Image and Video Analysis Exam

Identity verification through hand shape and geometry

Emanuele Vivoli
Francesca Del Lungo

University of Florence, Italy

May 2019

Outline

Biometric system

It is a system that uses information about a person to identify that person.

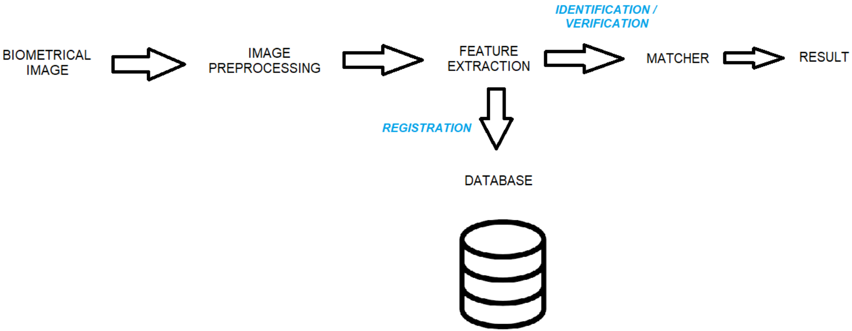


Figure: An illustration of a typical biometric system.

Multimodal biometric system

It is a system that uses a combination of two or more biometric modalities to identify a person.

Fingerprint Reader System

Iris Reader

Face Recognition System

Hand shape biometrics

Create a biometric system for personal identity verification that uses both **hand shape** and **hand geometry**.

Shape and geometry features are derived with the help of only contour of the hand image.

Hand shape biometrics

Why is hand shape and geometry biometrics attractive?

- Images can be taken in an user convenience, non-intrusive manner although cooperatively;
- Inexpensive sensors: no need of very high resolution images;
- Additional biometric features (such as palmprints and fingerprints) can be integrated.

Paper proposed method

Images acquisition

- Images can be captured through a simple camera or a document scanner.

- No images with missing parts of the hand (the approach is invariant to translation, rotation and scaling, but not to cropping effect)

Preprocessing I

It is done to *eliminate the noise* as well as *background of hand*.

Steps:

1. color captured image;

2. gray scale image;
3. 2-D median filtering with 3 x 3 filter (to eliminate salt and pepper noise);

Preprocessing II

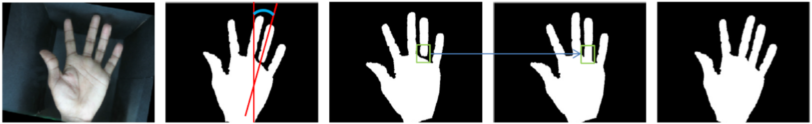
4. binary image using a threshold;
5. dilation with a diamond structuring element of size 2;
6. erosion with a square structuring element of size 3.

hand mask image

Joining the disjoint finger

For each disjoint finger of a hand mask the following steps are performed:

1. find the orientation of the disjoint finger, figure (b);
2. rotate the whole image such that the finger under consideration becomes vertical, figure (c);
3. fill the vertical gap between the disjoint finger and the hand mask, figure (d);
4. rotate back the whole image to its original orientation, figure (e).



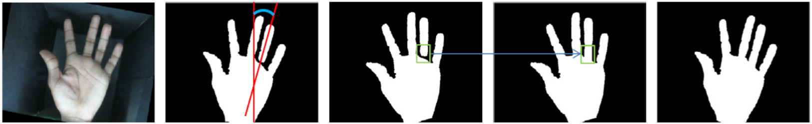
(a) (b) (c) (d) (e)

Joining the disjoint finger

For each disjoint finger of a hand mask the following steps are performed:

1. find the orientation of the disjoint finger, figure (b);
2. rotate the whole image such that the finger under consideration becomes vertical, figure (c);
3. fill the vertical gap between the disjoint finger and the hand mask, figure (d);
4. rotate back the whole image to its original orientation, figure (e).

⚠ How to find the disjoint finger is not specified in the paper



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

Extract centroid μ

The co-ordinates (x_μ, y_μ) of the centroid (μ) of the hand can be computed using the two first order Cartesian moments:

$$x_\mu = \frac{\sum_{\delta \in hand} x_\delta}{\sum_{\delta \in hand} hm} \tag{1}$$

$$y_\mu = \frac{\sum_{\delta \in hand} y_\delta}{\sum_{\delta \in hand} hm} \tag{2}$$

hm: hand mask

Orientation (θ)

It corrisponds to the orientation of major axis from the vertical one.

$$\theta = 0.5 tan^{-1}\left(\frac{2\xi_{1,1}}{\xi_{2,0} - \xi_{0,2}}\right) \tag{3}$$

where $\xi_{\alpha,\ \beta}$ are the central moments, defined as follows:

$$\xi_{\alpha,\ \beta} = \sum_{\delta \in hand} (x_\delta - x_\mu)^\alpha (y_\delta - y_\mu)^\beta \tag{4}$$

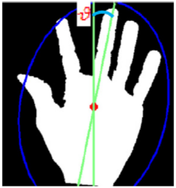


Figure: Centroid and orientation

Hand orientation registration

Then the image is rotated in such a way that the major axis of hm becomes vertical.

After the rotation of hm w.r.t. μ every point of hm will have new coordinates.

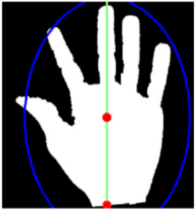


Figure: Rotated hand mask

Reference point extraction

The reference point **r** is calculated as the intersection of major axis of the hand mask with the hand contour. **r** is located in the opposite side respect to the middle finger.

We considered a point at the wrist boundary as a stable reference point because the points of the wrist boundary are the least affected by the rotation of the hand.

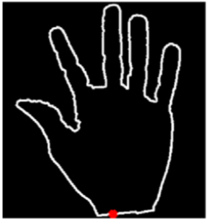


Figure: Hand contour with reference point

Peaks and valleys I

Let:

- **r** the extracted reference point at the wrist line of hand with co-ordinate (x_r, y_r)
- $b_r{}^{\mathbf{cw}}$ the boundary of the hand contour traced from the reference point **r** in clockwise direction.
  Every boundary component $b_r^{cw}(i) \in \mathbb{R}^2$. So the curve of the hand contour is parameterized with respect to a variable in one dimension whose values are in [0, n-1], where n is the number of points in the hand contour.

Peaks and valleys II

Than a distance map is created with the Euclidean distance of each boundary pixel from the reference point.

The local maxima of the distance map identified as p_j for j = 1,...,5 represent the five peaks (five hand fingers), and v_j for j = 1,...,4 the valleys between the fingers.

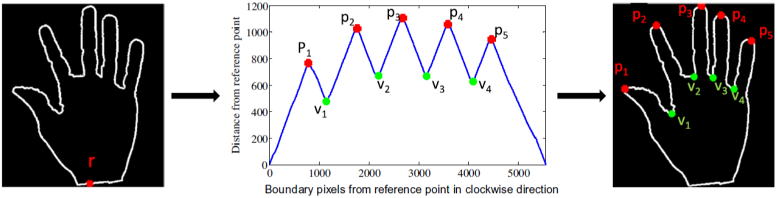


Figure: Peaks and valleys extraction through distance map

Complementary valleys

An important point for a finger is the complementary valley point, it is needed in order to delimitate the fingers in the other side of valley point.

The index Γ_{c_j} of complementary valley c_j corresponding to the valley v_j is extracted using the indexes of p_j and v_j as:

$$\Gamma_{c_j} = \begin{cases} \Gamma_{p_j} - (\Gamma_{v_j} - \Gamma_{p_j}), & \Gamma_{p_j} < \Gamma_{v_j} \\ \Gamma_{p_j} + (\Gamma_{p_j} - \Gamma_{v_j}), & \Gamma_{p_j} > \Gamma_{v_j} \end{cases} \quad \forall j \in [1,5] \tag{5}$$

Complementary valleys

The coordinates of the point c_j will be ($x_{b_r^{cw}(\Gamma_{c_j})}$, $y_{b_r^{cw}(\Gamma_{c_j})}$).

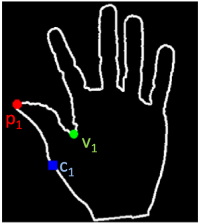


Figure: Complementary valley extraction for thumb finger

The complementary valley point is also needed with valley point to find middle points.

Middle points

Now that we have peaks, valleys and complementary valleys we calculate another end point (middle point) m_j, for each finger, from the pairs v_j and c_j.

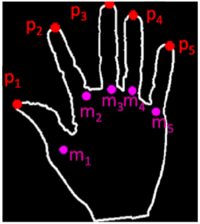


Figure: Mid-point of the valley and complementary valley extracted for each finger

Note that v_j and c_j lie on the contour of the hand whereas m_j lies inside the hand.

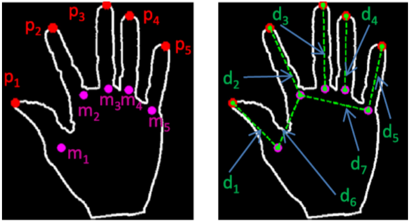
Features extraction

Geometrical features

Peaks (p_j), and middle of valleys (m_j) are used to design the geometrical features.

Geometrical feature are based on 7 distances (d_i i=1,...7)

If l is the number of distances that are used, then the dimension of final geometrical feature is: $l \times (l-1)/2$



These are distances ratio, that these are scale invariant.

Extraction of shape features: steps

Three steps:

1. **finger registration**: performed to achieve the finger rotation invariance;
2. generation of **distance and orientation maps** from the reference point and updated hand contour;
3. application of **wavelet decomposition** over distance and orientation maps to generate the distance and orientation features.

Finger registration I

Finger registration is performed to align the fingers to a particular orientation.

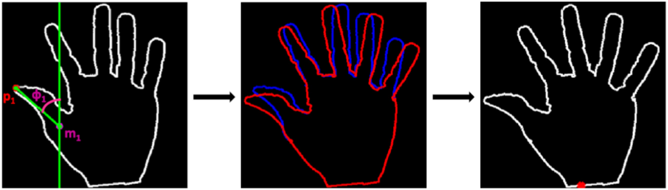
Let ω_j is the orientation of jth finger after finger registration for j=1,...,5. The values of ω_j are considered from the vertical axis in counter-clockwise direction and given as:

$$\omega_j = \begin{cases} +60, & j = 1 \\ +30, & j = 2 \\ +10, & j = 3 \\ -10, & j = 4 \\ -20, & j = 5 \end{cases}$$

Finger registration II

So each finger must be rotated by a proper angle ψ_j .
$\psi_j = \omega_j$ - ϕ_j
where, ϕ_j is the current orientation of the jth finger.

And then every pixel of the fingers contour must be updated (x_p', y_p').



(a) Contour with known finger feature points

(b) Finger alignments at particular orientations

(c) Updated contour with reference point

Distance and orientation map

The distance map (*dp*) and orientation map (*op*) are generated using reference point r and updated contours b_r^{cw}'.

dp and *op* of each boundary pixel from the reference point are calculated as:

$$dp(i) = \sqrt{(x'_r - x'_{b_r^{cw}(i)})^2 + (y'_r - y'_{b_r^{cw}(i)})^2} \quad (6)$$

$$op(i) = 90 + tan^{-1}\left(\frac{y'_r - y'_{b_r^{cw}(i)}}{x'_r - x'_{b_r^{cw}(i)} + \sigma}\right) \quad (7)$$

Wavelet decomposition

It is observed that increase in the number of pixels of contour will not benefit in the discriminative ability of the *dp* and *op*.

Wavelet decomposition is used to transform the higher dimension feature into low dimension feature and to select most discriminative features.

We applied **1-D wavelet decomposition at level 5 using Daubechies-1 wavelet filter** over *dp* and *op*.

We selected the first 50 coefficients of the wavelet decomposed *dp* and *op*.

Theoretical concepts I

Shapes and Landmarks I

Definition

Shape is all the geometrical information that remains when location, scale and rotational effects are filtered out from an object.
D.G. Kendall

Shapes and Landmarks I

Definition

Shape is all the geometrical information that remains when location, scale and rotational effects are filtered out from an object. *D.G. Kendall*

↓

shape is invariant to Euclidean similarity transformations.

Shapes and Landmarks II

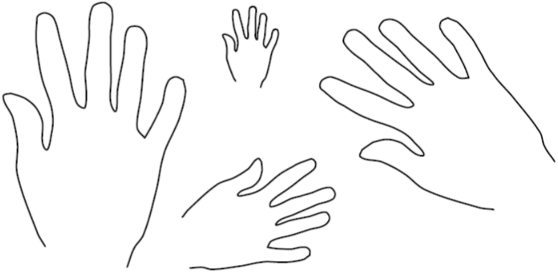


Figure: Four copies of the same shape, but under different Euclidean transformations.

Shapes and Landmarks III

How can we describe a shape?

One way to describe a shape is by locating a finite number of points on the outline.
So we can introduce the concept of landmark:

Definition

A **landmark** is a point of correspondence on each object that matches between and within populations.

Landmarks I

Landmarks can be divided in 3 main classes:

- **Anatomical landmarks**: are biologically-meaningful points in an organism. These points are assigned by an expert and correspond to distinctive signs of the specific class of objects (examples of anatomical landmark for the shape of a skull are the eye corner, tip of the nose, jaw, etc.).

Landmarks I

Landmarks can be divided in 3 main classes:

- **Anatomical landmarks**: are biologically-meaningful points in an organism. These points are assigned by an expert and correspond to distinctive signs of the specific class of objects (examples of anatomical landmark for the shape of a skull are the eye corner, tip of the nose, jaw, etc.).
- **Mathematical landmarks**: points located on an object according to some mathematical or geometrical property (high curvature or an extremum point can be considered as mathematical landmarks).

Landmarks I

Landmarks can be divided in 3 main classes:

- **Anatomical landmarks**: are biologically-meaningful points in an organism. These points are assigned by an expert and correspond to distinctive signs of the specific class of objects (examples of anatomical landmark for the shape of a skull are the eye corner, tip of the nose, jaw, etc.).
- **Mathematical landmarks**: points located on an object according to some mathematical or geometrical property (high curvature or an extremum point can be considered as mathematical landmarks).
- **Pseudo-landmarks**: constructed points on an object either on the outline or between landmarks. They can be automatically generated starting from previously identified anatomical or mathematical landmarks.

Landmarks II

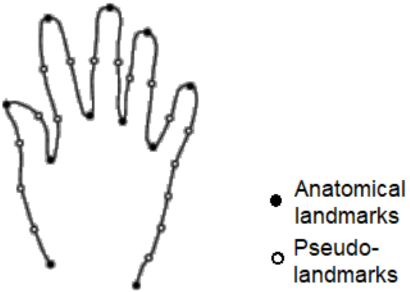


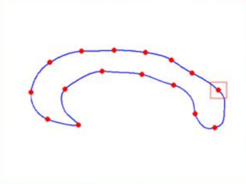
Figure: A hand annotated using 11 anatomical landmarks and 17 pseudo-landmarks.

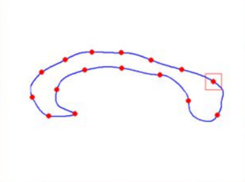
Landmarks III

- It's very important to identify the most appropriate landmarks to represent the shape of a family of objects.
- A wrong choice of the set of landmarks can lead to models that are unreliable and that do not reflect the actual variability between objects.

Landmarks IV

"Correspondence problem": given a population of shapes described through landmarks, it is necessary to define a criterion for choosing the labels so that the same value is associated to the corresponding points on all the forms.





This problem is usually solved in two dimensions with a manual annotation, but in 3D the difficulties of visualization and the high number of landkmarks required make the process long and subject to errors.

Landmarks V

The set of landmark coordinates, extracted for example from an image, is called "configuration".

Let's define the matrix $X \in \mathbb{R}^{nxm}$ as the **configuration matrix**, with n landmarks in m dimensions, where the space of configurations is the space of all possible landmark configurations.

Landmarks VI

Goodall (1991) gave a precise nomenclature to the shapes represented through landmarks, this was based on the transformations applied to them.
He defined:

- **Figure**: it corresponds to the configuration of the original landmarks, as extracted, without any transformation;
- **Form**: it corresponds to the equivalence class composed by the figures to which the translation and rotation information have been removed. A form is also called a *size-and-shape* (Dryden and Mardia, 1998).
- **Shape**: it corresponds to the equivalence class composed by the *figures* to which the translation, rotation and scaling information have been removed.

Shapes transformations

- **scale transformations** (isotropic, as the topology of the shape must be preserved) can be defined through a multiplication of the configuration matrix with a positive real number;
- **translation transformations** can be defined through a matrix sum between the configurations and a m-dimensional constant vector;
- **rotation transformations** can be interpreted as the multiplication the coordinates of the points by the sine and the cosine of the rotation angle θ.

Shape Alignment I

To obtain a true shape representation, according to our definition, location, scale and rotational effects need to be filtered out.

This is carried out by establishing a coordinate reference to which all shapes are aligned.

This process is commonly known as **Procrustes Analysis**, that brings the shape set into shape space.

Shape Alignment II

Definition

The **Shape Space** is the set of all possible shapes of the object in question.

- The shape space, unlike the form space, is similar to a hypersphere of unitary radius, as the figures are centered and normalized to the unitary dimension.
- In particular, the shape space is partitioned into fibers (each of which corresponds to an equivalence class [X]), where each pre-shape on the same fiber differs only by rotation transformations.
- Pre-shape can be seen as the last step towards true shape (rotational effects still need to be filtered out).

Shape space dimension I

What is the dimension of the shape space?

If we have n point vectors in m Euclidean dimensions the dimensionality is nm.

But the alignment procedure reduces the dimensionality, i.e. data now extend only on a subspace of nm.

Shape space dimension II

In particular:

- The translation removes m dimensions;
- the scale transformation removes one dimension;
- the rotation removes $m(m - 1)/2$ dimensions.

Shape space dimension II

In particular:

- The translation removes m dimensions;
- the scale transformation removes one dimension;
- the rotation removes $m(m - 1)/2$ dimensions.

So, the shape space dimensionality is:

$$M = nm - m - 1 - \frac{m(m-1)}{2}$$

Shape space and metric I

If a relationship can be established between the distance in shape space and the Euclidean distance in the original space, the set of shapes forms a Riemannian space containing the object class in question.

This is also denoted as the *Kendall shape space*, and the relationship is called a *shape metric*.

Shape space and metric II

Common shape metrics are:

- Hausdorff distance
- strain energy
- Procrustes distance

Shape space and metric II

Common shape metrics are:

- Hausdorff distance
- strain energy

] compare shapes with unequal amount of points

- Procrustes distance

→ requires corresponding point sets

Procrustes Analysis I

The **Procrustes analysis** make possible the alignment of different configurations through similarity transformations.

This is done to minimize the complete Procrustes distance between figures, using least squares optimization techniques.

Procrustes Analysis I

The **Procrustes analysis** make possible the alignment of different configurations through similarity transformations.

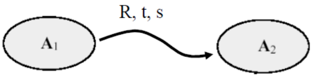
This is done to minimize the complete Procrustes distance between figures, using least squares optimization techniques.

There are two types of Procrustes analysis:

1. **classical or ordinary procrustes analysis**: deals with the matching between two figures;
2. **generalized procrustes analysis (GPA)**: it uses the first one and tries to analyze and align sets composed of more than two figures.

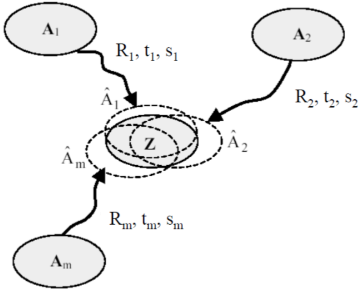
Procrustes Analysis II

Procrustes Analysis



Align one shape with another
(not symmetric)

General Procrustes Analysis



Align a set of shapes with respect to some unknown "mean" shape
(independent of ordering of shapes)

Ordinary procrustes analysis: Procrustes shape distance I

The Procrustes distance is a least-squares type shape metric that requires two aligned shapes with one-to-one point correspondence.

The alignment part involves four steps:

1. compute the centroid of each shape;
2. re-scale each shape to have equal size;
3. align w.r.t. position the two shapes at their centroids;
4. align w.r.t. orientation by rotation.

Ordinary procrustes analysis: Procrustes shape distance I

The Procrustes distance is a least-squares type shape metric that requires two aligned shapes with one-to-one point correspondence.

The alignment part involves four steps:

1. compute the centroid of each shape;
2. re-scale each shape to have equal size;
3. align w.r.t. position the two shapes at their centroids;
4. align w.r.t. orientation by rotation.

Then the squared Procrustes distance between two shapes, x_1 and x_2, is simply the sum of the squared point distances:

$$P_{\mathrm{d}}{}^2 = \sum_{j=1}^{n} [(x_{j_1} - x_{j_2})^2 + (y_{j_1} - y_{j_2})^2] \tag{8}$$

Ordinary procrustes analysis: Procrustes shape distance II

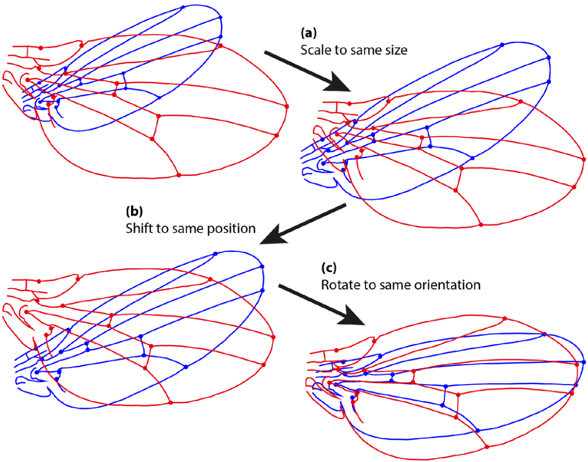


Figure: The figure shows the three transformation steps of an ordinary Procrustes alignment.

Step 1: compute the centroid

The centroid of a shape is the center of mass of the physical system consisting of unit masses at each landmark, so its coordinates $(\overline{x}, \overline{y})$ are calculated as:

$$(\overline{x}, \overline{y}) = \left(\frac{1}{n} \sum_{j=1}^{n} x_j, \frac{1}{n} \sum_{j=1}^{n} y_j \right) \tag{9}$$

Step 2: re-scale each shape I

In order to re-scale each shape we have to establish a *shape size metric*:

Definition

A **shape size metric S(x)** is any positive real valued function of the shape vector that fulfils the following property (like all other metrics):

$$S(ax) = aS(x)$$

Step 2: re-scale each shape II

Commonly used scale metric are:

- Frobenius norm (or 2-norm):

$$S(x) = \sqrt{\sum_{j=1}^{n} [(x_j - \overline{x})^2 + (y_j - \overline{y})^2]}$$

- centroid size:

$$S(x) = \sum_{j=1}^{n} \sqrt{(x_j - \overline{x})^2 + (y_j - \overline{y})^2}$$

Step 3-4: align position and orientation I

To do so a Singular Value Decomposition (SVD) is applied.
Steps:

1. Arrange the size and position aligned x_1 and x_2 as mxn matrices;
2. Calculate the SVD, $\mathbf{U\Sigma V^T}$, of $x_1^T x_2$ in order to maximize the correlation between the two sets of landmarks.

Step 3-4: align position and orientation II

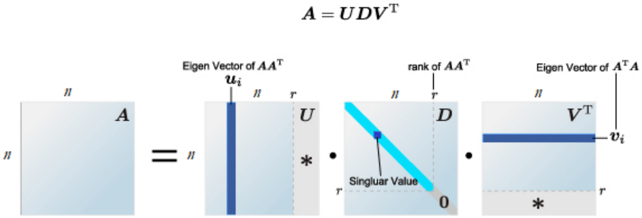
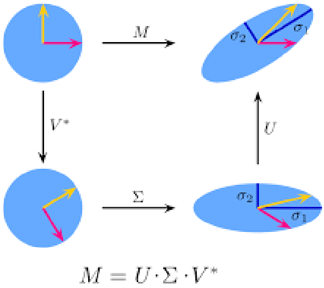


Figure: Left matrix is M, right matrixes are U, Σ, V^T. Left matrix is $x_1^T x_2$ nxn square with n size of dimensions in whitch landmarks are.

The expression $\mathbf{U\Sigma V^T}$ can be intuitively interpreted as a composition of three geometrical transformations: a rotation or reflection, a scaling, and another rotation or reflection.

Step 3-4: align position and orientation III



Upper left: The unit disc with the two canonical unit vectors.
Upper right: Unit disc transformed with M and singular values σ_1 and σ_2 indicated.
Lower left: The action of V^T on the unit disc. This is just a rotation.
Lower right: The action of ΣV^T on the unit disc. Σ scales in vertically and horizontally.
Finally, U is a rotation by an angle β.

Step 3-4: align position and orientation IV

3. the rotation matrix needed to optimally superimpose x_1 upon x_2 is then $\mathbf{VU^T}$:

$$\mathbf{VU^T} = \begin{bmatrix} cos(\theta) & -sin(\theta) \\ sin(\theta) & cos(\theta) \end{bmatrix}$$

Generalized Procrustes Analysis (GPA) I

To align a set of planar shapes the following iterative approach can be used (it is a generalization of the ordinary Procrustes analysis):

1. choose an initial estimate of the mean shape (e.g. the first shape in the set or the most representative: which one has minor distances between all the others);
2. align all the remaining shapes to the mean shape (that mean shape has to be re-normalized in order to represent it on the Kendall's hypersphere);

Generalized Procrustes Analysis (GPA) II

3. re-calculate the estimate of the mean from the aligned shapes, with the folling formula:

$$\overline{x} = \frac{1}{n} \sum_{i=1}^{n} x_i \tag{10}$$

4. If the estimated mean has changed return to step 2. Otherwise the true mean shape of the set has been found.

Convergence is obtained when the mean shape does not change significantly within an iteration.

Generalized Procrustes Analysis (GPA) II

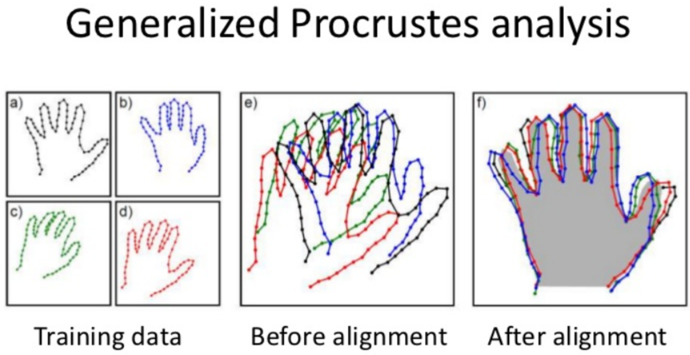
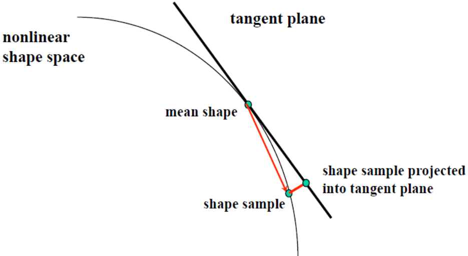


Figure: Example of alignment process with more than two shapes.

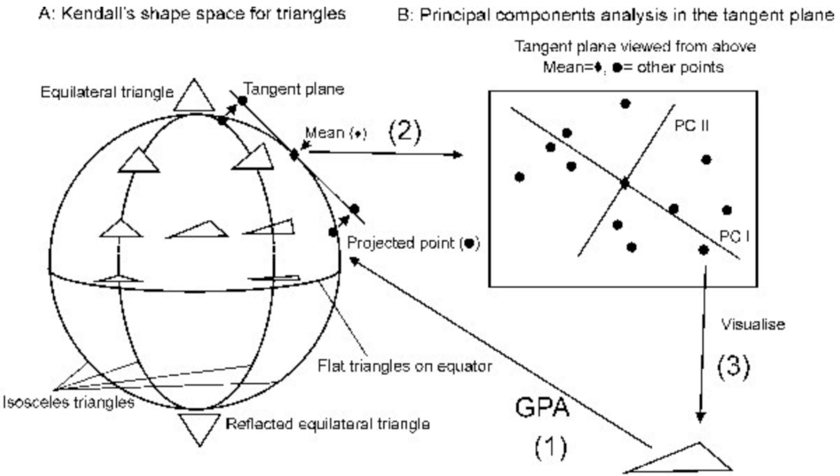
Generalized Procrustes Analysis (GPA) III

The shape space called Kendall's Shape Space for triangles (K=3, where K is the number of landmarks), can be seen as a sphere; for $K > 3$ the space is a hypersphere and so is extremely more complicated to visualize it.

So rather than measure the variation of the shape on the sphere (or hypersphere), it's common to project points into the tangent plane taken at the mean shape.



Kendall's Shape Space



Statistical Shape Models

Now that all the shapes are aligned we have to create a Statistical Shape Model.

To do so it's usually more efficient to reduce dimensionality where possible. After dimensionality reduction data lie in a subspace of reduced dimension.

This can be done with PCA.

Shape Models

Building Shape Models:

- Given aligned shapes x_i
- dimensionality reduction through PCA

$$x \approx \overline{x} + Pb$$

where:

- P: first t eigenvectors of covariance matrix
- b: Shape model parameters

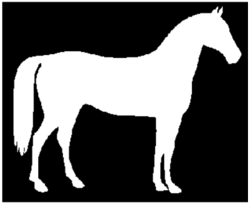
Theoretical concepts II

Theoretical concepts I

Definition

The **contours** of an image are a curve joining all the continuous points (along the boundary), having same color or intensity.

Contours is a list of all the contours in the image.
Each individual contour is a list of (x,y) coordinates of boundary points of the object.

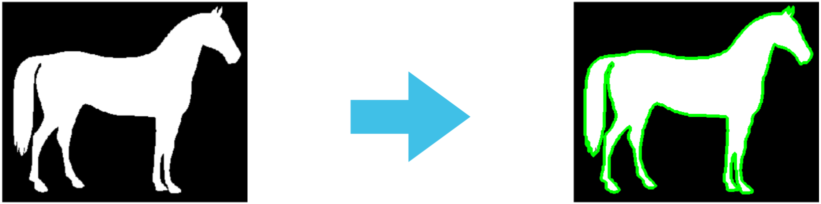


Theoretical concepts I

Definition

The **contours** of an image are a curve joining all the continuous points (along the boundary), having same color or intensity.

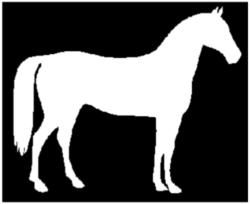
Contours is a list of all the contours in the image.
Each individual contour is a list of (x,y) coordinates of boundary points of the object.



Theoretical concepts II

Definition

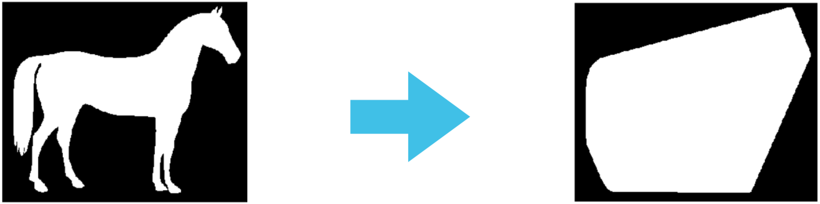
The **convex hull** of a binary image is the set of pixels included in the smallest convex polygon that surround all white pixels in the input.



Theoretical concepts II

Definition

The **convex hull** of a binary image is the set of pixels included in the smallest convex polygon that surround all white pixels in the input.

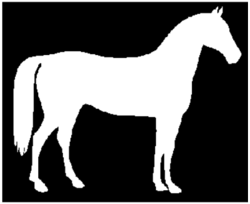


Theoretical concepts III

Definition

A **convexity defect** is a deviation of the object contour from this convex hull.

Any defect is defined as a list of 4 points, these are:
[start point, end point, farthest point, approximate distance to farthest point].
Note that the first three values (start point, end point and farthest point) are indexes of the contour.



Theoretical concepts III

Definition

A **convexity defect** is a deviation of the object contour from this convex hull.

Any defect is defined as a list of 4 points, these are:
[start point, end point, farthest point, approximate distance to farthest point].
Note that the first three values (start point, end point and farthest point) are indexes of the contour.

Our approach
(we use right hand images)

Preprocessing v.1

1. BGR image to GRAY image
2. median blur with kernel size 3
3. Otsu on subimages (grid 2 x 2) to BIN image
4. dilation with (16, 6) rectangle kernel
5. erosion with (9) square kernel
6. closure with (31, 31) ellipse structuring element

hand mask image

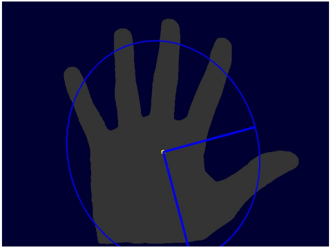
Preprocessing v.2

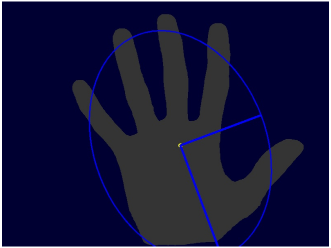
1. BGR image;
2. reshape image (cv2.resize);
3. changing color space to HSV;
4. median blur with kernel size 3 to each HSV channel;
5. segmentation of image through Gaussian mixtures;
6. creating BINARY image;
7. open with (10 x 10) ellipse kernel;

hand mask image

Not working approach

regionprops function returns not only center of mass, but also orientation and major axis but it doesn't work properly; as we can see:





Changing the way

In order to avoid the problem of getting a bad r stable reference point, we used a different approach than the paper.
The steps to find stable point and rotate hand image are:

1. contour and center of mass (centroid);
2. convex hull of contour (clockwise order)
3. importants convexity defects from contour and convex hull
4. find valley points and finger points

After that, with middle finger point and centroid we can find the stable reference point r on wrist contour.
Let's see how we can do this.

Contour

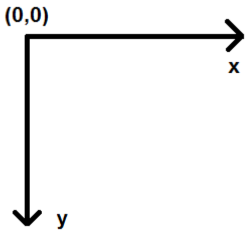


Figure: Axis orientation

The **Contour** starts from upper point (y co-ordinates), and has anti-clockwise direction. It is composed by all point (pixel) that a Pivot hits, starting in first contour element and moving on the margin of image.

Center of mass

Center of mass is calculated with *regionprops* function.
It calculates centroid (μ) co-ordinates (x_μ, y_μ) :

$$x_\mu = \frac{\sum_{\delta \in hand} x_\delta}{\sum_{\delta \in hand} hm} \tag{11}$$

$$y_\mu = \frac{\sum_{\delta \in hand} y_\delta}{\sum_{\delta \in hand} hm} \tag{12}$$

hm: hand mask
The ellipse has a certain orientation end semi-axis value, but as shown before it isn't usefull for finding middle finger - centroid axes.

I Contour and center of mass

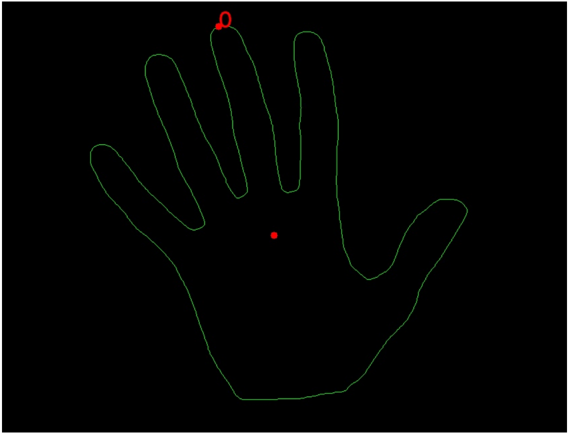


Figure: contour , center of mass, red number is the first element of contour

II Convex Hull

The convex hull is obtained by *convexHull* function of cv2, it starts from a different point respect to contour.

The starting point of convex hull is first left most point in x axes orientation, and move on in anti-clockwise way.

As we can see in next image, it will find a lot of convex hull points, and between all of these points there is a margin line of convex hull.

II Convex Hull of contour

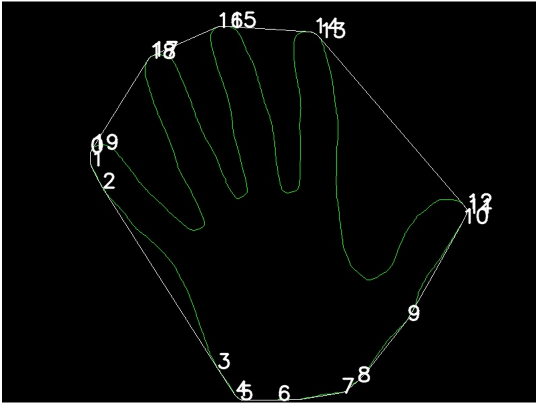


Figure: contour , convex hull, blue numbered points of convex hull

III Convexity Defects

Based on contour and Convex hull, it is important to get more information about and between these objects. As we have seen before, the convexity defects of a contour consists on finding internal convex hull points (belonging to contour) that have the biggest distance to convex hull margin. Between each couple of convex hull points we find a convexity defect point.
A convexity defect is composed by 4 params:

$$conv\_defects[i] = [s, e, f, d]$$

where s, e, f are respectively indexes of start, end and defect points on contour; and d is the distance between defect point and contour margin.

III Convexity Defects

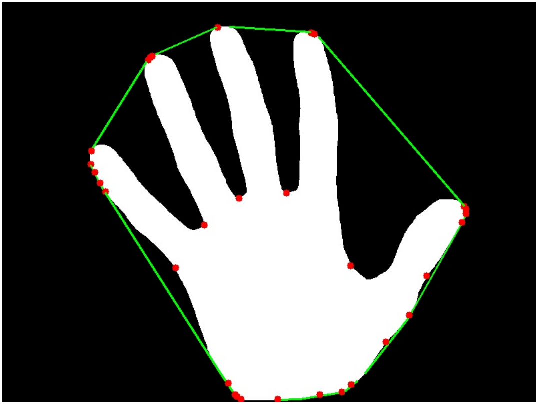


Figure: white hand mask , convex hull, points are starts, ends or defects points

III Convexity Defects (filtered)

In order to get finger points and valley points, we need to select only important defects. Notice that these important defects have larger distances between defect point and convex margin.

How many defects we need? Obviously 4.

Than we sort defects on distance parameter, and we get only firsts 4 elements. Remember the original order is important because we need now to re-sort in clockwise or anti-clockwise orientation. We choose clockwhise orientation and the result is shown below.

III Convexity Defects (filtered)

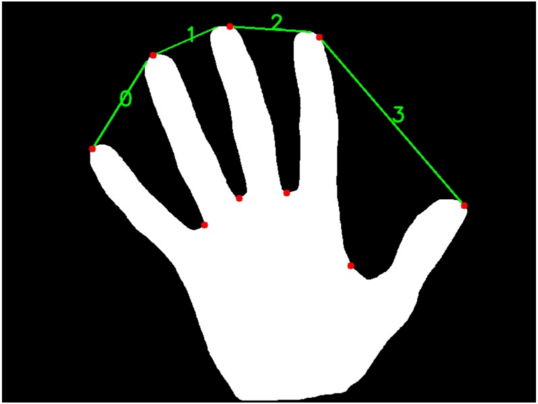


Figure: partial convex hull, red internal point are **valley** points, red external points are **finger** points

IV Peaks and Valleys

Obtained the 4 important defects from contour and hull of hand image (these correspond to the spaces between fingers) we define Peaks and Valleys. We treat these points always as indexes, in order to get real point with $contour[idx]$.

Valley indexes:

- the f defects index ($contour[f]$ is the valley point);

Peaks indexes:

- the *start* index if it is first defect;
- the middle index between *end* index of a particular defect and *start* index of their next defect;
- the *end* index if it is last defect;

IV Peaks and Valleys

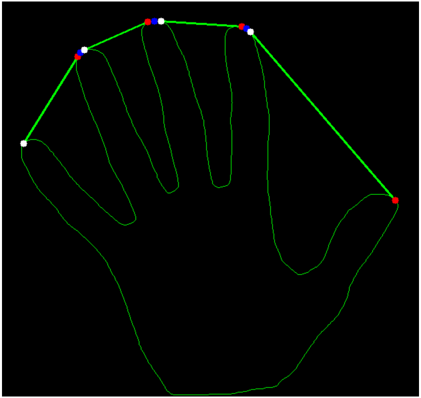


Figure: partial convex hull, white points are **start** points, red points are **end** points, blue points are finger points found by average of index of start and end (only for intermediate points.

IV Peaks and Valleys (Landmarks)

As we've seen in the theoretical introduction, here we've just detected some characteristic points for the shapes we're considering.

The only type of landmarks that will be used afterwords is peaks (finger points). Valleys are also important points that will be used to find another class of landmarks.

The location of the hand landmarks is an important source of information for all the studies we're going to do.

Note that this landmarks (finger points) are located at the contour.

V Middle finger point and center of mass

Having these Valleys and Peaks indexes, we can select middle finger point from Peaks list (the third will be middle finger index) and rotate the hand mask based on it and the centroid.

The rotation will affect all points of the mask, and also of contour but as we work with indexes nothing can happens to our indexes.

V Middle finger point and center of mass (before)

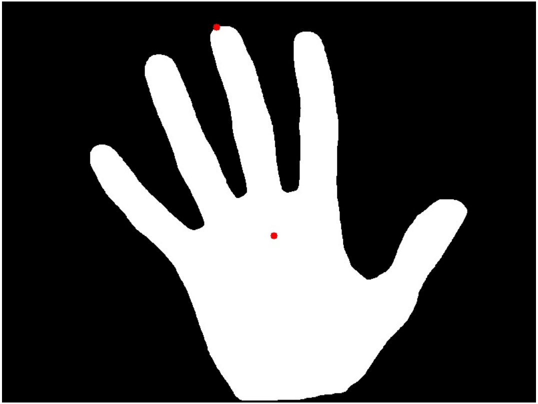


Figure: Based on middle finger point and center of mass we can rotate hand image in order to have vertical alignment of the axis passing from these two points

V Middle finger point and center of mass

Rotation formula is based on orientation angle of middle finger - centroid axes θ .

$$\theta = tan^{-1}\left(\frac{2\xi_{1,1}}{\xi_{2,0} - \xi_{0,2}}\right) \tag{13}$$

where $\xi_{\alpha,\ \beta}$ are the central moments, defined as follows:

$$\xi_{\alpha,\ \beta} = \sum_{\delta \in hand} (x_\delta - x_\mu)^\alpha (y_\delta - y_\mu)^\beta \tag{14}$$

V Middle finger point and center of mass

Calculate final angle ϑ :

$$\vartheta = \begin{cases} -(90+\theta), & 0 \leq \theta \leq 90 \\ 90-\theta, & \text{otherwise} \end{cases}$$

Than we need to rotate each contour's point, one by one as shown below depending on ϑ :

$$x'_{\varrho} = x_{\mu} + (x_{\varrho} - x_{\mu}) * \cos\vartheta - (y_{\varrho} - y_{\mu}) * \sin\vartheta \tag{15}$$

$$y'_{\varrho} = y_{\mu} + (x_{\varrho} - x_{\mu}) * \sin\vartheta + (y_{\varrho} - y_{\mu}) * \cos\vartheta \tag{16}$$

where (x'_{ϱ}, y'_{ϱ}) is the coordinate of any point ϱ on the contour, before rotation.

V Middle finger point and center of mass (after)

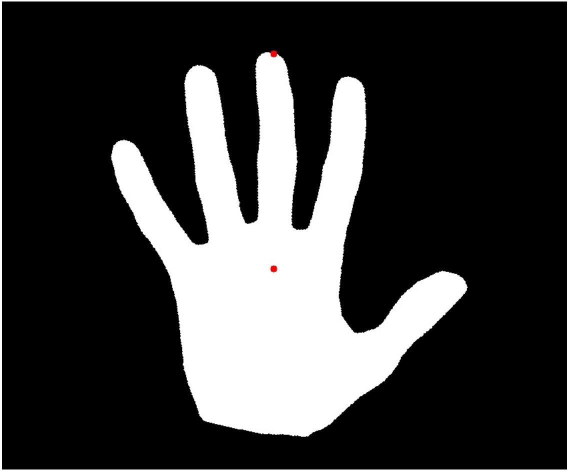


Figure: Hand mask with rotated contour

Rotation invariance

As we've seen in the theoretical introduction, with the operation we've just explained, we are able to get a model that is invariant to any rotation effect.

Whatever the original angle of the hand mask, this is rotated to get a vertical image.

Reference point

The coordinates of reference point (x_r, y_r) are calculated as the point of intersection between the line passing by the center of mass and middle finger point and the image contour in the opposite side.

Reference point

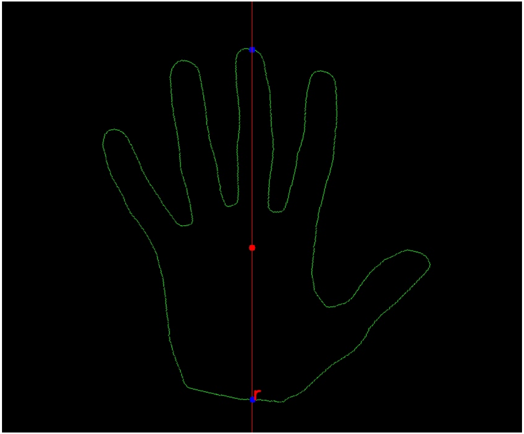


Figure: vertical middle finger-centroid axes, center of mass, contour, middle finger point and reference point (reference point marked with r)

Adjust points indexes

Now that we have the reference point, we can adjust all the points coordinates (rewrite the contour indexes, valley indexes, fingers indexes with respect to the reference point), so that the reference point becomes the contour component with index zero.

In this step only the indexes and the contour order have to be modified, and nothing happens to coordinates of contour's points.

Adjust points indexes

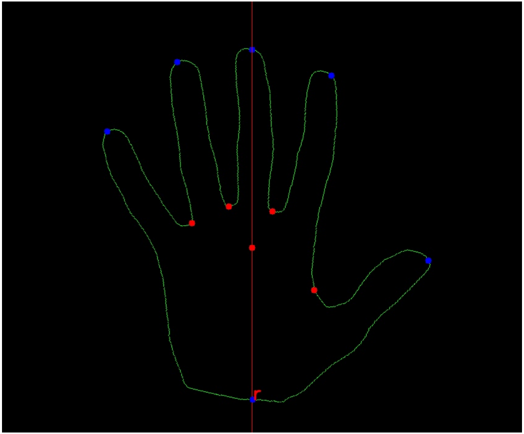


Figure: finger points and valley points

Complementary valley point

Once we have updated the contour to have it starting from reference point in anti-clockwise orientation, we can extract complementary valley points in order to calculate medium finger points.

Let's see how we can find them.

Complementary valley points

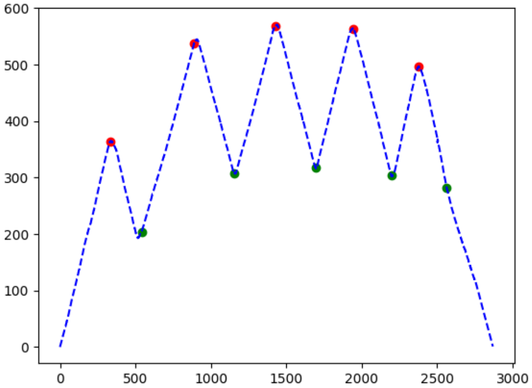


Figure: finger points and valley points

Complementary valley points

The image just seen shows fingers and valley points on a distance map created by calculating distance between each point of the contour and the reference point. In fact we can notice that 0-index point has distance 0, that is because we have updated contour to have it starts from the reference point.

The tecnique to find complementary valley points is based on indexes trick: count how many points there are between finger and valley points (like a distance), and marks a point that has the same distance to finger point in opposite side of the valley point.

Complementary valley points

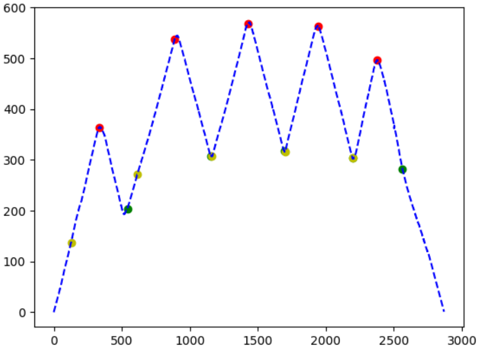


Figure: finger points, valley points and complementary valley points.

In some cases valley points and complementary valley points are exactly in the same point so we are not able to see them.

Medium points

Medium point are not points on the contour, but they are calculated starting from what we've just found.

Once we have the valley and the complementary valley point for each finger, we calculate the mean of these two points, and medium finger point is the result.

The mean formula we used is:

$$(m_x, m_y) = ((v_x + c_x)/2, (v_y + c_y)/2)$$

Complementary and medium points

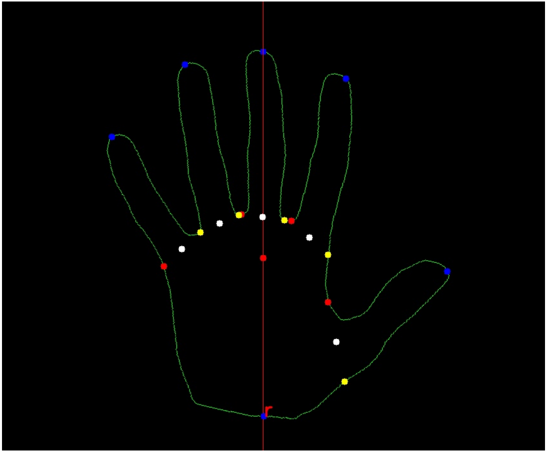


Figure: finger points, valley points, complementary valley points and white medium points

Final landmarks

As already said, valleys and complementary valleys are used to find medium points.

Medium points together with finger points (peaks) are the landmarks used for all the experiments.

Note that medium points are not located on the contour. So we have landmarks located both on the contour and inside the hand masks.

Traslation invariance

We consider now two different types of features, they are:

- geometrical features;
- shape features.

We consider for Geometrical features **medium points as reference**, and for Shape features **r as reference point**. Doing so all the hand masks that are used are invariant to translation (and rotation thanks to previously operations).

Geometrical features extraction

Distance calculation

The geometrical features of one hand image are based on the 5 distances calculated between the finger point and medium point of each finger. Together with these 5 distances we need two other ones in order to have geometrical features of one hand image.

The two other distances are:

- distance between medium point of little finger and medium point of index finger
- distance between medium point of index finger and medium point of thumb finger

Geometrical features

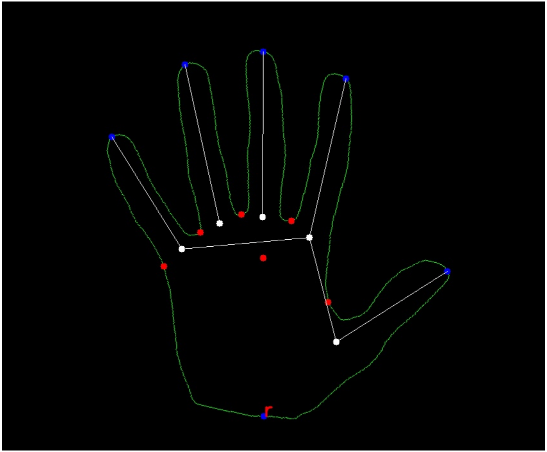


Figure: white lines represent the 7 distances used to calculate geometrical features

Geometrical features

The geometrical features are calculated as distances ratio between one distance and all the others.

This process generates 21 geometrical features for one hand mask.

Geometrical Features

0.8044	0.7930	0.8062	0.9910	1.2223	1.2293
	1.0260	0.9858	1.0021	1.2319	1.5162
		1.5195	1.2754	1.0165	1.2726
			1.2496	1.5413	1.2333
				1.2937	1.0352
					0.8393

Table: Upper triangle matrix composed by 21 geometrical features of one hand.

Shape features extraction

Fingers registration

As shown in the paper, we setted our W angles parameters. Our angles are different because we implements a generic function that has origin in vertical axes.

W is the following:

$$\omega_j = \begin{cases} +70 & ,j=1 \quad \text{little finger} \\ +80 & ,j=2 \quad \text{ring finger} \\ -80 & ,j=3 \quad \text{medium finger} \\ -60 & ,j=4 \quad \text{index finger} \\ -30 & ,j=5 \quad \text{thumb finger} \end{cases}$$

And after the rotation of all fingers, each i-th finger is based on W[i] angle around its medium point. We can see it in the next image.

Fingers registration

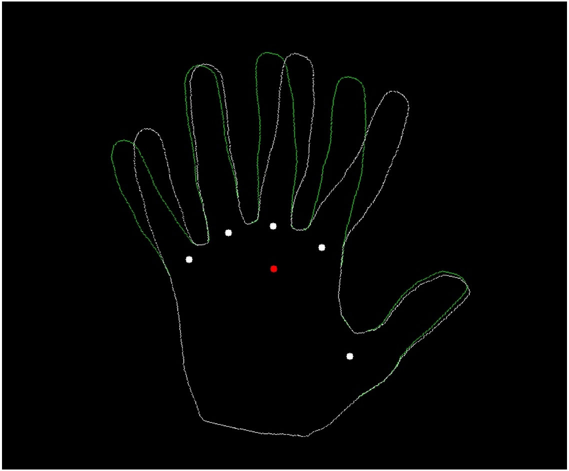


Figure: original contour and white updated contour

Rotation invariance

As we've seen in the theoretical introduction, with the operation we've just explained, we are able to get a model that is invariant to any finger rotation effect.

Whatever the original angle of the fingers, these are rotated to get a unified finger's angle position.

It is important to have every hand with open fingers in the same way.

Distance and orientation map

The distance map (*dp*) and orientation map (*op*) are generated using the reference point r and the updated contour.

dp and *op* of each boundary pixel from the reference point are calculated as:

$$dp(i) = \sqrt{(x'_r - x'_{{b_r}^{\mathrm{cw}}{}_{(\mathrm{i})}})^2 + (y'_r - y'_{{b_r}^{\mathrm{cw}}{}_{(\mathrm{i})}})^2} \tag{17}$$

$$op(i) = 90 + tan^{-1}\left(\frac{y'_r - y'_{{b_r}^{\mathrm{cw}}{}_{(\mathrm{i})}}}{x'_r - x'_{{b_r}^{\mathrm{cw}}{}_{(\mathrm{i})}} + \sigma}\right) \tag{18}$$

Distance map

As we can see the updated distance map is not very different from the older one. That is because the rotation of some part of finger points in a continued way does not cause discontinuity and brutal changes of position.

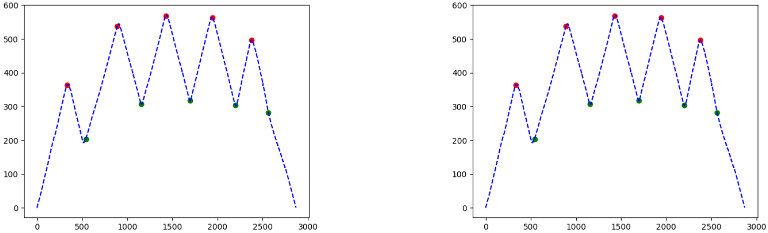
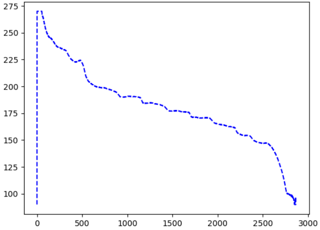


Figure: Example of distance map and updated distance map.

Orientation map

On the other hand, the updated orientation map is very different. In fact even if the distance has no changed a lot, the angle between all points and reference change based on W params.



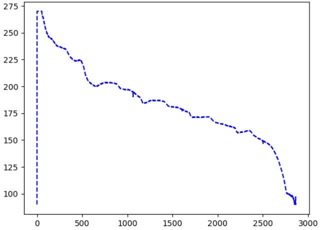


Figure: Example of orientation map and updated orientation map.

Wavelet decomposition

We applied **1-D wavelet decomposition at level 5 using Daubechies-1 wavelet filter** over *dp* and *op*.

We selected the first 50 coefficients of the wavelet decomposed *dp* and *op*. It is something like get *n* first coefficents of the interpolated function that interpolates your points, or get *n* first coefficents of Fourier transform.

Scale Invariance

It is observed that increase in the number of pixels of contour will not benefit in the discriminative ability of the *dp* and *op*.

Wavelet decomposition is used to transform the higher dimension feature into low dimension feature and to select most discriminative features (as we've seen in the theoretical introduction where this was made with SVD decomposition).

Both distance and orientation maps suffer from scale problem: they are very different for similar images if these have different scales (so different number of pixels in the contour).

Wavelet is a good solution to avoid this problem, doing so we obtain a model that is also invariant to scale.

Experiments

Dataset acquisition

We downloaded the right hand of 240 subjects with 5 images for each subject (totaling 1200 images) from IITD Dataset.

Figure: Some example images from the dataset

Verification and Identification

We introduce two scores distances that we will use: **Genuine** and **Imposter** scores: distances between the features of the hands of same person and different persons respectively.

Genuine
: We matched **each image pairs of the same person** to find genuine scores.

Imposter
: To generate the imposter scores, **the most representative image of each person** is collected and scores are generated for each pair of them.

Verification and Identification

We generated genuine and imposter scores for distance and orientation based shape features as well as geometrical features.

We present verification and identification using shape (distance and orientation) and geometrical features individually first and then we'll present the fusion method and the results.

But first let's introduce some definitions:

Performance measures

Definition

TAR is the True Acceptance Rate, it is measured as the number of occurrences when genuine images are matched correctly;

Definition

FAR is the False Accept Rate, measurement of the number of occurrences when imposter images are matched falsely;

Definition

FRR is the False Reject Rate and is measured on the basis of the number of false rejections of genuine matches and also given as:

$$FRR = 1 - TAR \tag{19}$$

Performance measures

Definition

EER is the Equal Error Rate, the point where FAR and FRR are equal.

Definition

DI is the Decidability Index, measurement of the distances between the genuine and imposter scores on basis of mean as well as standard deviation of both scores and defined as

$$DI = (|\mu_g - \mu_i|)/\sqrt{(\sigma_g^2 + \sigma_i^2)/2} \tag{20}$$

where, the genuine and imposter score means are represented by μ_g and μ_i respectively and the genuine and imposter scores standard deviations are represented by the σ_g and σ_i.

Performance measures

All of us knows what are the params True Positive (TP), False Positive (FP), True Negative (TN) and False Negative (FN).

Than see how simple is the calculation of FAR, FRR and TAR as:

- FAR = FP/(FP+TN)
- FRR = FN/(TP+FN)
- TAR = 1 - FRR

Performance measures

We calculate these measures on our dataset of images (knowledge base) to have verification results of our system.

For better verification system these parameters should be:

- TAR maximum
- FAR minimum
- EER minimum
- DI maximum

Similarity measures

What we need now is to calculate the Genuine and Imposter scores. To do so we perform the experiments with different similarity measures.

These are:
L1, Euclidean, Cosine and Chi-square (chisq or χ^2).

Evaluation procedure

Genuine

- The experiments consist on calculate distances between each genuine sets as the score of a couple of hand image.
- These distances are calculated between all features score we have (two shape and one geometrical).
- Than create from that a square matrix G of size nxn where n is the number of hand images we have for each person. At $G(i,j)$ cell there is the value of distance between the $i-th$ and the $j-th$ hand images of the same person.

Evaluation procedure

Imposter

- We calculate the centroid as the mean of all genuine scores (we refer to distances values as scores, from here).
- At this time we need to calculate distances between that mean score and all other centroid mean scores, one for each person.
- These distances are saved on a I square matrix of size mxm where m is number of different people. At $I(i,j)$ cell there is the value of distance between the $i-th$ centroid (centroid of $i-th$ person) and the $j-th$ centroid (centroid of $j-th$ person).

These steps are needed to have a way to perform FAR and FRR performance measures, and find a minimum EER.

Evaluation procedure (our proposed)

We use a big unique matrix B of dimension $nmxnm$ in order to count in a rapid way how many TP, FT, TN and FN there are.

We normalized our B matrix on min-max normalization method defined as

$$B'(i,j) = \frac{B(i,j) - min(B)}{max(B) - min(B)} \tag{21}$$

where $min(B)$ and $max(B)$ are defined as minimum and maximum value on B cells.

Evaluation procedure (our proposed)

In B, all diagonal sub matrixes every n rows-columns are Genuine G matrix, and all the upper triangular matrix of B minus diagonal G matrixes is called UB.

We set a proximity threshold, then every image that has lower values than the threshold his value will be set to 1 (these are hand images of the same person) or 0 if the value in higher than the threshold (these are hand images of the same person)
For the entire G genuine matrix we count the TP and FN as the number of 1s and 0s respectively.

We only consider the upper triangular matrix because it is necessarily simmetric, without counting diagonal cells (these are always 1, obviously).

For the UB we count the TN as 0s cells and FP as 1s cells.

Evaluation procedure (our proposed)

Inferior orange and white triangluar sub matrixes cells are not considered, red diagonal is always 1, upper orange and white cells are considered.

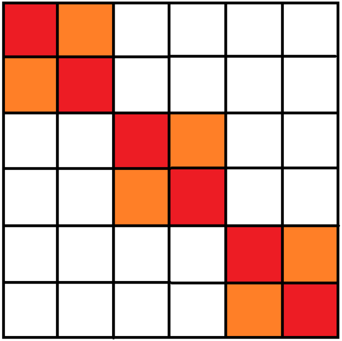


Figure: example with 3 people, two hand images each person

Thresholds

The verification method is based on finding TP, FP, TN, FN and it is possible only if we can set a threshold. We would like to have B matrix distances composed by 0 and 1, with a $B(i,j)$ set 1 if the $i-th$ and $j-th$ scores are very similar, 0 otherwise.

To do so we start with a threshold set to 0, and then we calculate the new matrix (of 0s and 1s) and the value of TP, FP, TN, FN. We increase that threshold up to 1, due to previous normalization, and save all data.

Score fusion

The score level fusion is applied both on genuine and imposter scores, in two levels:

First fuse genuine and imposter scores of the distance based shape features with the genuine and imposter scores of the orientation based shape features and computed the genuine and imposter scores after level 1 fusion.

Second in the second step of fusion the genuine and imposter scores obtained after first level fusion are fused again with the genuine and imposter scores of the geometrical features respectively to get the final fused genuine and imposter scores.

Score fusion

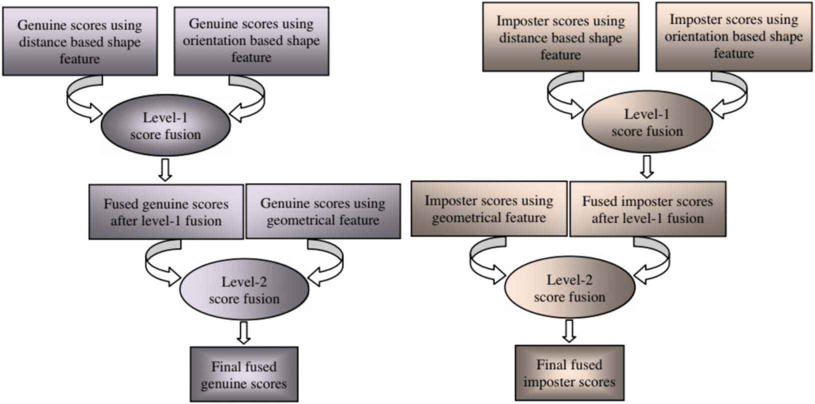


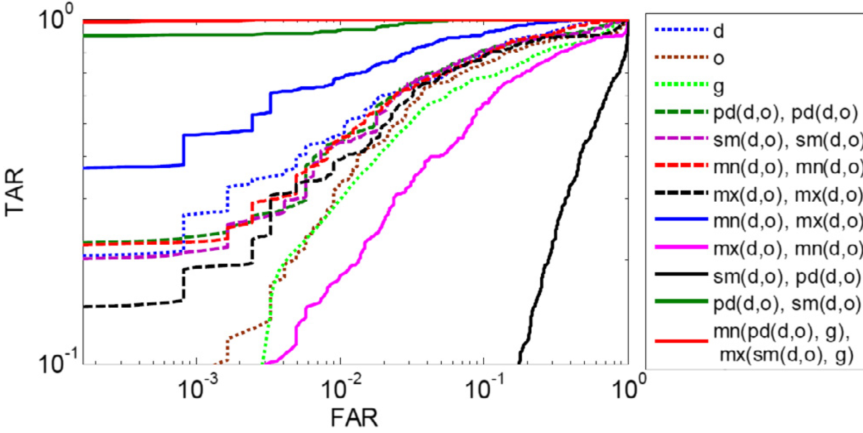
Figure: Two level score fusion of genuine and imposter scores using shape and geometrical features.

Score fusion

The score level fusion is done using the scores of shape distance, shape orientation and geometrical features.

After various tests, we found that multiplication (pd), summation (sm), minimum (mi) and maximum (mx) should be good operation for score fusion.
We combine all of these operations in all possible combinations in order to get the hightest ROC value while we fused shape and geometry features scores.

Score fusion



Score fusion

First we applied the fusion over scores of ‘d’ (distance map score) and ‘o’ (orientation map score) and we use pd(d, o) and sm(d, o) as combinations for genuine and imposter score fusion respectevely.

We fused the fusion result of ‘d’ and ‘o’ with the scores of ‘g’ using minimum and maximum rule for genuine and imposter scores.

- mn(pd(d, o), g) for genuine
- mx(sm(d, o), g) for imposter

Score fusion

Now we have genuine and imposter method to fuse the respective scores, we normilize the result fused genuine and imposter matrix, and apply the varing threshold on these.

Threshold

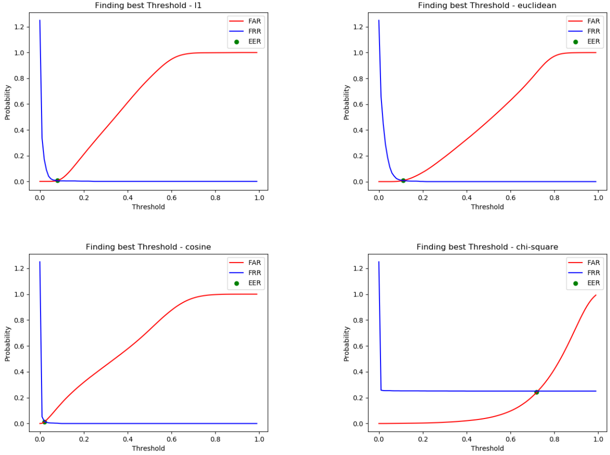


Figure: Threshold graph of fused scores with differet metric measure

Threshold

Now we have to set the threshold of the best measurements and use it to distinguish new images with respect to these similarity scores with the images of imposter.

ROC graphs

We need to choose only one similarity measure to use, and in order to get more performance measure we calculate a ROC graph: a graph with FAR and TAR in x and y axes respectively.

We would like to measure different features behaviour (shape, geometrical and fusion) with a particular similarity measure.

To do that let's see the following ROC graphs:

ROC graphs

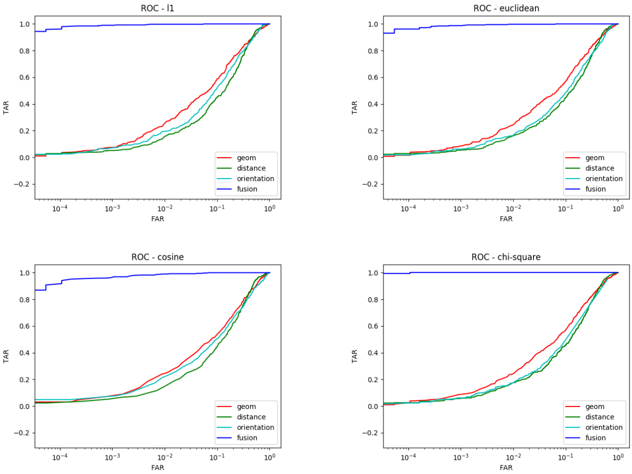


Figure: ROC plots with FAR x axis, TAR y axis

ROC graphs

Based on previously ROC plots we choosed both L1 and cosine distance measures.

Identification method

For identification, we start applying on the external images all the preprocessing we did on dataset images. We need shape and geometrical features, and than we can calculate the score (with both L1 and cosine distances) between this external images and all the centroids means of imposter set.

Now we take the threshold we set before to binarize the scores. We marks as one person hand image if the score distance value in 1 after threshold application. If there are more than 1 options open, we take the smaller one.

Results

Test image	score	dataset candidates	measure
001	fusion	001, 002	l1, euc, cos
001	fusion	003, 002	chi
002	fusion	002, 001	l1, euc, cos
002	fusion	003, 001	chi
003	fusion	003	l1, euc, cos
003	fusion	001, 002	chi

TEST success: 9

TEST insuccess: 3

Results

Try your own on Thursday.

Conclusion

Thanks for your attention !