Biometric Systems

Lukas Bach - lbach@outlook.de - lukasbach.com

1 Introduction

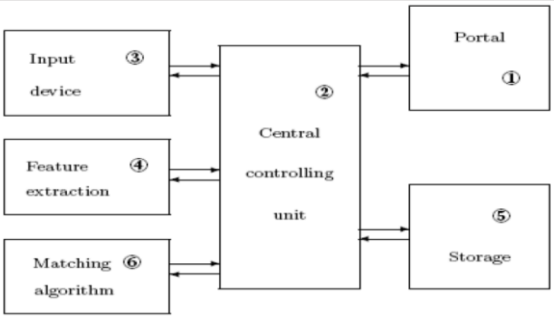
1.1 Background

1.1.1 Kinds of Biometrics

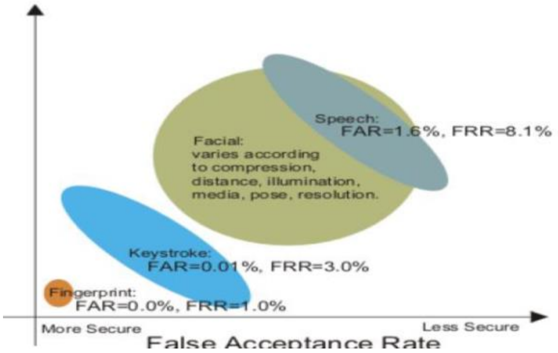
- Physical Biometrics
  - Fingerprints, Iris, Hand geometry, face, vein pattern, Retinal Scanning, Ear Shape
- Behavioral Biometrics
  - Voice, Signature, Typing Pattern, Gait recognition (how you walk), heart rate analysis

Specific kinds covered on slides in more detail.

1.1.2 Biometric System Components



- Enrollment: Create template for user, standardized procedure
- Identity Verification: Am I who I claim I am? Compare only with stored template.
- Identification: Who am I? Compare with all templates.
- Matching threshold needs to be learnt.
- Multi-modal Biometrics: Several combines biometrics.
- Cross-modal Biometrics: Match different sensor modalities, e.g. RGB, thermal BW, MRI-Xray.
- Performance measurements:
  - Usability (Failure to Enroll, Failure to Acquire)
    - E.g. sticky fingers, glasses, scars…
  - Performance metrics (False Acceptance Rate, False Rejection Rate)
    - FAR: Accept unregistered user or mistaking one user with another. Influence security.
    - FRR: Rejecting registered user. Influence usability.



2 Pattern Recognition and Deep Learning

2.1 Feature extraction

Extract feature descriptors that are

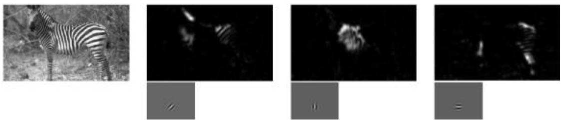
- Discriminative
- Robust against image transformations
- Robust against object transformations, viewpoints, occlusions
- Efficient to compute

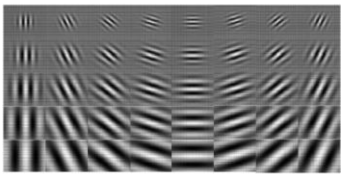
2.2 Pattern Recognition

2.2.1 Gabor Filters

Search for lines in different scales and orientations.

Gabor Wavelet Transform: Perform convolution with gabor filter kernels.





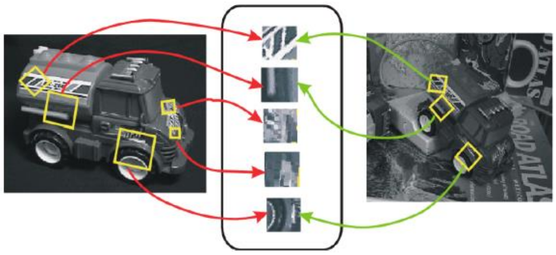
2.2.2 Edge Histogram Descriptor

Partition image into regions, for each region compute edge histogram.

2.2.3 Local Descriptors

Find local feature coordinates that are invariant to translation, rotation and scale.

Not chosen by random, but by relevant interest points (e.g. corners, …)



Approach (2-17):

1. Find set of distinctive key-points
2. Define a region around each point
3. Extract and normalize content
4. Compute local descriptor from normalized region

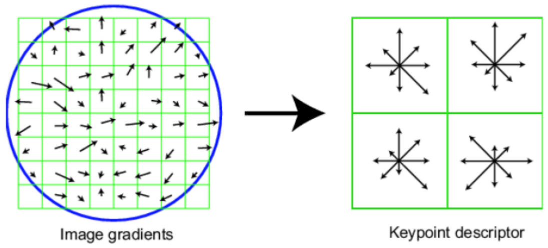
5. Match local descriptors

Local Descriptors SIFT, SURF

2.2.3.1 SIFT

Keypoints are found by blurring the image (and some other steps) and detecting outlier pixels in regards to their neighborhood.

- Divide area around keypoint in 4x4 subregions
- Build orientation histogram with 8 bins per subregion
- Results in a $4x4x8$ vector
- Normalize vector to unit length (invariance to multiplicative changes in lightning)



SIFT is invariant to scale and orientation and robust to illumination changes, noise and minor view-point changes

2.2.3.2 SURF

Fast approximation of SIFT.

2.2.4 Curse of dimensionality

Too many parameters for efficient sampling.

Reduction solutions: Principal Component Analysis (PCA, 2-26 to 2-33), Linear Discriminant Analysis (LDA, better than PCA but requires labels).

2.2.4.1 PCA

Find axis of most dispersion by finding covariance in data, denoted by covariance matrix (how much does every point vary with respect to itself and other points in that dimension). Covariance is estimated by subtracting mean of each point, then taking dot-product.

$$cov(V)_{i,j} = \frac{1}{n-1} VV^T$$

Goal of PCA: Make covariance as diagonal as possible, i.e. increase variance and reduce covariance. For n samples $v_i, |v_i| = d$:

$$\begin{pmatrix} var(v_1) & \dots & cov(v_1, v_n) \\ \vdots & \dots & \vdots \\ cov(v_n, v_1) & \dots & var(v_n) \end{pmatrix}$$

PCA Steps: 02-33

- Put mean subtracted data in a matrix $A = [n \times d]$
- Find covariance matrix $Cov = [d \times d]$
- Find eigenvectors U, S, V using SVD on Cov (Singular Value Decomposition)
- Set of all base vectors gives eigenvector matrix $U = [d \times d]$ (sort descending by eigen values. Note that eigen values correlates to variance of that dimension)
- Select $U = [d \times k]$ (cap first k values)
- Project d-dimensional vector x on to k-dim using $x' = U^T x,\ [k \times 1] = [k \times d] \cdot [d \times 1]$

2.3 Classification

Bayesian Classification: TODO 2-38ff.

Gaussian Classification: TODO 2-41ff.

- Easier estimation, only have to estimate μ and σ

Gaussian Mixture Models: TODO 2-46ff.

- Sum of multiple weighted gaussians
- Estimate multiple $\mu \& \sigma$ as well as weights with Expectation Maximization, repeat:
  - Expectation: Compute prob that datapoint i belongs to Gaussian j
  - Maximization: Compute new GMM parameters

Expectation Maximization: TODO 2-51.

2.3.1 Taxonomy

- Parametric vs non-parametric
  - Parametric need less training data, but require data to fit the model
  - Non-parametric works well for all types of distributions, but need more data.
- Generative vs discriminative

2.3.2 Instance-based Learning

Learning = Store all examples, Classification = assign target function to new instance. E.g. template-matching or k-nearest neighbor.

Nearest neighbor considerable when lots of training data available and less than 20 dimensions per example. Trains fast and learns complex functions, but is slow at query time and easily fooled by irrelevant attributes.

2.4 Background Deep Learning

See lecture CV or DL.

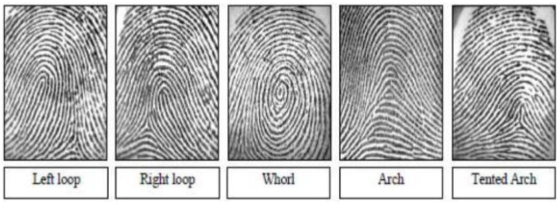
3 Finger Print Recognition

3.1 Background

Latent prints: Oil left behind when touching things.

3.1.1 Fingerprint types

“Singular regions”, where ridge lines assume distinctive shapes.



General ridge patterns: LOOP, WHORL, ARCH.

Type lines: Line surrounding the entire content of the fingerprint (around whorls or loops).

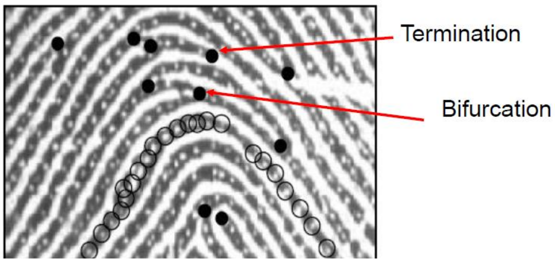
Delta: Triangle structure, see “Left Loop” bottom right.

Core: Pattern center, e.g. center of loop.

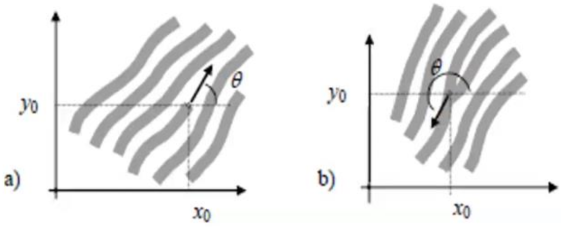
3.2 Feature Extraction

3.2.1 Minutiae

Most prominent ridge characteristics, subdivided into “Ridge termination” and “Ridge bifurcation”.



Minutiae orientation can be extracted and used as features:



Intra-ridge details (sweat pores) can be detected which are very distinctive, but require high-res images.

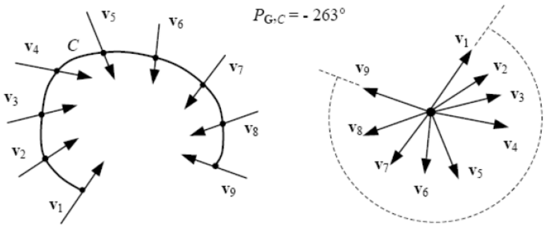
Local Orientation is computed based on gradient phase angles. Robust computation is based on local averaging of gradient magnitudes.

Singularity: fingerprint landmark due its scale, sift and rotation immutability. E.g. deltas, whorls, …

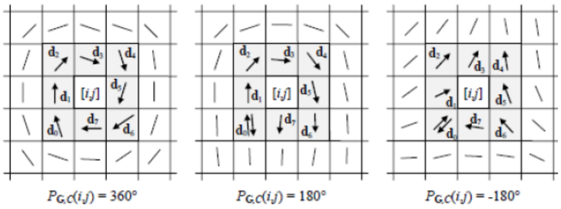
3.2.1.1 Poincare index

Poincare index: Total rotation of vectors in vector field G along curve C which is immersed in G.

Compute by going around the 8 vectors around the center point and summing up the relative angles between each pair of vectors.



$$P_{G,C}(i,j) = \begin{cases} 0° & \text{if } [i,j] \text{ does not belong to any singular region} \\ 360° & \text{if } [i,j] \text{ belongs to a whorl type singular region} \\ 180° & \text{if } [i,j] \text{ belongs to a loop type singular region} \\ -180° & \text{if } [i,j] \text{ belongs to a delta type singular region.} \end{cases}$$



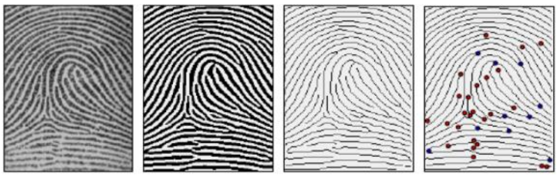
TODO 3-31ff regarding Poincare index, also look into paper

Core Detection: Clashing point of normals of ridges.

Contextual Image filtering used for enhancement, e.g. by using gabor filters.

3.2.1.2 Minutiae extraction

- Enhance, Binarize
    - E.g. gabor filters for enhancement
    - Use local threshold for binarization, global threshold does not work
    - Better binarization: Laplacian operator, set value based on pixel within/outside intensity
- Thinning (morphological operations, Erosion)
- Detect minutiae through pixel-wise computation of crossing number.



Crossing number: See protocol. half the sum of differences between pairs of adjacent pixels in 8-neighborhood

3.3 Matching

Tricky due to displacement, rotation, partial overlap, nonlinear distortion, changing skin condition, noise, feature extraction errors, …

- Correlation-based matching: Compute intensity-based correlation between fingerprint images.
    - Compute cross-correlation. Not reliable.
- Minutiae-based matching: Extract minutiae from two fingerprints, match minutiae pairings.
    - Most successful
    - Match Template and input minutiae sets, each consisting of minutiae, each minutiae consisting of x, y, φ
        - First pair each minutiae points to tuples, based

on spatial and direction difference.

- Ridge feature-based matching: Match local orientation/frequency/shape/texture… of ridges.
- CNN Based approaches (TODO subpart of ridge-based?):

  - Better than classic methods with complex/noisy backgrounds. Include enhancement, segmentation. Also for minutiae-extraction?
  - FingerNET: Unified model for enhancement, orientation estimation, segmentation, minutae detection

Pre-Alignment: Perform before RANSAC, find the core, find average ridge orientation, rotate fingerprint to minimalize orientation differences to template.

3.3.1 Minutiae-based matching

- Try to match input $I = \{m_i'\}$ and template $T = \{m_i\}$.
- Each minutiae is described by position and angle, $m_i = \{x_i, y_i, \theta_i\}$.
- Map input minutiae m_i' to m_i'' via

$$I_{new} = \left(I + \begin{pmatrix}\Delta x\\ \Delta y\end{pmatrix}\right) \cdot R,\ R = \begin{pmatrix}\cos\theta & -sin\theta\\ \sin\theta & \cos\theta\end{pmatrix}$$

- m_i and m_i'' "match", if spatial distance between x_i, x_i' and y_i, y_i' and direction difference between θ_i, θ_i' is smaller than some tolerances.
  - $sd(m_j', m_i) = \sqrt{(x_j' - x_i)^2 + (y_j' - y_i)^2} \leq r_0$
  - $dd(m_j', m_i)$ $= \min(|\theta_j' - \theta_i|, 360° - |\theta_j' - \theta_i|) \leq \theta_0$
- Maximize number of minutiae matches between input and mapped template by adjusting $\Delta x, \Delta y, \theta, P$ for pairing function P that find corresponding minutiae pair between input and template. Use RANSAC for that.
  - Use indicator function to get number of matches:
  - $mm(m_j'', m_i) = \begin{cases}1 & sd \leq r_0,\ dd \leq \theta_0\\ 0 & otherwise\end{cases}$

3.3.2 RANSAC

Objective robust fit of model to data set S which contains outliers. **Ran**dom **Sa**mple **C**onsensus. Use for finding optimal $\Delta x, \Delta y, \theta, P$. Video

- Select random sample of data points (minutiae points)
- Instantiate model ($\Delta x, \Delta y, \theta, P$) from points
- $ConsensusSet$ ≔ points within distance threshold of the model
  - If $|ConsensusSet| \geq t$ for some value t, reestimate model from Cons.Set. and terminate
  - Otherwise, rechoose sample and repeat. Select model with largest consensus set after N trials.

http://old.vision.ece.ucsb.edu/~zuliani/Research/RANSAC/docs/RANSAC4Dummies.pdf

3.4 Pre-Alignment

To reduce computational complexity, prealign prints.

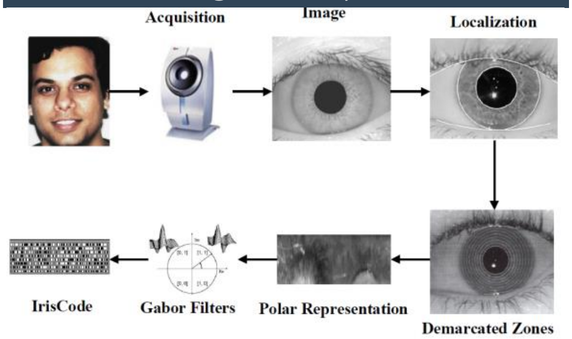
- Find core
- Find average ridge orientation on left/right sides of the core
- Rotate print around core such that difference between left and right ridge orientations are minimal

3.5 CNN based approaches

FingerNET: Unified model for enhancement, orientation estimation, segmentation and minutiae detection.

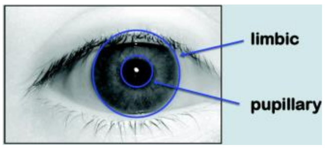
4 Iris Recognition

4.1 Iris Recognition System



4.1.1 Iris Segmentation

Main step, find pupillary (inner) and limbic (outer) boundaries.



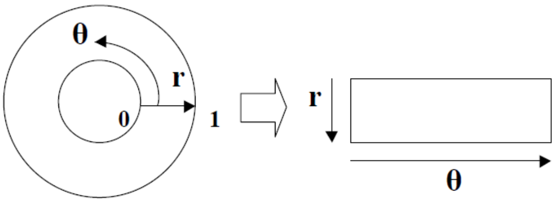
Historically found with Daugman's integro-differential operator, searches over image for maximum in blurred partial derivative. Operator:

$$\max_{r,x_0,y_0} \left| G_\sigma(r) * \frac{\delta}{\delta r} \oint_{r,x_ß,y_ß} \left(\frac{I(x,y)}{2\pi r}\, ds\right) \right|$$

For determining parameters of pupillary circle, for image $I(x,y)$. The operator searches over the image domain for the maximum in the blurred partial derivative with respect to increasing radius r. $*$ denotes convolution, $G_\sigma(r)$ is a smoothing function such as Gaussian of scale σ. x_0, y_0 are center coordinates of circle, r is radius.

4.1.2 Rubber Sheet model

Inner and outer boundaries are then used to "unwrap" circular iris region into a rectangular strip of a constant size (linearly stretching or compressing the imaged iris to a standardized frame; $(x,y) \mapsto (r,\phi)$). Divided into 8 subbands of equal thickness.



Binary mask on rectangle encodes occluded regions which bits should not be used (eyelids, reflections, eyelashes…). Mask has the same size as IRIS Code, e.g. 256 bytes.

4.1.3 Iris Code

IRIS Pattern is demodulated via quadrature 2D gabor filters (quadrature: four quadrants based on complex value: imag. +/- and real +/-). Results in complex number, code into binary code for each location based on real/imaginary value (sign function, is real/imag component ≥ 0 / < 0). Each phasor is quantized in 2 bits of phase information.

4.1.4 Comparing IRIS Codes

FHD: Fractional Hamming Distance between 0 (no difference) and 1 (totally different).

Computed as fraction of # of differing bits (algorithm: compute XOR-Map between enrolled map and probe map, then calculated (#1s / #total).

Actual function:

$$HD = \frac{||(codeA \otimes codeB) \cap maskA \cap maskB||}{||maskA \cap maskB||}$$

Degrees of freedom: Proof that it's very improbable for two eyes to match IRIS scores: Imposter matching score can be modelled with binomial with 249 degrees of freedom centered at hamming distance $.5$, that can be used to extrapolate claims. Conclusion: Hamming distance comparisons between different iris scores are binomial distributed with 249 degrees of freedom, with extremely high tails, making two different iris scores match extremely improbable.

Other explanation: We get that bernoulli model by tossing a coin 249 times (because it's centered around .5).

Decision: Genuine pair distribution of hamming distances still has a spread due to motion blue, noise, illumination…. Decision between genuine and imposter still easy.

- FAR (False Accept Rate) is used for identification $FAR = 1 - (1 - FMR)^N \cong N \cdot FMR$, where N is the # of degrees of freedom.
- FMR (False match rate) is used for identification

4.2 Recent Challenges

4.2.1 Occlusions

Observation: Noise is concentrated in different regions of the iris (reflections on sides, eye lashes in upper/lower regions). (TODO 04-41f)

4.2.2 Non-cooperative IRIS matching

Capture IRIS images at a distance.

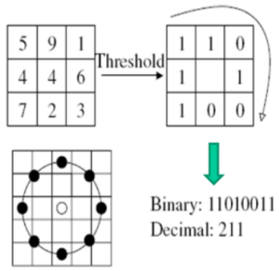
MICHE II competition: 75 users provided dataset of mobile device cameras, including different sources of noise.

Best performing method: compute iris code on segmented iris region, *extract LBP (Local Binary Patterns) features on periocular region*, fuse scores. (so not only iris, but also area around eyes)

4.2.2.1 Local Binary Pattern Histogram

Divide image into cells, compare each pixel with its neighbors. Replace with 1 if center pixel value is greater then neighbors value, otherwise 0. Compute histogram over cell, use histogram for classification (e.g. SVM or histogram distances).

Perform on larger neighborhood on circular points, interpolate values if point is between pixels.



4.3 IRIS Biometrics – Pros & Cons

- Few legacy databases
- high user cooperation required (small IRIS) or expensive devices
- Impaired performance by glasses, sunglasses, contact lenses
- Not left as evidence on scene of crime
+ Most accurate biometric, especially FA rates
+ Stable characteristics over a lifetime
+ Little negative press, more rapidly accepted
+ claimed to not involve high training costs

5 Face Recognition 1

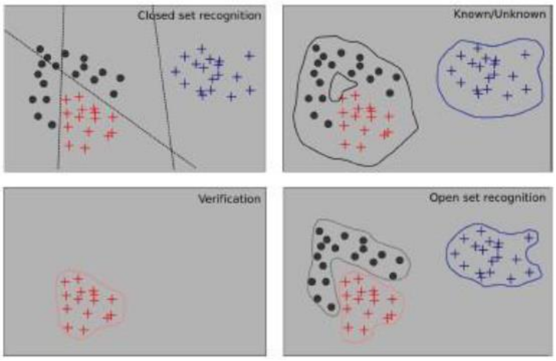
Advantages:

- No physical interaction with user
- Accurate
- No expert needed for interpreting results
- Leverage existing hardware
- Only biometric which allows passive identification in one-to-many scenario (identify terrorist in busy airport)

Not general object detection, but a single-class object recognition task (we can't assign one class for every person in the world).

Challenges: Variable illumination, low resolution, off-angle pose, heavy cosmetics, insufficient illumination.

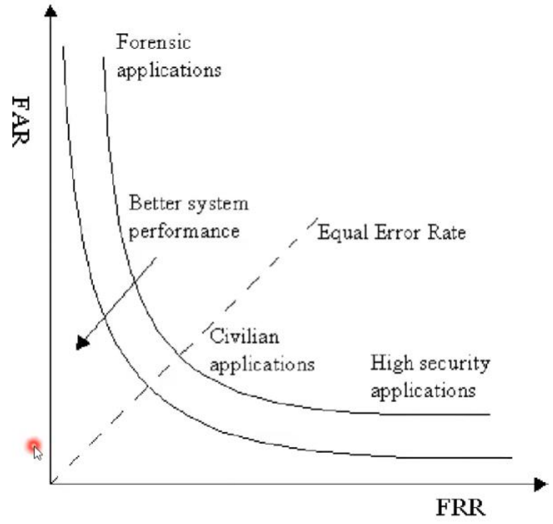
5.1.1 Closes Set vs Open Set



- Closed set Identification: Who is the reported person?
- Open set Identification: Is the person a known or unknown person? If known, who is he?
    - False accept: Invalid identity is accepted as known individual
    - False reject: Individual is rejected despite being known
    - False classify: Individual is correctly accepted but misclassified for incorrect person

Performance for Authentication/Verification: FR/FA, see FRR/FAR.

Receiver Operating Characteristics (ROC) curve:



5.2 Approaches

5.2.1 Appearance based

- Holistic approach: Processes entire face as input
- Local/fiducial approach: Facial features (eyes, mouth) are processed separately
    - Advantage: Local variations (expressions, occlusions, lightning) only affect local region
    - Facilitate weighting of local regions in terms of their effect on image recognition

Preprocessing: align with facial landmarks, normalize to common coordination by removing translation, rotation and scaling. Crop off background.

5.3 Holistic approaches

5.3.1 Eigenfaces

Image space is very high dimensional, subspace of potential faces/face features is relatively low dimensional. Idea: project face images into subspace and perform classification by similarity computation (e.g. distance).

Dimensionality reduction: Karhunen-Loeve transformation or **PCA**.

5.3.1.1 Principal Component Analysis (PCA)

Principal components: Eigenvectors of covariance matrix of the set of face images.

Algorithm: Suppose you have n face vectors, each of size d.

- Face matrix $Y = [y_i] \in R^{d \times n}$ of individual face imgs y_i
- Take the mean face $m = \frac{1}{n}\sum y \in R^d$ of all faces
- Compute covariance matrix
  $$C = (Y - m)(Y - m)^T \in R^{d \times d}$$
- Compute Eigenvalues $D \in R^{d \times d}$ and eigenvectors $U \in R^{d \times d}$ from $D = U^T C U$ per Eigendecomposition
    - D is a diagonal matrix with eigen values on diagonal
    - Cap U to the first k eigenvectors, sorted by eigenvalues, i.e. $U \in R^{d \times k}$
- Get representation coefficients $\Omega = U^T(y - m)$

5.3.1.2 Approach

- Training:
    - Acquire initial set of face images $Y = [y_1, y_2, \dots, y_k]$
    - Calculate Eigenfaces from training set, keep only M images corresponding to highest eigenvalues $U = u_1, u_2, \dots, u_M)$
    - Calculate representation of each known individual k in face space $\Omega_k = U^T \cdot (y_k - m)$
- Testing:
    - Project input image y into face space via
      $$\Omega = U^T \cdot (y - m)$$
    - Find most likely class k by distance computation $\varepsilon_k = \left|\left|\Omega - \Omega_k\right|\right| \forall \Omega_k$

Note: Formula for covariance between X and Y:

$$cov(X, Y) = E[(X - E[X])(Y - E[Y])]$$

5.3.1.3 Projections onto face space

Reconstruct images from their projection onto face space (from after PCA) via $Y_f = \sum_{i=1}^{M} \omega_i u_i$.

Faceness: Difference of mean-adjusted image $(Y - m)$ and projection Y_f, distance from face space, used for detecting faces.

TODO 05-27?

5.3.1.4 View-based Eigenspaces

Build separate Eigenspaces for every view (every angle), decide input view direction using distance from view space metric, classification in that view space.

5.3.2 Bayesian Face Recognition

Compares intrapersonal variations (within one class) with extrapersonal (between classes) variations.

- Reduce dimensionality of template/input with PCA ("Dual PCA")
- Subtract template from input
- classify this difference as "same" (intrapersonal, Ω_I) or "different" (extrapersonal, Ω_E)
- Use Bayes for final computation

$$S = P(\Omega_I|\Delta) = \frac{P(\Delta|\Omega_I)P(\Omega_I)}{P(\Delta|\Omega_I)P(\Omega_I) + P(\Delta|\Omega_E)P(\Omega_E)}$$

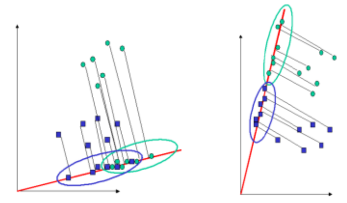
Dual PCA: TODO 05-32 !!

Subtract template from input to get difference, project it using PCA, classify this projection as "same" or "different".

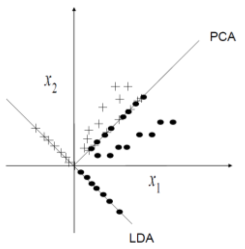
5.3.3 Linear Discriminant Analysis (LDA) – Fisherfaces

PCA does not use class information, thus can project different individuals too close together, also too much variation due to illumination.

Fisher's Linear Discriminant preserves class separability and projects away the within-class variation (lightning, expressions). *Maximizes between-classes scatter to within-classes scatter*. (Formulas: 05-34 !)


Poor Projection
Good Projection

PCA vs LDA:



5.4 Local Approaches

5.4.1 Modular Eigenfaces

Classification on fiducial (local) regions rather than holistic approach.

5.4.2 Local PCA (Modular PCA)

PCA seperatly on N subimages. Performs better than global PCA on variations of illumination and expression, not better under variation of pose.

5.5 Local feature-based Face Rec.

Local binary Pattern Histogram LBP, Gabor Feature, Discrete Cosine Transform DCT, SIFT.

5.5.1 Gabor Wavelet Transformation GWT

Construct Gabor kernels by different scales v and orientations u. Output of GWT is $O_{u,v}(x,y) = I(x,y) \cdot \psi_{u,v}(x,y)$ the convolution of the input image $I(x,y)$ and the gabor kernel $\psi_{u,v}(x,y)$. GWT emphasizes scale and direction of the image. $O_{u,v}(x,y)$ is high-dimensional with many scales u and orientations v, dimensions can be reduced with PCA.

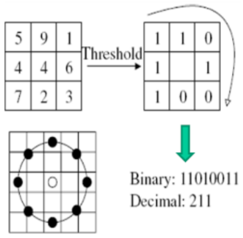
5.5.2 Elastic Bunch Graphs (EBG)

05-51?

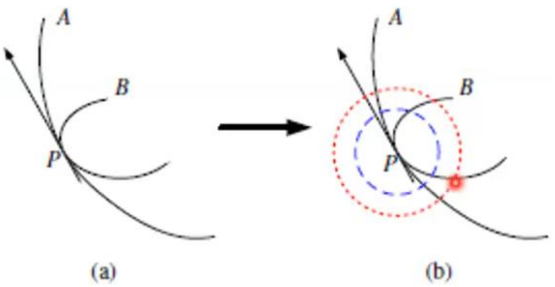
Uses bank of gabor filters, ?? not covered in depth in lecture, maybe look into literature

5.5.3 Local Binary Pattern Histogram

Image divided into cells, each pixel compared with its neighbors. For every neighbor pixel, use 1 if pixel is above threshold, otherwise 0. Use this binary pattern (8 neighbor pixels => 8bit), convert into decimal, compute histogram over the cell, use histogram for classification.



Why LBP?

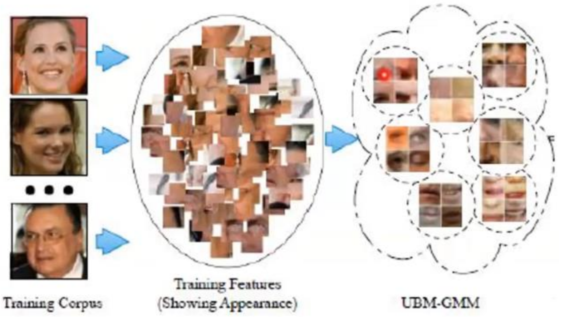


When comparing two gradients, their values might match at a certain point. With LBP, still detects differences there when taking the local neighborhood into account.

5.5.4 High-dim dense local feature extraction

Problem: Very high dimensionalities when working with many local patches. Especially when trying to detect at different scales of input image.

Solution: Encode into compact form, e.g. Bag of Visual Word Model (BoVW) or Fisher Encoding. (Fit thousands of input vectors into one combined vector)



Fisher Vector Encoding: Fit parametric generative model, e.g. Gaussian Mixture Model GMM. A Fisher vector is retrieved by stacking first- and second-order differences between dense features and GMM centers. (05-64 TODO)

6 Face Recognition 2

6.1 Face recognition across Pose

Problem: Different view-point or head-orientations.

Approaches:

6.1.1 Geometric pose normalization

Alignment on eye-positions insufficient. Find several facial feature to construct mesh, use that to normalize face.

2D Active Appearance model: Fitting via Inverse compositional (IC) algorithm. Shape is defined by a mesh (points). PCA on shape vector can find eigenvectors, do weighted sum on eigenvectors to get shape. TODO 06-7 and literature? TODO fitting goal.

Inverse compositional algorithm in 2d aa model:

- Shape is set of vertex coords $s = (x_1, y_1, x_2, y_2, \dots)^T$
- Do PCA over it to find eigenvectors s_i via
$$s = s_0 + \sum_i p_i s_i$$
p_i weights the eigenvectors s_i to get shape s. s_0 is mean vector of all examples.
- Appearance is modelled separately in similar way, with weights λ_i
- Fitting goal: Minimize difference between actual and model image by finding good weights p_i and λ_i

Varying eigenvectors directly varies shape of face.

Model can then be used to warp image into frontal pose.

Works well with local-DCT based approach, not so well with holistic approaches such as Eigenfaces.

6.1.2 3D face Model fitting

Simulate image formation in 3D space. Estimate 3D shape and texture of faces from a single image by fitting statistical morphable model of 3D faces to images.

Morphable face model is constructed such that any combination of shape vector S_i and texture vector T_i describes a realistic human face.

PCA is performed separately on shape and texture vectors. Eigenvectors form an orthogonal basis. (TODO?)

Fitting process: Find shape and texture coefficients s.t. 3D rendered face model resembles I_{input}, e.g. minimize

$$E_I = \sum_{x,y} \left|\left|I_{input}(x,y) - I_{model}(x,y)\right|\right|^2$$

6.2 DNNs for face recognition

See DL for CV.

- Hierarchical feature learning, e2e learning.
- CNN steps: Conv, Non-Linearity, Normalization, Pooling, Feature Maps, Repeat…
- GoogleNet, Inception Module, 1x1 convs

6.2.1 DeepFace

Learns a deep (7 layers) NN (20mio params) on 4 mio identity labelled face images directly on RGB pixels.

Alignment done before NN (detect 6 fiducial points for 2D warp, then 67 points for 3D model, then frontalize face)

Output is fed into k-way softmax to generate prob distribution over class labels. Maximize probability of correct class. After training last FC mapping representation to classes is thrown away, and distances on representations are computed.

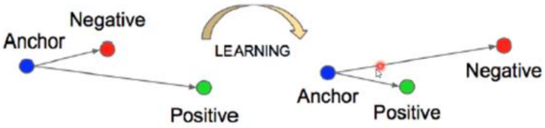
6.2.2 FaceNet

Idea: TODO 06-51-54

(Also uses GoogleNet/Inception Modules)

Metric Learning, Image Triplet Loss, Loss Function: TODO

- Minimize distance of feature vectors if coming from the same class, maximize otherwise. Try to learn this distance metric.
- Output of FaceNet are face embeddings
- Triplet Loss: Consider 3 vectors, 2 from the same person (anchor, positive) and one from another (negative).



- Loss formular:
  - $||f(anc) - f(pos)||_2^2 + \alpha < ||f(anc) - f(neg)||_2^2$
  - Anc = Anchor image, pos = positive image, neg = negative image, α = desired margin between pos/neg pairs

TODO read paper

7 Soft Biometrics

Traits to aid identification

- Behavioral traits (Gait/way of walking, speech, …)
- Physical traits (age, gender ethnicity)
- Semantic traits (long/short hair, body weight, clothing, color, glasses

7.1 Gait biometrics

Available at a distance or on low-res data. Needs moving features extraction. Many researchers, datasets and approaches.

7.1.1 Average Silhouettes Signature

1. Background is taken from each frame, pixels are thresholded → binary image
2. Normalize silhouette by height to account for camera distance, end with standardized frame with silhouette taking up entire frame
3. Average silhouettes (add all and divide by number of frames)
4. Resulting image is the signature

Analysis: Generate average gait silhouette, perform recognition on that.

Centroid of silhouette can also be used to give invariance to noisy silhouette data, as centroid is fixed on flat walking surfaces (07-12).

Experiment results: TODO lecture

7.2 Other soft biometrics

Use semantic descriptions (TODO lecture) of physical traits, semantic terms, visible at a distance.

Often done on silhouettes (ethnicity requires color-silhouettes)

- Advantages:
  - No (feature/sensor) ageing
  - *Available at distance/low res*
  - Fit with human descriptions
  - Complement automatically perceived measures
  - Need for search mechanisms ? (TODO lecture)
- Disadvantages
  - Psychology/perception
  - Needs labelling

Traits are chosen s.t. they are visible at distance, and are mentioned consistently in judgements.

Commonly used features:

- Global Features
  - Commonly mentioned in witness statements
  - Sex and age
  - Ethnicity (unstable, we chose 3 main subgroups +2 for UK police)
- Body features
  - Based on whole body description stability analysis by MacLeod
  - 5-point quality measures (very thin → very fat, very short → very long)
  - Most likely candidate for gait associaton
- Head features
  - Hair length and color

7.2.1 Age estimation

Basic Approach: Image Preprocessing → Feature extraction → PCA → Support Vector Regression.

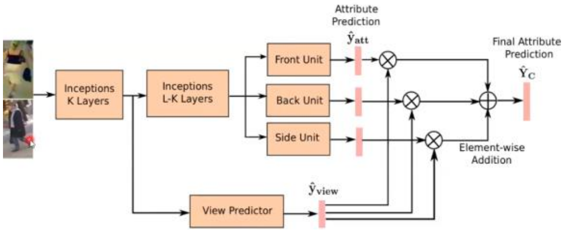
Today you can just use CNNs.

Two-Stage Regression: Estimate with 2 stages. Get rough result from global classifier and more exact age from another set of classifiers, each in this set matching for a precise age region (10-19, …).

7.2.2 Semantic Attribute Recognition

- Infer attributes (wearing backpack, female, longhair…)
- Person Re-Identification
- Efficient query and retrieval

VeSPA: View Specific Pedestrian Attribute Inference. Deep CNN, GoogleNet Structure. Incorporates pose information. Separate layers/units for front view, back view, side view. Those outputs are then weighted by view predictor and then summed. TODO relook into lecture.



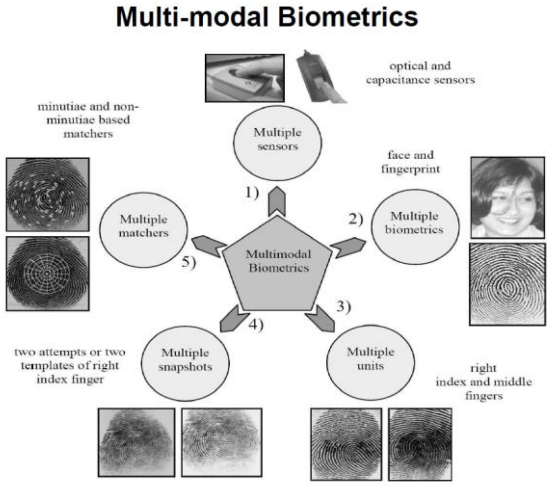
8 Multi-Biometrics

- Multimodal Biometrics: Combine 2+ biometrics traits to improve accuracy
- Crossmodal Biometrics: Match 2+ biometrics traits acquired with different sensor modalities

8.1 Multimodal Biometrics

Why multimodal? Better reliability, data is noisy, easy to do, FTE (failure to enroll rate) can be reduced, more robust against spoofing attacks.

- Combining metrics:
  - Intra-Modal Combination: Combine multiple matchers within single biometric
  - Multi-Modal Combination: Combine single/multiple matchers among multiple biometrics



8.1.1 Fusion Levels

- Fusion prior to matching
  - Sensor level
  - Feature level
    - Weighted avg of individual feature vectors
    - Concatenating non-homogenous vectors
    - Difficult to achieve
- Fusion after matching
  - Decision Level
    - Majority Voting
    - Dynamic Classifier selection (choose best classifier)
  - Matching Score Level
    - Score normalization required
    - Classification approach or Combination approach (scores are combines by sum/max/…)

Architecture image TODO? 08-12

8.1.2 Face vs Ear

- Image cropping, normalization
- Images are Masked
- Images are histogram equalized
- PCA computes eigenvectors and eigenvalues

TODO 08-16-28, important? yeah

Results show that face-based and ear-based biometrics perform similarly, but a multimodal biometric using both can outperform one using only one.

8.1.3 Finger vs Fingerprint

- Alignment stage: Transformations between template in database and input are estimated, then input minutiae are aligned with template minutiae
- Matching stage: Both minutiae are converted to strings, "elastic" string matching is performed

- FAR is estimated by computing impostor distribution
- Decision Fusion: TODO 08-32?

8.2 Cross-modal Biometrics

Different sensors or imaging modalities.

Computationally efficient, can work in darkness, but large research gap (!). TODO look into problems with research gap.

Motivation & Examples: Maybe not relevant? 37-45

8.3 Deep Perceptual Mapping (DPM)

Non-linear mapping between visible and thermal domain, FC FF DNN, learns a parametric non-linear perceptual mapping function.

- TODO formular? 08-47
- Dense SIFT vectors

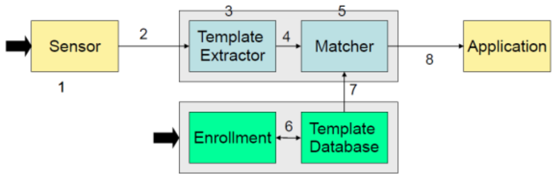
8.4 Polarimetric Thermal

Material emits radiation that exhibits linear polarization. Acquire polarization state imagery in thermal spectrum. Provides additional textual and geometric information.

Mapping direction is from thermal to visible for DPM.

9 Biometric System Attacks

Automated Biometric System model:

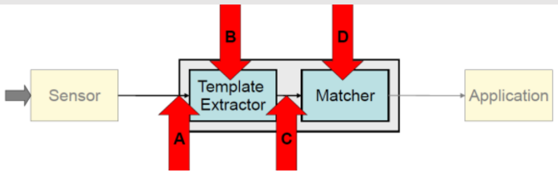


9.1 Attacks

9.1.1 Kinds of attacks

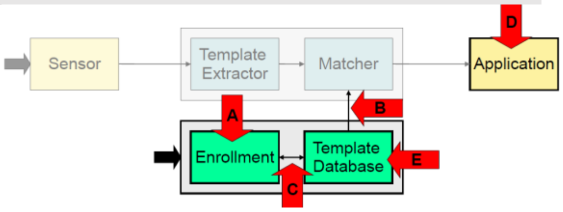
- Artificial samples (Spoofing)
- Coercive attack
  - Genuine user is forced to authorize attacker (fix by e.g. stress analysis, guards…)
- Impersonation attack (Type 1 attack)
  - Unauthorized individual changes his apperance to appear like an authorized one
  - Voice/face changes, fake fingerprints
- Replay attacks
  - Recording of true data presented to sensor
  - Fixed by e.g. prompt to read random text, require change of expression
- Denial of service
- Changes in the database

9.1.2 Front-end attacks



- A: Replay attack
  - Recording of true data transmitted to extractor
- A: Electronic Impersonation
  - Injection of artificial image created from features
- B: Trojan Horse
  - Extracted features are replaced
- C: Communication
  - Attacks during transmission to remote matcher
- D: Trojan Horse
  - Match decision is manipulated
- Attacks on the Matcher→Application arrow
  - Collusion: Use of or agreement with “super-users”
  - Covert Acquisition: Biometric stolen, but just parametric data used (so no impersonation)
  - Denial: Authentic user is denied by the system

9.1.3 Back-end attacks



- A: Enrollment attack: Collusion, Covert Acquisition, Denial
- B, C: Communication attack: During transmission
- D: Viruses, Trojans
- E: Hacker’s attack: Modification of registers, gathering information

9.1.4 Other attacks

- Password attacks (for psw-protected systems)
- Hill Climbing: Repeatedly submit biometric data with slight differences, preserve modifications that improve score. Prevent by not giving out score or limiting attempts.
- Swamping: E.g. submit print with hundreds of minutiae hoping a threshold amount of them matches. Prevent by normalizing the number of minutiae.
- Piggy-back: Gain unauthorized access by simultaneous entry with legitimate user.

9.2 Counter attacks

- Data protection: Standards, template encryption
- Liveness Detection & Anti Spoofing: Detect finger pores/pulse/temperature, detect pupil size change on light, texture analysis on face.
- Combine smartcards and biometrics: Use biometrics for reliable authentication and smartcards to safely store biometrics and other data.
  - Authentication done locally: No DB communication
  - Information never leaves the card
  - Attacks occur locally and are treated locally
  - Keeps privacy
- Challenge Response Protocol: System issues a challenge for the user, makes recorded biometrics harder to use
- Cancellable Biometrics: intentional and repeated distortion of biometric features to protect sensitive user-specific data
  - Usually: Biometric identifier compromised → compromised forever
  - Non-invertible distortion used on both input and template

9.2.1 Anti-Spoofing and Liveness Detection

Spoofing: Using fake biometric samples.

Obfuscation: Hiding your identity, e.g. mutilation of fingerprint, texture-contact lens to hide iris pattern, makeup.

- Ways to mitigate spoofing risk: Multi-factor authentication, multi-biometrics, liveness detection, anti-spoofing

9.2.1.1 Liveness Detection

Liveness not necessary to measure biometric itself.

- Hardware based: Use specialized hardware integrated into biometric sensor
  - Temperature
  - Pulse
  - Blood pressure
  - Odor
  - Electrocardiogram
  - Multispectral imaging (09-49)
- Software based: Use information already measured from biometric sensor.

(TODO: 09-54-55 lists performance metrics, might be interesting to look into those)

Performance vocab:

- Biometric performance
  - False reject rate: Error of rejecting a genuine user
  - False accept rate: Error of accepting imposter
- Anti-spoofing
  - False reject rate: Error of anti-spoofing detection rejecting genuine user
  - Spoof false accept rate: Error of accepting a spoof

10 Standards

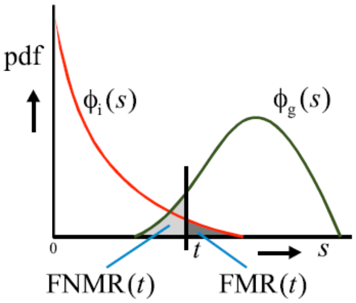
Why? To tell if deployment is secure, allowed in gov scenario, …

10.1 Relevant standards

(Probably not relevant?)

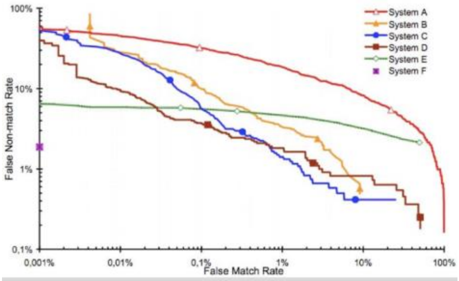
10.2 Biometric Performance Testing

- Technology Testing
  - Algorithmic level verification error
    - False-match-Rate (FMR): algorithm accepts zero-effort-imposter
    - False-non-match-rate (FNMR): algorithm rejects true identity
- Scenario Testing and operational testing
  - System level verification error
    - False-Accept-Rate (FAR)
    - False-Reject-Rate (FRR)
  - System level error requires observing:
    - Sample generation: Failure to Capture (FTC)
    - Enrollment: Failure-to-Enroll (FTE) – no reference for subject
    - Verification: Failure-to-Acquire (FTA) – no probe feature vector



For Probability density Distribution Function PDF, PDF of genuine similarity score $\Phi_g(s)$ and PDF of imposter similarity score $\Phi_i(s)$.

DET curve: Detection error trade-off curve, modified ROC plotting error rates on both axis, FPs on x and FN on y:



10.3 Vulnerability Testing

Liveness detection, maybe not relevant? TODO

10.4 Template Protection

Don't store fingerprints, iris or face images, instead transform templates to pseudonymous identifiers (PI), reaching:

- Secrecy: Can be compared without decryption
- Diversifiability/Unlinkability: prevents database cross-comparison
- Renewability: Can revoke and renew template data
- Noise-robustness: works with noisy biometric samples
- Non-invertibility: Cannot reconstruct original