said to be in correspondence with minutia *b* in the query minutiae set if and only if their **distance** is within a **predefined** distance threshold (say, 15 pixels) and the **angle** between their directions is within another predefined angle threshold (say, 20 degrees)

# SCORE

- We need to compute a **match score**, which is then compared to a **predefined threshold** to classify the two fingerprints as a **genuine** match (they come from the same finger) or an impostor match (they come from two different fingers).
- This problem can be viewed as a two-class **classification** problem with genuine match as class-1 and impostor match as class-2.

# MATCH SCORE



(a) Match score = 614 (b) Match score = 7

'ig. 2.30 Fingerprint matching by a commercial matcher. (a) A genuine pair of fingerprints with 1 matched minutiae, and (b) an imposter pair with 6 matched minutiae. Corresponding minutiae etween the two images are connected by lines. The match score is computed as some function of he number of matched minutiae and some other parameters that are proprietary to the commercial natcher.

# Palm Print Recognition System

## Features of Palm Print

- Friction ridge patterns on the palm of the hand and the sole of the foot have also been claimed to be **unique** and permanent and thus can be used for personal identification.
- Palm prints and Sole prints have **fewer applications** than fingerprints since it is not as **convenient to capture** them compared to fingerprints.
- In fact, sole prints are only used to register **newborns** in a hospital since it is easier to capture the sole print of newborns than fingerprints or palmprints (newborns tend to keep their fists closed!).

# PALMPRINT

- Forensics application is concerned, the main benefit of using palmprints is in **latent palmprint** matching since it is estimated that about 30% of the latent found at crime scenes are those of palms.

# Palm print features

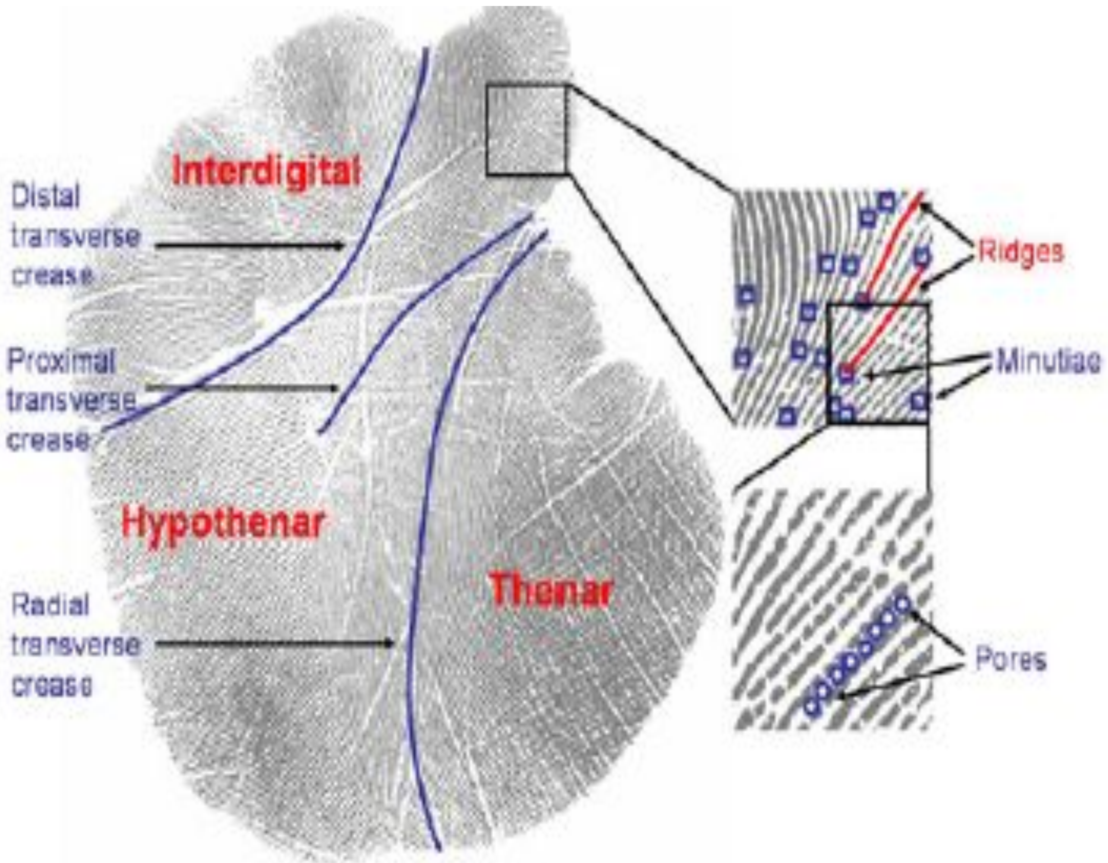



Fig. 2.39 Regions (interdigital, thenar and hypothenar), major creases (distal transverse crease, proximal transverse crease and radial transverse crease), ridges, minutiae and pores in a palmprint.

# Challenges in Palm Print Recognition

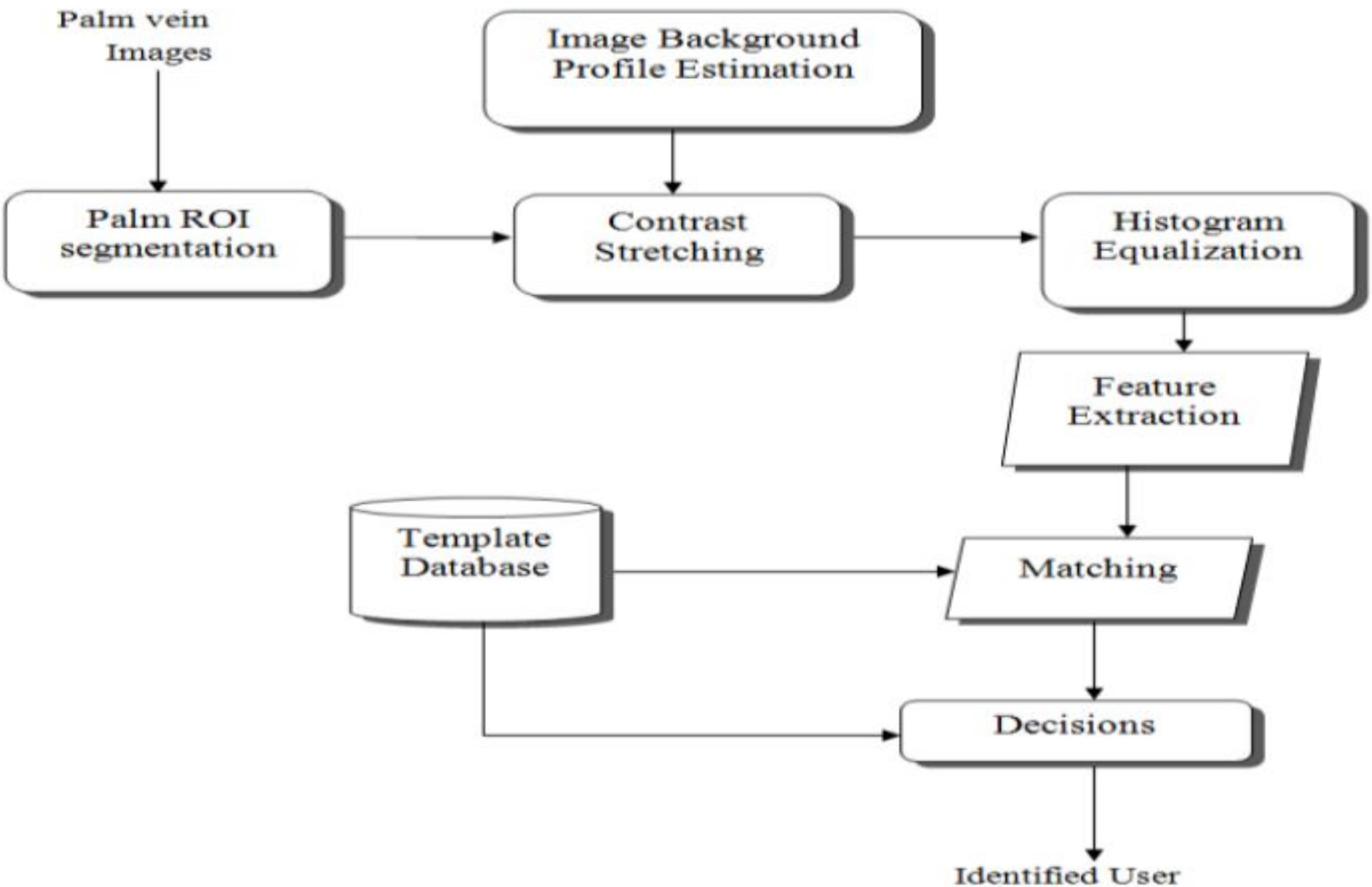
- Matching palmprints is more **complicated** than matching fingerprints.
- First of all, in terms of image acquisition, it is more **challenging** to capture palmprints than fingerprints since the sensors are **larger** and more **expensive** and greater **user cooperation** is required to ensure a good quality palmprint image due to the **concavity** of the palm surface.

# Friction ridge patterns

(a) (b)

Fig. 2.38 Friction ridge patterns on palm and sole [11].

# Palm Print Recognition Framework



# Palm Print Recognition

- The palm vein images employed in identification using palm vein obtained under near-infrared illumination (NIR) and the images normally emerge darker with low contrast.
- So, image development more evidently demonstrates the vein and texture patterns.
- First, approximate the condition intensity profiles by segmenting the image into a little overlapping. In 32 X 32 blocks, three pixels are overlapped between two blocks to address the blocky effect.
- The average gray-level pixels in each block are forecasted.

# Palm Print Recognition

The main modules of a palm-print verification system preforms

a) Palm-print senses the palm print of a human obtained by a palm print scanner.
b) Pre-processing in which the input palm print is improved and adapted to shorten the process of feature extraction;
c) Feature extraction in which the palm print is further processed to produce discriminative features also called feature vectors
d) Matching in which the feature vector of the input palm print compared against with one or more existing templates.

# Face recognition

*Face recognition and acquisition*

*Face detection, feature extraction and matching*

(a) Twin[1]

(b) Family[2]

# Face recognition

- Face recognition can be defined as the process of establishing a person’s identity based on their facial characteristics
- The problem of face recognition involves comparing two face images and determining if they are of the same person
- Face images of a person may have variations in age, pose, illumination, and facial expressions exhibit changes in appearance due to make-up, facial hair, or accessories (e.g., sunglasses, hat etc)

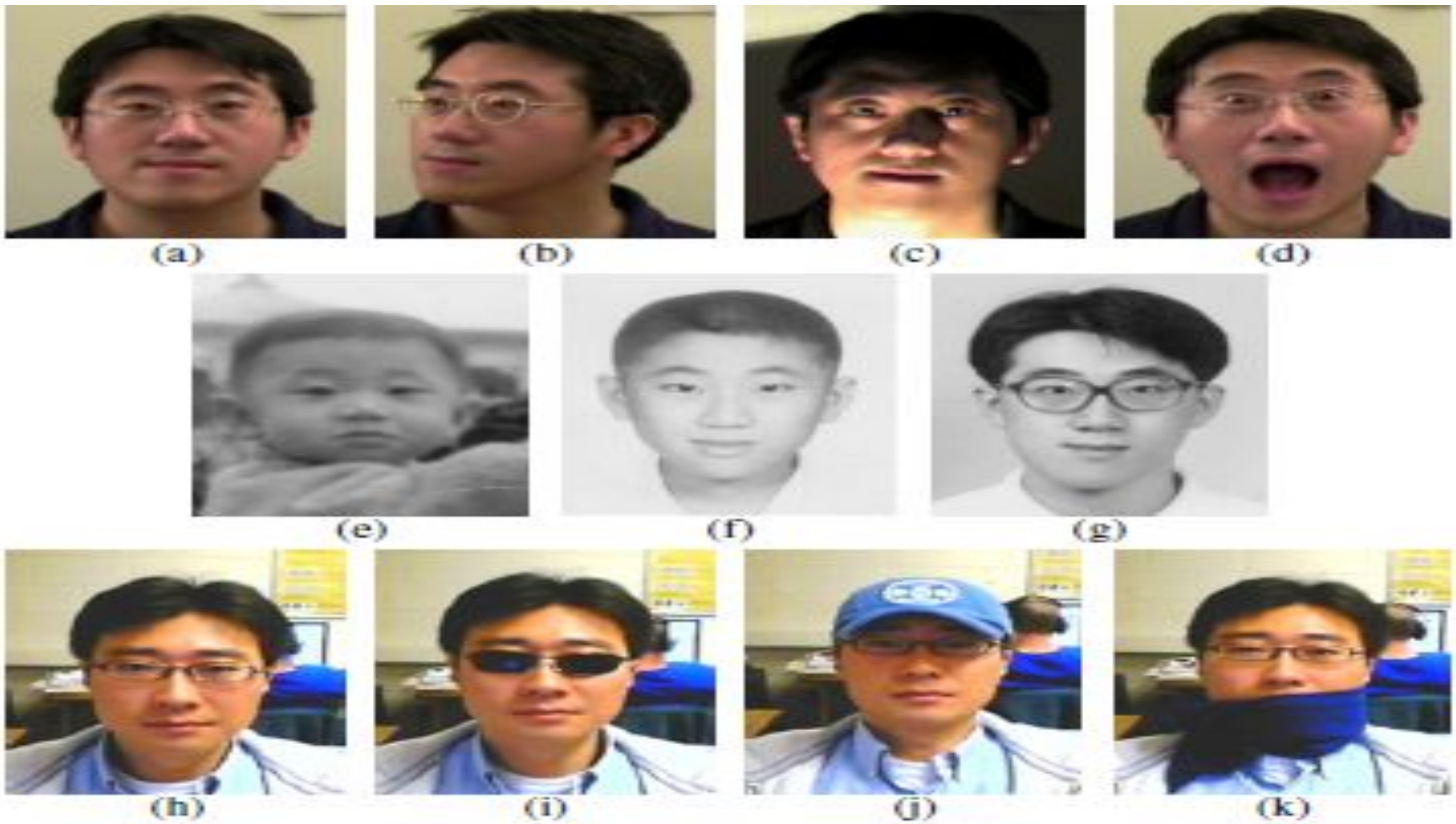



**Fig. 3.1** The problem of intra-class (i.e., intra-user) variations is quite pronounced in the context of face recognition. The face image of an individual can exhibit a wide variety of changes that make automated face recognition a challenging task. For example, the face images in (b), (c), and (d) differ from the frontal face image of the person in (a) in terms of pose, illumination, and expression, respectively. The second row shows the variability introduced due to aging. Here, the images in (e), (f), and (g) were acquired when the person in (a) was 32, 21, and 15 years younger, respectively. The third row depicts the problem of occlusion of some facial features due to the person wearing accessories such as (h) prescription glasses, (i) sunglasses, (j) cap, and (k) scarf.

# ADVANTAGES

- Face can be captured at a longer standoff distance using non-contact sensors.
- The face conveys not only the identity, but also the emotions of a person (e.g., happiness or anger) as well as biographic information(e.g., , gender, ethnicity, and age).
- Face recognition has a wide range of applications in law enforcement, civilian identification, surveillance systems, and entertainment/ amusement systems

# Level 1 details

- Consists of gross facial characteristics that are easily observable.
- Examples include the **general geometry of the face** and global **skin color.** Such Features can be used to quickly discriminate between
- (a) a short round face and an elongated thin face;
- (b) faces exhibiting predominantly male and female characteristics;
- (c) faces from different races.
- These features can be extracted even from low resolution face images (*< 30 interpupilary distance (IPD)1).*

# Level 2

- **Level 2 details** consist of localized face information such as the structure of the face components (e.g., eyes, nose, mouth etc), the relationship between facial components and the precise shape of the face.
- These features are essential for accurate face recognition, and they require a higher resolution face image (30 to 75 IPD).
- The characteristics of local regions of the face can be represented using geometric or texture descriptors.

# Level 3

- Level 3 details consist of unstructured**, micro level features** on the face, which includes **scars, freckles, skin discoloration, and moles**.
- One challenging face recognition problem where Level 3 details may be critical is the discrimination of identical twins

# All three levels of facial features

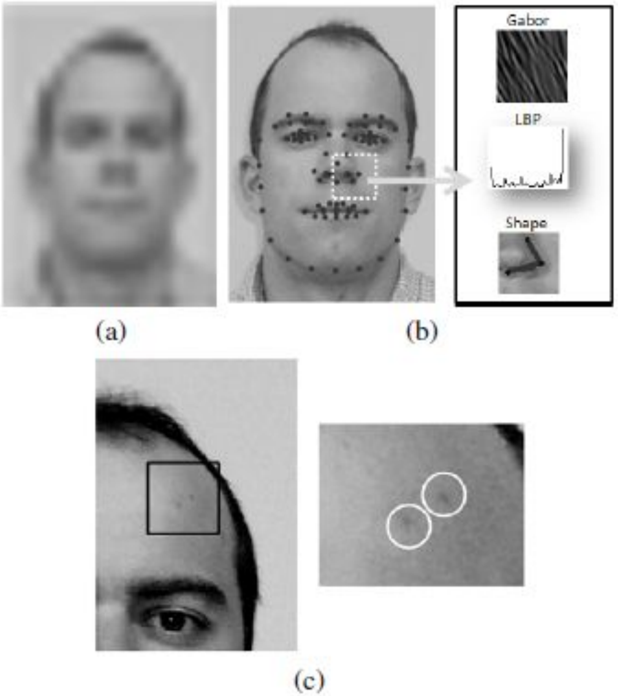



**Fig. 3.5** Examples of the three levels of facial features. (a) Level 1 features contain appearance information that can be useful for determining ethnicity, gender, and the general shape of a face. (b) Level 2 features require detailed processing for face recognition. Information regarding the structure and the specific shape and texture of local regions in a face is used to make an accurate determination of the subject's identity. (c) Level 3 features include marks, moles, scars, and other irregular micro features of the face. This information is useful to resolve ambiguities when distinguishing identical twins, or to assist in forensic investigation scenarios.

# Design of a face recognition system

A typical face recognition system is composed of three modules:

(a) image acquisition,

(b) face detection, and

(c) face matching

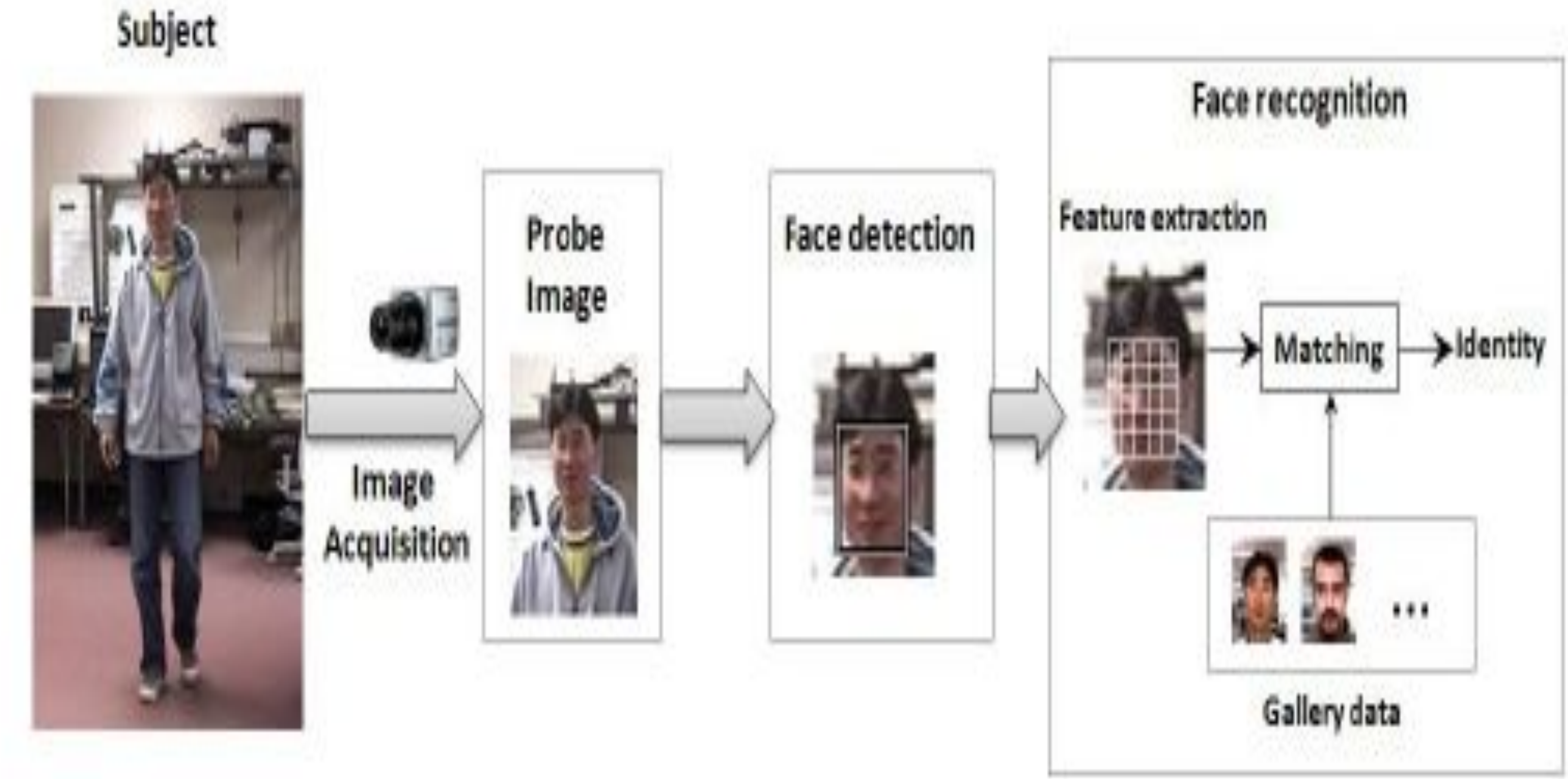



Fig. 3.6 Schematic of the face recognition process.

# Image acquisition

Face image acquired from a sensor can be categorized based on

a) the spectral band (e.g., visible, infrared, and thermal) used to record the image

b) the nature of the image rendering technique (e.g., 2D, 3D, and video).

- Since most of the automated face recognition systems make use of 2D images acquired in the visible spectrum

# Face matching

- Face matching is usually carried out by comparing the features extracted from the probe and gallery images

# FACE DETECTION

- Face detection is the first step in most face-related applications including face recognition, facial expression analysis, gender/ethnicity/age classification, and face modeling.
- There are a number of approaches for detecting faces in a given image
- State-of-the-art face detection methods are typically based on extracting local texture features from the given image and applying a binary (two-class) classifier to distinguish between a face and non-face

# Face Detection:

***Viola-Jones face detector:*** The Viola-Jones face detector scans through the input image with detection windows of different sizes and decides whether each window contains a face or not.

- Viola-Jones face detector is not perfect and can produce both false positive and false negative errors
- A false positive error refers to the detection of a face where none exists, while a false negative error indicates that a face present in the image was not detected.

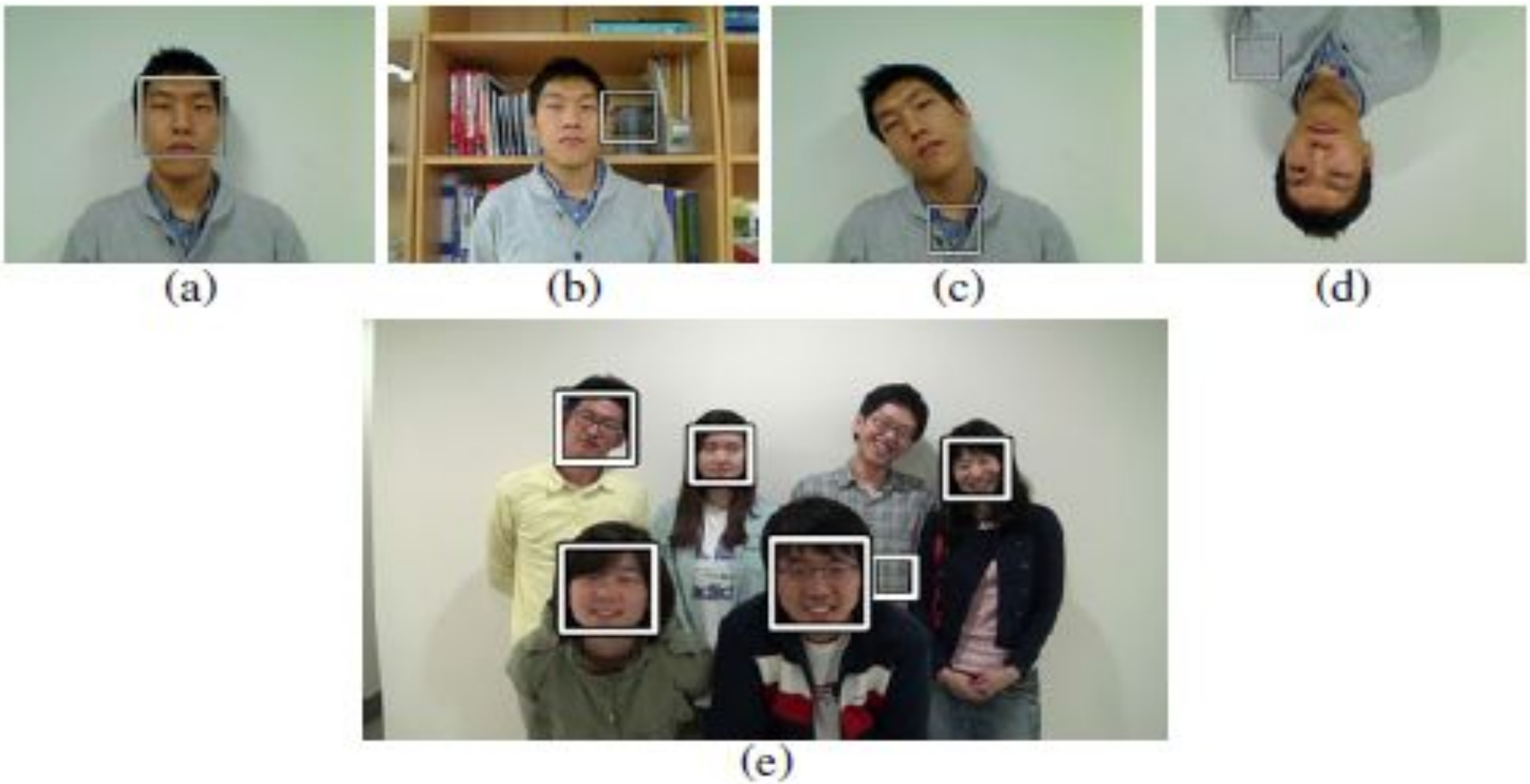


Fig. 3.13 The problem of face detection involves detecting a face in an image. Face detection algorithms have to be robust to variations in illumination, background, rotation, and image resolution. In this figure, the output of the Viola-Jones face detection algorithm, as implemented in the Open Computer Vision Library (OpenCV), is shown for different scenarios: (a) simple background, (b) cluttered background, (c) tilted face, (d) inverted face, and (e) multiple faces. Figures (b) through (e) have both false negatives (faces that are not detected) and false positives (non-face regions are wrongly categorized as faces).

# Viola-Jones face detector

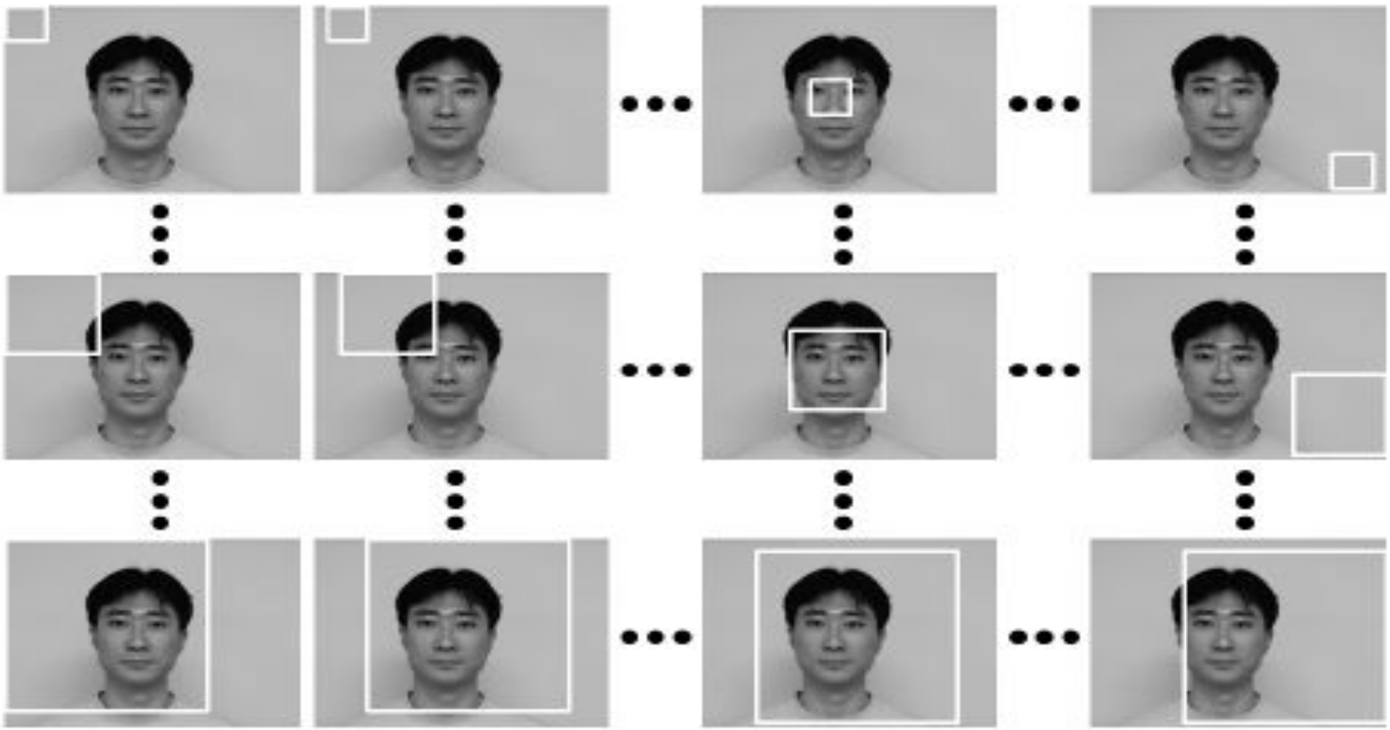


Fig. 3.14 The Viola-Jones method uses windows of different sizes to "scan" the entire image in order to determine the location of the face. In the above example, the window in the third column of the second row is likely to be detected as a face.

# Viola Jones Face Detector

- The Viola-Jones face detector scans through the input image with detection windows of different sizes and decides whether each window contains a face or not.
- Figure shows the scanning process ranging from small to large windows.
- In each window, the existence of a face candidate is decided by applying a classifier to simple local features derived using rectangular filters

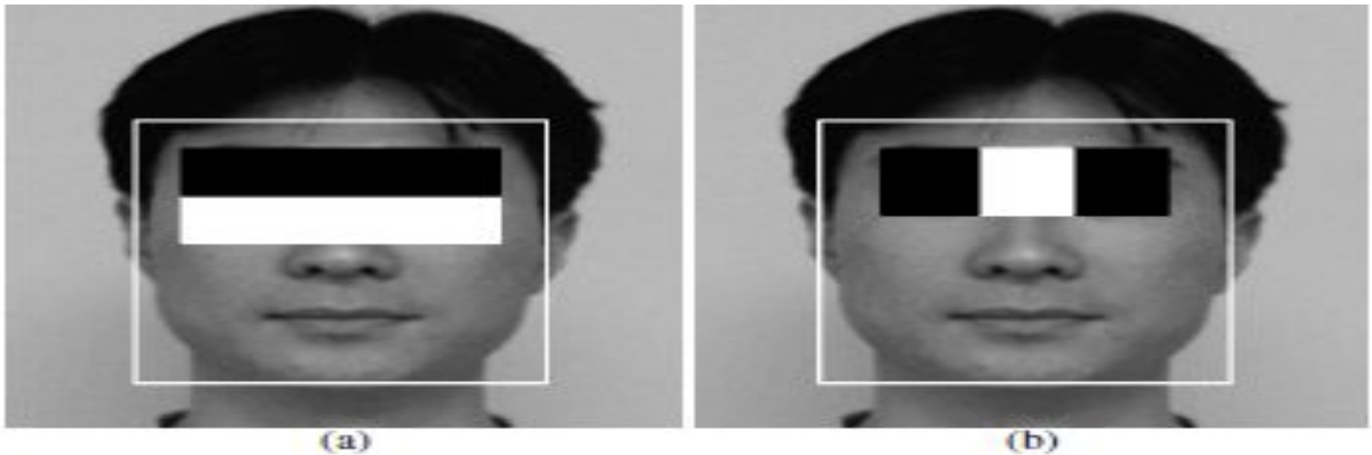



**Fig. 3.17** Two most discriminative Haar-like features overlaid on an input image.

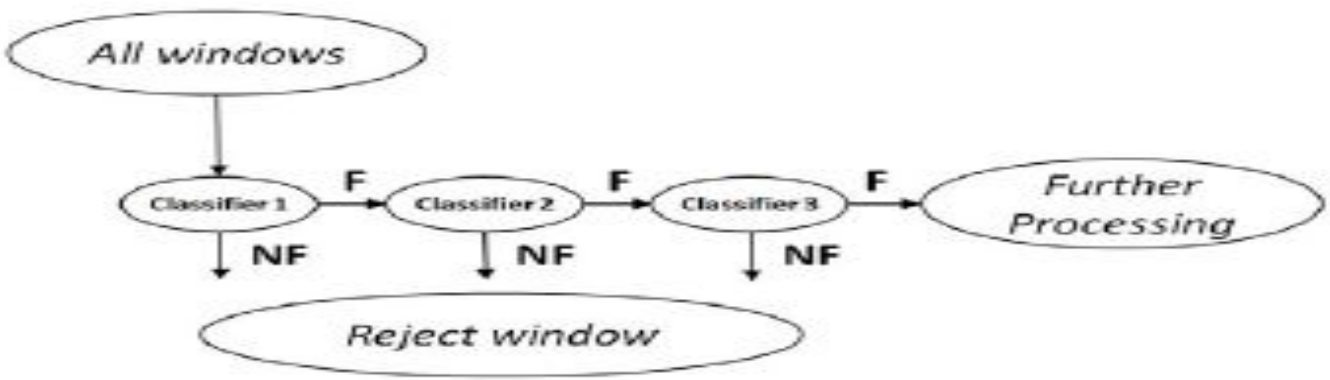



**Fig. 3.18** Schematic of a cascaded classifier to speed-up the face detection process. The initial classifier uses only a few features and eliminates a large number of non-faces with minimal processing. Subsequent classifiers consider increasingly more features and further eliminate the remaining non-faces at the cost of additional processing. Here, F indicates that a classifier decides that the tested window contains a face candidate, while NF represents the classifier decision that there is no face candidate within the tested window.

# Feature Extraction and Matching

There are three main approaches to match the detected face images.

1. Appearance-based,
2. Model-based, and
3. Texture-based methods

# Appearance based

- ***Appearance-based techniques*** *generate a compact representation of the entire* face region in the acquired image by mapping the high-dimensional face image into a lower dimensional sub-space
- This sub-space is defined by a set of representative basis vectors, which are learned using a training set of images the mapping can be either linear or non-linear, commonly used schemes such as
- **Principal Component Analysis** (PCA),
- **Linear Discriminant Analysis** (LDA), and
- **Independent Component Analysis** (ICA)
- involve linear projections

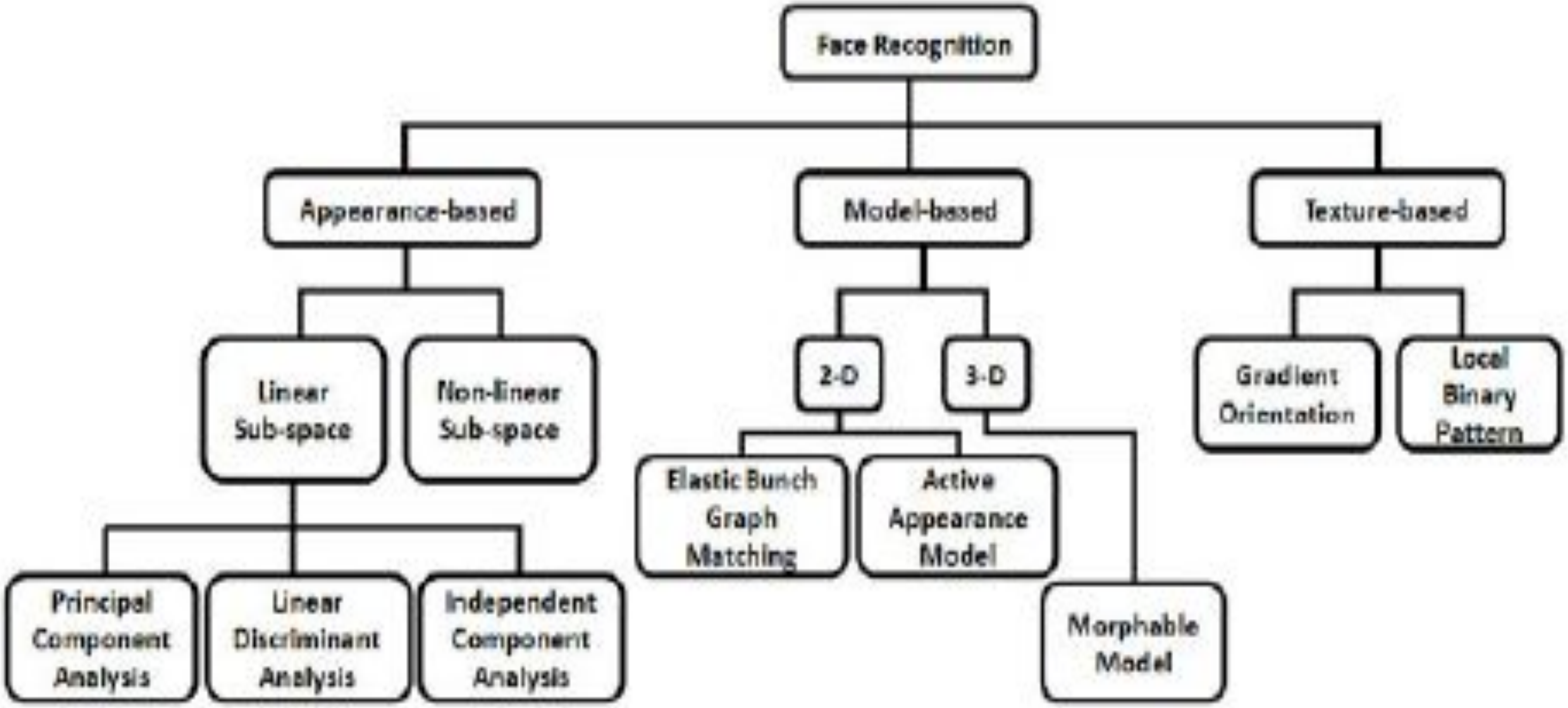
# Model based techniques

- *Model-based techniques* attempt to build 2D or 3D face models that facilitate matching of face images in the presence of pose variations.
- While the Face Bunch Graphs (FBG) and Active Appearance Model (AAM) are examples of 2D face models, the morphable model is a 3D model.

# Texture-based methods

- *Texture-based approaches* try to find robust local features that are invariant to pose or lighting variations.
- Examples of such features include gradient orientations and Local Binary Patterns (LBP).

# CATEGORIZATION OF FACE RECOGNITION TECHNIQUES



# Appearance-based face recognition

- **Principal Component Analysis**
- **Linear Discriminant Analysis**

- **Principal Component Analysis:**
- Principal Component Analysis (PCA) is one of the earliest automated methods proposed for face recognition. PCA uses the training data to learn a subspace that accounts for as much variability in the training data as possible. This is achieved by performing an Eigen value decomposition of the covariance matrix of the data.

- PCA can be referred to as an unsupervised learning method, because the class label (user identity information) is never used during the learning of the basis faces.
- Hence, the face recognition accuracy based on PCA cannot be expected to be very High

# Linear Discriminant Analysis

- Linear Discriminant Analysis (LDA) explicitly uses the class label of the training data and conducts subspace analysis with the objective of minimizing intra-class variations and maximizing inter-class variations

LDA

- can generally be expected to provide more accurate face recognition when sufficient face image samples for each user are available during training.

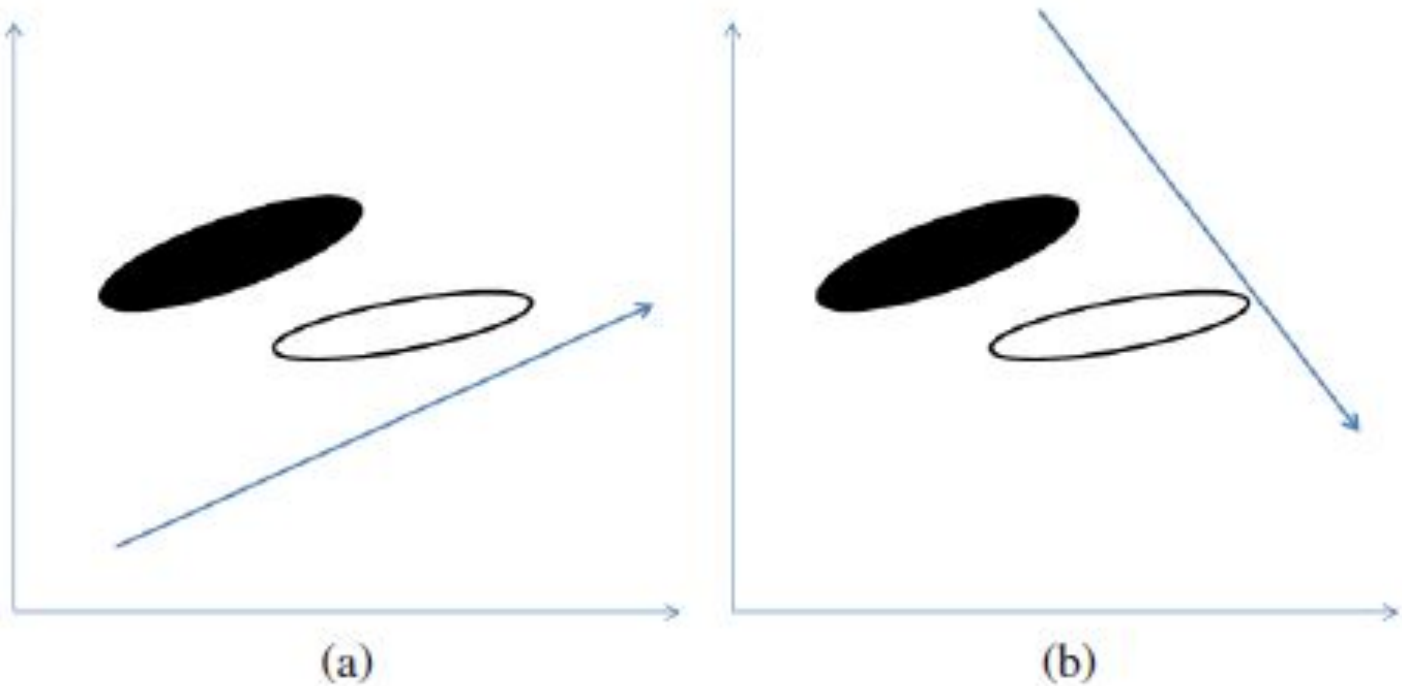



**Fig. 3.23** Comparison of PCA and LDA for a two-class problem with two-dimensional data. Here, the data corresponding to the two classes are assumed to exist within two ellipses. (a) The principal axis in PCA is aligned such that when the data is projected onto this axis, the variance is maximized. (b) The principal axis in LDA is aligned such that when the data is projected onto this axis, the variance within each class is minimized and the separability between the two classes is maximized.

# Model-based face recognition

- Model-based techniques try to derive a pose-independent representation of the face images that can enable matching of face image across different poses.
- These schemes typically require the detection of several fiducial or landmark points in the face (e.g., corners of eyes, tip of the nose, corners of the mouth, homogeneous regions of the face, and the chin)

# Model-based face recognition

**Elastic Bunch Graph Matching:** The Elastic Bunch Graph Matching (EBGM) scheme represents a face as a labeled graph with

> **Each node** of the graph is labeled with a set of **Gabor coefficients** (also called a jet) that characterizes **the local texture information around the landmark point**, and

> The **edges** of the graph is labeled based on the **distance** between the **corresponding fiducial points**

> The **Gabor coefficient** at a location in the image can be obtained by convolving the image with a complex 2D Gabor filter centered at that location.

> By varying the orientation and frequency of the Gabor filter, a set of coefficients or a Gabor jet can be obtained

A **Face Bunch Graph** (FBG) model can be constructed in two stages from a **training set** of face images with a specific pose

- In the first stage, the designer has to **manually mark** the desired **fiducial points** and **define** the **geometric structure** of the image graph for one (or a few) initial image(s).
- The image graphs for the **remaining images** in the training set can be obtained semi-automatically, by comparing the new images to model graphs (images that have been already marked) based on the **extracted Gabor jets**.

- During this process, manual intervention is required only if the fiducial points are identified incorrectly
- Since all the training images have the **same pose**, the graphs corresponding to these face images will have the same structure

**In the second stage, a FBG** is obtained from the individual image graphs by combining a representative set of individual graphs in a stack-like structure.

- Each node in the FBG is labeled by a set of Gabor jets that represent the **local variations** in the associated fiducial point among the population of users in the training set

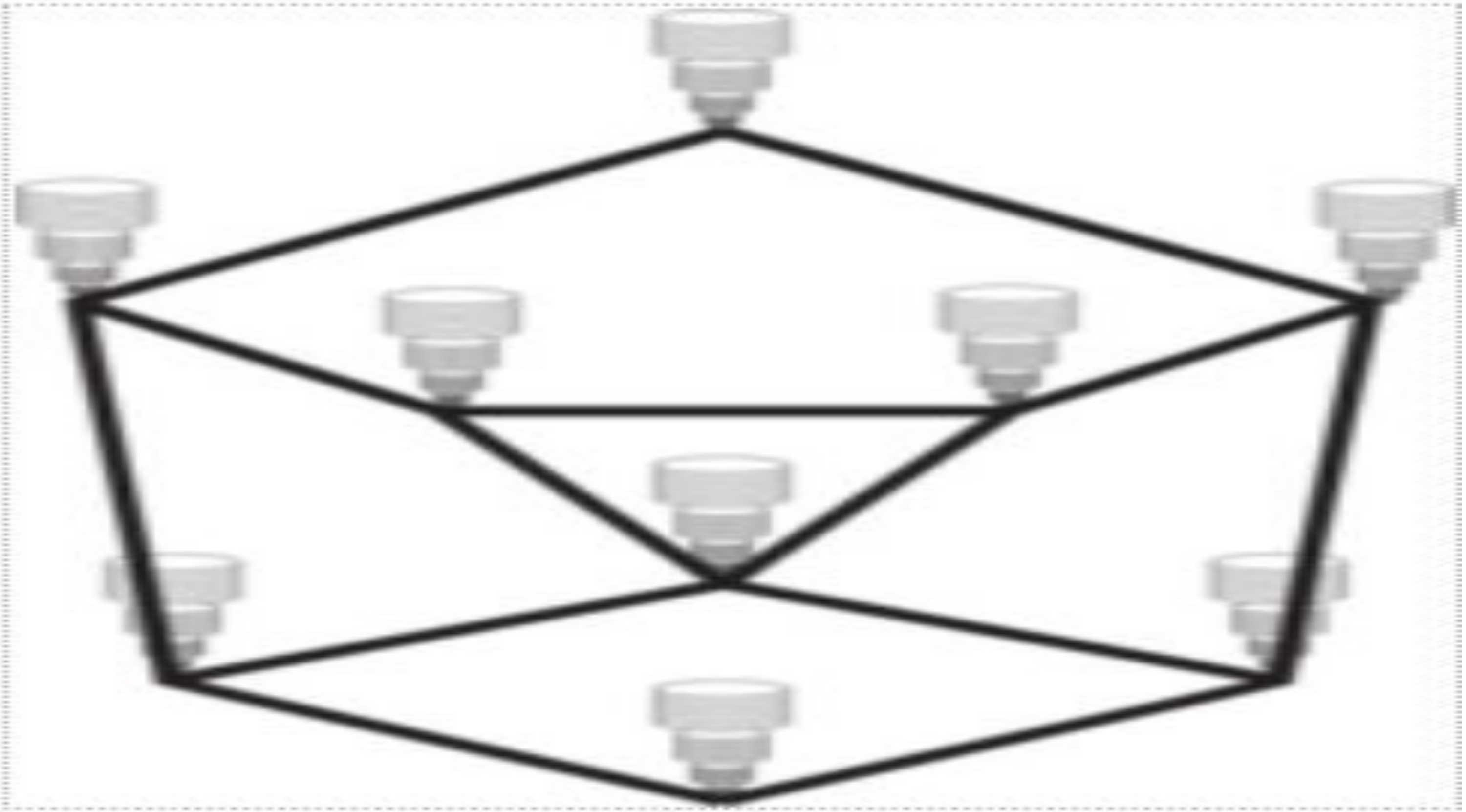


Figure 7: Schema of a model graph. On a fixed graph structure all nodes are labeled with the respective jets.

- **For example,** an eye bunch may include jets from open, closed, male and female eyes, etc. that cover the variations in the local structure of the eye.
- An edge between two nodes of the FBG is labeled based on the **average distance** between the corresponding nodes in the training set.
- Typically, a **separate FBG is constructed for each pose** and the correspondence between the nodes of bunch graphs belonging to different poses is specified manually.

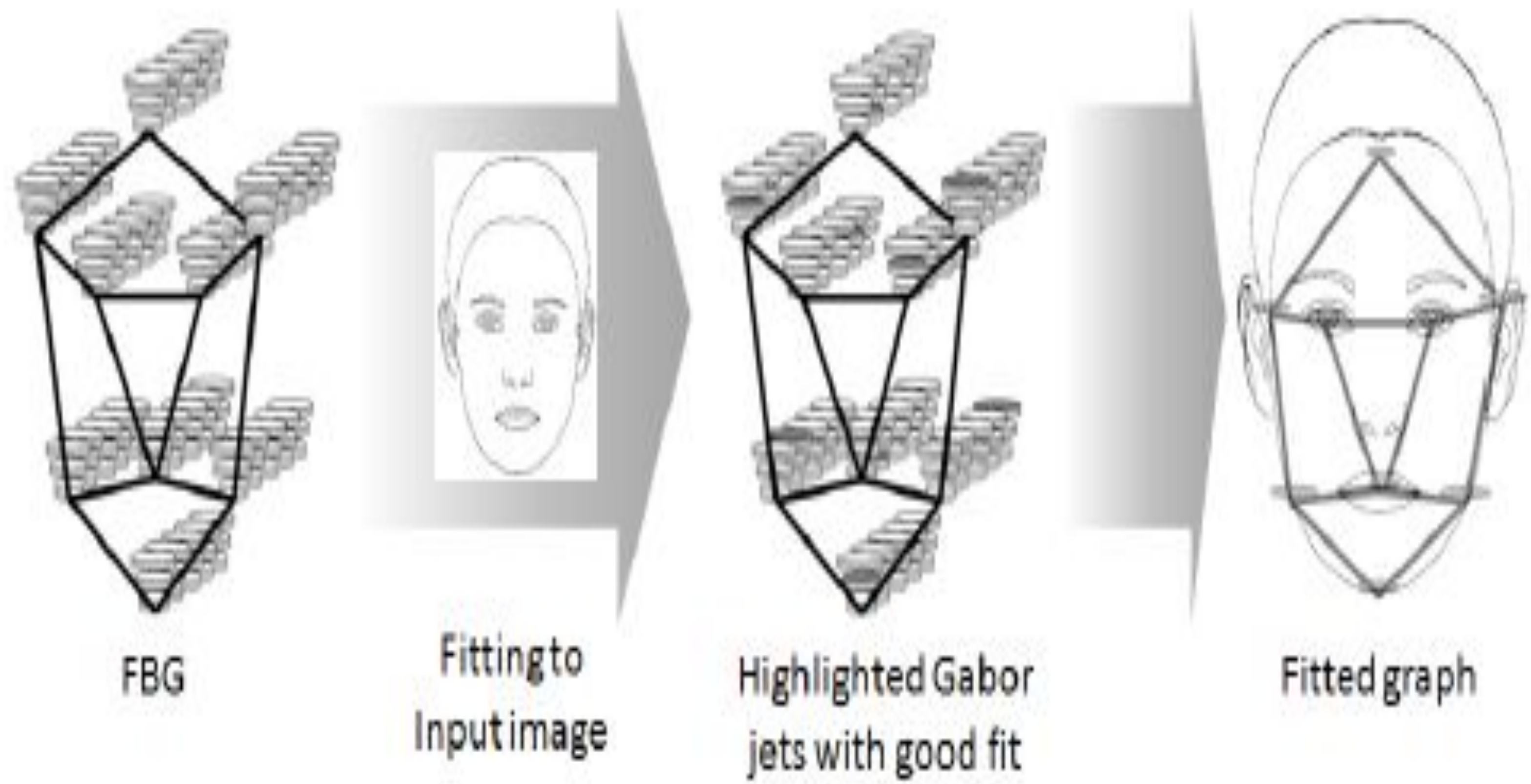



Fig. 3.26 The Face Bunch Graph (FBG) serves as a general representation of faces. Each stack of discs represents a jet. From a bunch of jets attached to each node, only the best fitting one is selected for computing the similarity and such jets are indicated by the gray shading.

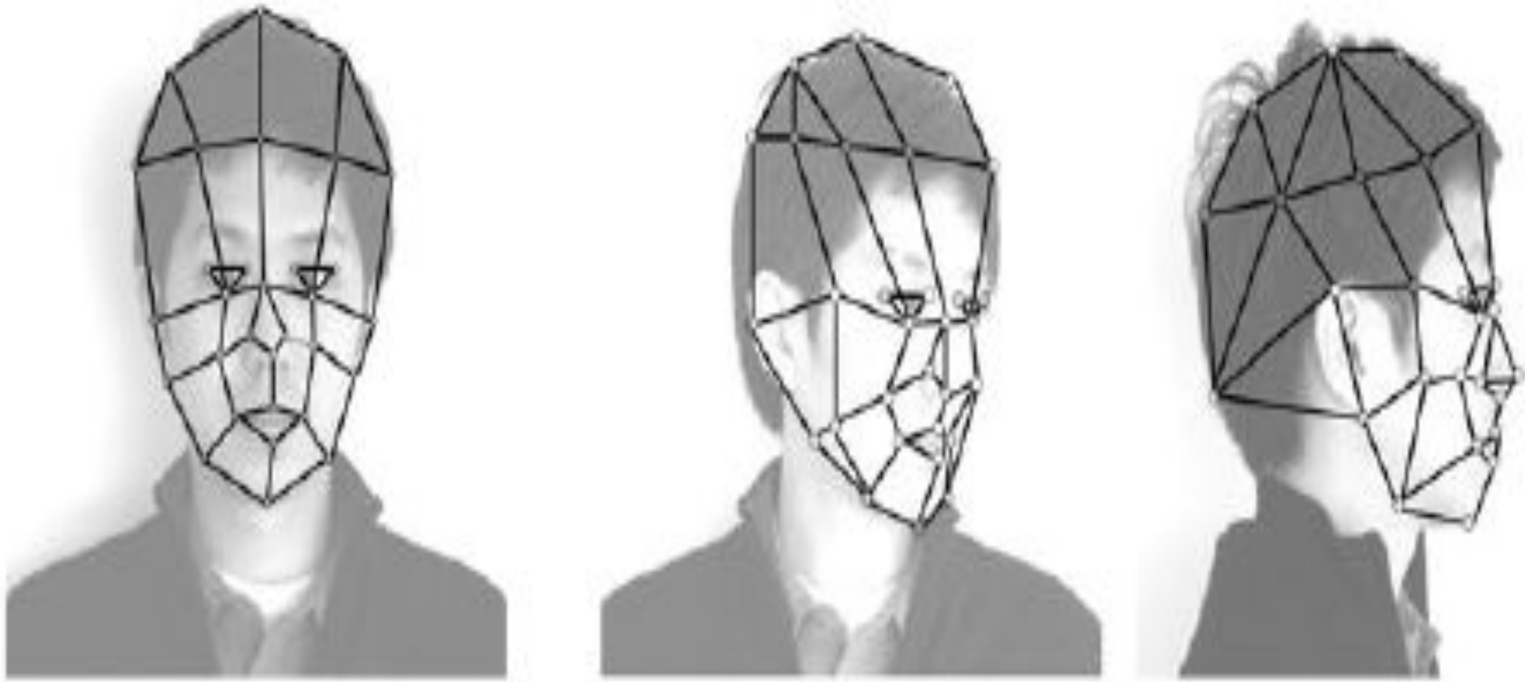


**Fig. 3.25** Determining the image graphs for face images with different poses. The nodes are positioned automatically based on comparing the Gabor jets extracted from the given image to those in the model graphs. One can observe that, in general, the fitting process finds the fiducial points quite accurately. However, errors may occur as can be observed in the case of the middle face image above. In the middle image, the chin was not found accurately; further, the leftmost node and the node below it should ideally be located at the top and the bottom of the ear, respectively.

# STEPS

*1.Find the approximate face position by coarsely scanning the input image with* a condensed FGB (an average graph obtained by taking the mean Gabor jets at each bunch).

- This is achieved by extracting the Gabor jets at some discrete locations in the given image and comparing it to the jets in the condensed FBG, while taking into account the geometric structure of the FBG

# STEPS

*2. Refine the **position** and **size** of the face by searching the image again with the full* FBG, whose size and aspect ratio is systematically varied.

- When computing the similarity between a Gabor jet in the given image and a bunch of jets in the FBG, only the FBG jet that best matches with the given image jet is considered.

*3. Precisely locate the fiducial points by moving all the nodes locally and relative* to each other to optimize the graph similarity further.

# FBG

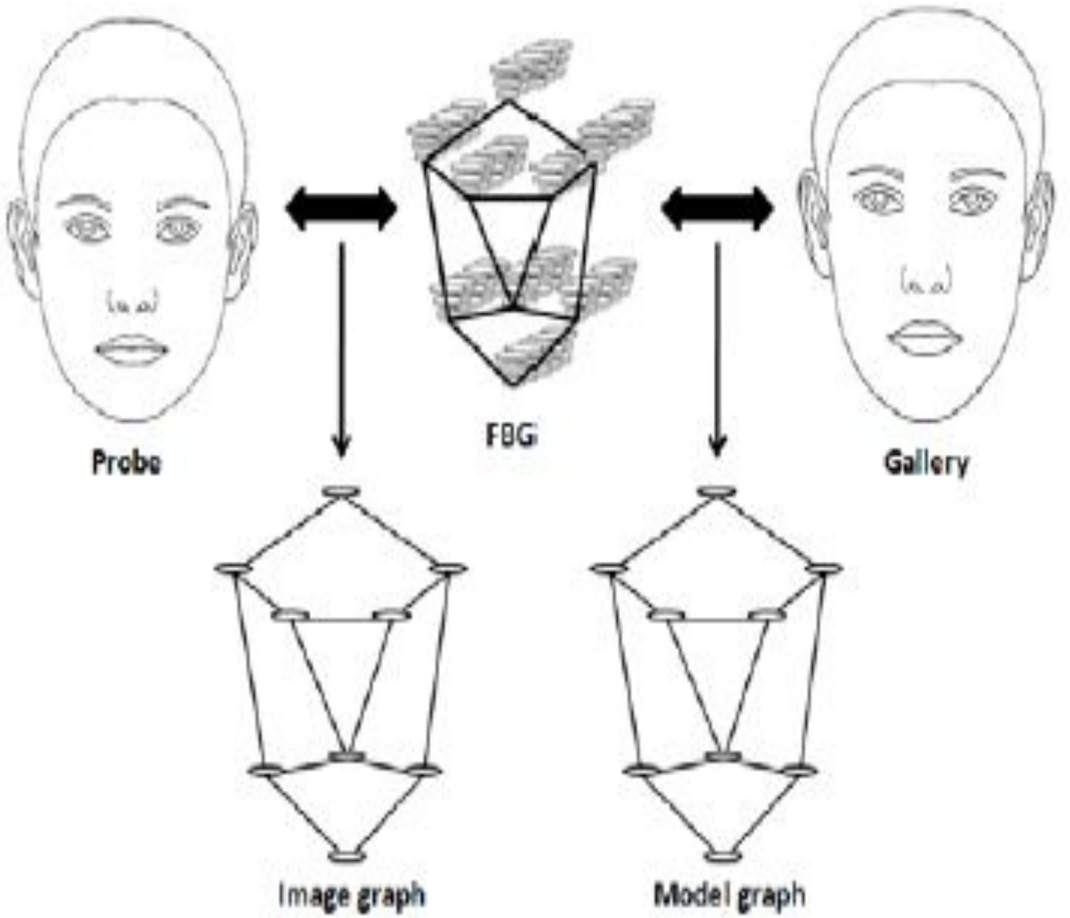



Fig. 3.27 Schematic of generating image and model graphs from probe and gallery images using the Face Bunch Graph (FBG). Note that there are bunches (set of jets) at each node in FBG, but only one jet at each node in the image and model graphs.

## *Texture-based face recognition*

- Texture of an image uses the distribution of local pixel values.
- **Scale Invariant Feature Transformation** (SIFT) and **Local Binary Pattern** (LBP) are two most well known schemes for analysis of local texture

**Scale Invariant Feature Transform:** Scale Invariant Feature Transform (SIFT) is one of the most popular local representation schemes used in object recognition.

**(a) key point extraction**,

(b) **descriptor calculation** in a local neighborhood at each key point.

## *Texture-based face recognition*

- Just like the fiducial points in the model-based approach, the **key points** can be used to achieve **tolerance against pose variations**
- The face image is typically divided with **multiple patches** and the **SIFT descriptor** is constructed from each patch.
- The **final descriptor** is obtained by **concatenating** all the descriptors from all the patches

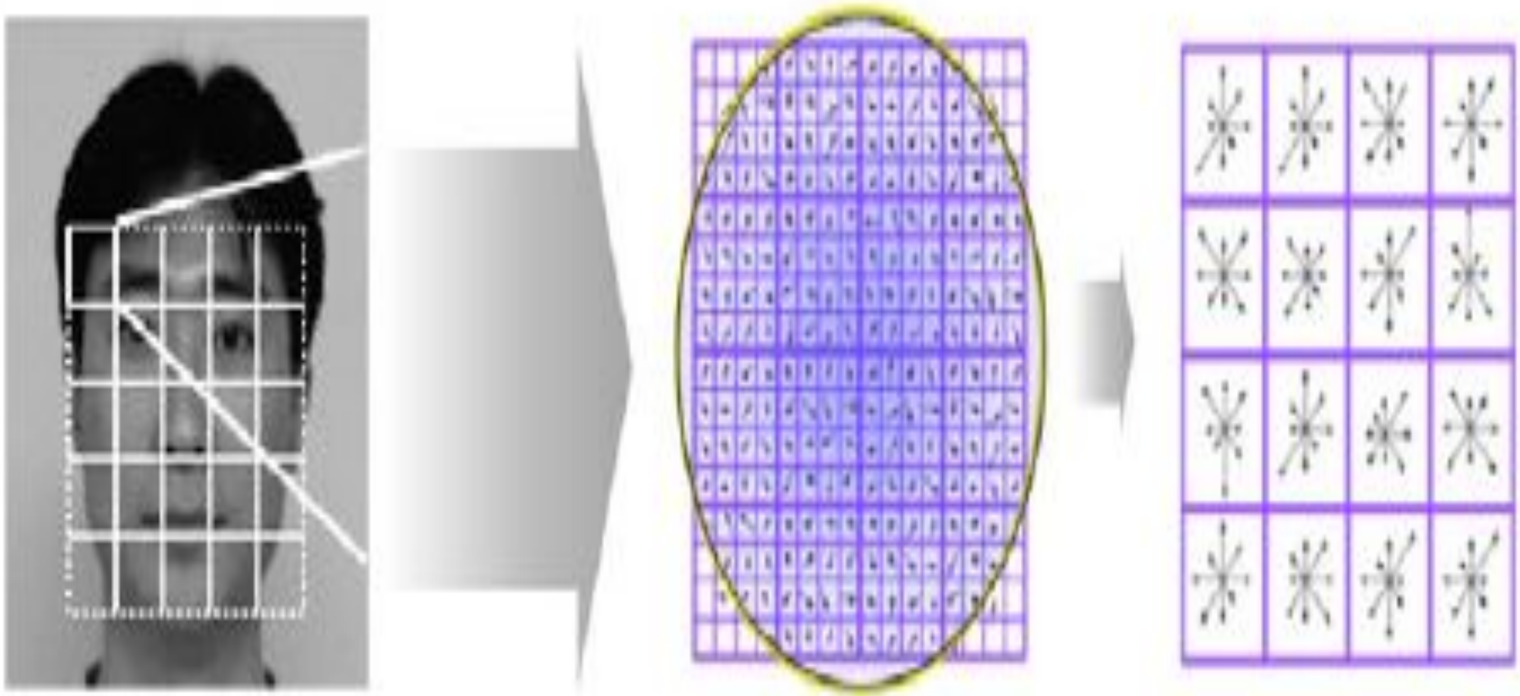


Fig. 3.28 Schematic diagram of SIFT descriptor construction. The key point detection process can be bypassed if the face images are pre-aligned.

# LBP

## Local Binary Pattern

- ***Local Binary Pattern*** (LBP) has been successfully used as a **local texture descriptor** in general object recognition as well as in face recognition
- LBP features are usually obtained from image pixels of a ***3×3 neighborhood region***
- The basic LBP operator **compares** the 8 neighboring pixel intensity values to the intensity value of the central pixel in the region and represents the result as a **8-bit binary string**.

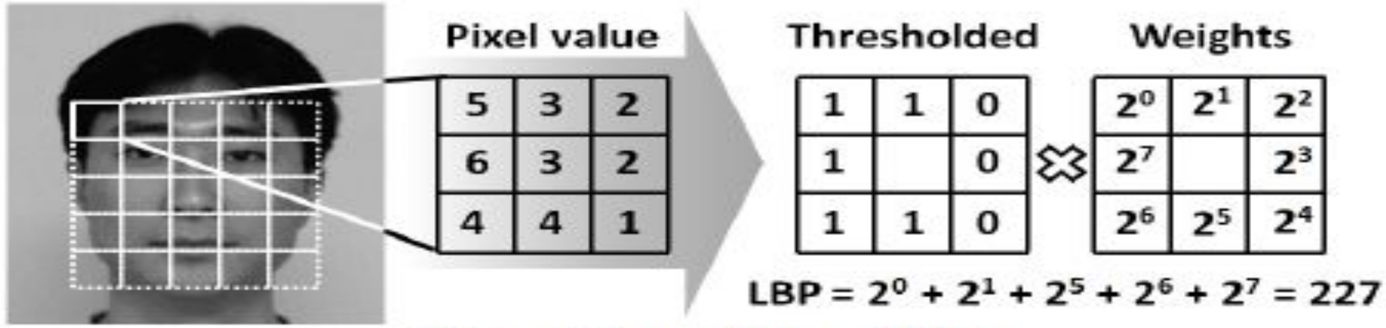
# MLBP

**Multi-scale LBP (MLBP) is an extension of the basic LBP**

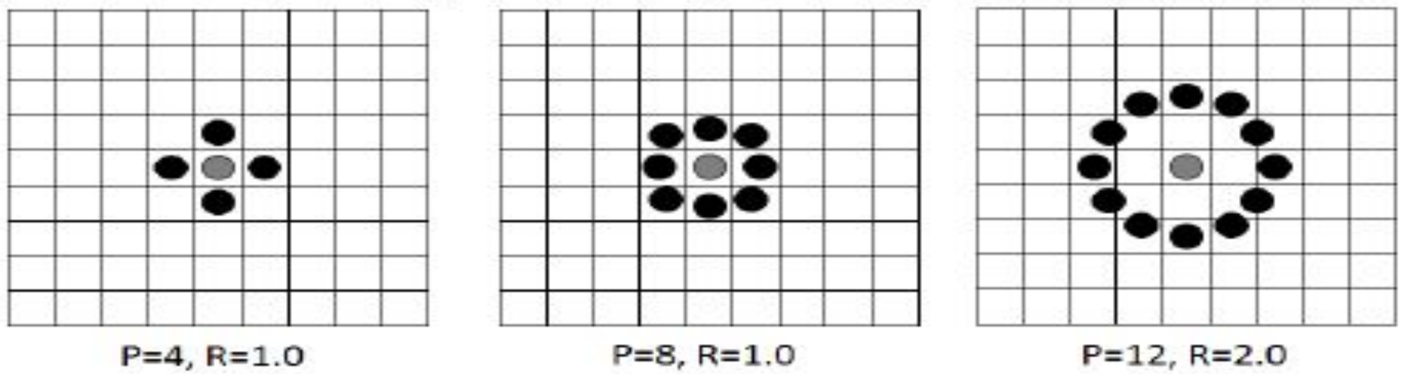
- MLBP introduces a **radius parameter *R***, *which means that the compared* neighbors are *R pixels away from the center pixel. There is also another parameter **P,** which is the **number of sampling points** along the circle of radius R.*
- *If a sampling* point is **off** the pixel grid, **bilinear interpolation** of pixel values can be applied to obtain the **intensity value** of the sampling point.

**The MLBP operator with parameter**

- *Generally,MLBP with larger value of P provides* more **detailed** information about the **local region**. However, when *P becomes larger,* the dimension of the descriptor also increases.
- MLBP operators with different values of *R encode different local image structures, ranging from micro to macro details*




(a) Local Binary Pattern (LBP)




(b) Multiscale Local Binary Pattern (MLBP)

**Fig. 3.29** Schematic diagram of (a) Local Binary Pattern (LBP) and (b) Multiscale LBP calculation. $P$ and $R$ represent the distance of the sampling points from the center pixel and the number of the sampling points to be used, respectively.

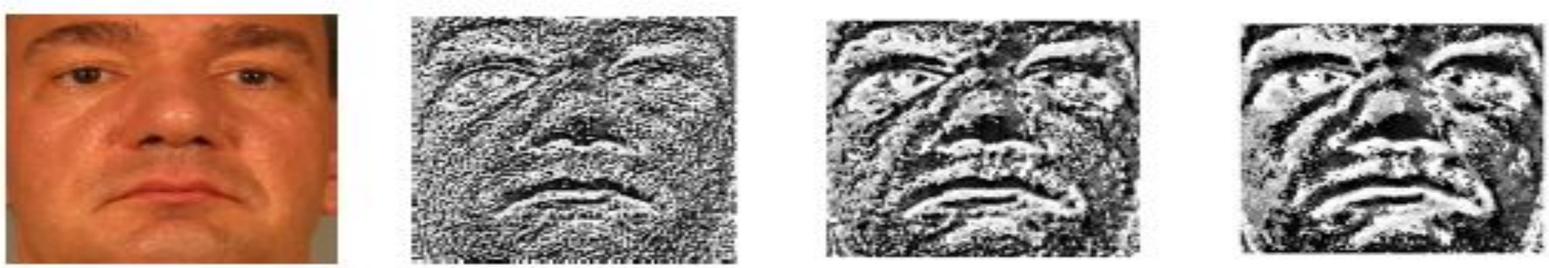


**Fig. 3.30** Local Binary Pattern (LBP) images encoded at different scales.. From left to right: original image; images encoded using $LBP_{8,1}$, $LBP_{8,3}$, and $LBP_{8,5}$ operators, respectively.

# IRIS

- *Iris recognition and acquisition*
- *Iris Segmentation, normalization and matching*

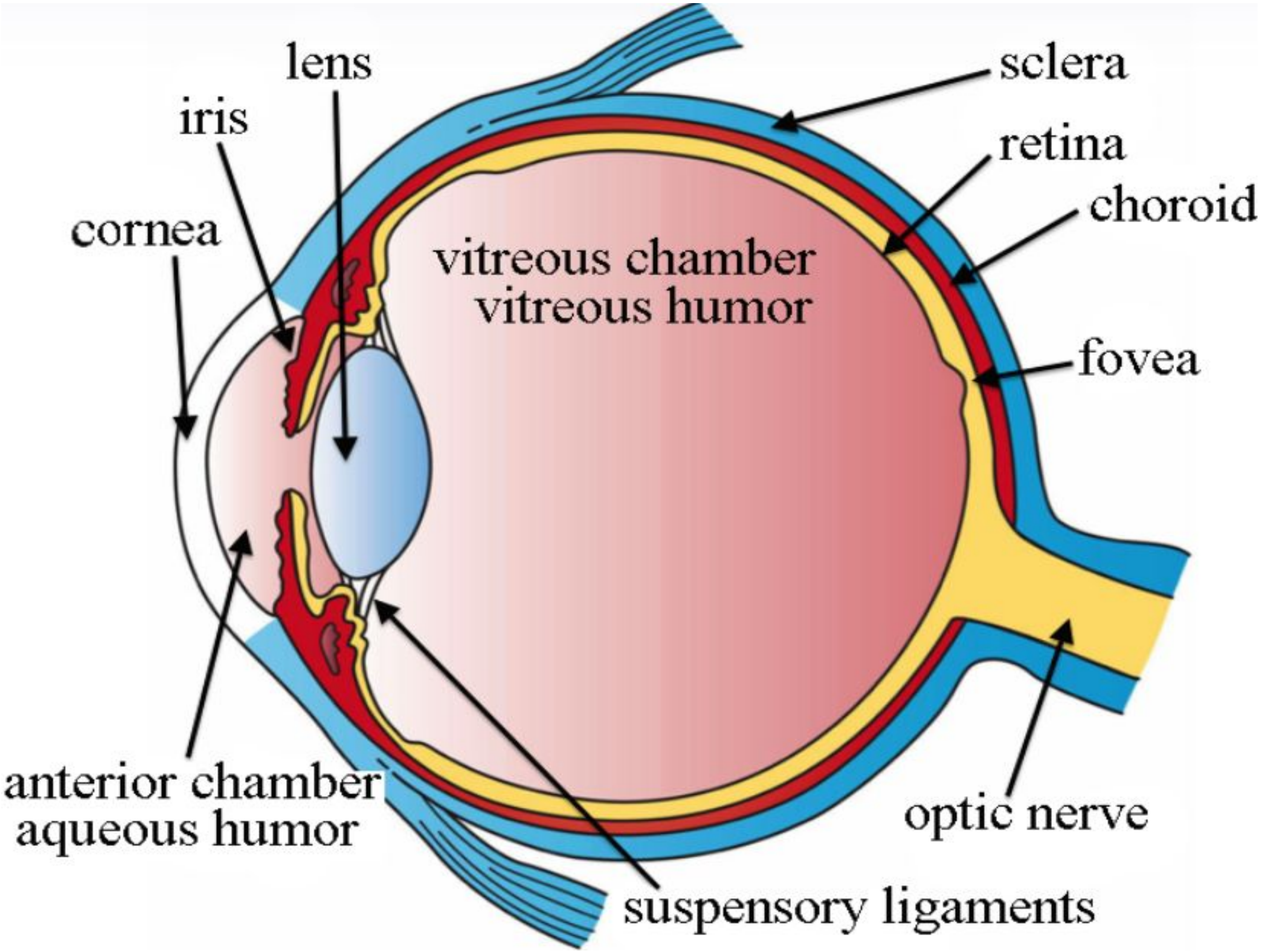
# IRIS RECOGNITION

- The use of the **ocular region** as a biometric trait has gained importance, especially due to significant advancements made in iris recognition since 1993.
- The **ocular region** of the human face consists of the eyes and the surrounding structures such as facial skin, eyebrows, and nose bridge

# IRIS

- The **iris** is an internal organ of the eye that is located just behind the **cornea** and **in front of the lens**.
- The primary function of the iris is to **regulate the amount of light entering the eye** by **dilating or contracting** a small opening in it called the **pupil**.
- The iris contracts the pupil when the ambient illumination is high and dilates it when the illumination is low.

# IRIS


lens
iris
sclera
retina
choroid
cornea
vitreous chamber
vitreous humor
fovea
anterior chamber
aqueous humor
optic nerve
suspensory ligaments

# LAYERS OF IRIS

- The iris is a multilayered structure and a cross-section of the iris reveals the following layers:
- The **posterior layer:** at the back, which is two cells thick, contains heavily pigmented epithelial cells, making it impenetrable to light.
- The **muscle layer:** Above it consists of the sphincter and dilator muscles that contract and dilate the pupil

# LAYERS OF IRIS

- The **Stromal layer**: located above the muscles, is made up of collagenous connective tissue (arranged in an arch-like configuration) and blood vessels (arranged along the radial direction).
- The **anterior border layer**:  is the foremost layer and has an increased density of chromatophores (i.e., pigment containing cells) compared to the stromal layer

# ANTERIOR PORTION

- The anterior portion of the iris - consisting collectively of the muscles, stroma, and the border layers - is the foremost visible portion of the iris.
- Therefore, it can imaged by a **camera** and is the focus of all automated iris recognition systems.

# IRIS IMAGE

- Figure shows an image of the iris as viewed by a near-infrared camera.
- The iris, from this perspective, appears as an annular entity bound by the pupillary boundary (that separates it from the dark pupil) and the limbus boundary (that separates it from the white sclera).

# IRIS IMAGE

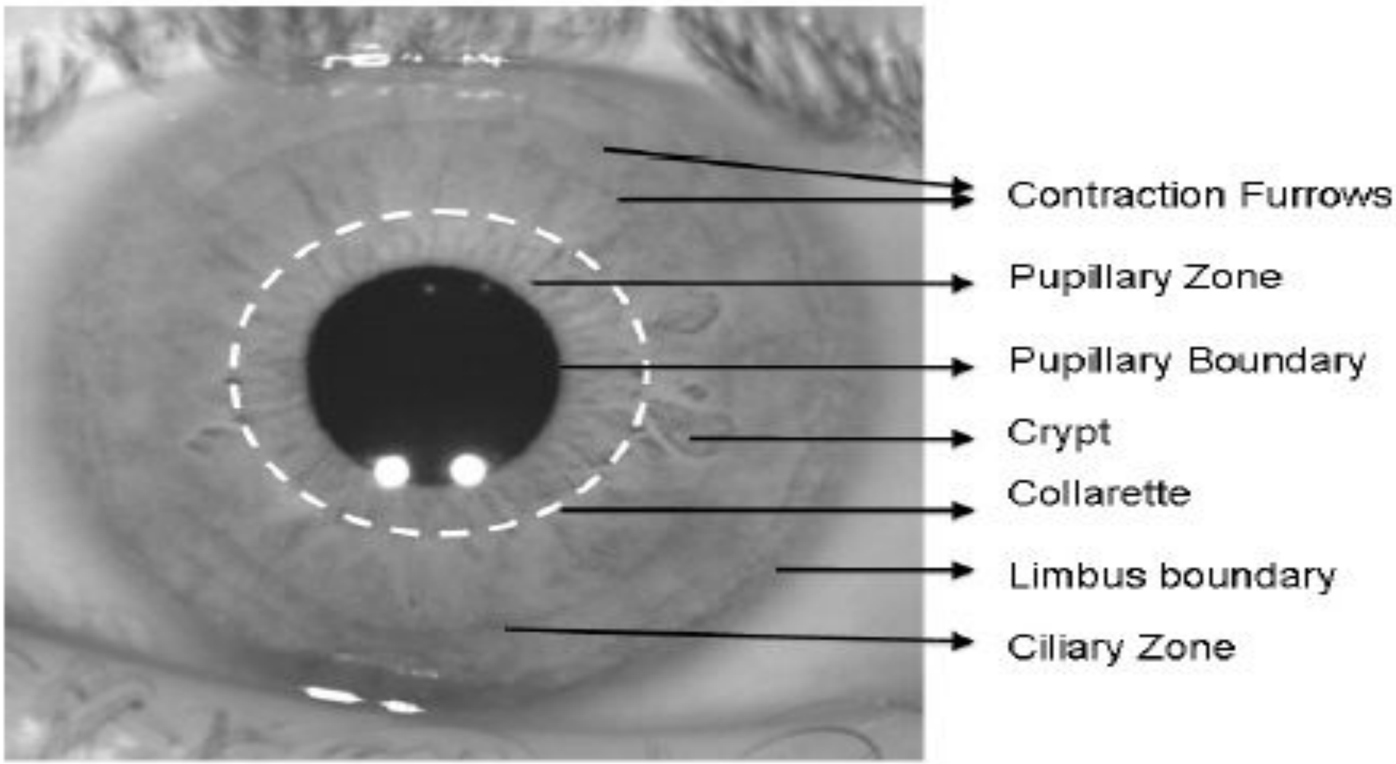



**Fig. 4.2** Anatomy of the iris as observed by a near-infrared camera placed in front of the eye. The rich texture of the iris is due to its anterior portion consisting of the musculature, stroma, and the anterior border layers. When the anterior border layer recedes, the collarette becomes visible as a zig-zag line separating the ciliary zone from the pupillary zone. Crater-like irregular structures called crypts are often observed when the anterior layer thins out, thereby revealing the heavily pigmented and much darker posterior layer.

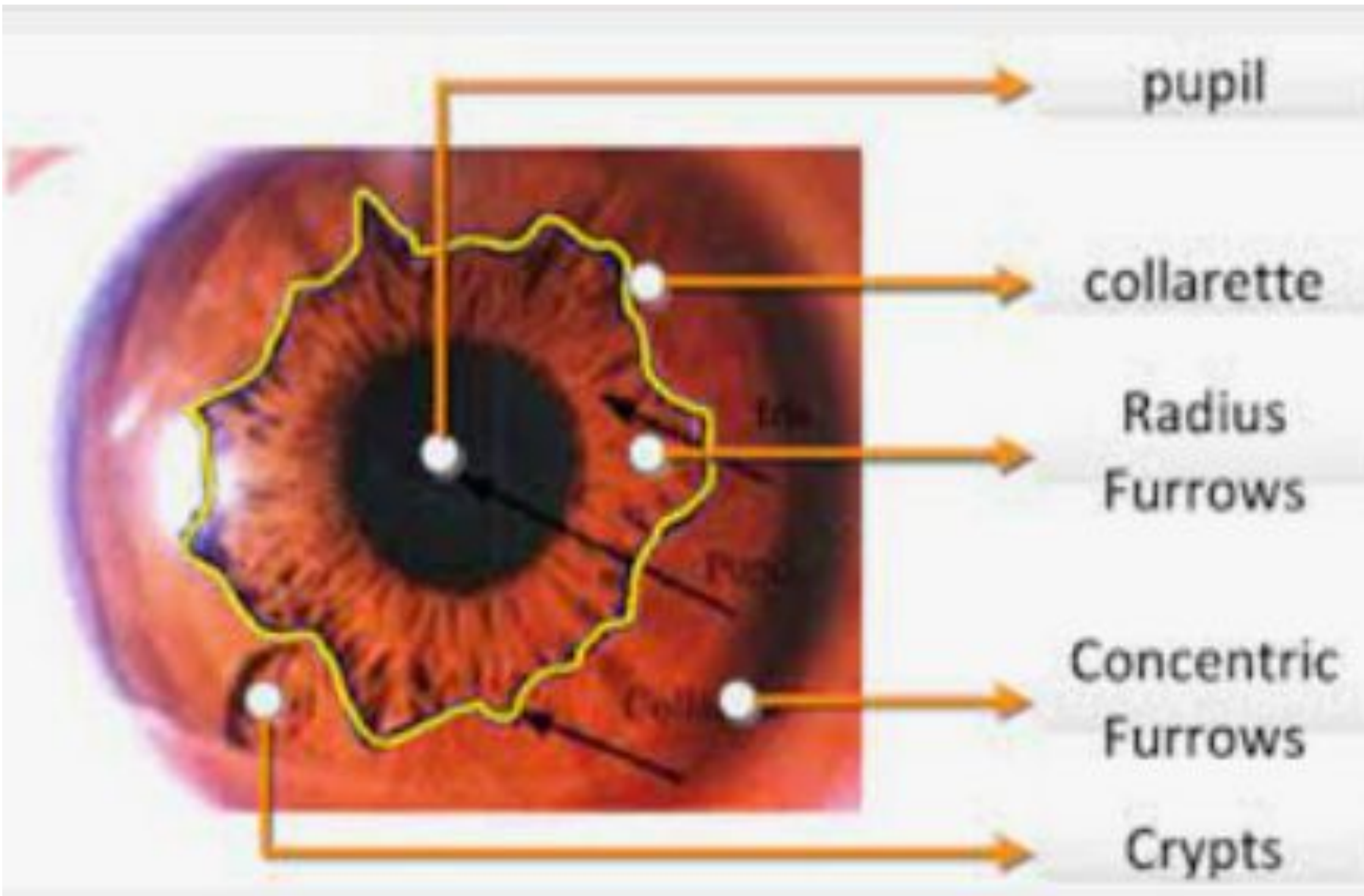
---

[1] The plural form of iris is *irides* although the term *irises* has also been extensively used in the biometric literature.

# IRIS ZONES

- The iris image is partitioned into two zones:  the central pupillary zone and the surrounding ciliary zone.
- These two zones are divided by a **circular zigzag ridgeline known as the collarette.**
- Many pit-like irregular structures appear mainly in the region around the collarette. These structures are called **crypts** (Fuchs crypts) and they **permit fluids** to quickly enter and exit the iris during dilation and contraction of the pupil.
- Near the outer part of the ciliary zone, concentric lines can be seen, especially in case of darkly pigmented irides1. These lines become deeper as the pupil dilates and are called contraction **furrows**. In the pupillary zone, radial furrows are observed.

# IRIS ZONES



pupil
collarette
Radius Furrows
Concentric Furrows
Crypts

# IRIS

- The color of the iris is primarily defined by the **pigmentation** present in it. The pigmentation itself is controlled by the number of **melanin granules** - a genetically determined factor.
- However, other factors, such as the cellular density of the stroma, can also affect the color of the iris.
- The color of the iris does *not* play a significant role in iris recognition systems.
- It is the texture detail present in the anterior portion of the iris that is useful for recognition.

# Design of an Iris Recognition System

- An iris recognition system may be viewed as a pattern matching system whose goal is to compare two irides and generate a match score indicating their degree of similarity or dissimilarity.
- Thus, a typical iris recognition system has four different modules:

1. The acquisition
2. Segmentation
3. Normalization, and
4. Encoding/matching modules.

# BLOCK DIAGRAM OF IRIS RECOGNITION SYSTEM

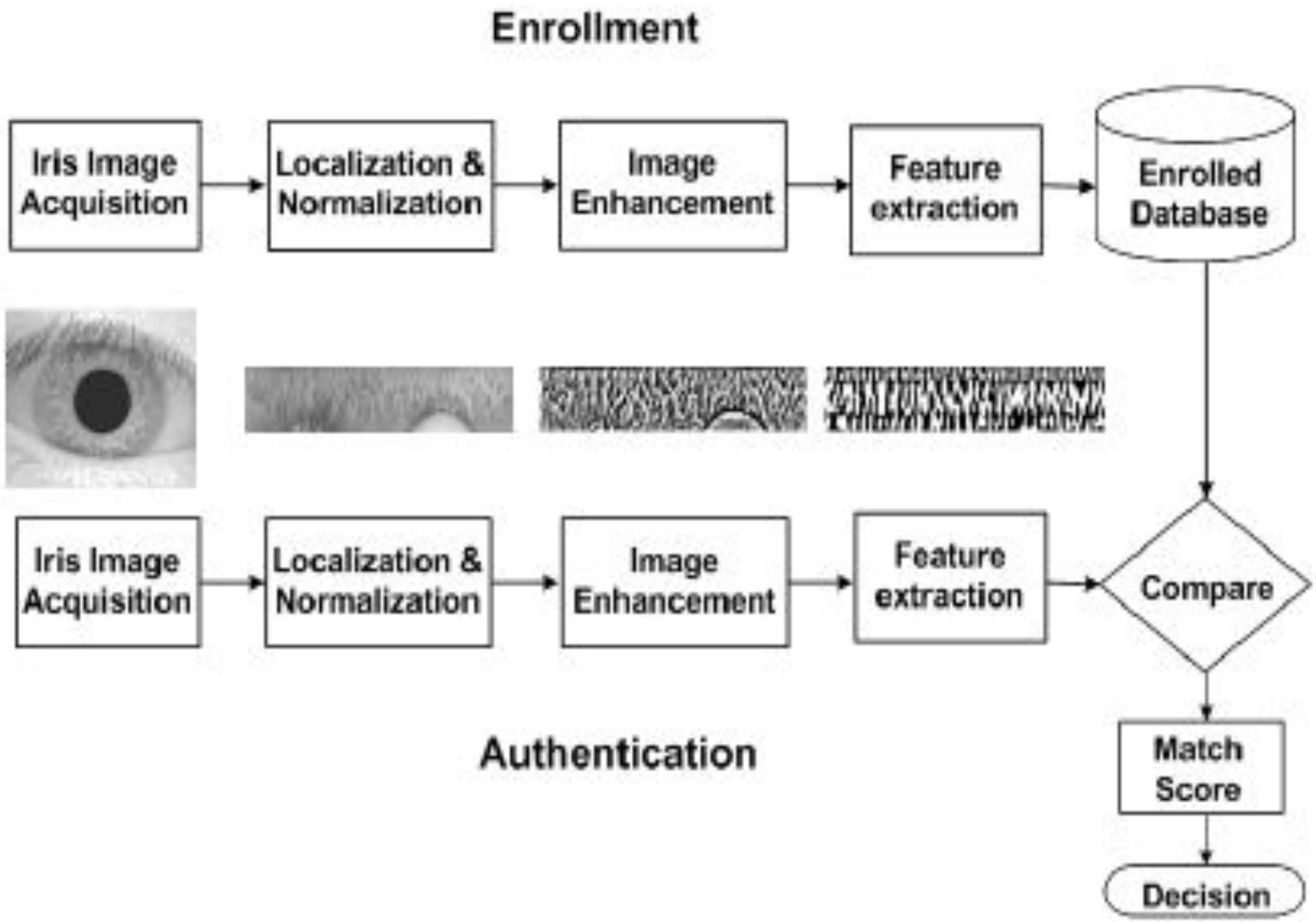



**Fig. 4.3** The block diagram of an iris recognition system. The matching performance of an iris recognition system can be impacted by the accuracy of the segmentation module, which detects the iris and establishes its spatial extent in an image of the ocular region.

# ACQUISITION

- ***Acquisition***: The role of the acquisition module is to obtain a **2D image** of the eye using a mono chrome **CCD camera** that is sensitive to the **near-infrared** (NIR) range of the electromagnetic spectrum.
- An external source of NIR light, often co-located with the acquisition system, is used to illuminate the iris.
- Most iris recognition systems require the participating individual to be cooperative and to place their eye in **close proximity** to the camera.

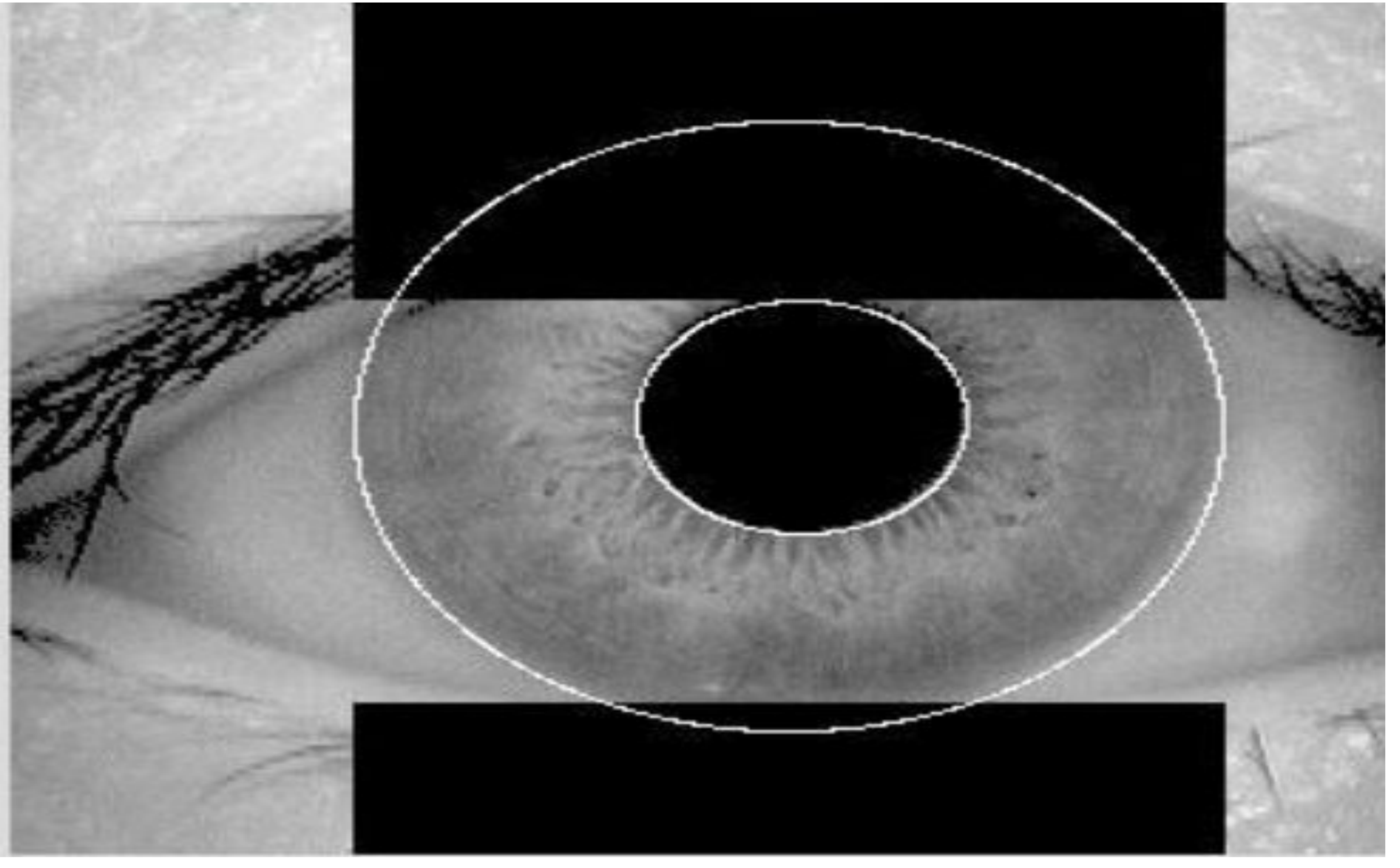
# SEGMENTATION

- ***Segmentation***: The segmentation module localizes the **spatial extent of the iris** in the image of the eye by isolating it from other structures present in its vicinity. These structures include the sclera, the pupil, the eyelids, and the eyelashes.
- Typically, segmentation is accomplished by **detecting the inner and outer boundaries** of the iris (commonly referred to as the pupillary boundary and the limbus boundary, respectively), and the eyelids and eyelashes that can interrupt the circular contour of the limbus boundary.

# SEGMENTATION

- These segmentation techniques model iris boundaries and the two eyelids with simple geometric models.
- **Pupil** and **limbus** are often modelled as **circles** and the **two eyelids** are modelled as **parabolic** arcs.
- It is desired to distinguish the iris texture from the rest of the image.
- The **integro-differential operator**, **Hough transform** and **active contour models** are commonly used to detect the boundaries of the iris.

# SEGMENTATION



# NORMALIZATION

- ***Normalization***: Once the inner and outer boundaries of the iris are estimated, a geometric normalization scheme is invoked to transform the iris texture within the annular region from **cartesian coordinates to pseudo polar coordinates** via a **Rubber Sheet Model**. This process is called as the "**unwrapping of the iris**"
- It results in a **rectangular entity** whose **rows** correspond to the **angular direction** in the original iris and whose **columns** correspond to its **radial direction**.

# NORMALIZATION

# Encoding and Matching:

- ***Encoding and Matching*:** While the unwrapped iris may be directly used to compare two irides (e.g., by using correlation filters), typically a feature extraction routine is used to encode its textural content.
- Most encoding algorithms perform a **multi-resolution analysis** of the iris by applying **wavelet filters** and examining the ensuing response.
- A commonly used encoding mechanism uses **quadrature 2D Gabor Wavelets** to extract the local phasor information of the iris texture.
- Each phasor response is then encoded using two bits of information based on the quadrant of the complex plane in which it lies. The resulting 2D binary code is referred to as the **iris code**

- Two such iris codes can be compared using the **Hamming distance**, which computes the number of corresponding bits that are different across them;
- The **binary mask** computed in the segmentation module is used to ensure that only bits corresponding to valid iris pixels are compared.

# Ear

- *Ear recognition*
- *Ear detection*

# Ear

- The human ear is observed to exhibit variations across individuals as assessed by the curves, surfaces, and geometric measurements pertaining to the visible portion of the ear, commonly referred to as the **pinna**.
- The structure of the pinna depicting various anatomical features can be seen in Figure

# EAR

(1) Helix Rim
(2) Lobule
(3) Antihelix
(4) Concha
(5) Tragus
(6) Antitragus
(7) Crus of Helix
(8) Triangular Fossa
(9) Incisure Intertragica

**Fig. 5.1** External anatomy of the ear. The visible flap is often referred to as the *pinna*. The intricate structure of the pinna coupled with its morphology is believed to be unique to an individual, although large-scale evaluation of automated ear recognition systems has not been conducted.

# The Ear offers several advantages

(a) the structure of the ear has been observed to be **stable despite aging**, and ear growth is almost linear after the age of four

(b) the ear, unlike other facial features, is **minimally impacted** by changes in facial expression; and

(c) image acquisition does not involve explicit contact with the sensor.

# EAR RECOGNITION SYSTEM

A typical ear recognition system consists of the following components:

(a) An ear detection module (also known as segmentation) that localizes the position and spatial extent of the ear in an image

(b) a feature extraction module that extracts discriminative features from the ear

(c) a matching module that compares the features extracted from two ear images and generates a match score; and

(d) a decision module that processes the match score(s) and establishes the identity of the subject.

# Ear detection

- 1. Template Matching
- 2. Model-based Detection
- 3. Morphological-operator-based Detection
- 4. Face-geometry-based Detection

# TEMPLATE MATCHING

**Template Matching**:

- In a template matching scheme, a template of a typical ear is constructed and is matched with each location in the query image.
- The location giving the **highest score** is considered as the region containing the ear.
- The template may consist of an **edge image** of the ear or a **set of descriptors extracted** from the ear such as the response to a **set of filters** or a **histogram of shape curvatures** in case a 3D image of the ear is being used for recognition

# Model based detection

- A model-based detection technique assumes certain characteristics of the shape of the ear and tries to find **regions that manifest** such characteristics.
- The shape of the helix, for example, is usually elliptical so a generalized **Hough transform** tuned for detecting ellipses can be used to locate the ear in an edge image.

# Morphological-operator-based Detection

- Since the structure of the ear is usually more intricate than the structure of the remaining region in a profile face image, morphological transformations such as the **Top-hat transformation** can be used.
- A Top-hat transformation essentially **subtracts** a morphologically smoothened version of an image from itself, thereby highlighting finer details.

# Face-geometry-based Detection

- Since in a profile image the nose can be easily detected as the point with high curvature, it is possible to constrain the search for the ear in an appropriate location relative to the nose.

# Ear recognition

- Subspace analysis-based techniques
- Sparse representation-based techniques
- Point set matching-based techniques
- Image filtering-based techniques
- Geometric measurements-based techniques
- Transformation-based techniques
- 3D techniques

# Subspace analysis-based techniques

- Projecting the ear image onto a **set of principal directions** is an effective way to obtain a salient and compact representation of an ear.
- Subspace projection techniques such as **PCA,** and **LDA** have been successfully used in literature for matching ear images.
- Manifold learning based techniques such as **Locally Linear Embedding** (LLE) and **Kernel PCA** have also been used to perform ear recognition.

# Sparse representation-based techniques

- Optimization techniques that **minimize L-1 norm** of the distance vector between the transformed query and all the transformed templates in a database have been shown to provide high recognition accuracy in object recognition studies.
- This technique has also been successfully used for ear recognition.

# Point set matching-based techniques

- **Elastic bunch graph matching** is an effective technique to recognize faces based on responses to a bank of Gabor filters at some nodal points on the face.
- This technique has also been successfully used for matching ear images where a number of landmark points can be easily detected, thanks to its complex structure.
- **Scale Invariant Feature Transform** (SIFT) is a well known technique for matching two images where a set of salient points can be reliably and repeatably extracted from them.

Fig. 5.2 Comparing two ear images by using the SIFT key points matching scheme. Here, the SIFT keypoints are first extracted from each image prior to comparing them.

# Image filtering-based techniques

- In certain techniques, the ear image is first **enhanced** to highlight the discriminative features and **suppress** the noise.
- Two common techniques that use this basic procedure are

a) force field transformation and

b) local binary patterns

# Force field transformation

- A force field transformation obtains the **intensity** of **the forces** at each location in the image where each pixel is considered as a **source of the force** with intensity proportional to its value.
- **Force field transformation** has been shown to effectively remove the **noise** in the ear image leading to significant improvement in recognition accuracy.
- Further, a set of lines indicating the gradients of this force field can be extracted and used for matching.
- Figure for a depiction of force field extracted from an ear image with the force field lines marked.

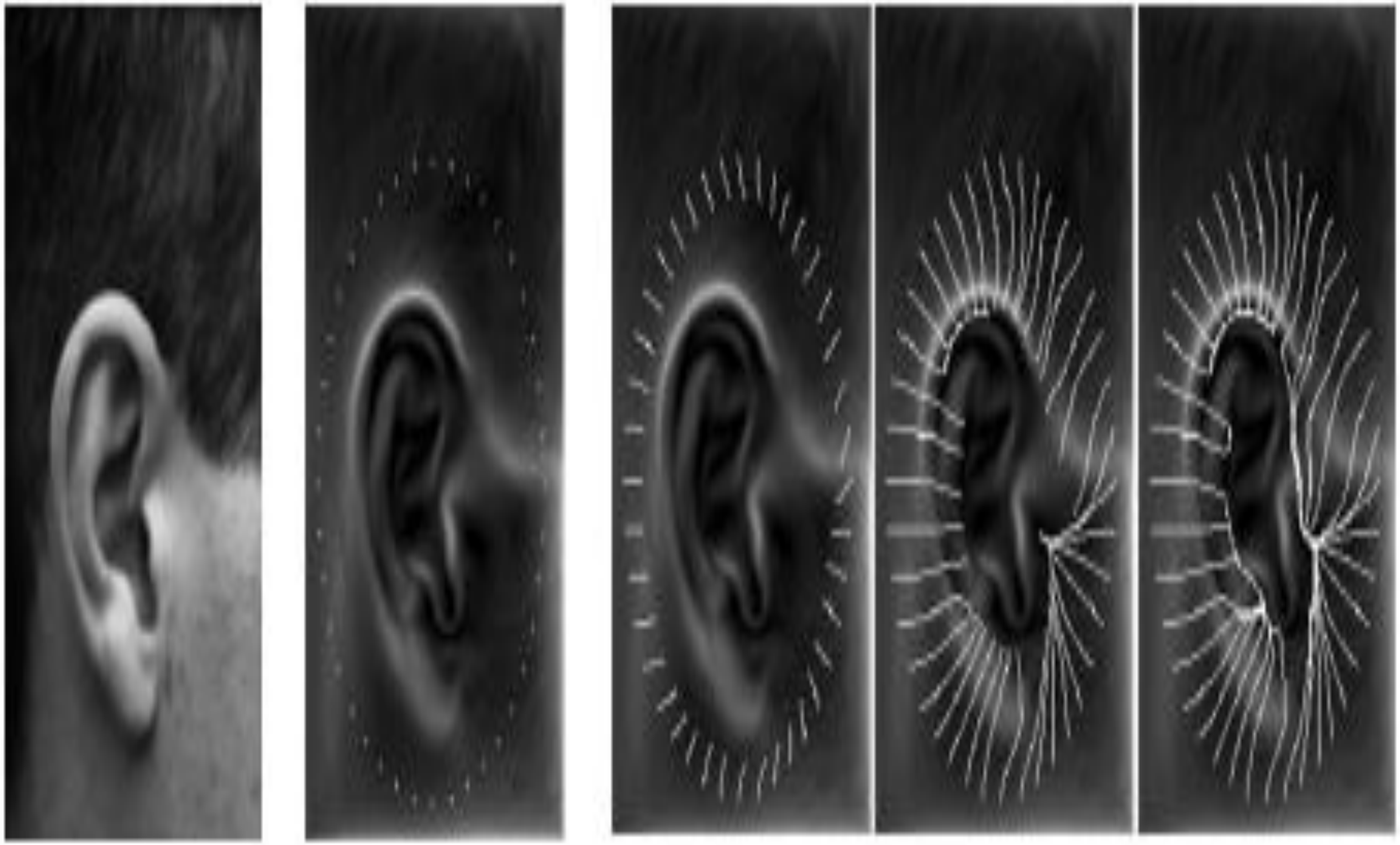


Fig. 5.3 Extraction of force field lines from an ear image using an iterative approach.

# Local binary patterns

- Local binary patterns essentially characterize each pixel based on **variation of intensity** of that pixel along a **set of directions**.
- The variation along each direction is encoded as a **single bit** indicating whether the **intensity** is **increasing or decreasing**, and the set of bits associated with each direction is used to obtain an **integer value for each pixel**.
- Such a transformation effectively reduces the **effect of illumination variations** and other sources of noise, thereby generating an enhanced image which can be used for robust matching.

# Geometric measurements-based techniques

- Features obtained by measuring certain **geometric characteristics** of the ear can also be used as a set of discriminative features.
- As an example, the **centroid of a near image** obtained from its edge image can be used as a center to draw concentric circles with pre-specified radii.
- Various measurements, such as **number of points on a circle intersecting the edge image** or **distance** between two consecutive intersections, can be used as a **feature vector**.

# Transformation-based techniques

- Various image transformation techniques such **as Fourier transform or wavelet transform** can also be applied to extract discriminative features from an ear image.
- Fourier transform can also be applied in order to obtain a rotation and translation invariant representation of the ear –
- For example, by using **a polar coordinate** system and extracting only the **magnitude** of the Fourier transform.

# 3D Techniques

- 3D images offer **depth information** that can be used in conjunction with the 2D texture information to improve the recognition accuracy.
- In the case of 3D ear images, **local histograms** of shape curvature values can be used to match two ear images.
- The **Iterative Closest Point** (ICP) algorithm is commonly used to register and match 3D ears.
- In the ICP technique, each point in the input ear image is used to obtain a corresponding point in the template image and the input is then rotated and translated in order to minimize the distances between corresponding points.

# Challenges in ear recognition

- There are no commercial biometric systems at this time that explicitly utilize features of the ear for human recognition.
- But the performances of ear recognition algorithms have been tested on some standard ear datasets. Experiments suggest that ear images obtained under controlled conditions can result in good recognition accuracy.
- 1. Ear Occlusion
- 2. Earprint Identification

# Ear Occlusion

- One of the main challenges faced by an ear recognition system is occlusion due to the subject’s hair.
- One way to address such occlusion is by capturing the thermogram along with the visible light image.
- In a thermogram, the hair can be easily detected (and possibly isolated) as its temperature is usually lower than that of the skin.

# Ear print identification

- Earprints, or earmarks, are marks left by secretions as a result of pressing the ear against a flat surface.
- These marks mainly consist of impressions of the helix, anti-helix, tragus, and anti-tragus.
- Other features include the earlobe and the crus of helix, but they are less frequently observed.
- Earprints can be compared based on details such as the notches and angles in imprinted samples, the positions of moles, folds, and wrinkles, and the position of pressure points

- Earprint identification is usually manually performed by identifying and matching a set of geometric features from the earprint such as the intersection points of  ear curves with a regular grid or locations of certain other landmark points.

# Hand Geometry

- Hand geometry, as the name suggests, refers to the geometric structure of the hand.
- This structure includes width of the fingers at various locations, width of the palm, thickness of the palm, length of the fingers, contour of the palm, etc.
- Although these metrics do not vary significantly across the population, they can still be used to verify the identity of an individual.
- Hand geometry measurement is non-intrusive and the verification involves a simple processing of the resulting features.
- Unlike palmprint, this method does not involve extraction of detailed features of the hand

# Hand Geometry

A typical hand geometry system consists of four main components:

- Image acquisition,
- Hand segmentation and alignment,
- Feature extraction,
- Feature matching.

# Image capture

- Contact-based vs Contactless:
  - A typical system requires the user to place her hand on a flat surface prior to capturing. Such systems are contact-based and require explicit cooperation of the subject being identified.
  - To address these issues, contactless recognition systems have been proposed. However, such systems are required to address the intra-class variability in the captured images due to the articulation of the hand in all three dimensions.

# Image capture

- ***Dorsal vs Palmar***
  - An image of the hand is acquired by placing the hand on a flat surface and imaging the back of the hand with a CCD camera.
  - However, there has been interest in capturing the ridge patterns present on the palm and fingers along with the hand shape by imaging the palmar aspect of the hand
  - Drawback of such a system is that it is rather inconvenient for a user to place the hand on a platen with the palm facing upward.

# Image capture

- **Peg-based vs Pegless:**
  - In order to guide the positioning of the hand on the platen for imaging purposes (e.g., to prevent the fingers from touching each other), a few pegs are usually placed on the sensor platen.
  - It adds to the complexity of using the system and, thus, greater inconvenience to users.

# Image capture

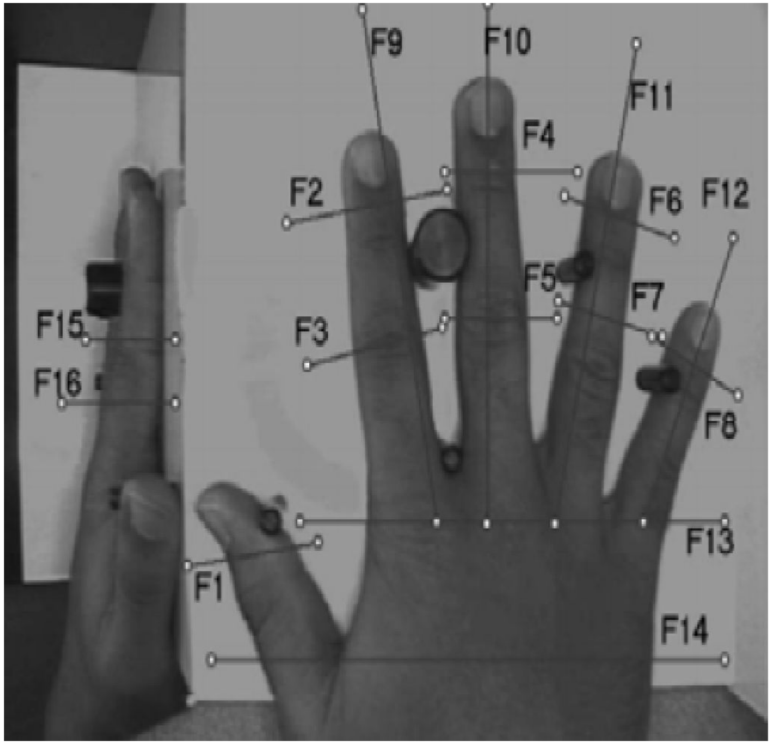
- The image acquisition system comprises of a light source, a camera, a single mirror and a surface (with five pegs on it)
- Pegs serve as control points for appropriate placement of the right hand of the user
- A 640 by 480 8-bit grayscale image of the hand is captured

# Hand segmentation

- The hand boundary must be extracted in order to determine the **region of interest.**
- The segmented hand may be further divided to obtain smaller segments corresponding to individual fingers.
- Features can then be extracted separately from each of these segments.

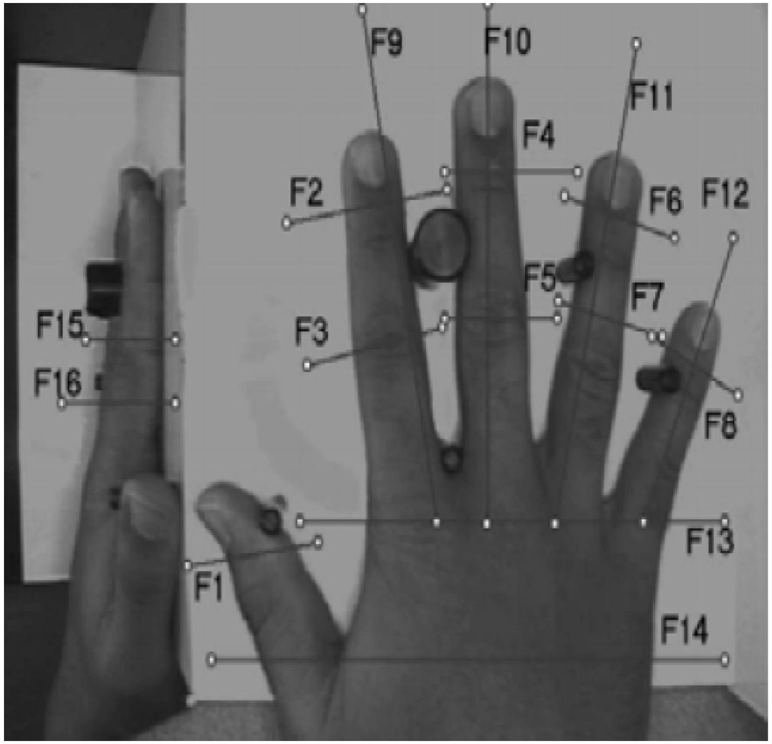
# Feature Extraction

- Two kinds of features are extracted from a hand or a finger silhouette:
  - one-dimensional geometric measurements and two-dimensional shape-based features.
- The geometric measurements include length and width of fingers, length and width of the palm, and thickness of the fingers.



# Feature Extraction

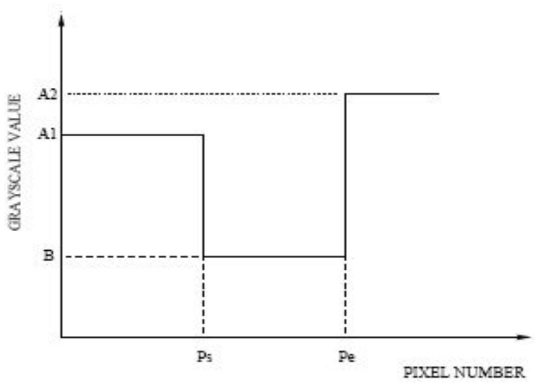
- It is also possible to use the set of points along the contours of the silhouette (or the segmented image itself) as features.
- The dimensionality of these features can be reduced in order to obtain a more discriminative and compact representation of the hand.
- 16 axes along which the various features mentioned above have been measured
- The hand is represented as a vector of the measurements selected above



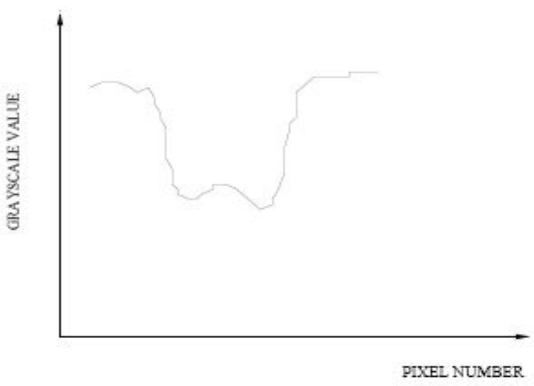

$$F = \left(f_1, f_2, \ldots, f_{16}\right)$$

# Feature Extraction

- $X = (x_1, x_2, x_3, \ldots, x_{Len})$ is the gray values of the pixels along a axis
- A sequence of pixels along a measurement axis will not have an ideal gray scale profile
  - Background lighting, color of the skin, noise and etc.



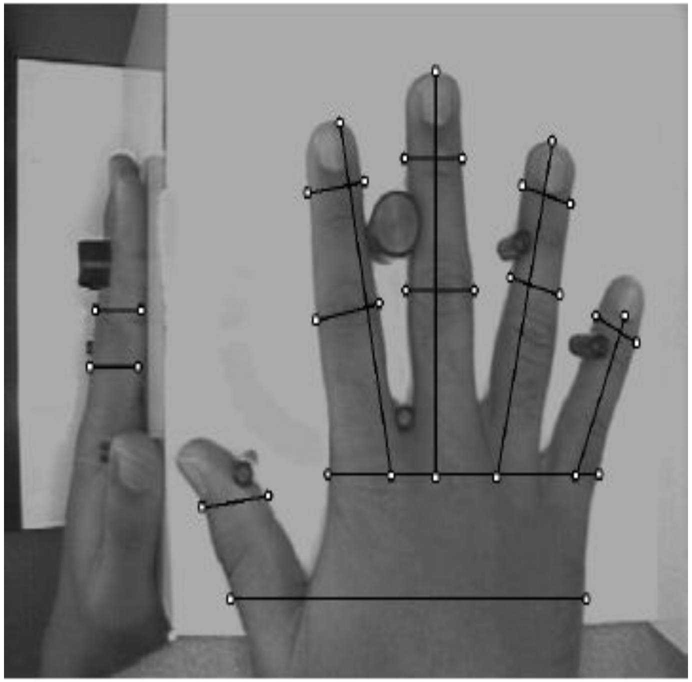

The ideal profile




An observed profile

# Feature Extraction

- Result



# Feature matching

- The features extracted from a segmented hand image can often be denoted as a feature vector in the Euclidean space.
- Common distance measures such as Euclidean and Manhattan distances can be effectively used to compare two hand images
- Mahalanobis distance
- ML algorithms such as SVM

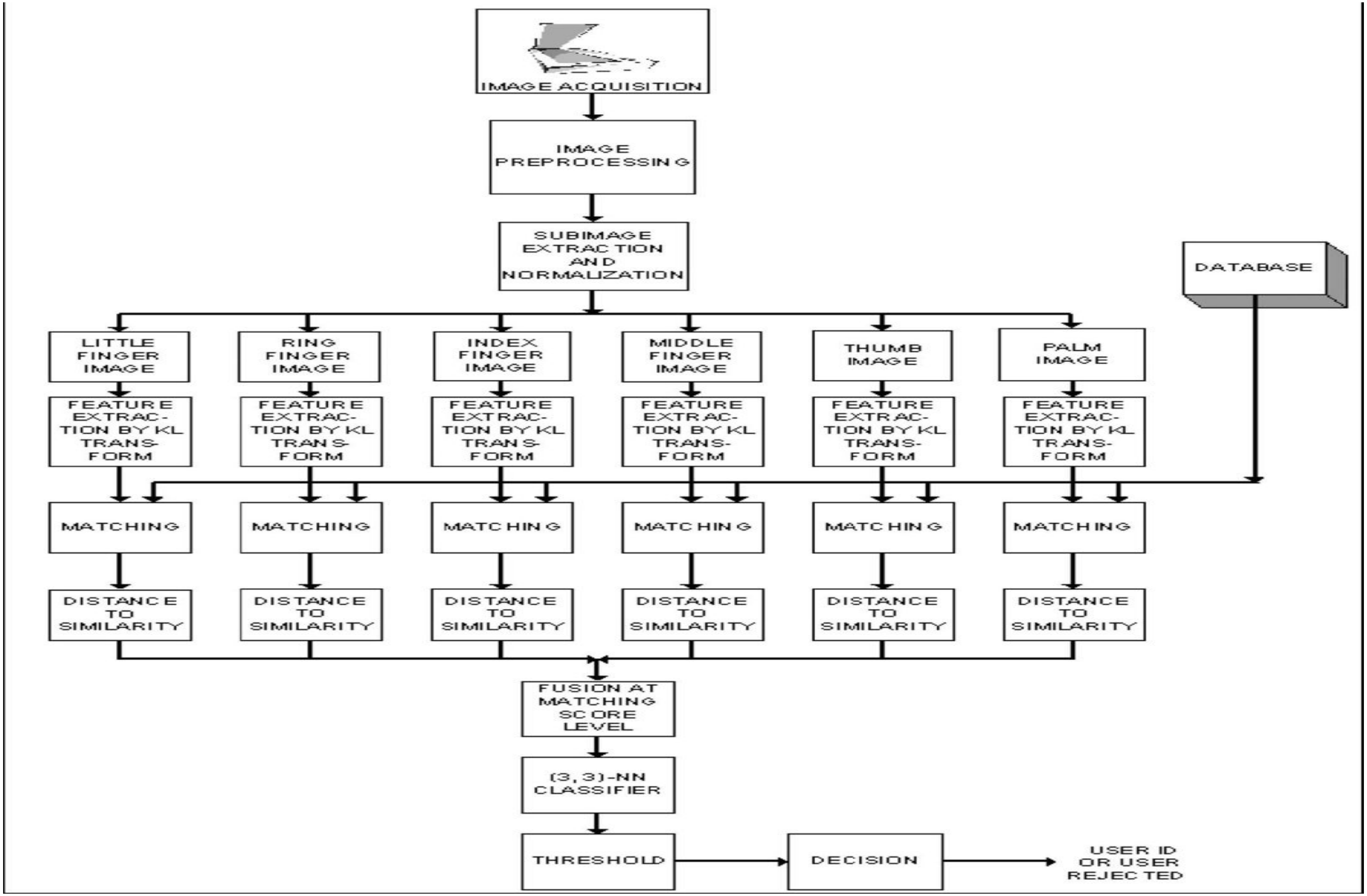
# Why Hand Geometry

- What is the most effective biometric measurement?
  - Each biometrics has its strengths and limitation
  - It’s depend on application
  - User acceptability is the most significant factor
- For many access control applications, like dormitory meal plan access, very distinctive biometrics, e.g., fingerprint and iris, may not be acceptable
- Hand geometry-based verification systems are not new and have been available since the early 1970s

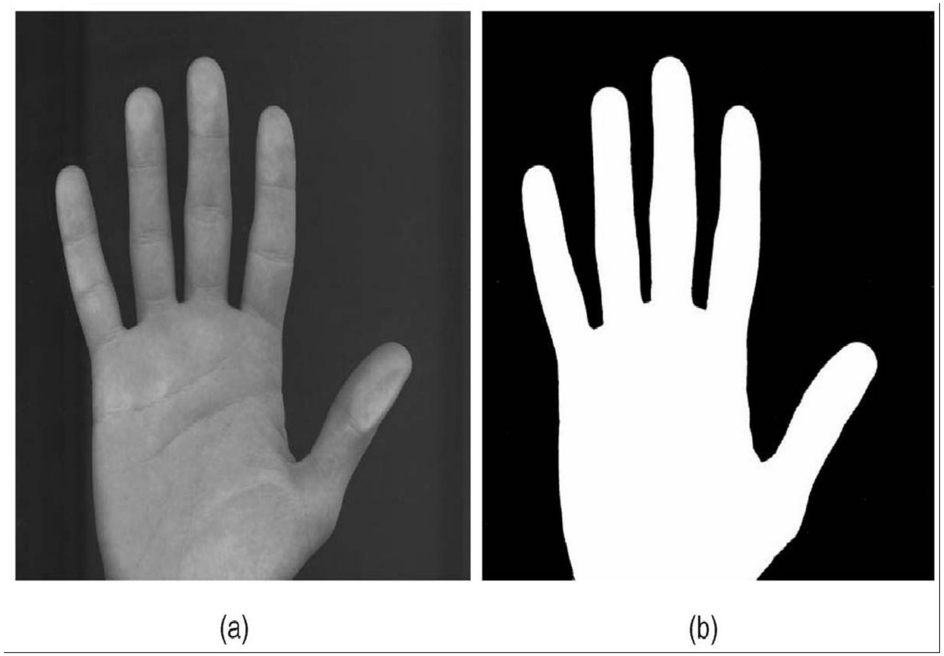
# A Biometric Identification System Based on Eigen-palm and Eigen-finger Features

- The identification process can be divided into the following phases:
  - Capturing the image
  - Preprocessing, extracting and normalizing the palm and strip-like finger subimages
  - Extracting the eigenpalm and eigenfinger features based on the PCA
  - Matching and fusion
  - A decision based on the (k, l)-NN classifier and thresholding

# System Description

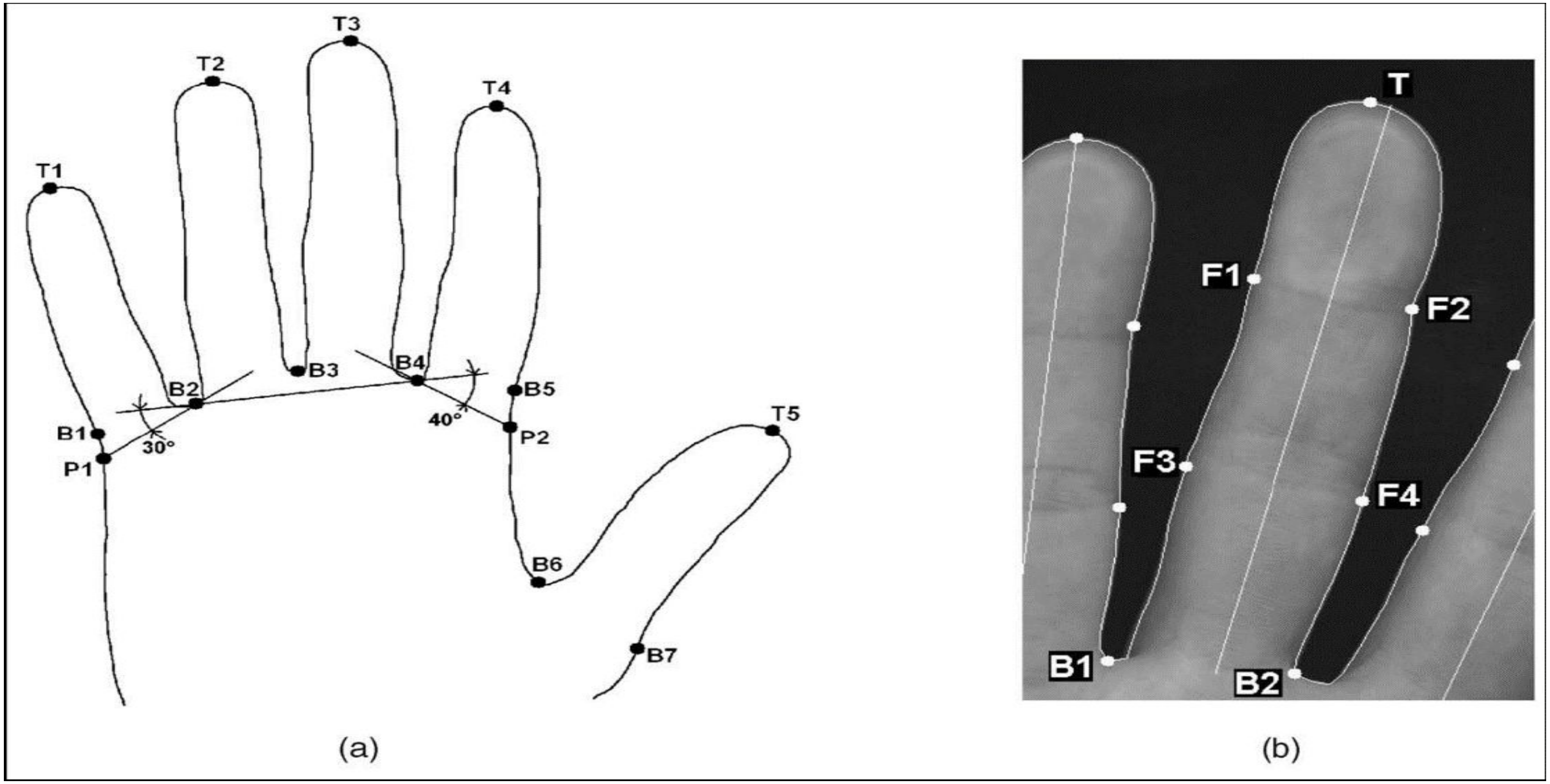


# Preprocessing and Sub-images Extraction



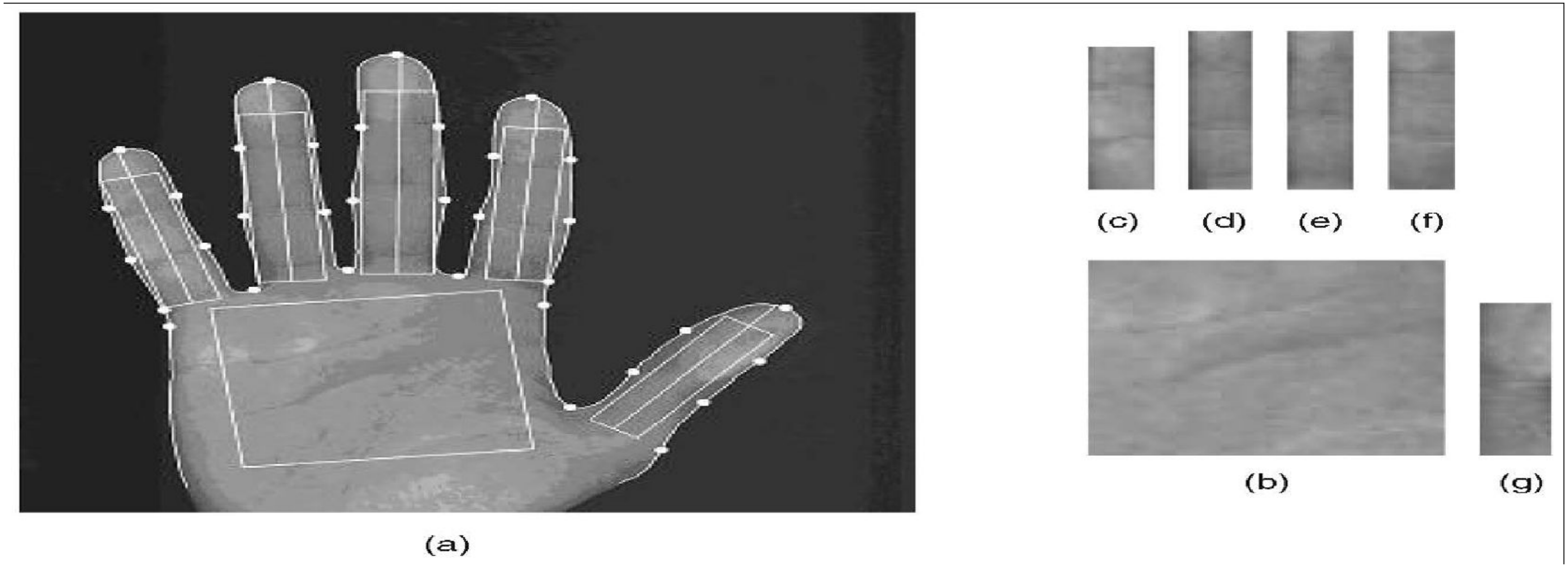

(a) Example of a scanned hand image. (b) Binerized hand image.

# Preprocessing and Sub-images Extraction




(a) The hand contour and the relevant points for finding the regions of interest. (b) Processed finger on the hand contour.

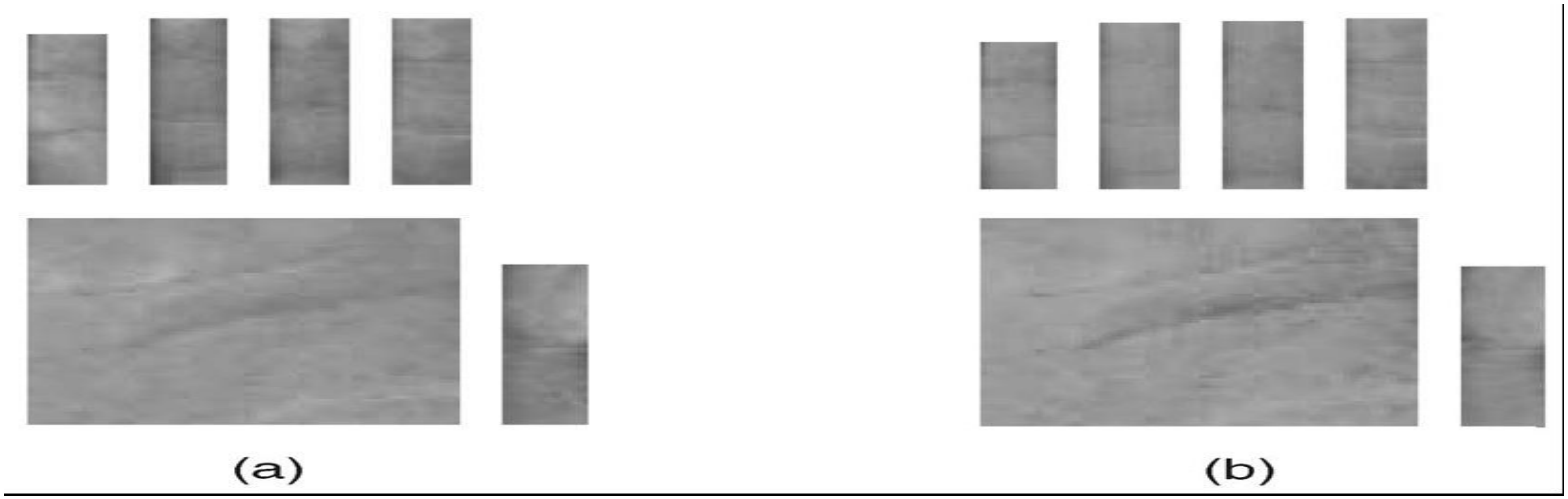
# Geometry and Lighting Normalization




(a) Original hand image with the regions of interest marked on it. (b) Palm subimage. (c) Little-finger subimage. (d) Ring-finger subimage. (e) Middle-finger subimage. (f) Index-finger subimage. (g) Thumb subimage.

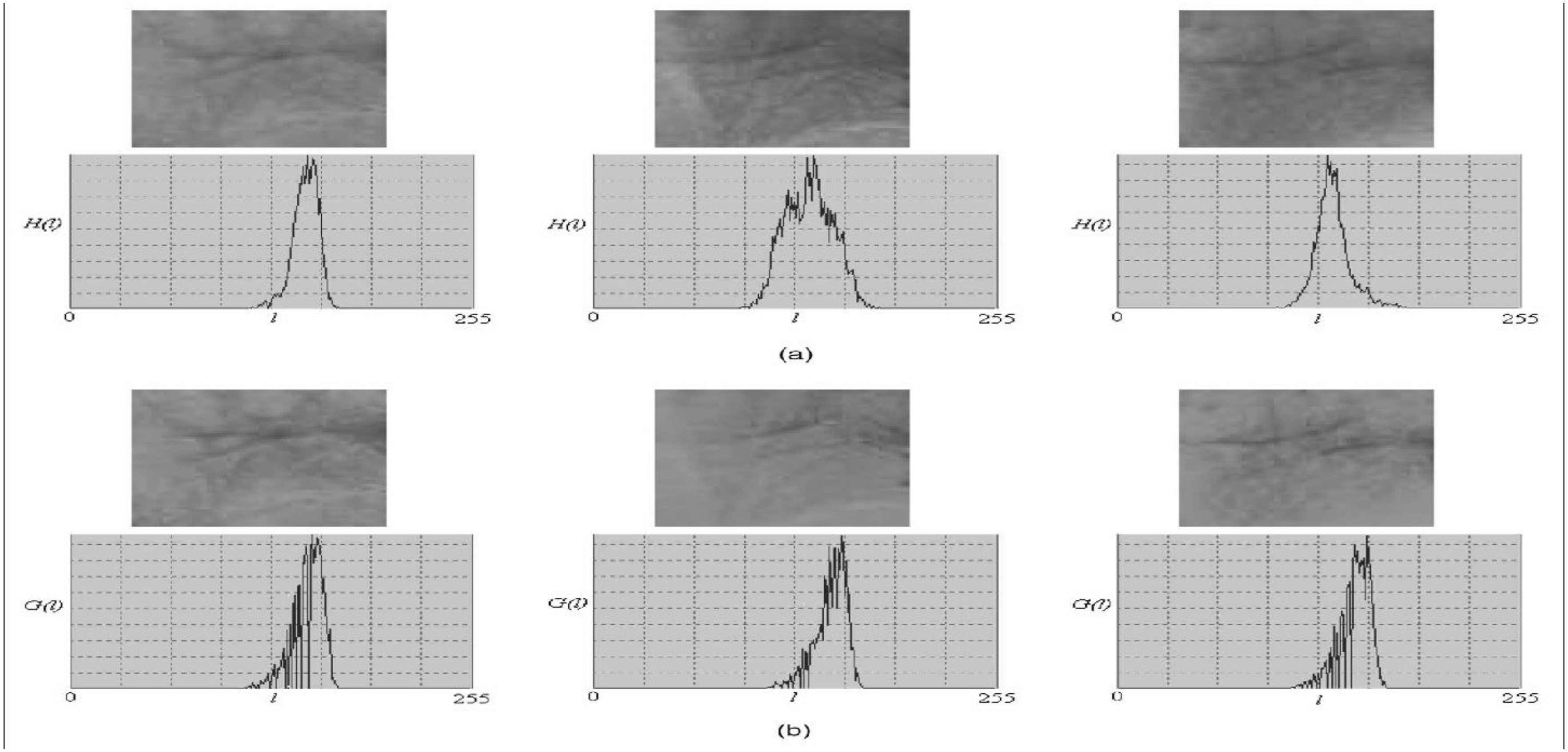
# Geometry and Lighting Normalization

- After the subimages have been extracted, a lighting normalization using histogram fitting is applied.
- In this process, a target histogram *G(l)* is selected for each of the six subimage



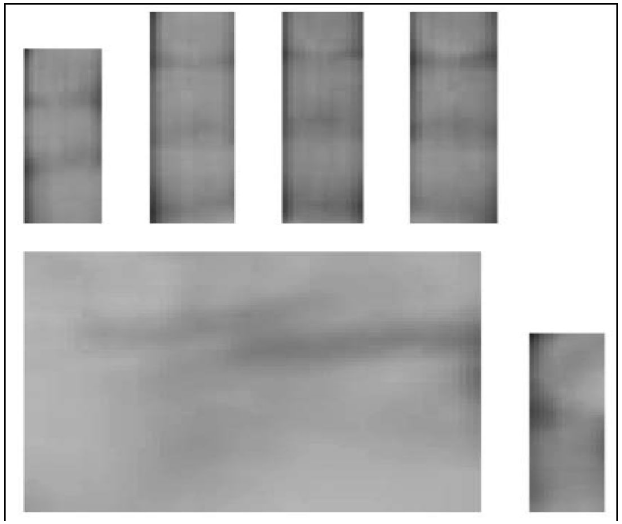

Subimages (a) before and (b) after the histogram fitting.

# Lighting Normalization

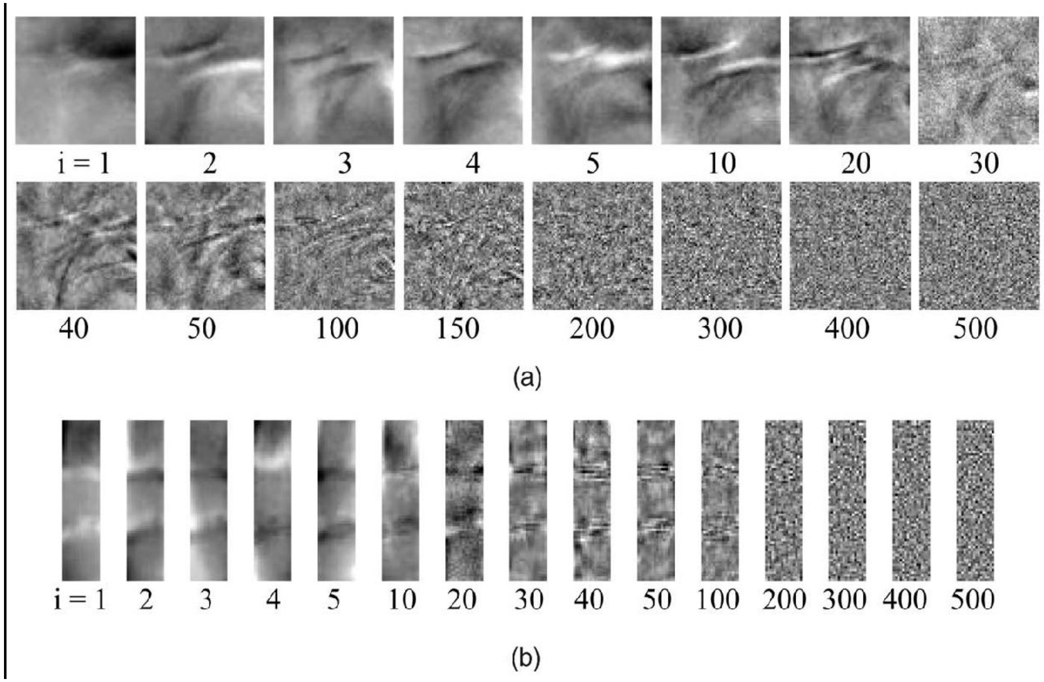



Palm sub-images with the corresponding histograms (a) before and (b) after the histogram fitting.

# Extraction of the Eigenpalm and Eigenfinger Features



Average palm and finger images from database.




Eigenpalms and eigenfingers obtained from the database (indicated by the corresponding ordinal numbers): (a) the eigenpalms and (b) the little-finger eigenfingers.

# MATCHING AND FUSION AT THE MATCHING-SCORE LEVEL

- The matching between corresponding feature vectors is based on the Euclidean distance
- The distances are normalized and transformed into similarity measures
  - The frequency distribution of the distances of templates of the same person is calculated.