Biometrics in ID management

Characteristics of Biometrics. Why are they used?

Statistics and detection. What is new in these measurements?

Methods:

- How they work
- Physical phenomena being measured
- How they can fail

System complexity and use logistics

Where Biometrics fits into ID management

We've already looked at:

- Passwords, PINs and other secrets that one knows.
- Tokens, tags, cards and things that you have.

Biometrics represent methods to establish identity based on what you are.

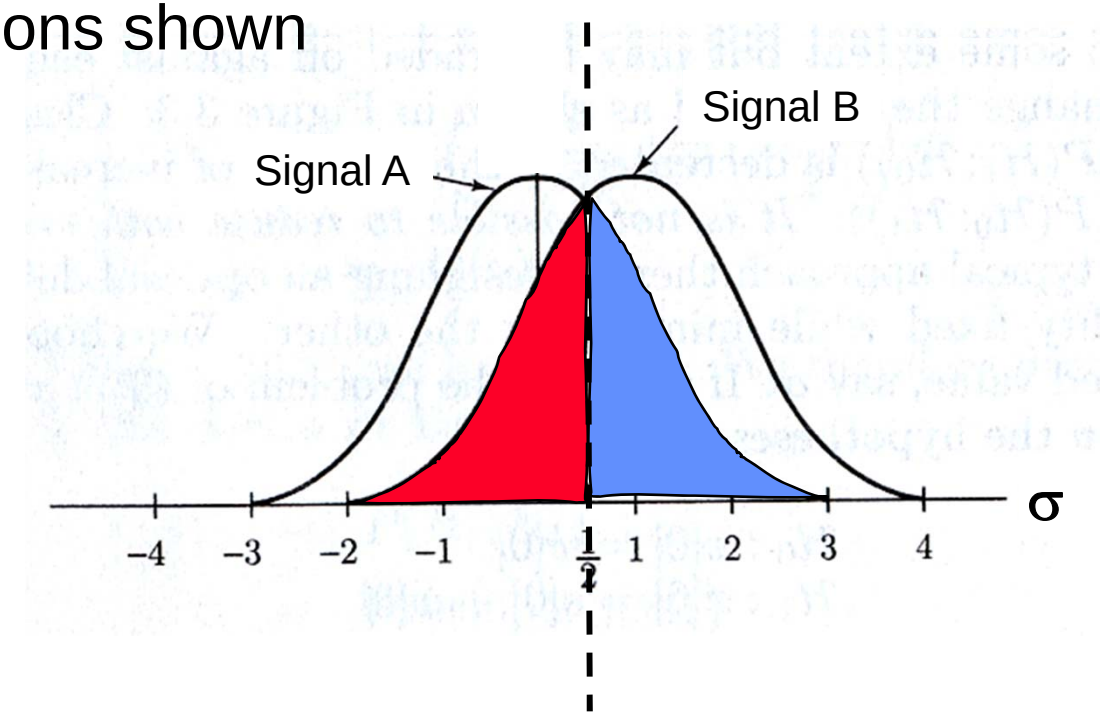
Sensors and systems form a lot of the support for successful biometrics. As before, you are sensing something physically definable. Heat, light, sound, motion, etc. In biometrics, you want to correlate these with a person's identity.

Bio-noise and precision

- Biometrics depends on there being unique variations in physical features between individuals. For example like fingerprints. Biometrics works because of this.
- Often proving uniqueness is a problem.
- For some features, there is significant variation in the same individual. Consider variation in facial images brought on by:
  - aging
  - injury
  - fashion additions
- You can consider this as a form of *bio-noise*.
- In addition, biometric measurements are often not performed under controlled conditions. Consider the effects of lighting or clothing.
- In the practice of pervasive computing the context and conditions under which measurements are made will vary.
- These will have an impact on your measurement precision.

Precision and Pe revisited

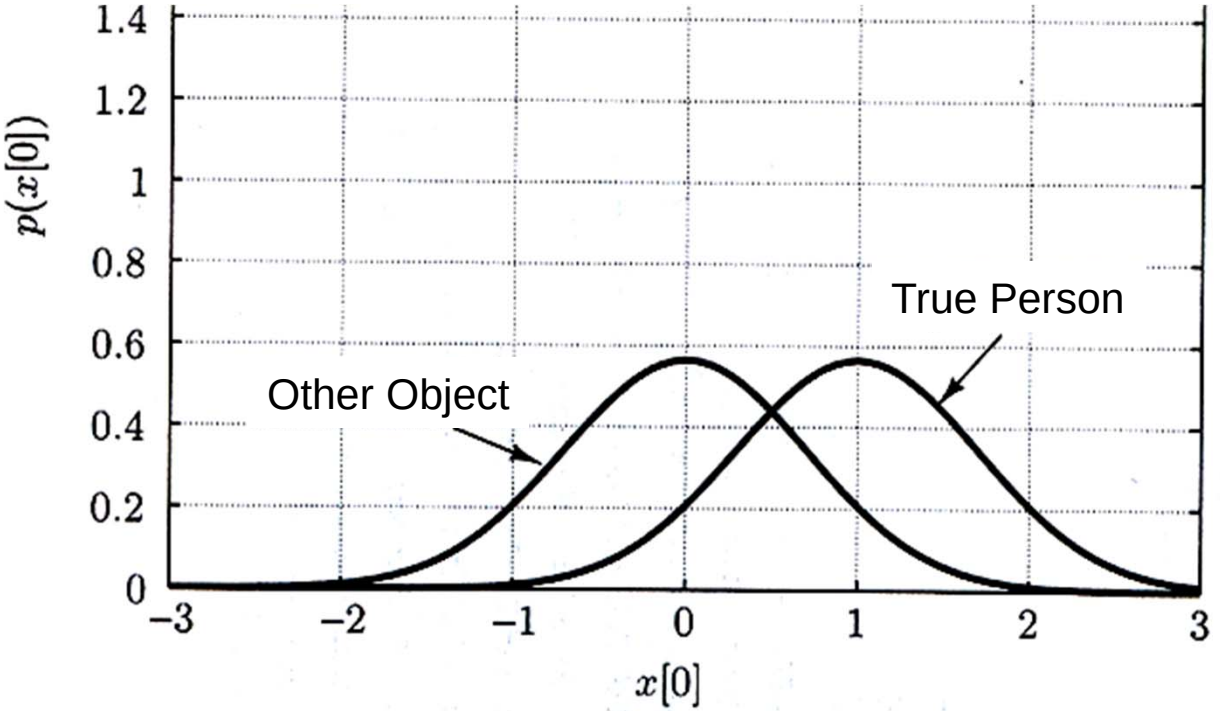
One new consideration is how to deal with the uncertainty caused by the variation in biometric measurement. As an example, remember our previous data from measuring two equally likely values. After measuring them repeatedly, we ended up with the two distributions shown



Because we decided that the outcome of any measurement could be from signal A or signal B with equal likelihood, we made our decision criteria to result in equal Pe for guessing wrong in the event of either signal A or signal B.

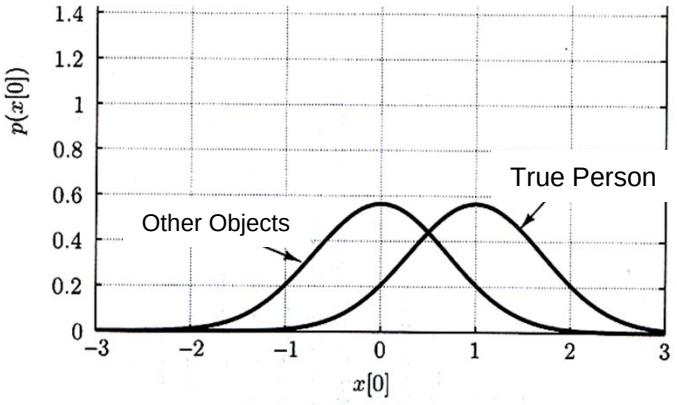
Precision and Pe revisited

But, what if we have the following scenario: Your application is one where you have to verify the identity of a person presenting themselves to your biometric measurement system. The person claims to be a particular person, and you have to decide if that is true. You make two sets of measurements. One of the 'true person' and the other of 'other object'. The measurement distributions are shown below.



Graph from: Steven M. Kay, *Fundamentals of Statistical Signal Processing*, 1998, Prentice-Hall, ISBN 0-13-504135-X

False Accept, False Reject



What you do to deal with the uncertainty depends on the consequences of guessing wrong.

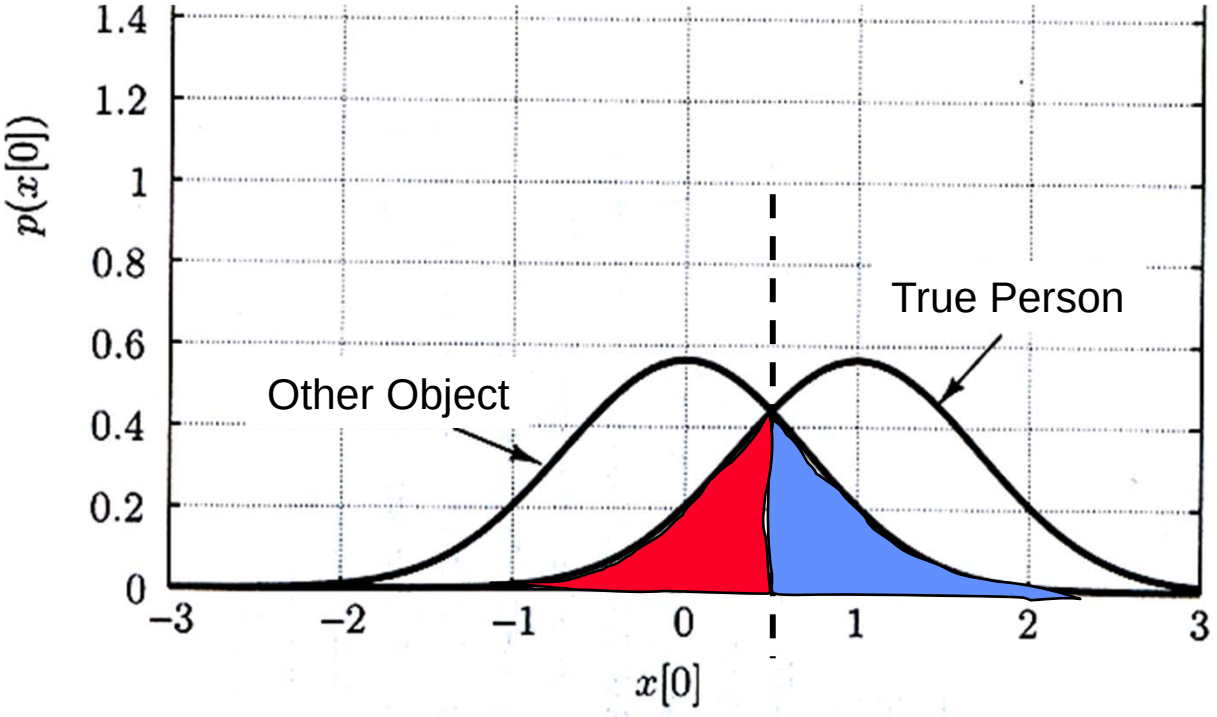
If we guess ‘true person’ when in fact a measurement was due to ‘other object’, that is a *False Accept*.

If we guess ‘other object’ when in fact a measurement was due to the ‘true person’, that is a *False Reject*.

Often you’ll see in biometric ID system literature a reference to the False Acceptance Rate or the False Rejection Rate. These are just related to the Pe.

Example: False Accept, False Reject

You can control this by changing how you decide. Suppose we decide as we have in previous examples as shown below:

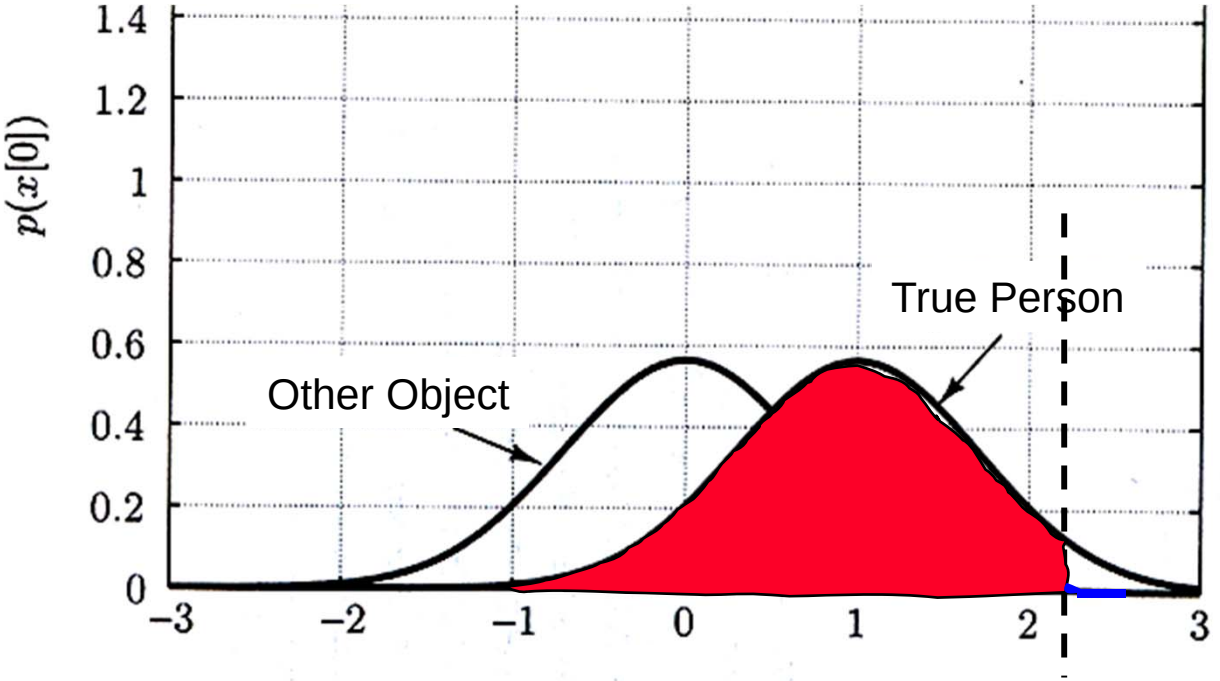


The blue area is the Pe resulting in a False Accept. The red is the Pe resulting in a False Reject. In this case they are equal.

$$\mathbf{P_{False\_Reject} = P_{False\_Accept} = \Phi(\infty) - \Phi(0.5) = 0.3085 \text{ or } 30.85\%}$$

Example: False Accept, False Reject

If this is a non-threatening consumer application, a false accept of almost 31% might be OK. If it isn't, move your decision point.

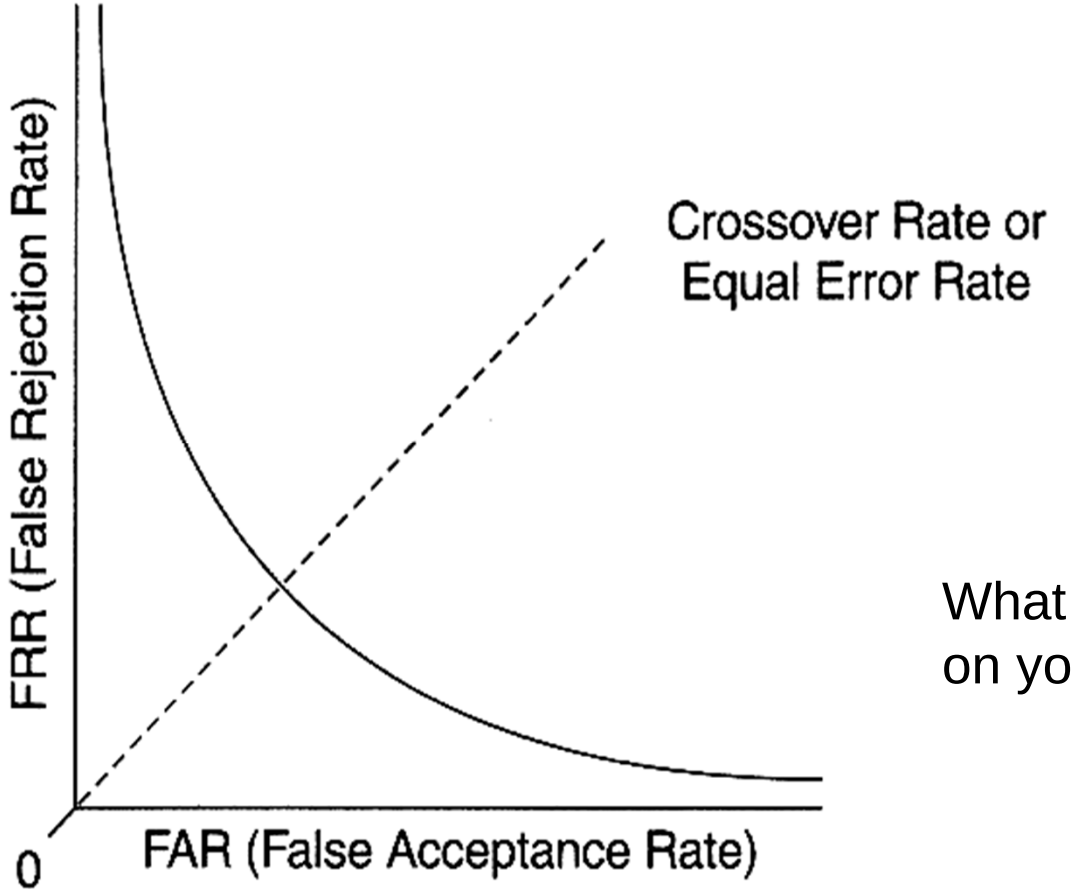


Things are a lot different now (and kind of extreme):

$$\mathbf{P_{False\_Accept} = \Phi(\infty) - \Phi(2.3) = 0.0107 \text{ or } 1.07\%}$$
$$\mathbf{P_{False\_Reject} = \Phi(1.3) = 0.9032 \text{ or } 90.32\%}$$

Accept, reject and application

The point is, you can improve one error but at the expense of the other. You can't improve both simultaneously without using other techniques such as more or better sensors or a different decision algorithm.



What makes sense depends on your application.

Graph from: John D. Woodward Jr. et al. *Biometrics*, 2003, McGraw-Hill/Osborne ISBN0-07-222227-1

Biometric measurement phenomena

Biometric measurement methods are generally based on:

- Temperature
- Pressure
- Light and image processing
- Sound
- Motion dynamics
- Molecular structure
- Electro-magnetics
- Charge

There can be others as well as new methods evolve

Systems using Biometrics must also perform:

In addition to sensing you need these to create a system:

- Enrollment: Creating the reference data or templates that will be used to match users against.
- Signal Processing: Matching the current measured data with previously stored templates. This has to be done in an acceptable time using available resources.
- Decision Processing: Determining admission or rejection based on the signal processing.
- Data security: Both for representation and storage. This can be especially hard if either the signal processing or decision processing is done remotely.

Another concern is cost. It makes no sense to protect an asset with a biometric system costing just as much or more.

Fingerprint Biometrics

Fingerprint systems work by analyzing characteristics such as:
The pattern of the image ridges, valleys, loops and whorls.
The *minutiae*, such as ridge endings and bifurcations.

FBI Image Size Requirements:
(Eight bits/pixel)

- Rolled Impressions:
  - Width: 1.6 inches 800 pixels
  - Height: 1.5 inches 750 pixels
- Plain Thumb:
  - Width: 1.0 inch 500 pixels
  - Height: 2.0 inch 1000 pixels
- Plain 4 fingers
  - Width: 3.2 inches 1600 pixels
  - Height: 2.0 inches 1000 pixels
- Expected 30 to 40 minutiae/finger

From Image Quality Specification, IAFIS, US FBI

Fingerprint image algorithms

Methods used for image capture are:

- Optical
- Ultrasonic
- Thermal
- Capacitive
- Pressure / mechanical sensing

Steps:

1. Acquire Image and remove background
2. Find and enhance ridge edges, ie 2D FFT
3. Represent the data in some form. There are many.
4. Perform a matching algorithm. There are many.

Enrollment template data

- In fingerprint biometric systems, typically a template of minutiae data is stored rather than the full finger image.
- The template stores location and feature details of the minutiae
- Need only about 4 bytes per minutia. Considerable space savings. This can be further compressed using any lossless method.
- The fingerprint itself cannot be reconstructed from the template. Makes theft or fraudulent use of a fingerprint difficult.

Because the position, orientation and apparatus doing the measurement can change each time a measurement is made the templates will not match exactly. This is one source of noise that is seen as resulting in non-perfect measurement precision. Other major noise sources are the condition of the finger (worn, dirty, injured, etc) and the sensor itself.

Matching Algorithms

There are a number of methods that can be used to match a user measurement against reference data, for example:

1. Image correlation methods based on the loops, whorls, etc
2. Image correlation methods based on minutiae.
3. Spatial methods based on minutiae.

Example: Minutiae spatial matching:

Minutiae data is represented as a function of X,Y location and rotation for both the reference template and current measurement.

$$T = \{m_1, m_2, ..., m_m\} \text{ where } m_i = \{x_i, y_i, \theta_i\} \text{ for } i = 1..m$$

$$I = \{m'_1, m'_2, ..., m'_n\} \text{ where } m'_j = \{x'_j, y'_j, \theta'_j\} \text{ for } j = 1..n$$

Matching Algorithms

In this example, minutiae m'_j in I and m_i in T are said to be matching if the *spatial distance* (sd) between them is less than some amount and the *direction difference* (dd) is smaller than some angle.

$$sd\left(m'_j, m_i\right) = \sqrt{\left(x'_j - x_i\right)^2 + \left(y'_j - y_i\right)^2} \quad \leq \quad \text{min set distance}$$

$$dd\left(m'_j, m_i\right) = \min\left(\left|\theta'_j - \theta_i\right|,\ 360^o - \left|\theta'_j - \theta_i\right| \right) \quad \leq \quad \text{min set angle}$$

The goal is to maximize the number of “matches” for each minutiae, where a match is a measurement less than the minimum set distance or angle.

Matching Algorithms

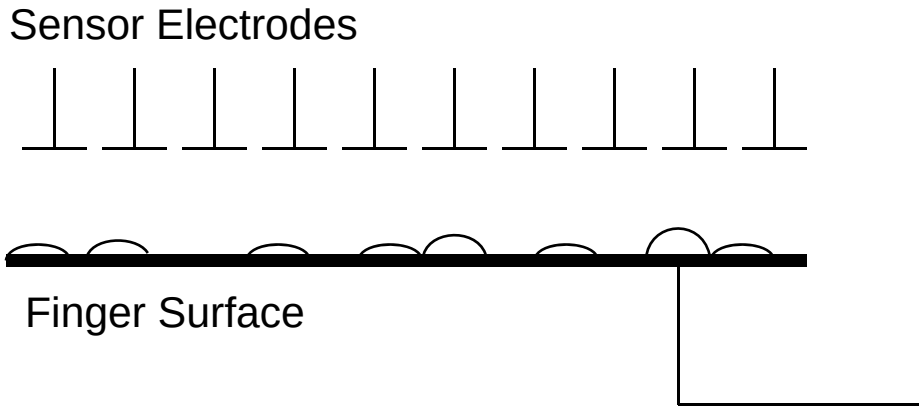
Examples of things that will complicate the matching process are:

1. Image scaling: Different optics, sensors and other conditions will affect how the image is scaled or sized.
2. Geometric transformations: The image may have to be rotated or translated. This needs to happen without adding new noise in the form of image distortions.
3. Large numbers of minutiae that could have more than 1 matching minutiae. As the potential data set goes up, the computational complexity can get very large, potentially exponential with the number of minutiae.

Optical sensors

- Sensors are a type of digital camera. The system has to include an illumination source as well.
- Digital camera technology is a commodity. Sensors are easily obtained.
- Required resolution and pixel depth are easy.
- Susceptible to dirt or other contaminants.
- Latent print can be left on the sensing plate that can be stolen or re-utilized to gain access to an asset.
- Can’t read though gloves or excessive dirt on fingers.
- Without extra “liveness” sensors, can be fooled by images or imitations.

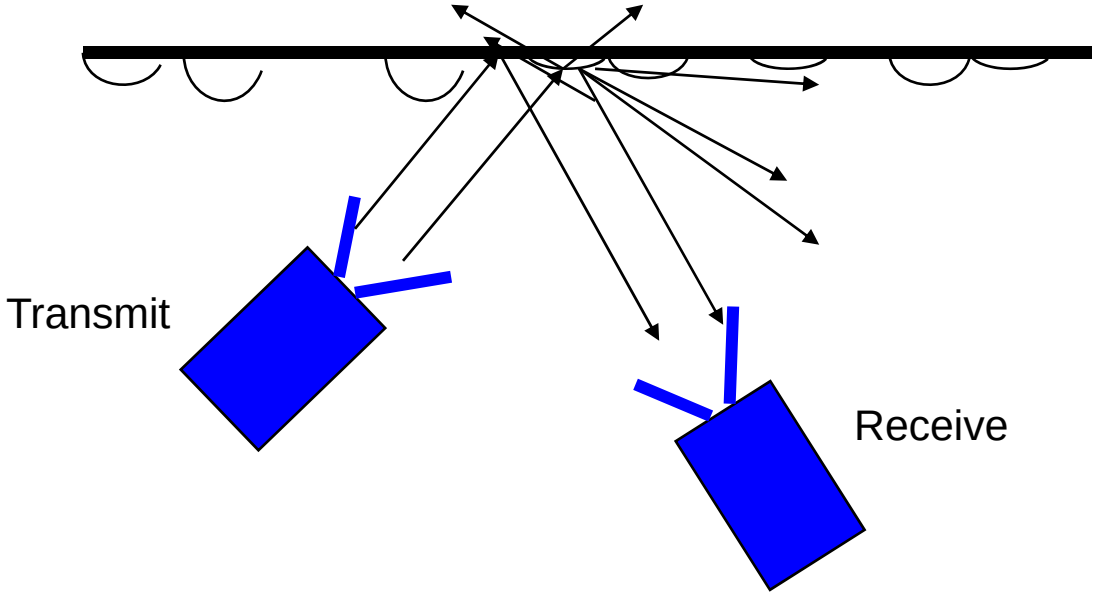
Capacitive Sensing



- Fingertip is placed against an array of charge sensitive elements.
- Variations in the dielectric between a ridge and no ridge look like differences in local capacitance.
- Because q=CV you can extract an image of the ridges.
- Susceptible to dirt or other contaminants on sensor.
- Possible to read though thin gloves or thin dirt on fingers.
- Without extra liveness sensors, can be fooled by imitations.

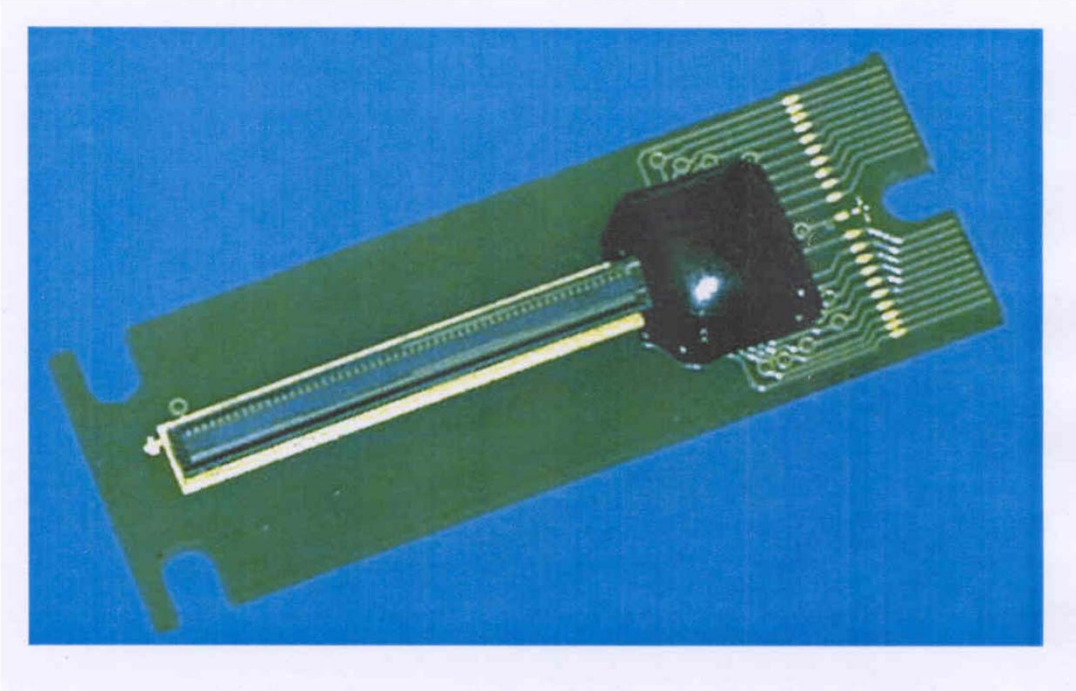
Sensor image from: http://www.st.com/stonline/products/support/touchip/index.htm

Ultrasonic sensors



- Ultrasound energy at several wavelengths is beamed at fingertip.
- Variations in the energy scattering between a ridge and no ridge modulates the received ultrasound.
- Possible to read though gloves, dirt or other contaminants.
- Without extra liveness sensors, can be fooled by imitations.

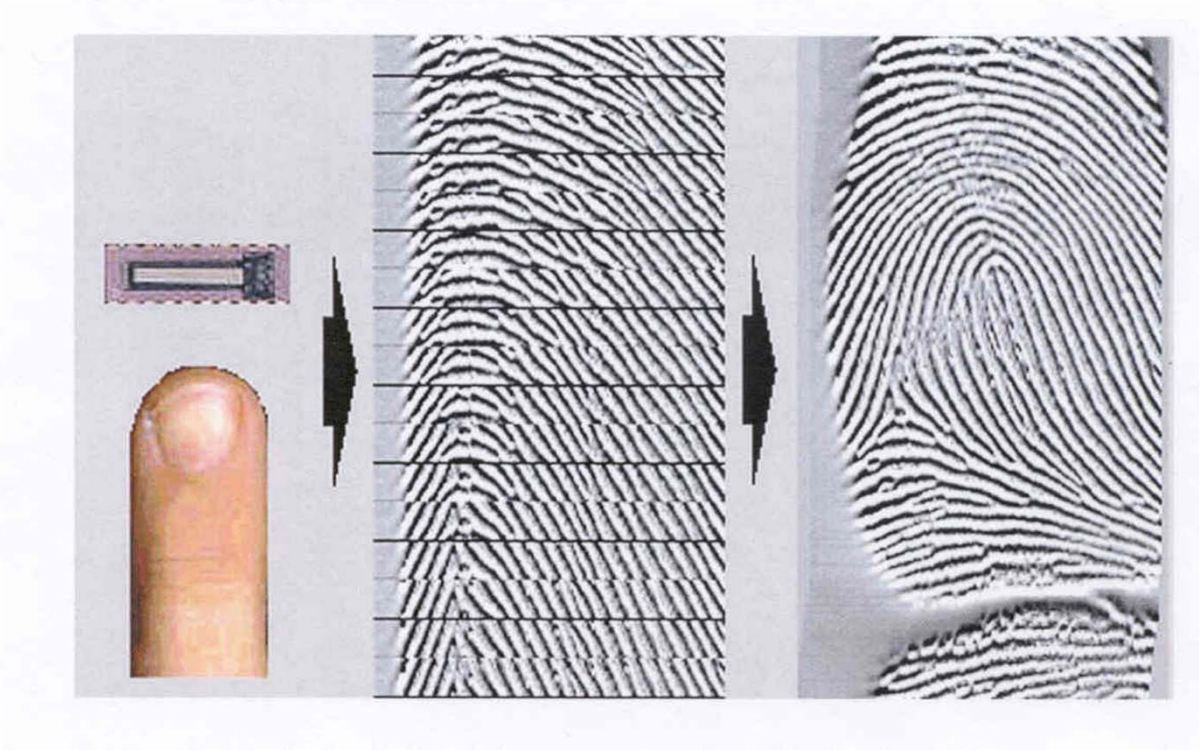
Thermal sensors



- Fingertip is placed against an array of heat sensitive elements.
- Ridges are warmer, as they are in contact with the sensor.
- Non-ridges are not, so they are cooler.
- Can’t press too hard, or leave finger in contact too long.
- Possible to read though thin gloves or thin dirt on fingers.
- Also works as a kind of liveness sensor, so it’s harder to fool with imitations.

Sensor Image: http://www.atmel.com/dyn/resources/prod_documents/FingerChip_WhitePaper_11_12_02.pdf

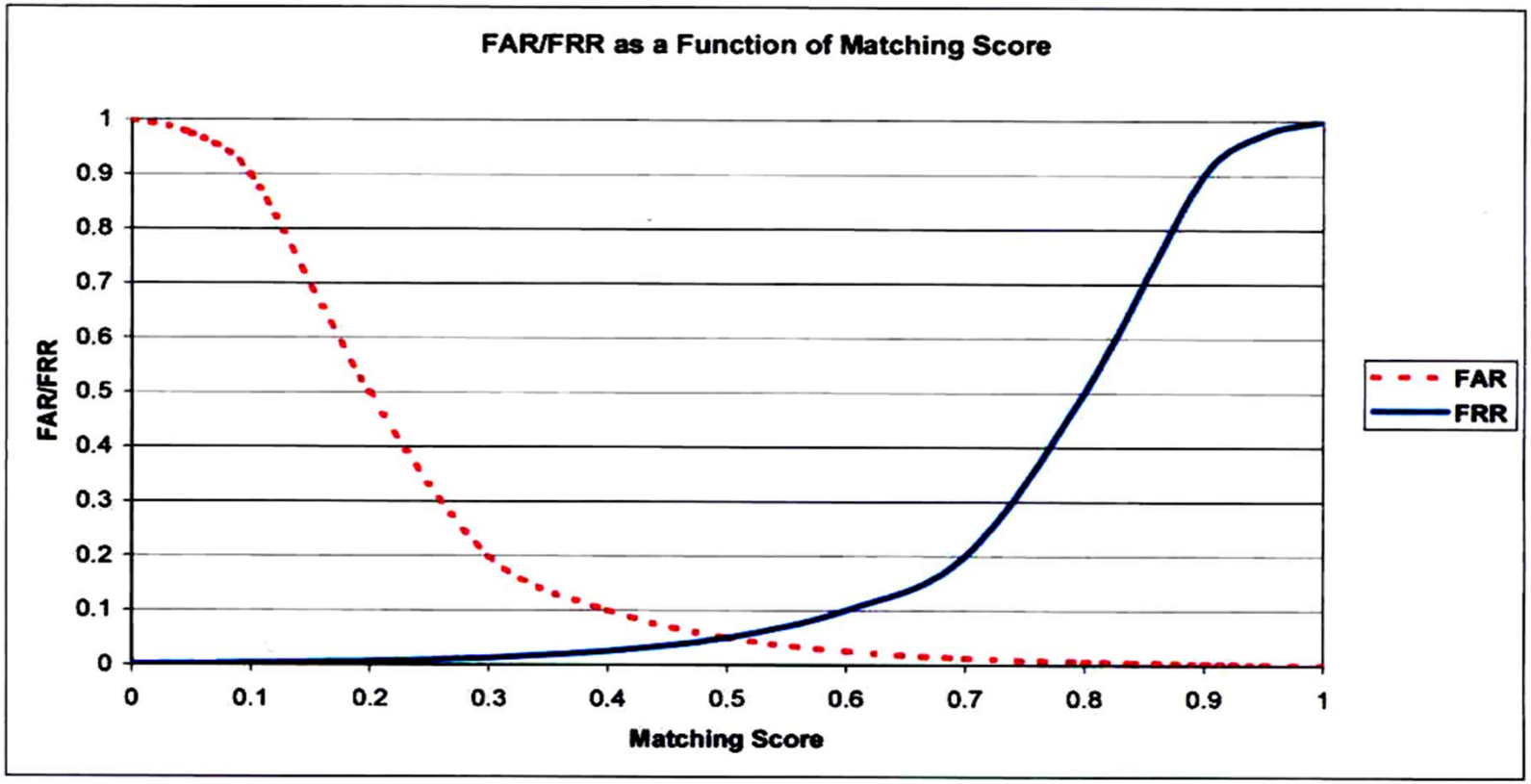
Thermal sensor example



This is an example of a sensor that is formed as a linear array. You scan your finger by moving it across the sensor. Data is clocked out scan by scan and re-assembled. There are cost advantages to this but the algorithms and use can be more complex.

Image from: http://www.atmel.com/dyn/resources/prod_documents/FingerChip_WhitePaper_11_12_02.pdf

Example: Decision Criteria



- Template matching software generates a score that is used to decide if the 'true person' is being measured.
- These curves give an example of distributions for false acceptance and false rejection.
- Software can ask for more fingers if the matching score is low.

Data from: http://www.atmel.com/dyn/resources/prod_documents/FingerChip_WhitePaper_11_12_02.pdf

Example: PDA with thermal fingerprint sensor

The IPAQ 5455 is an example of an information device where ID mgmt is provided by a combination of password and fingerprint methods.

Example: User feedback for fingerprint protected PDA

Features I didn't like or didn't need.

Biometrics.

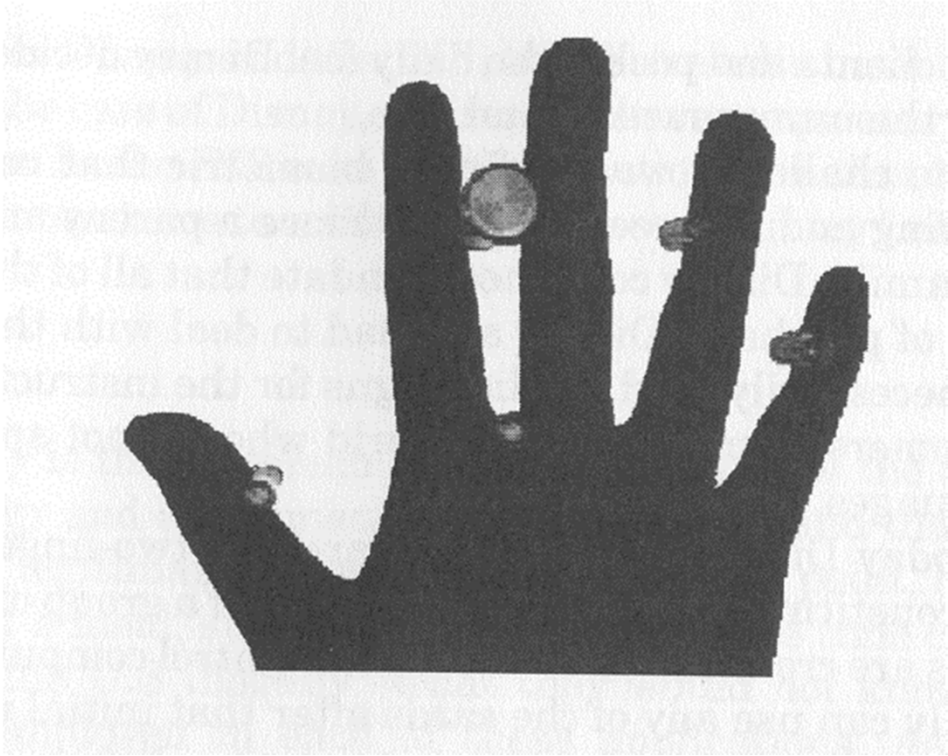
This is some pretty advanced security technology that is finally making its way into consumer applications and devices. The most popular applications of this technology are in the military, law enforcement, secure corporate networks and healthcare / hospitals. There are not many consumer needs for such high security measures. This is how it works. You scan in one or two fingerprints into the device, for each authorized user of the device. You can then set the desired frequency of verifying such access. (Once a day, once an hour or every time the unit is powered on, etc.) There are 3 settings for accuracy of fingerprint scans. On the highest setting, you may need to scan your finger more than once to access the device. This technology is not a theft deterrent, but it will prevent a stranger from accessing your data. (A strong password would do the same) BESIDES, after too many incorrect logon attempts, the IPaq prompts you to do a full reset that deletes any user data. (Which means that a thief will get a fresh clean IPaq)

From: http://hardwarecentral.dealtime.com/xPR-Compaq_iPAQ_h5455~RD-91917225604

Handprint biometrics

- Handprint is the second only to fingerprint as the most used biometric method.
- It is based entirely on hand geometry. Individual features on hands are not resolved.
- Simple optical based sensing is good enough. The images are just black and white.
- To help with background removal, scaling and translating, the hand to be measured is place on a special platform in a forced orientation.
- Like fingerprint, it requires the cooperation of the user.
- Can be combined with liveness testing for added security.

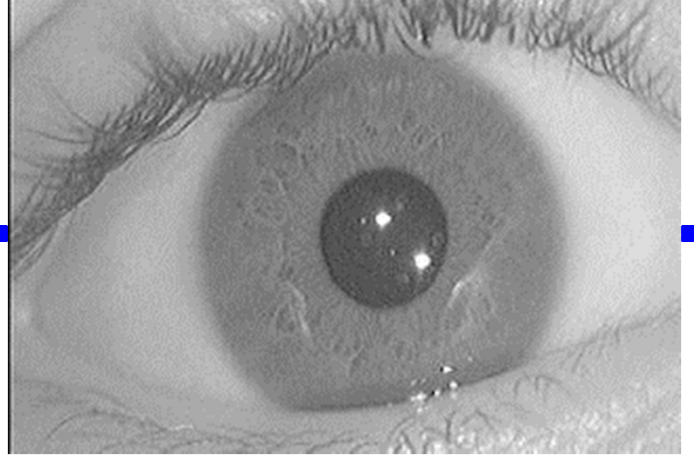
Handprint biometrics



Note the pins used to align the hand.
Measured parameters are length, width, thickness, shape and surface area of the imaged 4 fingers (thumb isn't used).
To measure finger thickness, some systems use images both of the top and side of the hand.
Reference templates need up to about 20 bytes to represent hands.

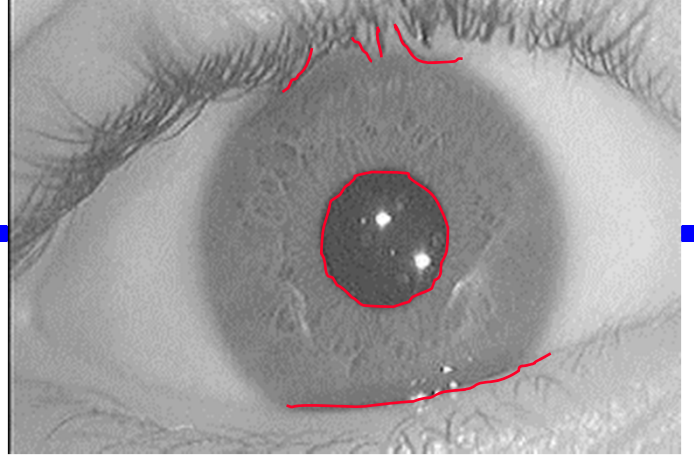
Image from: John D. Woodward Jr. et al. *Biometrics*, 2003, McGraw-Hill/Osborne ISBN0-07-222227-1

Iris scans



- Very related to fingerprint biometrics, iris scanning uses features represented in a person's iris.
- Arching ligaments, ridges, furrows, rings, freckles, zigzag collarettes are typical features that collectively are considered as texture.
- Focal distance is about 3 to 7 inches. User stands close to the camera.
- Also like finger information, the iris feature information is difference from the left and right eye.
- Color information is not used.
- Liveness testing is easy by just looking for eye motion, or adjusting the amount of light used and looking for iris response.
- It is invasive. Some users find exposing their eye to the sensor to be somewhat unnerving.

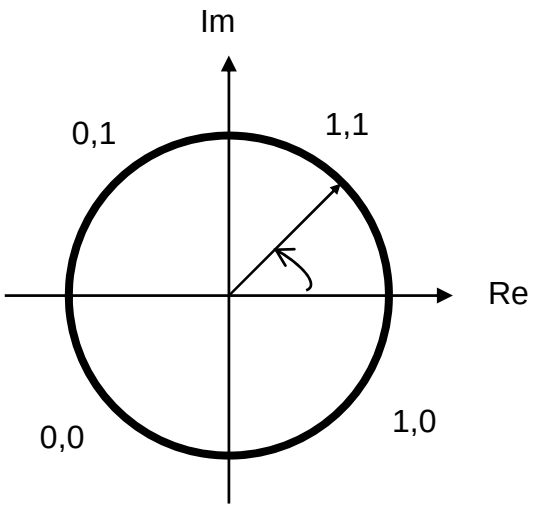
Iris matching algorithms



1. Find the iris. This is related to background extraction in fingerprint sensing. Various filters can be used to find the iris outline.
2. Advanced techniques will account for image noise such as eyelashes, eyelids or foreign objects.
3. Extract the iris as a bitmap image.
4. Process all X,Y points in the image through a filter that extracts image texture information.
   - Orientation
   - Spatial frequency
   - 2D position

Iris matching algorithms

5. An example described in the literature* is a wavelet demodulation based on 2D Gabor filters. These filters take as inputs:
   - Starting position in the image
   - Image size
   - Spatial frequency



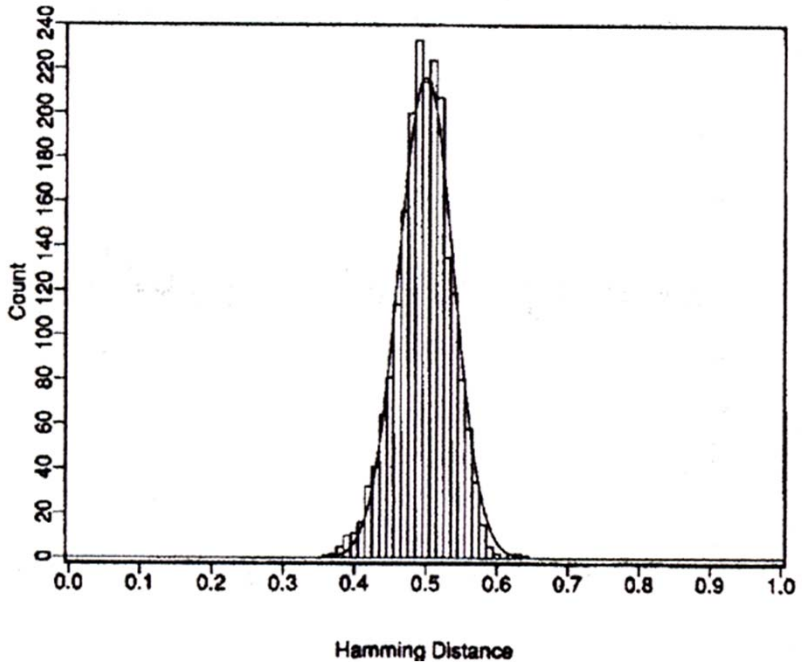
6. The output transformation describes a phasor in the complex plane.
   - Only the phase data is used. To do this the complex plane is broken into 4 quadrants, and encoded with 2 bits.
   - The transform is run over many sizes, frequencies and orientations to generate the template data or “Iriscode”.

*Daugman, *High confidence Visual Recognition of Persons by a Test of Statistical Independence*, IEEE Trans. on Pattern Analysis and Machine Intellegence, Vol 15, No. 11, Nov 1993 pp 1148-1161

Iris matching algorithms

7. The templates are then compared bit by bit using Hamming Distance (HD) as the difference metric. HD is just the fraction of bits out of the entire template that don't match:

$$\mathrm{HD} = \frac{1}{x}\sum_{y=1}^{x} T_y \otimes I_y \quad \text{where } \mathrm{x} = \#\,\text{template bits}$$



8. Your admission or rejection decision is then made based on the Hamming Distance. The example* here shows the distribution of HDs from over 2000 different pairs of irises. From this a reasonable admission decision might be HD < 0.3.

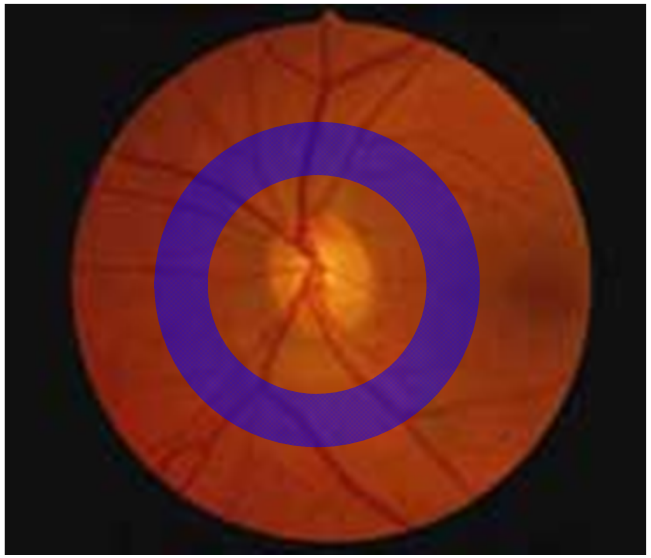
*Daugman, *High confidence Visual Recognition of Persons by a Test of Statistical Independence*, IEEE Trans. on Pattern Analysis and Machine Intellegence, Vol 15, No. 11, Nov 1993 pp 1148-1161

Retinal scans

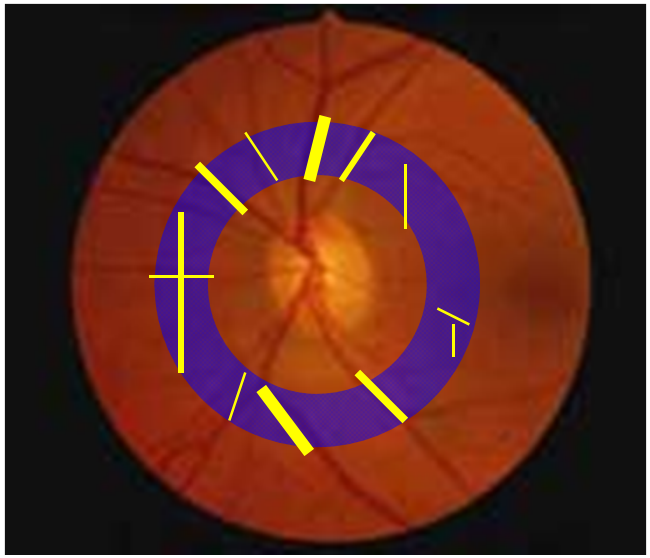
- Related mostly to bar codes, retinal scanning uses the vascular pattern on the retina.
- Near IR illumination is used as the retina appears transparent to these wavelengths. The vessels are readily imaged.
- It exploits the anatomy of the retina. Blood vessels radiate outward from the optic nerve area. The pattern they make are thought to be unique and unchanging across individuals.
- Also like iris information, the vessel feature information is difference from the left and right eye.
- Can be combined with other sensors or algorithms to measure liveness.
- Like iris scans, it is invasive. Light is being directly beamed into the user's eye. Some find this disturbing.
- Some feel that retinal imaging can reveal certain medical conditions, resulting in medical information privacy issues.

Retinal scans

1. Image the Retina



2. Find area around optic nerve.



3. Find blood vessels

4. Flatten it out preserving size spatial relation and direction of blood vessels. 2D Bar code! Templates typically 100 bytes.

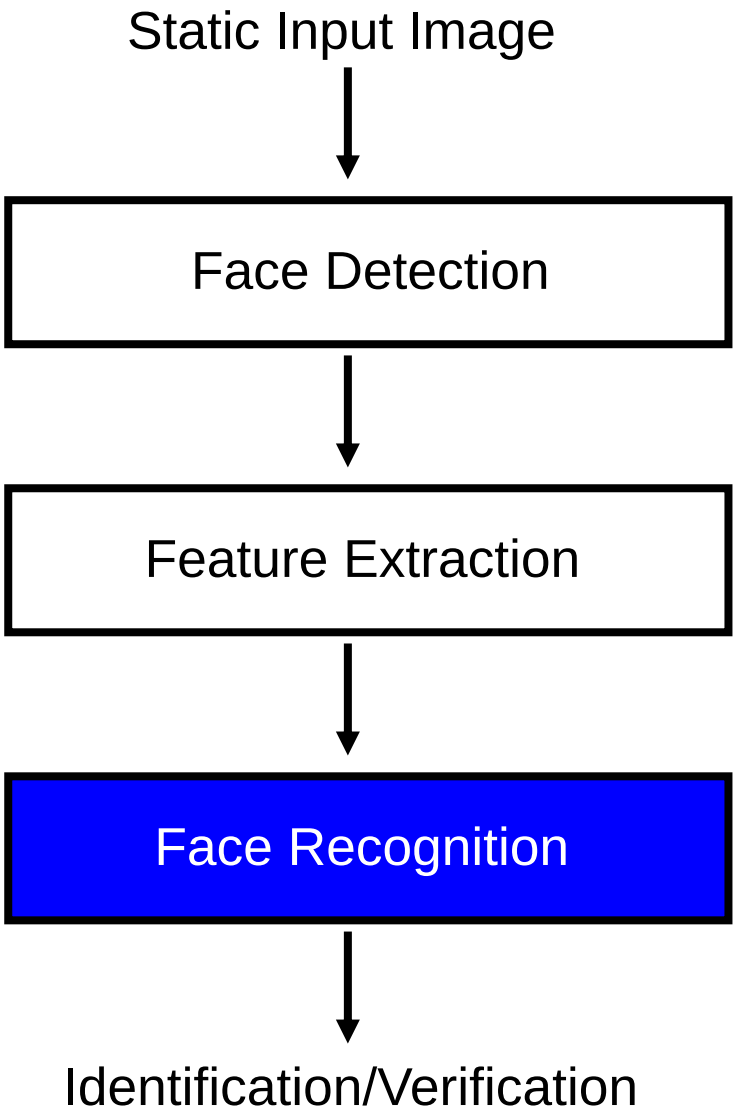
To see examples of commercial retinal scanning devices, see:
http://www.retinaltech.com

Face Recognition

- Face recognition can be opportunistic. Many methods do not require a cooperative subject.
- All use optical sensors, although what is imaged can vary.
- Most are 2D intensity maps, but other components such as color, gender, heat and motion have been proposed.
- Most algorithms seek to minimize the effects of facial changes on the matching process using various estimation methods.
- Comparisons between methods are possible using standard data bases of face images.
- Very active area of research.

An excellent overview:
W. Zhao, *Face Recognition: A Literature Survey*
http://citeseer.ist.psu.edu/cache/papers/cs/17462/http:zSzzSzwww.cfar.umd.eduzSzftpzSzTRszSz
FaceSurvey.pdf/zhao00face.pdf/

Typical face recognition design



Other applications of face detection:
- Tracking
- Pose estimation
- Compression
- HCI systems

Other applications of feature extraction:
- Facial feature tracking/animation
- Emotion recognition
- Gaze estimation (remember distraction!)
- HCI systems

Most used and emerging methods

As examples, we will consider:

Methods based on static images

- Correlation of local features
- Methods based on Principle Component Analysis (Eigenfaces)
- Methods based on video or image sequences

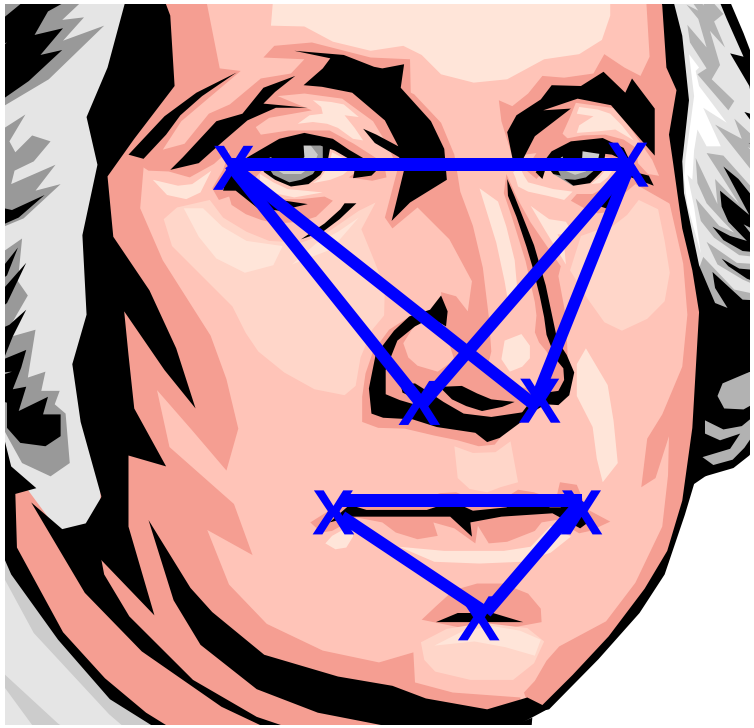
There are many others in addition to these, and they can be combined into hybrid approaches as well.

Local feature analysis based on direct measurements

Identification performs correlation of:

1. Heights and width of head.
2. Width of neck and shoulders.
3. Distance between eyes
4. Distance from top of head to eyes
5. Distance from eyes to nose
6. Distance from eyes to mouth

Local feature analysis based on geometric measurements



Identification performs correlation of distances and angles between points such as:

1. Eye corners
2. Mouth extremeties
3. Nostrils
4. Chin top

The analysis takes into account ratios of distances, areas and angles. This allows some degree of compensation between varying sizes of pictures.

Advantages/disadvantages with local feature analysis

+ Somewhat illumination robust. As long as you can find the features, the level of illumination doesn't matter.
– Very pose dependant. Head tilt, side views, etc.
– Dependant on facial expression.
– Problems with image size/scaling.
  + Geometrical analysis is somewhat better here.
– Dependant on changes in hair, clothes, glasses or other accessories.

Eigenfaces

- Eigenfaces is one of the most referenced methods for face recognition. See reference below.
- It uses optical sensing (a camera) to acquire an image of a face. Thus, a face signal can be considered as a bunch of intensity values mapped in a 2D space.
- The problem boils down to trying to take an image of a person, and determine whether or not it matches an image in a data base of a person with a known identity.
- Sounds straight forward. But it isn't.

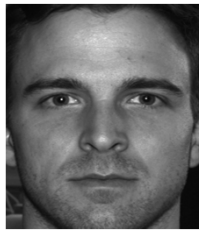
M. Turk and A. Pentland, *Eigenfaces for Recognition*
Journal of Cognitive Neuroscience, Vol 3, No. 1 1991

What Eigenfaces Does

- From the previous examples, there is potentially an enormous amount of data, and relationships between the data that has to be accounted for. Orientation, distance, lighting, expression, etc. Lots of dimensions.
- Eigenfaces attempts to deal with this data space complexity in a couple of ways:
- Constrain what the camera sees, ie it has to be centered in the 2D frame and you need to know what signals in the frame correspond to the eyes.
- Do something to reduce the size of the data space. Rather than searching and correlating over zillions of points in 2D intensity maps, is there some way to represent faces where the signals are expressed relative to something significant about the face? Hopefully, whatever those significant things are, there won't be very many of them (a low degree of dimensionality). Principle Component Analysis is used here.

M. Turk and A. Pentland, *Eigenfaces for Recognition*
Journal of Cognitive Neuroscience, Vol 3, No. 1 1991

Start by collecting your reference, or ‘training’ set of faces


- Reposition the 2D data in each training image to become a vector of pixel values:

N by N image

Reordered into t
N^2 by 1 vector.
Call this vector Γ

Normalize the training face data

- Do this by computing an ‘average face vector’:

$$\Psi = \frac{1}{\mathrm{M}} \sum_{i=1}^{M} \Gamma_i \text{ where } \Gamma_\mathrm{i} \text{ is the i}^{\text{th}} \text{ training face vector}$$

- Use this to normalize the training face vectors’:

$$\Phi_i = \Gamma_i - \Psi$$

Use the normalized training face vectors to form a covariance matrix

$$C = \frac{1}{M}\sum_{n=1}^{M}\Phi_n\Phi_n^{T} = \mathrm{AA}^{t} \quad (\mathrm{N}^2 \text{ by } \mathrm{N}^2 \text{ matrix})$$

$$\text{and where } \mathrm{A} = \left[\Phi_1\Phi_2\cdots\Phi_M\right]$$

- This is the same way you would calculate any covariance matrix. It's a big matrix, having N^2 dimensions formed from M training faces.
- The matrix shows you what strongly varies from face to face.

Compute the Eigenvectors and Eigenvalues

- For such a huge covariance matrix, this is challenging to do, but there are numerical methods to do this.

- This will give N^2 eigenvectors and eigenvalues.

- Choose and keep the eigenvectors with the largest eigenvalues (the things where the co-variance is most strong).
- These are the *Principle Components.* These will be used to form a new data space of lower dimensionality than N^2. Lets say you keep the eigenvectors corresponding to the K largest eigenvalues. Lets call each eigenvector u_j, where j ranges from 1 to K and where $K << N^2$.

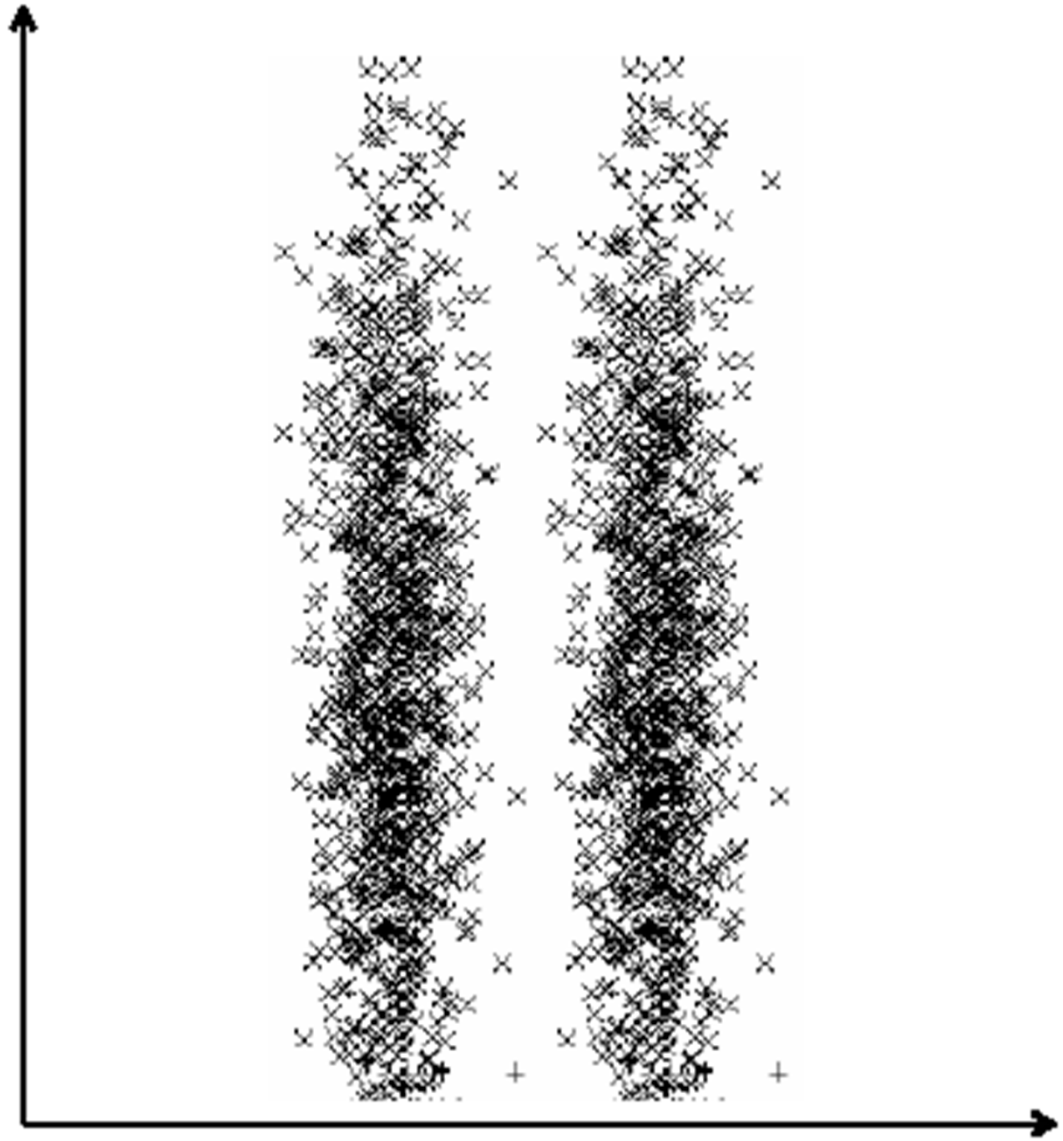
Now, map your training faces into this new lower dimensional space

$$\Omega_i = \sum_{j=1}^{K} w_j u_j \text{ where } w_j = u_j{}^t \Phi_i$$

- This means that each normalized training face now is represented as a linear combination of the K eigenvectors:

$$\Omega_i = \begin{bmatrix} w_1{}^i \\ w_2{}^i \\ \vdots \\ w_K{}^i \end{bmatrix} \text{ where } i = 1, 2, ..., \text{M training faces}$$

They look something like this



Using this new eigenspace

- Capture an unknown face and form Γ
- Normalize with the average face vector: $\Phi = \Gamma - \Psi$
- Use the eigenvectors to project this unknown face into the new eigenspace: $\Omega = \sum_{i=1}^{K} u_i u_i^t \Phi$
- Find the eigenvector with the minimum spatial distance between itself and this unknown face vector:

$$d = \min \left\| \Omega - \Omega_i \right\| \text{ where } i = 1, \ldots, M \text{ training faces}$$

- If d < threshold, then Γ is recognized as face i from the training set.

Eigenfaces

- The eigenvectors u_i are referred to in the literature as *Eigenfaces*. There's nothing magic about them. They're just eigenvectors of the N2 by N2 matrix formed by the training face vectors.
- Because they are vectors formed from that space, they tend to have a ghostly face-like appearance if they are reformed into a 2D image. Thus the name Eigenfaces.