Utrecht University

Evaluation of Existing Methods for Earprint Recognition

Hiroo Azadi

3439879

Information and Computing Sciences Department

Faculty of Science

Utrecht University

Supervisor: Prof. Remco Veltkamp

January 2014

Table of Contents

Abstract……. 1

Chapter 1 Introduction……..... 2

1.1 Goal…… 4

1.2 Overview…..……... 4

Chapter 2 Database and Performance Measures……..…….. 6

2.1 Design of the database……. 6

2.2 Performance measures……. 6

2.2.1 ROC curve and Equal Error Rates (EER).…….. 7

2.2.2 Hitlist behavior…… 8

Chapter 3 Scale Invariant Feature Transform (SIFT)……..... 9

3.1 Creating the Difference of Gaussian Pyramid..…….. 9

3.2 Extrema Detection..…… 10

3.3 Keypoints Elimination..……..…… 11

3.4 Orientation Assignment..…… 12

3.5 Descriptor Computation..…… 13

3.6 Experiment Result……..……. 13

3.6.1 Pre-processing of images……. 13

3.6.2 Algorithm……. 14

3.6.3 Matching…….. 15

3.7 Evaluation of performance.……. 16

3.7.1 EER…… 16

3.7.2 Hitlist behavior…..…… 18

Chapter 4 Curvature Scale Space (CSS).…… 20

4.1 The CCS representation…… 20

4.2 Extracting maxima of CSS contours…… 22

4.3 CSS matching…… 22

4.4 Experiment Result……. 24

4.4.1 Pre-processing of images..……. 26

4.4.2 Algorithm..…… 26

4.4.3 Matching…….. 26
4.5 Evaluation of performance…… 27
4.5.1 EER..…….. 27
4.5.2 Hitlists...…… 28
Chapter 5 Intensity-based Image Registration...…….. 30
5.1 Images...…….... 30
5.2 Metrics...……... 31
5.3 Interpolators...…….. 33
5.4 Transforms...……. 33
5.5 Optimizers..…….. 35
5.6 Multi-Resolution..……..... 36
5.7 Experiment Result..……. 37
5.7.1 Pre-processing of images..…….…… 37
5.7.2 Algorithm..…… 37
5.7.3 Matching…… 37
5.8 Evaluation of Performance..……..... 39
5.8.1 EER…….... 39
5.8.2 Hitlists..……..... 41
Chapter 6 Conclusion…….. 43
References…… 45

List of Figures

Fig. 1. Example of an annotated earprint……..……. 3

Fig. 2. Example: original print, clicked polyline and calculated ‘superstructure’….…….. 4

Fig. 3. Variants of the ROC plot…….……..…….. 8

Fig. 4. An example of hitlist.…….…….. 8

Fig. 5. Creating the Difference of Gaussian Pyramid…….…. 10

Fig. 6. Extrema Detection…….…... 11

Fig. 7. Orientation Assignment……..... 12

Fig. 8. Descriptor Computation……... 13

Fig. 9. Illustration of the pre-processing of images……. 14

Fig. 10. Earprint with Sift features…….…. 16

Fig. 11. Matched keypoints including outliers and inliers……..… 16

Fig. 12. ROC curve for SIFT before applying RANSAC.……. 17

Fig. 13. Results matrix showing the number of matching keypoints…….. 18

Fig. 14. Smoothing of the curve and the CSS image of the shape…….. 21

Fig. 15. Procedure of earprint recognition using CSS..…….. 24

Fig. 16. Procedure of curvature scale space…… 25

Fig. 17. ROC curve for CSS cost function.……. 28

Fig. 18. Results matrix showing the CSS cost..…….. 29

Fig. 19. Fixed image and moving image in image registration…….. 30

Fig. 20. Basic registration components…….. 31

Fig. 21. Different transformations…….. 34

Fig. 22. Iterative optimization…….…… 35

Fig. 23. Multi-resolution strategies using a Gaussian pyramid……...…… 36

Fig. 24. Optimal alignment illustrated by the RGB image..…… 38

Fig. 25. ROC curve for DSC metric……...…….…… 40

Fig. 26. ROC curve for CC metric……..…… 41

Fig. 27. Results matrix showing the correlation coefficient……..…. 42

Abstract

Ear as a part of human body has been used in forensic practice but the use of earprints as evidence in criminal trials remains arguable. The Forensic Ear Identification (FearID) research project was started in order to study the strength of evidence of earprints found on crime scenes. A limited number of publications exist related to computerized methods used for earprint identification. The study presented here compares existing methods for earprint image identification. Two of them are point based methods that use Scale Invariant Feature Transform (SIFT) feature and Curvature Scale Space (CSS) feature for recognition process. The other is based on the image intensity value that compares two images by registration algorithm. We applied two different similarity metrics for evaluating registration. All methods carried out on a subset of FearID database. Equal error rate and hitlist behavior on our small dataset show that CSS is not useful in earprint recognition whereas SIFT and image registration technique have promising results.

Chapter 1

Introduction

Several signs are used in a crime scene to identify the criminal who committed crime. These can be fingerprints, DNA material, shoe prints, earprints and so on. When criminal listen at a window or a door, oils on the ear leave a print that can be made visible using some techniques. Until now, due to a lack of scientific basis, earprints have been used very few times and most of the time without much success. A good review article on the lack of scientific research up to 1999 with respect to earprint identification can be found in [1]. The three cases discussed briefly below highlight this point.

In the US, in the case against David Wayne Kunze [2], the Appeal Court considered the admissibility of earmarks as evidence, according to the Frye general acceptance standard. The court concluded that earmarks failed the Frye criteria considering that identification based on earmarks are not generally accepted by the scientific community.

In the case of Mark Dallagher [3], in his first trial the defendant was sentenced to life imprisonment on the basis of earmark evidence. The two experts called to testify on behalf of the prosecution both expressed the view that Mark Dallagher was the source of the recovered earmarks, either in absolute terms (‘‘identification’’) or by degree (‘‘very likely’’). The decision was reversed six years after the conviction on the ground that the Court had, at the time of the first trial, no opportunity to hear a dissenting view. This was caused by the shortage of experts in this specialized area available for the defense. On appeal, it was also argued by the defense that there was a paucity of empirical research to support a conclusion based on the comparative examination of earmarks against earprints.

The UK court of Appeal also considered the case against Mark Kempster [4]. During this last appeal, the Court faced a prosecution expert who identified Kempster based on a mark recovered from the scene. The defense expert testified in favor of exclusion because of the presence of differences that could not easily be reconciled on the ground of distortion or other deposition factors.

To provide the scientific basis, the Forensic Ear Identification research project (FearID), a collaboration of several European institutes, was initiated a couple of years ago. The FearID research project aims to obtain estimators for the strength of evidence of earprints found on crime scenes and the development of methods to match and classify earprints. For this purpose earprints from over 1229 donors from three different countries have been collected to set up a database. Standard operating procedures were designed for the recovery and lifting of donor earprints, laid down in [5]. During the FearID research project, several methods have been developed which determine the similarity between two earprints. E.g. Alberink and Ruifrok [6] developed a semi-automatic recognition system. In their work, the digital images are processed in three different ways: Annotated print (Fig. 1), Operator axis (Fig. 2) and Medial axis to form the feature vector. The medial axis is a polyline derived from the superstructure.

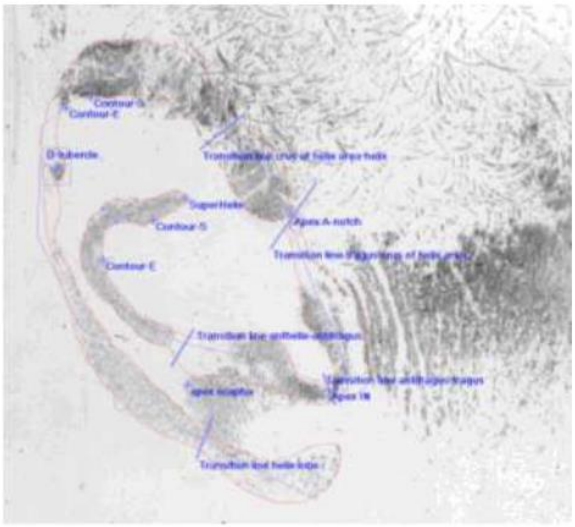


Fig.1. Example of an annotated earprint.

Fig. 2. Example: original print, clicked polyline and calculated ‘superstructure’.

1.1 Goal

The goal of this project is to compare a set of methods for earprint recognition. The main task is performance comparison between point based approach and image based approach. Example of point based approach are Scale Invariant Feature Transform (SIFT) [7] and Curvature Scale Space (CSS) [8] that extract some key features from the image and image registration technique as an image based method that works based on the superimposition of images. Image registration is performed using an iterative multi-resolution algorithm, based on the image intensity [9].

1.2 Overview

The first chapter gives an introduction and the main goal of our project.

In chapter two the design of FearID dataset and the ways in which performance is reported (equal error rates and hitlist behaviour of the system) is briefly described.

Chapter three describes Scale Invariant Feature Transform (SIFT) features extracted from images to help in reliable matching between earprint images. The extracted features are invariant to scale and orientation, and are highly distinctive of the image. They are extracted in four steps that are described in more detail in this chapter. Experimental result and performance measure are shown by graph and table.

In chapter four the other method of recognition is introduced: Curvature Scale Space (CSS). The original images are used to represent the shapes of boundary contours of earprint. Then, the CSS representations of the contours of earprint images are computed. From that we extract the CSS features. Finally, CSS matching performs between a set of input CSS features and the stored CSS features for earprint identification. Performance is represented in the form of EER and hitlist behavior.

Chapter five introduces image registration technique used in this project. Image registration is the process of aligning two images into a common coordinate system. It computes transformations to set correspondence between the two images. The registration process requires specifying a pair of images, one as the fixed image f(x) and the other as the moving image m(x) as input images, a transform, a metric, an interpolator and an optimizer. This chapter will be ended with the achieved experimental results.

Chapter six concludes the whole project and the performance of the given methods are compared.

Chapter 2

Database and Performance Measures

To enable earprint to be evaluated in different methods we require a database contains earpint images coming from the same and different sources. On the basis of the above, we implemented some methods for extracting useful features in the matching process of earprint images. Performance of these methods should be reported in some ways that are discussed as follows.

2.1 Design of the database

The database consists of 7364 prints of 1229 donors, taken according to the standard procedures to ensure that all prints have the same (high) quality. This means that all donors provided first three left, then three right earprints on the FearID ‘listening box’, which would be consecutively lifted using Black Gel Lifter. 216 of 7364 prints are without instructions to the donor and have low quality and function as simulated crime scene marks which are called marks.

The original images contained in the FearID dataset are at 600 dpi in greyscale (8 bits) and dimension 2100*3000. Access to this large-scale dataset was not possible due to the difficulty in data transferring. On the other hand doing the operation on the small dataset can be much faster and less expensive than the large one. The following numbers of earprint images are in our database as described in Table1.

Table1. Number of prints in our database.

Side of ear	Earprint images
Left	9
Right	9

2.2 Performance Measures

Performances were measured on the query prints, the prints to test the system, using ROC curve, equal error rates and hitlist behavior.

2.2.1 ROC curve and Equal Error Rates (EER)

Receiver Operating Characteristic (ROC) is a standard information retrieval curve in order to visualize the quality of the classification system. Here we measure the performance of a classification system with two classes of outcomes: matching (or positive or acceptance) and non-matching (or negative or rejection).

The ROC curve is the parametric curve given by the false positive rate (FPR) against the false negative rate (FNR). False positive corresponds to the cases in which the system declares a match in case of non-matching prints (also called a false acceptance). False negative corresponds to the cases in which the system declares a non-match in case of matching prints (also called a false rejection). Since the FPR and FNR are threshold-dependent, we rather use the equal error rate (EER), which is the point on the ROC curve that corresponds to have an equal probability of miss-classifying a positive or negative sample (FPR(t) = FNR(t) with threshold t). This point is obtained by intersecting the ROC curve with a diagonal of the unit square. The lower the equal error rate value, the higher the accuracy of the system. The Area Under the Curve (AUC) is another indicator of the overall quality of a ROC curve. For example, the ROC of the ideal classifier has AUC equal to 1. ROC rand would be the performance of a random classifier (in the limit of infinite data). An example of this is given in Fig. 3.

According to VLFeat library [10] different variants of the ROC plot can be produced (Fig. 3). In this paper we used false positive/false negative plot. However the AUC is probably a misnomer for that plot. In practice, the function reports always the same AUC measure regardless of the particular variant of the plot. Anyway we did not use AUC and ROC rand in our calculations.

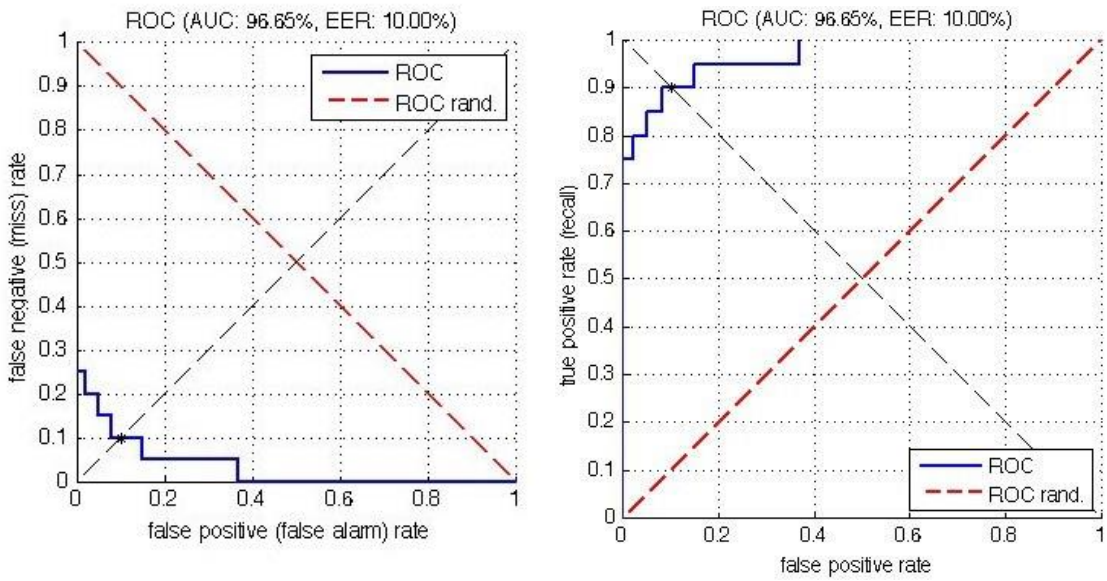


Fig. 3. Variants of the ROC plot.

2.2.2 Hitlist behavior

The second way in which the performance of the system is tested is by looking at hitlist behavior of the system. Here, query prints are compared to a reference database and for any query print the relative position of the corresponding matching print that comes highest in the hitlist is noted. A possible result of a query search is depicted in Fig. 4 with two matching prints at places 4 and 15 and the best hit at place 4. In the current setting, we have two reference databases, for left and right prints, respectively; those consist of two prints per ear, and use the remaining prints as query prints to test the system. We are not so much looking for a matching print of the query print at place 1 of the hitlist, but rather in, the top 3.

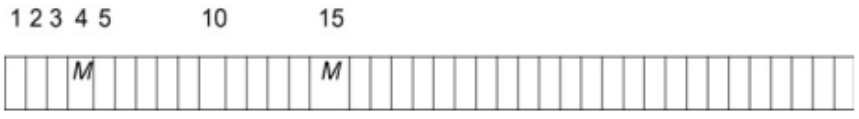


Fig. 4. An example of hitlist.

Chapter 3

Scale Invariant Feature Transform (SIFT)

Scale Invariant Feature Transform (SIFT) is an algorithm presented by Lowe [7] takes an image, detect keypoints and compute its descriptors. These are invariant to scale, rotation and translation of the image.

Meijerman et al. [11] applied Sift features to automatically match earprints. Their method consists of the following steps: image preprocessing to resample the images and apply filter, keypoint detection using the Sift algorithm of Lowe, keypoint matching defined as the minimum Euclidean distance in the Sift feature space, and similarity metric definition as the number of accepted keypoint matches found for a pair of images [11]. The method has been initially tested on a limited sample of 36 right earprints from six pairs of identical twins. It has been given promising hitlist results in which a print from the correct individual is ranked at the top of the hitlist in 36 out of 36 cases.

Here is an outline of what happens in Sift.

3.1 Creating the Difference of Gaussian Pyramid

The first stage is to construct a Gaussian scale space from the original image [7]. For this, the image is convolved with Gaussian functions of varying widths, and then the difference of Gaussian (DoG) images are taken as the difference between two filtered images, one with k multiplied by scale of the other.

$$D(x, y, \sigma) = L(x, y, k\sigma) - L(x, y, \sigma)$$

Where $L(x,y,\sigma)$, is the convolution of Gaussian functions, $G(x,y,\sigma)$, with an original image, $I(x,y)$.

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y)$$

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} \exp\{-(x^2 + y^2)/2\sigma^2\}$$

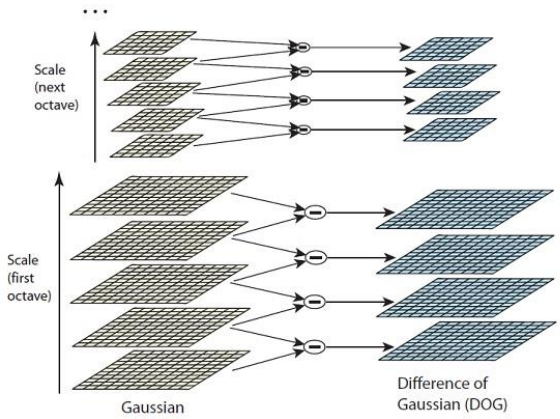


Fig. 5. Gaussian pyramid in the left with neighboring images separated by a constant scale factor *k*. Difference of Gaussian (DoG) pyramid on the right.

An efficient approach to construction of $D(x, y, \sigma)$ is shown in Fig. 5. The convolved images are grouped by octave and separated by a constant scale factor *k*. Each octave is divided into an integer number, *s*, of intervals, so $k = 2^{1/s}$. Then the Difference of Gaussian (DoG) images are taken from subtracting the adjacent Gaussian-blurred images per octave. Once an octave has been completed, the Gaussian image is down-sampled by a factor of 2, and the process repeated.

3.2 Extrema Detection

In this stage extrema points are identified in the DoG pyramid. This is done by comparing each pixel in the DoG images to its eight neighbors at the same scale and nine corresponding neighboring pixels in each of the neighboring scales (Fig. 6). If the pixel value is the maximum or minimum among all compared pixels, it is an extrema [7]. Subpixel extrema are found by the Taylor expansion of the image around the approximate keypoint.

$$D(\mathbf{x}) = D + \frac{\partial D}{\partial \mathbf{x}}^T \mathbf{x} + \frac{1}{2}\mathbf{x}^\mathbf{T}\frac{\partial^2 D}{\partial \mathbf{x}^2}\mathbf{x}$$

Taking the derivative of this function with respect to x and setting it to zero, gives the true extrema point.

$$\hat{\mathbf{x}} = -\frac{\partial^2 D}{\partial \mathbf{x}^2}^{-1}\frac{\partial D}{\partial \mathbf{x}}.$$

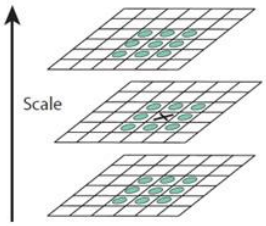


Fig. 6. An extrema is detected as a minimum or maximum value between all its neighbours in DoG scale.

3.3 Keypoints Elimination

Some keypoints generated in the previous step lie along an edge, or they don't have enough contrast. In both cases, they are not useful as features. So in this stage they will be eliminated [7].

The value of the keypoint in the DoG pyramid at the extrema is given by:

$$D(\hat{\mathbf{x}}) = D + \frac{1}{2}\frac{\partial D}{\partial \mathbf{x}}^T \hat{\mathbf{x}}.$$

If the value at x in the DoG image is less than a threshold, it is rejected.

In Keypoints that are located on edges the principal curvature across the edge would be much larger than the principal curvature along it. So they should be eliminated from the keypoint list. A 2x2 Hessian matrix, *H*, computed at the location and scale of the keypoint is used to compute the principal curvature.

$$H = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix}$$

$$\frac{D_{xx} + D_{yy}}{D_{xx}D_{yy} - (D_{xy})^2} < \frac{(r+1)^2}{r} \qquad (1)$$

So if inequality (1) fails, the keypoint is rejected.

3.4 Orientation Assignment

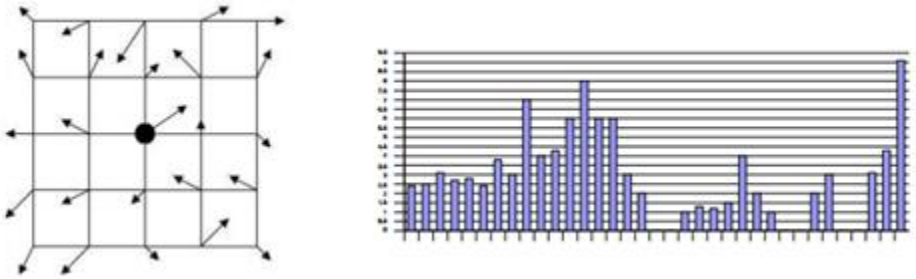


Fig. 7. Left: An orientation is assigned to the candidate keypoint to achieve invariance to image rotation. Right: An orientation histogram with 36 bins covering 360 degrees is created.

In this step, one or more orientations are assigned to each keypoint based on local image gradient directions. First, For an image sample *L(x,y)* at scale σ, the gradient magnitude and orientation are calculated using these formulae:

$$m(x, y) = \sqrt{(L(x+1, y) - L(x-1, y))^2 + (L(x, y+1) - L(x, y-1))^2}$$

$$\theta(x, y) = \arctan((L(x, y+1) - L(x, y-1)) / (L(x+1, y) - L(x-1, y)))$$

The magnitude and orientation is calculated for all pixels around the keypoint. Then, a histogram is created for this [7]. In this histogram, the 360 degrees of orientation are broken into 36 bins (each 10 degrees) as illustrated in Fig. 7.

Each sample is weighted by its gradient magnitude and by a Gaussian-weighted circular window with a σ that is 1.5 times that of the scale of the keypoint.

The histogram will have a peak at some points. The peaks in this histogram correspond to dominant orientations. The orientations corresponding to the highest peak and local peaks that are within 80% of the highest peaks are assigned to the keypoint. In the case of multiple peaks, a new keypoint is created having the same location and scale as the original keypoint. But it's orientation is equal to the additional peak. So, orientation can split up one keypoint into multiple keypoints.

This is the key step in achieving invariance to image rotation. Till now the keypoints are invariant to image location, scale and rotation. In Fig. 7, keypoints are indicated as arrows. The length of the arrows indicates the magnitude of the contrast at the keypoints.

3.5 Descriptor Computation

In this stage, a descriptor vector for each keypoint is computed such that the descriptor is highly distinctive. To do this, a 16×16 window is computed around the keypoints [7]. It is divided into 16 sub-blocks of 4×4 sizes. Within each 4×4 window, gradient magnitudes and orientations are calculated. These orientations are put into an 8 bin histogram. The magnitudes are further weighted by a Gaussian function with σ equal to one half the width of the descriptor window. Since there are $4 \times 4 = 16$ histograms each with 8 bins, the vector has 128 elements. This 128-dimensional vector forms keypoint descriptor that is then normalized to unit length in order to achieve more invariance. This process is indicated in Fig. 8.

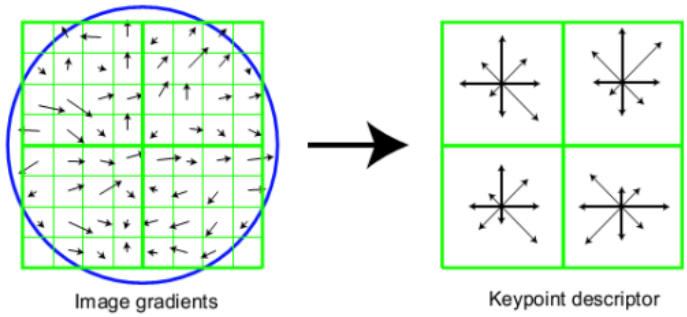


Fig. 8. Left: The magnitude and orientation values of samples in a square region around the keypoint. Right: The orientations are put into an 8 bin histogram indicated by the arrows and is computed from 4×4 sub regions.

3.6 Experiment Result

3.6.1 Pre-processing of images

As explained in section 2.1, the images used by us for testing and comparing the SIFT keypoints and descriptors are 18 earprint images of 6 persons with 3 images per person taken from FearID dataset. The original images are at 600 dpi in greyscale (8 bits) and 2100*3000 in dimensions

(Fig.9, left). For the speed optimization purpose the images were resampled to 220*300. The region of interest (ROI) is specified by drawing a polygon on the image at any location of interest (Fig.9, right) then denoising filter are applied. Finally the images are manually rotated to align vertically (Fig.9, middle).

Each print in the database which follows the same pre- processing to obtain ROI was taken in turn as a query print or a test image. The image is then compared to a database containing other prints. Therefore considering the number of prints in the database, 81 independent runs were carried out for each side of ear.

Fig. 9. Illustration of the pre-processing of images.

3.6.2 Algorithm

Matlab was used to implement the SIFT matching algorithm. The code for extracting SIFT features was available from David Lowe's website [12].

3.6.3 Matching

Given two prints with extracted keypoints, the matching algorithm is as follows:

First, SIFT features are obtained from each print automatically using the SIFT algorithm described above. Fig. 10 shows the detected SIFT keypoints. Each feature in the test image or query print is matched to all the SIFT features obtained from the other print. This feature matching is done through a Euclidean-distance based nearest neighbor approach. A feature is considered the best matched with another feature when the ratio between the distance to that feature and the distance to the next nearest feature is minimum. But this matching gives many wrong matches (outliers). In order to remove the outliers, the keypoints with low ratio (less than 0.6) are used to find the transformation model. This transformation (rotation and translation) are used to improve the number of good matches found between the images. For this purpose, the pixel distance threshold should be set on the found transformation to remove bad keypoint matches.

We defined the number of accepted keypoints as a similarity metric to compare pair of images. We expect to have few keypoint matches in the prints come from the different sources however prints with the same source have many keypoints matches [11].

We found the matches consistent with a geometric transformation using RANSAC (Random Sampling and Consensus) algorithm [13] implemented by the function *GeometricTransformEstimator* [14]. So many wrong matches (outliers) automatically will remove in this way. Fig. 11 represents the matched keypoints before and after applying RANSAC.

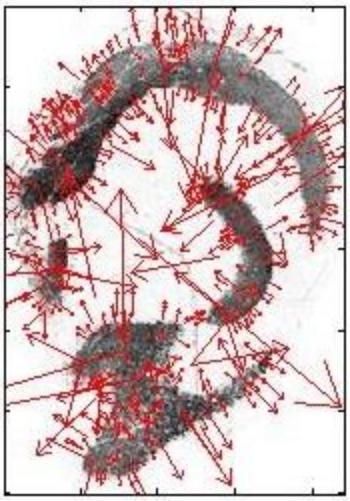


Fig. 10. Earprint with Sift features.

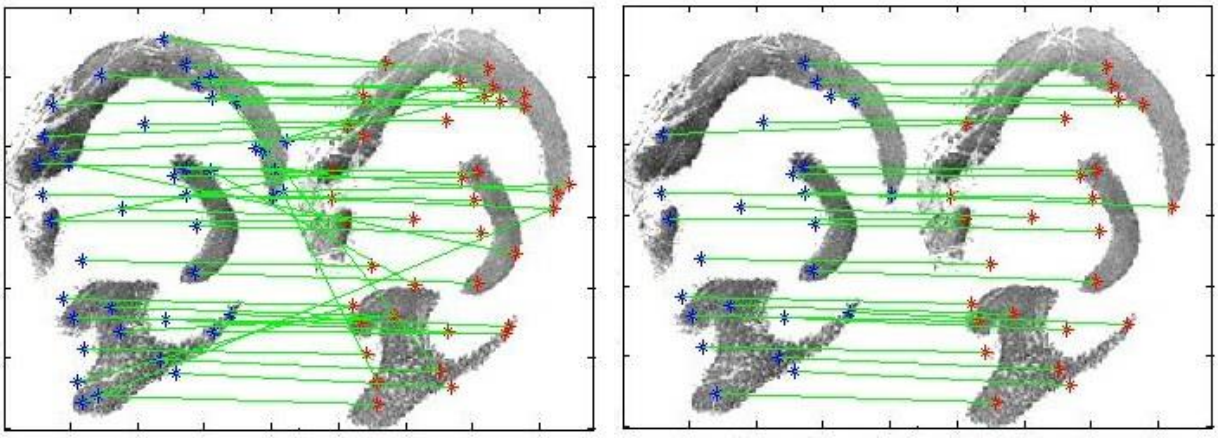


Fig. 11. Matched keypoints including outliers (left) and matched keypoints including inliers only (right).

3.7 Evaluation of performance

3.7.1 EER

We evaluate the system performance by comparing a query print to all prints in the reference database. The results are shown using false positive and false negative rate in a ROC curve. The value of equal error rate on the ROC curve indicates the performance. The smaller this rate, the more efficient the system.

Fig. 12 shows the ROC curve for the similarity metric. Graph in green color is for left ear and blue is for right ear. The difference between EER for left prints and right prints shows that the discrimination of left ears is better than that of right ones before applying RANSAC.

Results in terms of equal error rates (EER), for all comparisons are given in Table 2. Note that a value of EER = 0 indicates perfect detection.

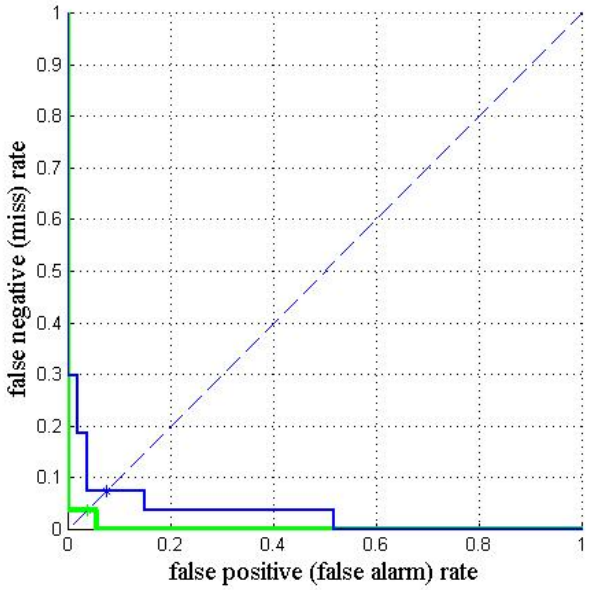


Fig. 12. ROC curve for SIFT before applying RANSAC.

Table 2. EER performance of the system.

Results obtained with **SIFT** using our dataset		
Sample	before RANSAC[%]	after RANSAC[%]
Right	7.41	0.0
Left	3.70	0.0
Total	**5.55**	**0.0**

3.7.2 Hitlist behavior

For the method the numbers of keypoint matches found for each print comparison are provided in a results matrix (Fig. 13). In Fig. 13 the three prints from each donor are grouped together and the two or more comparisons that give the highest keypoint matches are highlighted.

Comparisons between prints from different donors show few keypoint matches. Among this group, only 1 of 54 comparisons showed ten keypoint matches and no comparison showed more than 10 matches. Conversely, comparisons between prints from the same donor show relatively high numbers of keypoint matches. Among this group, 18 from 18 comparisons showed the highest matches; the maximum number of matches was 53.

The results shown in Fig. 13 can be investigated in the hitlist. All comparisons made for a query print were ranked by the number of keypoint matches in descending order. As a result, a print from the correct donor is ranked at the top of the hitlist in all cases (i.e., 9 out of 9 prints). Also in 100% cases, the two prints from the correct donor are returned as the first two prints in the hitlist.

		Donor 1			Donor 2			Donor 3		
		P1	P2	P3	P1	P2	P3	P1	P2	P3
Donor 1	P1		10	22	5	4	4	5	4	3
	P2	10		13	4	7	10	4	5	4
	P3	19	13		5	5	4	4	3	5
Donor 2	P1	4	3	4		13	19	6	5	5
	P2	4	3	4	14		30	6	4	5
	P3	4	4	4	16	26		7	5	4
Donor 3	P1	3	4	4	6	7	8		48	27
	P2	6	4	6	6	7	7	53		36
	P3	5	3	5	7	6	5	28	33	

Fig. 13. Results matrix for 9 left earprints showing the number of matching keypoints found for each print comparison. Two or more matches are highlighted in the matrix and the print matched with itself left blank.

Hitlist results for all comparisons are presented in Table 3 and Table 4. For each query print, its matching position in the list is noted. These results therefore show the percentage of query print, according to its matching position in the list. Thus, for right and left earprint comparisons with RANSAC, 100% of the first matches and 100% of the second matches are in the first and the two position in the list. Whereas 77.7% of the second matches is in the two position and 22.2% of the matches in the three position before applying RANSAC.

Table 3. Hitlist results including outliers using Sift descriptor in our dataset.

Searches			
Position	Left	Right	Overall
1	9 out of 9 (100%)	9 out of 9 (100%)	100%
2	8 out of 9 (88.8%)	6 out of 9 (66.6%)	77.7%
3	1 out of 9 (11.1%)	3 out of 9 (33.3%)	22.2%

Table 4. Hitlist results including inliers only using Sift descriptor in our dataset.

Searches			
Position	Left	Right	Overall
1	9 out of 9 (100%)	9 out of 9 (100%)	100%
2	9 out of 9 (100%)	9 out of 9 (100%)	100%
3	0 out of 9 (0.0%)	0 out of 9 (0.0%)	0.0%

Chapter 4

Curvature Scale Space (CSS)

In many applications, the user wishes to retrieve similar images from the database. The goal is to capture image information in the form of feature vector which describe shape, texture and color properties of the image. These vectors are compared to one another to find images from the database. A considerable amount of information exists in two dimensional boundaries of objects which enable us to recognize objects without using further information. Curvature scale space is an efficient and robust shape representation method. The CSS image consists of several arch-shape contours representing the inflection points of the shape as it is smoothed. The maxima of these contours are used to represent a shape [8].

The construction of the CSS image and the properties of using its maxima as the shape representation are explained as follows.

4.1 The CCS representation

Consider a parametric vector equation for a curve:

$$\boldsymbol{\Gamma}(u) = (x(u), y(u)) \ ,$$

Where u is an arbitrary parameter. The formula for computing the curvature function can be expressed as:

$$\kappa(u) = \frac{\dot{x}(u)\ddot{y}(u) - \ddot{x}(u)\dot{y}(u)}{(\dot{x}^2(u) + \dot{y}^2(u))^{3/2}} \qquad (1)$$

Convolution each component of $\boldsymbol{\Gamma}$ with $g(u, \sigma)$, a1D Gaussian kernel of width σ, gives

$$g(u,\sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-u^2}{2\sigma^2}} \qquad X(u,\sigma) = x(u) \star g(u,\sigma)$$

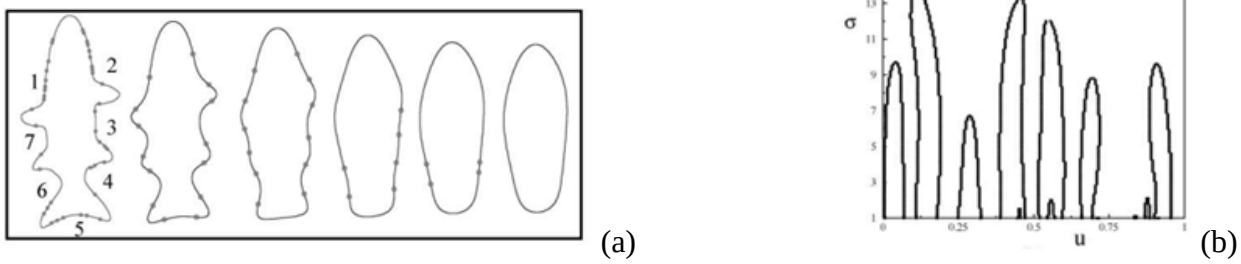
$$X_u(u,\sigma) = x(u) \star g_u(u,\sigma) \ ,$$

$$X_{uu}(u,\sigma) = x(u) \star g_{uu}(u,\sigma) \ ,$$

Similar formulas are used to compute $Y(u, \sigma), Y_u(u,\sigma)$ and $Y_{uu}(u, \sigma)$. Thus the curvature of the smoothed curve can be computed easily:

$$\kappa(u,\sigma) = \frac{X_u(u,\sigma)Y_{uu}(u,\sigma) - X_{uu}(u,\sigma)Y_u(u,\sigma)}{(X_u(u,\sigma)^2 + Y_u(u,\sigma)^2)^{3/2}} \quad (2)$$

Every object is represented by the x and y coordinates of its boundary points. Due to the various numbers of these points for images, the curve is first resampled and then smoothed by Gaussian function. The smoothed curve is called $\boldsymbol{\Gamma}_\sigma$, where σ denotes the width of the Gaussian kernel, $g(u, \sigma)$. Then Eq.2 is used to find the locations of curvature zero crossings on the smoothed curve. Curvature-zero crossings of a curve are points where the sign of curvature changes. The process starts with σ =1, and at each level, σ is increased by $\Delta\sigma$. As σ increases, $\boldsymbol{\Gamma}_\sigma$ shrinks and becomes smoother, and the number of curvature zero crossing points on it decreases. Finally, when σ is sufficiently high, $\boldsymbol{\Gamma}_\sigma$ will be a convex curve with no curvature zero crossings (Fig.14.a). The location of curvature zero crossing of every $\boldsymbol{\Gamma}_\sigma$ are displayed by some points in (u, σ) plane, where u is an approximation of the normalized arc length and σ is the width of the Gaussian kernel. The result can be represented as a binary image called the CSS image of the curve (Fig. 14.b). The intersection of every horizontal line with the contours in this image indicates the locations of curvature zero crossings on the corresponding evolved curve $\boldsymbol{\Gamma}_\sigma$ [8].



(a) (b)

Fig. 14. (a) Shrinkage and smoothing of the curve and decreasing of the number of curvature zero crossings during the evolution, from left: σ=1, 4, 7, 10, 12, 14. (b) The CSS image of the shape.

4.2 Extracting maxima of CSS contours

Every image in the database is represented with the locations of its major CSS contour maxima. For example, in Fig.14.b there are seven major maxima, and therefore the image will be represented by seven pairs of integer numbers. The CSS contours are usually connected everywhere except in a neighborhood of their maxima. We find the peaks of both branches of a contour in the CSS image and consider the midpoint of the line segment joining the pair as a maximum of the CSS image. In order to avoid complicated and inefficient matching, small maxima are not included in the representation. In our system, if a maximum is less than 1/6 of the largest maximum of the same CSS image, it is considered as noise [8]. As a result, only major concavities and convexities of a shape will contribute to the representation.

The benefits of this representation are that it is invariant under rotation, uniform scaling, noise and translation of the curve. It is unique (or there is a 1 to 1 correspondence between a curve and CSS image) and it is stable (slight changes in the curve does not significantly affect the CSS image and vice versa).

4.3 CSS matching

In this section, I explain the basic concepts of the matching algorithm, which compares two sets of maxima and assigns a matching value to them. A complete description of the CSS matching algorithm can be found in [15]. The matching value represents the similarity measure between the actual boundaries of objects. I call the input image as image and the images in the database as models.

Let $M_{image} = \{(t_1, \gamma_1), (t_2, \gamma_2), ..., (t_L, \gamma_L)\}$ be the maxima of the image, parameterized by arc length t and arranged in descending order of the scale γ. Let $M_{model} = \{(s_1, \sigma_1), (s_2, \sigma_2), ..., (s_N, \sigma_N)\}$ be the maxima of the model, parameterized by arc length s and arranged in descending order of scale σ . The matching algorithm is as follows [15]:

1. Compute the CSS shift parameter using the largest scale maximum in the image and the model. $\alpha = s_1 - t_1$

   Calculate the cost for the match as: $C = |\sigma_1 - \gamma_1|$

Create two lists. One contains maximum pair from image and the other with the maximum pair from the model.

2. Apply the CSS shift parameter α for the second largest maximum in the image and then find the closest maximum in the model to this shifted image maximum.

$$(s_i, \sigma_i) = arg\ min\ ||(s_i, \sigma_i) - (t_2+\alpha, \gamma_2)||, (s_i, \sigma_i)$$

Add the two new maxima to their respective list.

3. The cost of the match is updated as follows:

$$C = \begin{cases} C + \big||(si, \sigma i) - (t2 + \alpha, \gamma 2)|\big| & (1) \\ C + \big||\gamma 2|\big| & (2) \\ C + \big||\gamma 2|\big| & (3) \end{cases}$$

If the two maxima are in a reasonable horizontal distance (0.2 of the maximum possible distance), then follow (1) otherwise (2) and if there are no more image curve CSS maxima left, follow (3).

4. Repeat 2 for all the elements in the image.

5. Calculate the CSS shift parameter for maxima in the model which have a scale close (within 80%) to the highest maximum of the image. Also do the same using the second highest scale maximum (t_2, γ_2) of the image and the respective maximum in the model (s_2, σ_2). Repeat 1-3 to compute the cost of match.

6. Reverse the place of the image and the model and repeat steps 1 to 4.

7. Consider the lowest cost from all these matches as the best matching cost between the image and the model.

4.4 Experiment Result

The overall procedure of earprint recognition based on the curvature scale space (CSS) representation is shown in Fig. 15.

First, input the original earpint image. Second, the image of earprint is segmented into the binary contour images via some image processing techniques. Third, compute the CSS representation of the contours of earprint images. Then, extract the CSS features. Finally, perform features matching and recognition between the input feature vectors and the stored feature vectors for reference database.

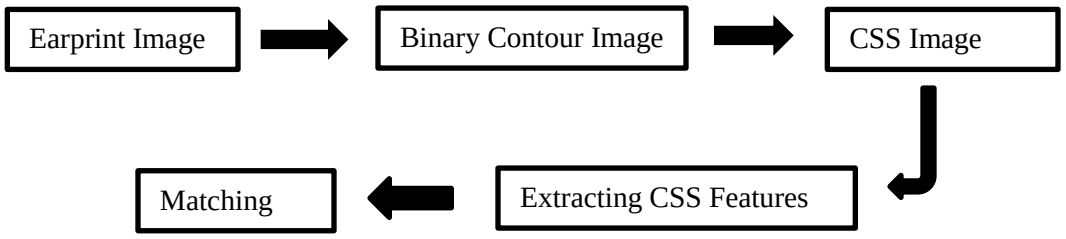
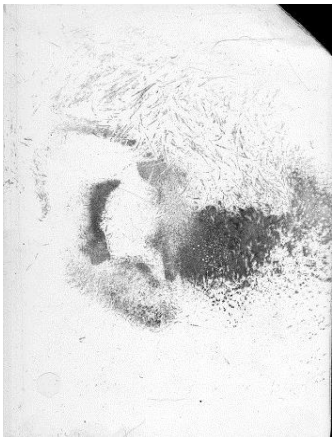


Fig. 15. Procedure of earprint recognition using CSS.

The first four steps are shown in Fig. 16(a) -16(e). Fig. 16(a) shows an input earprint and Fig. 16(b) is the contour of the earprint. Fig. 16(c) shows the resulting contours performing a convolution with the (1:0.1:20) kernel. The red points in the figure mean the locations of zero crossing points in the CSS image corresponding to the locations of the smoothed contours. Fig. 16(d) shows the generated CSS image. Fig. 16(e) represents the extracted maxima of the CSS image as the CSS features.

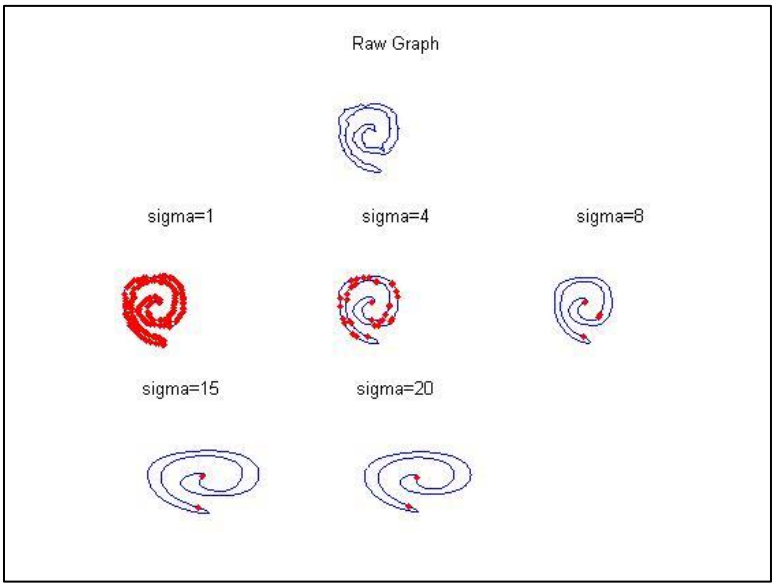
As the CSS image shows the two points on the smoothed contour do not join each other even in the high sigma, so we do not consider these two points in our calculations.



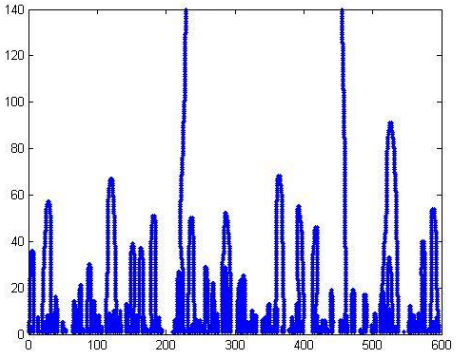
(a)

Extract the contour

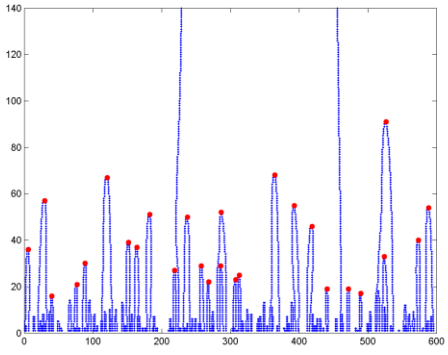
(b)



(c)



(d)



(e)

Fig. 16. Procedure of curvature scale space.

4.4.1 Pre-processing of images

As explained in the previous section, the antihelix contour should be extracted from the original images by an image editing software. The antihelix area was chosen because of its prevalence on prints found on crime scenes. Our dataset for CSS descriptor is the same as that I explained in section 3.6.1 just here we have contour images with dimension 350*500. The CSS image should be computed for all contour images in the dataset (Fig. 16(d)). One of the CSS images is taken as a test image. The CSS test image is then compared to a database containing other CSS images. Therefore considering the number of CSS images in the database, 81 independent runs were carried out for each side of ear.

4.4.2 Algorithm

Given an earprint image, the antihelix area as the region of interest (ROI) is specified in an image editing software. This part of the work has been done by the user. All operation performed after this step is fully automatic and do not require any user input. The boundary of the antihelix is traced and then gradually smoothed using Gaussian function. The process starts with sigma=1, the width of the Gaussian kernel. With increasing the value of sigma, chosen as 0.1 in our experiments, the resulting contour gets smoother and the number of zero crossings of the curvature along it decreases, until finally the contour is convex. The approaching zero crossing points create a contour in the CSS image, the joint points represent the maximum of the relevant contour.

The two sets of extracted maxima are compared and a matching value is assigned to them according to the matching algorithm in section 4.3.

4.4.3 Matching

The CSS features or sets of maxima are extracted from all the CSS images in the database (Fig.16 (e)). Then, given a CSS test image or CSS query, the extracted maxima from that is compared against the sets of maxima as a CSS features from each CSS image in the database.

The cost of the match is defined as the summation of the straight-line distances between the matched pairs plus the vertical coordinates of the unmatched maxima. The matching cost

represents the similarity measure between the actual boundaries of earprints. The CSS image in the database with the lowest cost is considered as the nearest CSS image.

4.5 Evaluation of performance

4.5.1 EER

We compared all query prints to all reference prints (per ear side).The results are illustrated by ROC curve. The value of equal error rate on the ROC curve indicates the performance. The smaller this rate, the more efficient the system.

Fig. 17 shows the ROC curve for the cost fucntion. The curve for left and right ear comparison are presented in green and blue color respectively. Results in terms of equal error rates (EER), for all comparisons are given in Table 5.

Table 5. EER performance of the system.

Results obtained with CSS using our dataset	
Sample	Cost[%]
Right	29.63
Left	14.81
Total	**22.22**

Note that the EER are significantly larger in right ears compare to the left ears. Overall EER is also high (22.22%). One reason for that would be the extraction of low quality contours.

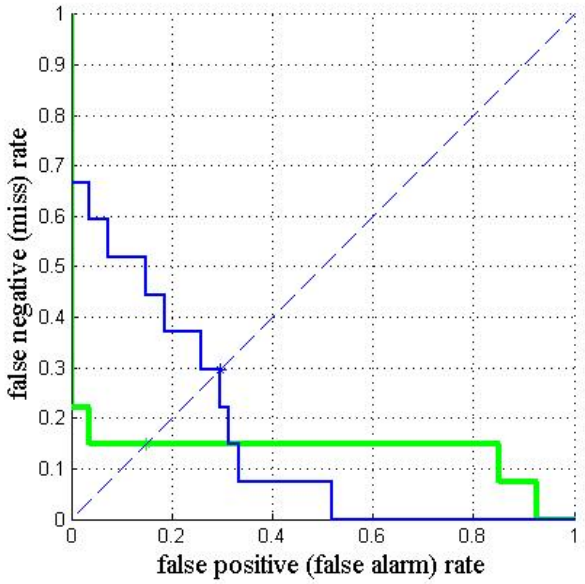


Fig. 17. ROC curve for CSS cost function.

4.5.2 Hitlists

The cost of the matches found for each CSS image of left earprints comparison is provided in a results matrix (Fig. 18). In Fig. 18 the three prints from each donor are grouped together and the two comparisons that give the lowest cost are highlighted. These two are the best and the second best matches between the others.

If a hitlist is made for each query by ranking the prints according to the cost of the matches, a print from the correct donor is ranked at the top of the hitlist in 8 out of the 9 prints. Furthermore, the two prints from the correct donor are returned as the first two prints in the hitlist for 6 out of the 9 prints; the two prints in which they are not come from the same donor belong to donor 2.

Hitlist results splitted out in left and right for all comparisons are given in Table 6. To achieve these results, the position of the true donor in the hitlist was noted. These results show that for only 66.6% of cases the best resulting hit will be in the top 0.1% of the hitlist. 72.2% of hits are in position three which means they are ranked from incorrect donors.

		Donor 1			Donor 2			Donor 3		
		P1	P2	P3	P1	P2	P3	P1	P2	P3
Donor 1	P1	0	147	135	349	202	191	192	183	214
	P2	147	0	179	360	188	185	228	216	237
	P3	135	179	0	309	193	201	187	234	236
Donor 2	P1	349	360	309	0	313	272	221	282	250
	P2	202	188	193	313	0	165	210	174	240
	P3	191	185	201	272	165	0	250	191	234
Donor 3	P1	192	228	187	221	210	250	0	154	144
	P2	183	216	234	282	174	191	154	0	171
	P3	214	237	236	250	240	234	144	171	0

Fig. 18. Results matrix for 9 left CSS images showing the cost found for each CSS image comparison.

Table 6. Hiltlist behavior for CSS.

Searches			
Position	Left	Right	Overall
1	8 out of 9 (88.8%)	4 out of 9 (44.4%)	66.6%
2	6 out of 9 (66.6%)	5 out of 9 (55.5%)	61.05%
3	4 out of 9 (44.4%)	9 out of 9 (100%)	72.2%

Chapter 5

Image Registration

It is a process of aligning two images into a common coordinate system in order to monitor subtle changes between the two.

Junod et al. [16] proposed an automatic system for comparison of earprints and earmarks, based on the superimposition of images by a registration algorithm. In their method, region of interest (ROI) is indicated by a user in form of polygon and then denoising filter will be applied. Then processed images are aligned using an iterative multi-resolution algorithm, based on the image intensity. Closeness between aligned images is measured by 2D Pearson–Bravais correlation coefficient normalized by a Z-score.

The following section introduces the mathematical formulation of the registration process and gives an overview of the components of which a general registration method consists.

5.1 Images

Image registration is all about images. Two images are involved in the registration process. One image, the moving image $I_M(x)$, is deformed to fit the other image, the fixed image $I_F(x)$. Registration is the problem of finding a transformation $T(x) = x + u(x)$ that makes $I_M(T(x))$ spatially aligned to $I_F(x)$. The transformation is defined as a mapping from the fixed image to the moving image.

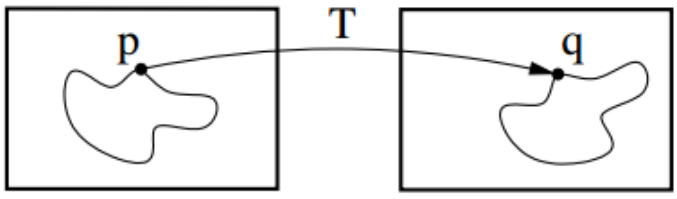


Fig. 19. Image registration is the task of finding a spatial transformation mapping one image to another. Left is the fixed image and right the moving image.

Commonly, the registration problem is formulated as an optimization problem in which the cost function C is minimized w.r.t. $\boldsymbol{T}$:

$$\boldsymbol{T} = arg\ min\ C(\boldsymbol{T}; I_F, I_M)$$

$\boldsymbol{T}$ is the number of possible transformation

The components involved in image registration are shown in Fig. 20.

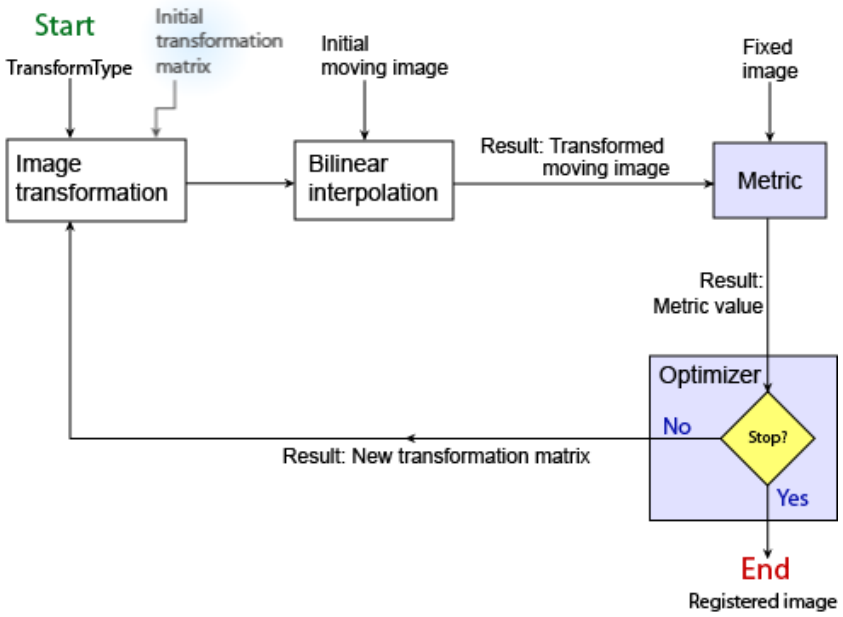


Fig. 20. Basic registration components.

5.2 Metrics

The metric defines the image similarity metric for evaluating the accuracy of the registration. This image similarity metric takes two images and returns a scalar value that describes how similar the images are. Several choices for the similarity metric can be found in the literature. Some common choices are described below.

Sum of Squared Differences (SSD): The SSD is defined as:

$$\mathrm{SSD}(\boldsymbol{\mu}; I_F, I_M) = \frac{1}{|\Omega_F|} \sum_{\boldsymbol{x}_i \in \Omega_F} \left(I_F(\boldsymbol{x}_i) - I_M(\boldsymbol{T}_{\boldsymbol{\mu}}(\boldsymbol{x}_i))\right)^2,$$

With $|\Omega_F|$ the number of pixels of the fixed image. The vector μ contains the values of the transformation parameters. For example, when the transformation is modelled as a 2D rigid transformation, the parameter vector μ contains one rotation angle and the translations in x and y direction.

Given a transformation T, this measure can easily be implemented by looping over the pixels in the fixed image, taking $I_F(x_i)$,calculating $I_M\ (T_\mu(x_i))$ by interpolation, and adding the squared difference to the sum.

Normalized Correlation Coefficient (NCC): The NCC is defined as:

$$\mathrm{NCC}(\boldsymbol{\mu}; I_F, I_M) = \frac{\sum_{\boldsymbol{x}_i \in \Omega_F} \left(I_F(\boldsymbol{x}_i) - \overline{I_F}\right)\left(I_M(\boldsymbol{T}_{\boldsymbol{\mu}}(\boldsymbol{x}_i)) - \overline{I_M}\right)}{\sqrt{\sum_{\boldsymbol{x}_i \in \Omega_F} \left(I_F(\boldsymbol{x}_i) - \overline{I_F}\right)^2 \sum_{\boldsymbol{x}_i \in \Omega_F} \left(I_M(\boldsymbol{T}_{\boldsymbol{\mu}}(\boldsymbol{x}_i)) - \overline{I_M}\right)^2}},$$

with the average grey-values $\overline{I_F} = \frac{1}{|\Omega_F|} \sum_{\boldsymbol{x}_i \in \Omega_F} I_F(\boldsymbol{x}_i)$ and $\overline{I_M} = \frac{1}{|\Omega_F|} \sum_{\boldsymbol{x}_i \in \Omega_F} I_M(\boldsymbol{T}_{\boldsymbol{\mu}}(\boldsymbol{x}_i))$.

Mutual Information (MI): For MI

$$\mathrm{MI}(\boldsymbol{\mu}; I_F, I_M) = \sum_{m \in L_M} \sum_{f \in L_F} p(f, m; \boldsymbol{\mu}) \log_2 \left(\frac{p(f, m; \boldsymbol{\mu})}{p_F(f) p_M(m; \boldsymbol{\mu})} \right),$$

p is the discrete joint probability, and p_F and p_M are the marginal discrete probabilities of the fixed and moving image, obtained by summing p over m and f , respectively. The joint probabilities are estimated using B-spline Parzen windows [17]:

$$p(f, m; \boldsymbol{\mu}) = \frac{1}{|\Omega_F|} \sum_{\boldsymbol{x}_i \in \Omega_F} w_F(f/\sigma_F - I_F(\boldsymbol{x}_i)/\sigma_F) \times w_M(m/\sigma_M - I_M(\boldsymbol{T}_{\boldsymbol{\mu}}(\boldsymbol{x}_i))/\sigma_M),$$

Mutual Information is the most popular image similarity measures for registration of multimodality images. Normalized Correlation Coefficient and Sum of Squared Differences are commonly used for registration of images in the same modality.

5.3 Interpolators

During the optimization the value $I_M\ (T_\mu(x))$ is evaluated at non-pixel positions, for which intensity interpolation is needed. Several methods for interpolation exist, varying in quality and speed.

***N-th* order B-spline**: The higher the order, the better the quality, but also requiring more computation time. Zero-order B-spline interpolation is called nearest neighbor ($N=0$) and first-order one is linear interpolation ($N=1$) [18]. Nearest neighbor is the simplest technique, low in quality, requiring little resources. The intensity of the pixel nearest in distance is returned. In linear, the returned value is a weighted average of the surrounding pixels, with the distance to each pixel taken as weight. Bilinear interpolation is an extension of linear interpolation in two directions. During registration linear interpolation, often gives satisfactory results but to generate the more accurate result, a higher order interpolation e.g. $N=3$ is usually required.

5.4 Transforms

The transformation type defines the type of 2-D transformation that brings the misaligned image (called the moving image) into alignment with the reference image (called the fixed image).

Translation: The translation is defined as: $T_\mu(x) = x + t$, with t the translation vector. The parameter vector is simply defined by $\mu = t$.

Rigid: A rigid transformation is defined as: $T_\mu(x) = R(x - c) + t + c$, with the matrix R a rotation matrix, c the center of rotation, and t translation again, which means that the image can translate and rotate but cannot be scaled or stretched. The parameter vector μ consists of the Euler angles (in rad) and the translation vector. In 2D, this gives a vector of length 3: $\mu = (\theta_z, t_x, t_y)^T$, where θ_z denotes the rotation around the axis normal to the image.

Similarity: A similarity transformation is defined as $T_\mu(x) = sR(x - c) + t + c$, with s a scalar and R a rotation matrix. This means that the image is treated as an object, which can translate, rotate, and scale isotropically. The rotation matrix is parameterized by an angle in 2D. The parameter vector μ consists of the angle, the translation vector, and the isotropic scaling factor. In 2D, this gives a vector of length 4: $\mu = (s, \theta_z, t_x, t_y)^T$

Affine: An affine transformation is defined as: $T_\mu(x) = A(x - c) + t + c$, where the matrix A has no restrictions. This means that the image can be translated, rotated, scaled, and sheared. The parameter vector μ is formed by the matrix elements a_{ij} and the translation vector. In 2D, this gives a vector of length 6: $\mu = (a_{11}, a_{12}, a_{21}, a_{22}, t_x, t_y)^T$.

B-splines: For the category of non-rigid transformations, B-splines are often used as a parameterization [19]:

$$T_\mu(x) = x + \sum_{x_k \in N_x} p_k \beta^3 \left(\frac{x - x_k}{\sigma} \right),$$

with x_k the control points, $\beta^3(x)$ the cubic multidimensional B-spline polynomial, p_k the B-spline coefficient vectors , σ the B-spline control point spacing, and N_x the set of all control points within the compact support of the B-spline at x.

See Fig. 21 for an illustration of different transforms.

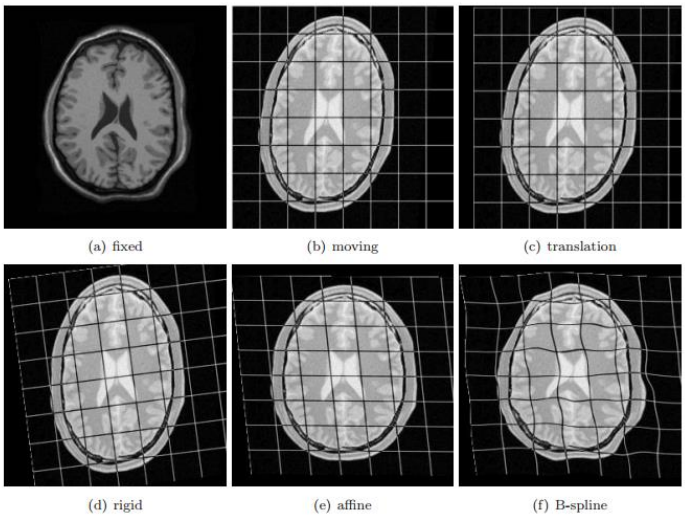


Fig. 21. Different transformations [20].

5.5 Optimizers

To obtain the optimal transformation parameter vector μ, commonly an iterative optimization strategy is employed. The optimizer defines the methodology for minimizing or maximizing the similarity metric. $\boldsymbol{\mu_{k+1}} = \boldsymbol{\mu_k} + a_k \boldsymbol{d_k}, \qquad k = 0, 1, 2, \cdots$

with $\boldsymbol{d_k}$ the 'search direction' at iteration k, a_k a scalar gain factor controlling the step size along the search direction. The optimization process is illustrated in Fig. 22. Examples are quasi-Newton (QN), nonlinear conjugate gradient (NCG), gradient descent (GD), and Robbins-Monro (RM) [21]. Bellow I explain gradient descent used in our experiment.

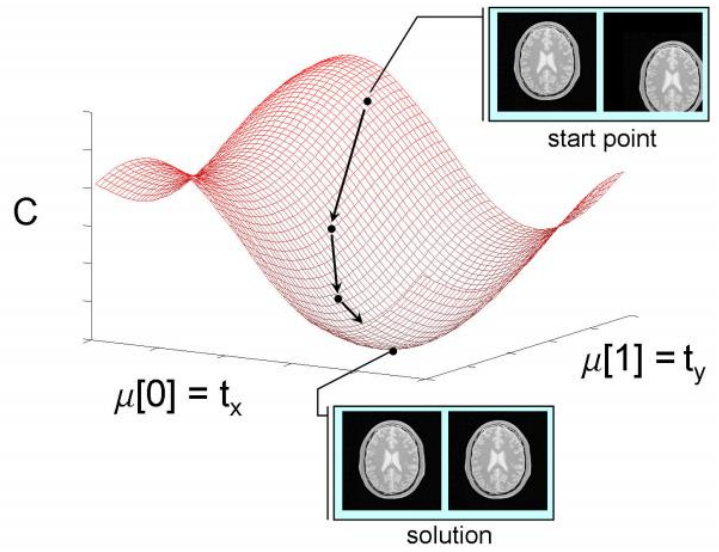


Fig. 22. Iterative optimisation. Example for registration with a translation transformation model. The arrows indicate the steps $a_k \boldsymbol{d}_k$ taken in the direction of the optimum, which is the minimum of the cost function [20].

Gradient descent (GD): Gradient descent optimization methods take the search direction as the negative gradient of the cost function:

$$\boldsymbol{\mu_{k+1}} = \boldsymbol{\mu_k} - a_k \boldsymbol{g(\mu_k)},$$

with $\boldsymbol{g(\mu_k)} = \boldsymbol{\partial C / \partial \mu}$ evaluated at the current position $\boldsymbol{\mu_k}$. Several choices exist for the gain factor a_k. It can for example be determined by a line search or by using a predefined function of k.

5.6 Multi-Resolution

Registration is a multi- resolution algorithm. For multi-resolution strategy, two hierarchical methods are distinguished: reduction of data complexity, and reduction of transformation complexity.

- Data complexity

It is common to start the registration process using images that have lower complexity, e.g., images that are smoothed and possibly downsampled. This increases the chance of successful registration. A series of images with increasing amount of smoothing is called a scale space. If the images are not only smoothed, but also downsampled, the data is not only less complex, but the amount of data is actually reduced that is called pyramid. The Gaussian pyramid is the most common one that applies smoothing and down-sampling. Fig. 23 shows the Gaussian pyramid with and without downsampling.

- Transformation complexity

The second multiresolution strategy is to start the registration with fewer degrees of freedom for the transformation model. The degree of freedom of the transformation equals the length (number of elements) of the parameter vector μ. An example of this is the use of a rigid transformation prior to nonrigid (B-spline) registration.

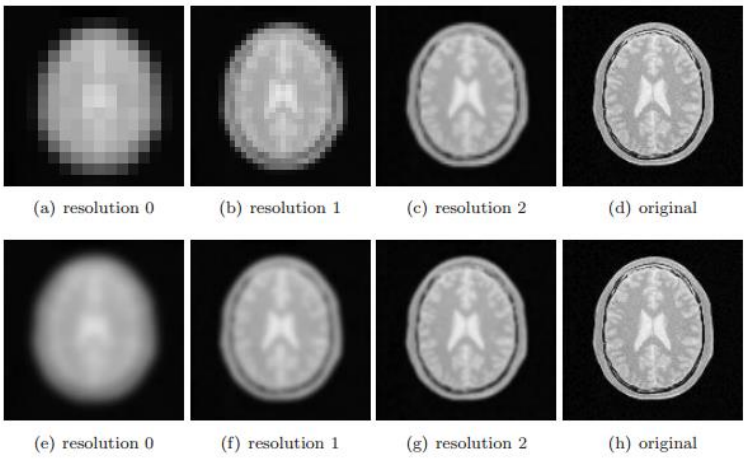


Fig. 23. Two multi-resolution strategies using a Gaussian pyramid (σ = 8.0, 4.0, 2.0 pixels). The first row shows multi-resolution with down-sampling, the second row without [20].

5.7 Experiment Result

Image registration is an automatic way to compare earprint images based on superimposition of images, without any manual extraction of key features. Image registration is performed using an iterative multi-resolution algorithm, based on the image intensity as described in the previous section.

5.7.1 Pre-processing of images

All pre-processing steps for image registration experiment are the same as section 3.6.1. So I do not repeat them.

5.7.2 Algorithm

As described before intensity-based image registration is an iterative multi-resolution process. It requires specifying a pair of images, a metric, an optimizer, and a transformation type.

The process begins in each level of pyramid with the transform type you specify and an internally determined transformation matrix. Together, they determine the specific image transformation that is applied to the moving image with bilinear interpolation. Next, the metric compares the transformed moving image to the fixed image and a metric value is computed.

Finally, the optimizer checks for a stop condition. A stop condition is anything that warrants the termination of the process. In most cases, the process has reached a point of diminishing returns or it has reached the specified maximum number of iterations. If there is no stop condition, the optimizer adjusts the transformation matrix to begin the next iteration.

Implementation of image registration was done in Matlab using Image Processing Toolbox by the function *imregister* [14]. We used mean squared difference as the similarity metric, affine transformation as the transform type, number of pyramid levels = 3 and gradient descent as the optimizer with maximum iteration equals to 100.

5.7.3 Matching

Matching process is fully automatic. The registration algorithm previously explained is used as a mechanism to find the closest alignment between a test image and prints contained in

the database. The alignment is performed between greyscale images. The result of a comparison between a test image and a print can be represented for illustration purposes as a composite RGB image including the test image and the print image. The overlapping parts between the test and the print are shown in black and non-overlapping parts of the test and the print appear as magenta and green (Fig. 24).

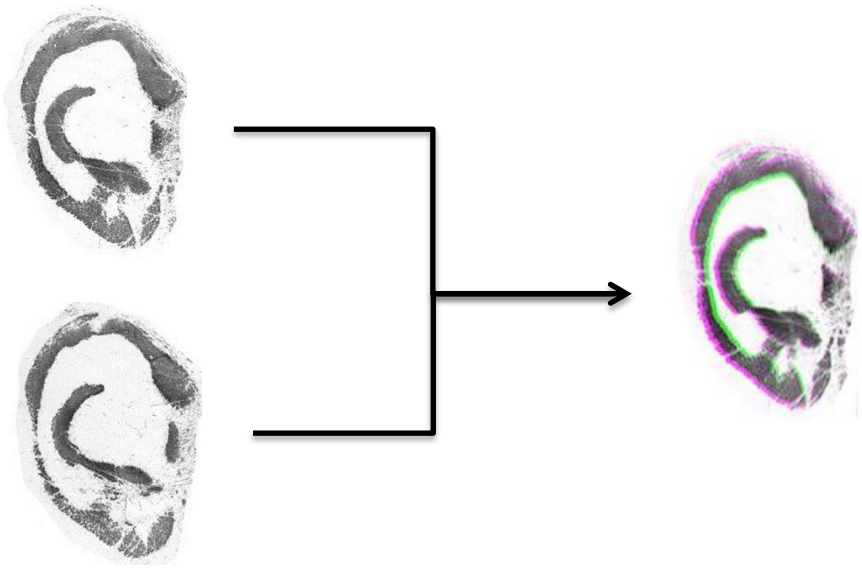


Fig. 24. Illustration of an automatic comparison between a test image and print (left) leading to the optimal alignment illustrated by the RGB image (right).

For each comparison, a score is calculated to reflect the level of closeness obtained following the alignment between a test image and print image. To find the most discriminating score, two metrics were used : Dice Similarity Coefficient (DSC) and Correlation Coefficient (CC).

Correlation Coefficient (CC)

A measure of the strength and direction of the linear relationship (dependence) between two variables X and Y that is defined as the (sample) covariance of the variables divided by the product of their (sample) standard deviations.

$$r = \frac{\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^{n}(X_i - \bar{X})^2}\sqrt{\sum_{i=1}^{n}(Y_i - \bar{Y})^2}}$$

Giving a value between +1 and −1 inclusive, where 1 is total positive correlation, 0 is no correlation, and −1 is total negative correlation.

Dice Similarity Coefficient

Compute the overlap of segmented anatomical structures after registration. The better the overlap, the better the registration.

$$\mathrm{DSC}(X, Y) = \frac{2|X \cap Y|}{|X| + |Y|},$$

Where *X* and *Y* represent the binary label images, and | · | denotes the number of pixels that equal 1. A higher DSC indicates a better correspondence. A value of 1 indicates perfect overlap; a value of 0 means no overlap at all.

The metrics are calculated on the resulting images in binary form obtained from alignment and test image in each run.

5.8 Evaluation of Performance

5.8.1 EER

As previously indicated, EER is a performance metric for the system.

In Fig. 25 and Fig. 26, ROC curve for the two metrics calculated with combination of two graphs, one for left ear in green color and the other for right ear in blue color. Results in terms of equal error rates (EER), for all comparisons are given in Table 7.

Table 7. EER performance of the system.

Results obtained with registration using our dataset		
Sample	DSC[%]	CC[%]
Right	7.41	1.85
Left	7.41	7.41
Total	**7.41**	**4.63**

The system shows better performance for comparisons using correlation coefficient, with equal error rate of 4.63% whereas the equal error rate for dice similarity coefficient is 7.41%.

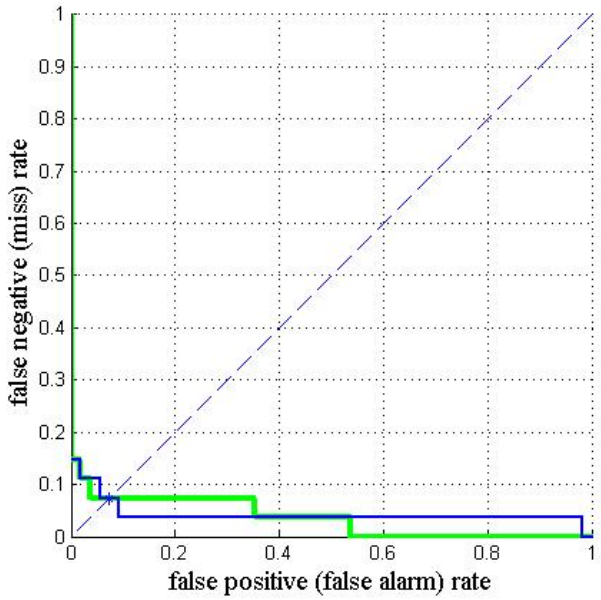


Fig. 25. ROC curve for DSC metric.

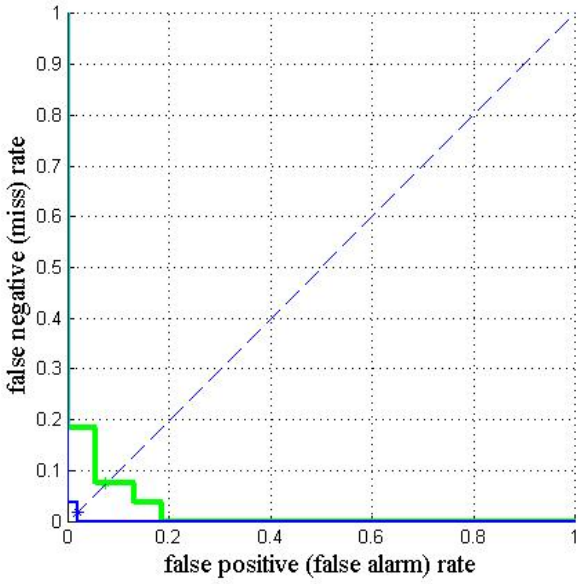


Fig. 26. ROC curve for CC metric.

Note that the EER are significantly larger in left ears using correlation coefficient comparison than for right ears.

5.8.2 Hitlists

The second way in which the performance of the system is tested is, as explained in Section 2.2, by constructing hitlists for every query print compared to its reference database. Between two similarity metrics, the better results was obtained with the correlation coefficent so we construct matrix and then hitlist on the results of CC. For this purpose the value of correlation coefficient of the matches found for each right ear image comparison is provided in a results matrix (Fig. 27). In Fig. 27 the two comparisons that give the highest value are highlighted. These two are the best and the second best matches between the others.

		Donor 1			Donor 2			Donor 3		
		P1	P2	P3	P1	P2	P3	P1	P2	P3
Donor 1	P1	1	0.238	0.157	-0.294	-0.284	-0.296	-0.162	-0.014	-0.107
	P2	0.350	1	0.133	-0.328	-0.297	-0.282	-0.124	-0.030	-0.138
	P3	0.080	0.343	1	-0.127	-0.324	0.055	-0.222	0.019	-0.010
Donor 2	P1	-0.148	-0.150	-0.143	1	0.064	0.101	-0.117	-0.044	-0.285
	P2	-0.182	-0.098	-0.168	0.175	1	0.019	-0.386	-0.321	-0.353
	P3	-0.096	-0.086	-0.068	0.645	0.155	1	-0.209	-0.255	-0.248
Donor 3	P1	-0.060	-0.002	-0.057	-0.104	-0.092	-0.061	1	0.261	0.193
	P2	-0.108	-0.117	-0.101	-0.077	-0.079	-0.108	0.673	1	0.151
	P3	-0.068	-0.089	-0.121	-0.081	-0.039	-0.006	0.430	0.194	1

Fig. 27. Results matrix for 9 right ear images showing the correlation coefficient found for each image comparison.

The results matrix on the correlation coefficient value can be sorted in descending order of scores to provide a hitlist. For any query print, we look for the relative position of the corresponding matching print that comes up highest in the hitlist. For a right query print this means it is compared to all 9 prints in the right earprint database, and the best position of the matching prints in the resulting hitlist is taken. The results are presented in Table 8. The two right prints from the correct donor are returned as the first two prints in the hitlist for 9 out of the 9 prints.

Comparison shows that on average, 94.4% of cases gained the best match in the first position in which for right ear all first and second matches (100%) were ranked correctly.

Table 8. Hiltlist behavior for CC.

Searches			
Position	Left	Right	Overall
1	8 out of 9 (88.8%)	9 out of 9 (100%)	94.4%
2	7 out of 9 (77.7%)	9 out of 9 (100%)	88.85%
3	3 out of 9 (33.3%)	0 out of 9 (0%)	16.65%

Chapter 6

Conclusion

Main aim of this project is comparing the performance of the existing methods for automatic recognition of earprint images. These methods are based on the scale invariant feature transform (SIFT), curvature scale space (CSS) features and image registration techniques. Among these methods, SIFT features and registration based on the image intensity have been used for earprint recognition. But this is the first time that CSS is used as a feature extraction approach for earprint recognition.

The recognition rates shown in Table 9 and the EER presented in Table 10 are the average results of both sides of the ears.

According to the Table 9, using SIFT features, we got 100% recognition rate. This result is in agreement with those obtained by Meijerman et al. [11] for matching right earprints of twins. We both used the SIFT algorithm implemented using the executable code provided by Lowe [12] and also both defined the number of accepted keypoint matches as a similarity metric. However Meijerman et al. applied the SIFT algorithm on the whole image of the earprint but we applied the algorithm on the specified region of interest (ROI) showing the antihelix area.

In case of image registration, the best performance obtained using correlation coefficient similarity metric in which in 94.4% cases the first matches was recognized correctly. This result can be compared with Junod et al. [16] work. With these differences that they applied Intensity based image registration algorithm on the ROI two times. One time to construct a model from all earprints coming from the same donor and the other to do matching between a query mark and a model. Whereas we applied image registration algorithm once for comparison of query print with other prints. In Junod et al. work, 2D Pearson–Bravais correlation coefficient normalized with Z-score was used to reflect the level of closeness between an aligned print and another print or model. As a result, 97.6% of cases have the correct match in the first position of the hitlist. To evaluate the registration we used Correlation Coefficient and Dice Similarity Coefficient which we got the better result using Correlation Coefficient (EER = 4.63 compared to EER = 7.41) that this equal error rate of 4.63% is rather high compared to 0.51% for Junod et al. work. The

reasons for our high rate is probably the limited number of prints in our dataset and also we did not make any model of the prints in order to compare with other models or prints.

CSS has too low recognition rate on the first (66.6%) and second best matches (61.05%) and also it has too high EER of 22.2%. These results show CSS is not useful in earprint recognition. This is probably due to the shape of the contour of earprint and also having two inflection points in the CSS image which do not join.

Overall, SIFT algorithm applied on our small dataset achieved high recognition rate. However, the 128-dimensional SIFT feature descriptor contains redundant information and is slow to compute. Thus, our results in consistent with those results reported by Meijerman et al. for 6 pair monozygotic suggest that SIFT is the most promising method for a small dataset. Although for a large-scale dataset image registration would be more potential approach [16].

Table 9. Recognition rate on the methods.

Method	First matches	Second matches
SIFT	**100%**	**100%**
CSS	**66.6%**	**61.05%**
Image registration using CC	**94.4%**	**88.85%**

Table 10. EER on the methods.

Method	EER(%)
SIFT	**0.0**
CSS	**22.2**
Image registration using CC	**4.63**

References

[1] Champod, C., Evett, I.W., Kuchler, B. Earmarks as evidence: a critical review, J. For. Sci. 46, 1275–1284 (2001).

[2] State, V., Kunze, D.W. Court of Appeals of Washington, Division 2, 97 Wash. App. 832, 988 P.2d 977 (1999).

[3] Dallagher, R.V.M.A., In the Supreme Court of Judicature – Court of Appeal (Criminal Division) – No. (2002) EWCA Crim 1903, Case No. 2000/5024/Z2, London (2002).

[4] Mark, R.V., Kempster, J. UK Court of Appeal, [2003] EWCA Crim 3555 (2003).

[5] Johnson, Z. Standard Operating Procedure for the Taking of Earprints. *Internal FearID Report*, (2003), http://forensic.to/fearid/Procedure.doc.

[6] I. Alberink, A. Ruifrok , Performance of the FearID earprint identication system, Netherlands Forensic Institute, Internal publication.

[7] Lowe, D. G. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis*. **60**, 91-110 (2004).

[8] Abbasi, S., Mokhtarian, F., Kittler. J. Curvature scale space image in shape similarity retrieval. Multimedia Sys. **7**, 467–476 (1999).

[9] Image registration techniques using MATLAB.

[10] Vedaldi, A., Fulkerson, B. {VLFeat}: An open and portable library of computer vision algorithms (2008).

[11] Meijerman, L., Thean, A., van der Lugt, C., van Munster, R., van Antwerpen, G., Maat, G. Individualization of earprints. variation in earprints of monozygotic twins, *Forensic Sci. Med. Pathol.* **2**, 39–49 (2006).

[12] David Lowe's website http://www.cs.ubc.ca/~lowe/

[13] Fischler, M., Bolles, R. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography, *Comm. of the ACM*. **24**, 381–395, (1981).

[14] MATLAB and statistics toolbox release, The MathWorks, Inc., Natick, Massachusetts, United States (2012).

[15] Abbasi, S. Curvature. scale space in shape similarity retreival, Ph.D. thesis, Centre for Vision, Speech and Signal Processing, University of Surrey, Guildford, GU2 5XH, England (1998).

[16] Junod, S., Pasquier, J., Champod, C.The development of an automatic recognition system for earmark and earprint comparisons. *Forensic Sci. Int*. **222**, 170-178 (2012).

[17] Thevenaz, P., Unser., M. Optimization of mutual information for multiresolution image registration. *IEEE Trans. Image Process.* **9**, 2083–2099 (2000).

[18] Unser, M. Splines: A perfect fit for signal and image processing. *IEEE Signal Process. Mag.* **16**, 22 – 38 (1999).

[19] Rueckert, D., Sonoda, L.I., Hayes, C., Hill, D.L.G., Leach, M. O., Hawkes. D. J. Nonrigid registration using free-form deformations: Application to breast MR images. *IEEE Trans. Med. Imag*. **18**, 712 – 721, (1999).

[20] Klein, S., Staring, M., Murphy, K., Viergever, M.A., Pluim, J.P.W. "Elastix: a toolbox for intensity based medical image registration," *IEEE Transac. Med. Imag*. **29**, 196 - 205, (2010).

[21] Klein, S., Staring, M., Pluim, J.P.W. Evaluation of optimisation methods for nonrigid medical image registration using mutual information and B-splines. *IEEE Trans. Image Process*. **16**, 2879–2890 (2007).