# The University of Calgary
# Department of Electrical and Computer Engineering

## ENCM 509 - Fundamentals of Biometric Systems Design

## Laboratory Experiment #5

***Fingerprint Biometrics part II: Feature Extraction and Matching***

# 1 Introduction

In fingerprint verification/identification, most commonly only one feature is used, namely the minutiae. There are variable number of occurrences of minutiae in each fingerprint. Hence the feature vector contains a vector of minutiae, each represented as triplets $x, y, \theta$ ($x$-location, $y$-location and orientation $\theta$). A fingerprint matching algorithm is used to produce a similarity score between two pairs of fingerprint samples. Traditionally, the fingerprint matching algorithm computes a relative distance (in certain domain) between two samples of the fingerprint. To automatically make a decision of match (same finger) or non-match (different finger), a statistical model assumes the availability of a training set. For example, a data set may consist of 100 fingerprints, 8 samples for each. From these, some same and some different fingerprint pairs can be taken, and half of these numbers can be used for the purpose of learning, and remaining for evaluation.

The accuracy of the matching depend on the accuracy of the collected fingerprints, and the quality of the pre-processing algorithms.

The preprocessing involved:

- extraction of the fingerprint from the background,
- pre-classification (in some matchers) which is dividing images to the classes according to the presence of a whorl, arch, right loop or left loop (look at your fingers: you may have a ringlet or lock (whorl), unlocked loop (loop), or arch).

The processing and feature (minutia) extraction involves:

- Orientation estimation,
- Ridge orientation,
- Applying Gabor filters on the fingerprint image which have frequency-selective and orientation-selective properties. This enhances the ridges.

Finally, the fingerprint comparison or matching, includes:

- alignment,

- the comparison between two user-selected fingerprints.

In this lab, we will use the fingerprints acquired in Lab 4, and the same preprocessing and processing steps as in lab 4. These steps prepare data for fingerprint matching. Next, we will investigate the matching algorithm that uses a score to compare the images, represented by their feature vectors. The score is a numerical data expressing the similarity of two feature vectors. The idea of matching score is the basis for a simple identification system, performing one-against-many comparisons. Such a system is created using the following steps:

- We assume you have collected the database of 10–30 templates, each representing one fingerprint.
- Identification of a newly submitted, or probed, fingerprint. This is performed using so-called “matching”, or comparison of the feature vectors, - the database and the probed one (one to be identified). The score is calculated for each pair “database fingerprint – probed fingerprint”, so given a database of 10 fingerprints, we have to have 10 pairs, and therefore, 10 numerical scores.

  Again, it should be noted that is we have $N$ templates for each person in the database, we may have to calculate $N$ scores within this set of templates, and then choose the best score for this set.

The matching of fingerprints in this laboratory exercise include:

- alignment of the compared images, and
- comparison based on the relative distance, resulting in a numerical score.

The matching can be used in verification or identification. In both applications, you may need to know the threshold for your score value, needed for the appropriate conclusion about the matching. In the identification, you may need to detect the highest score, and then again compare it against some threshold.

In this laboratory, we investigate two matching algorithms: one based on the Minutiae matching with prior alignment, and another based on the Gabor filtered features. In both cases, matching is based on calculating the Euclidean distance between the two templates: the database one and the probed one.

The `.m`-files for fingerprint image processing and plotting can be found in the Windows directory on the N drive
*\ENCM\509\lab4-5*.

# 2 The laboratory procedure Part I (from Lab 4): processing and feature extraction

## 2.1 Fingerprint acquisition

To create a database for further matching, collect:

- 30 of your left (or right) thumb finger images,
- 30 of your other thumb or the other fingers.

## 2.2 Fingerprint image processing

The fingerprint processing was implemented in the demo `Lab4Fingerprint1.m` file, which calls other functions available in the Windows directory on N drive \*ENCM*\*509*\*lab4-5*.

In the demo (`file Lab4Fingerprint1.m`), the fingerprint image was segmented by isolating the foreground from the background. Then, the ridge orientation is calculated , and the singularity points (core and delta) are localized. After thinning and skeleton cleaning, the minutiae were found.

We also used the demo file Lab4Fingerprint2gab.m to perform Gabor filtering on your fingerprint image.

# 3 The laboratory procedure Part II: Fingerprint matching

The matching, or the scoring algorithm in this lab is based on two approaches:

- Features obtained based on the Gabor filtering, and
- Feature obtained from localization of the minutiae (two types only: ridge end, and bifurcation).

In the first case, script `MatchGaborFeat.m` is used. Matching score is calculated as the mean of the differences between two feature vector, while the features are calculated using the absolute deviation from the mean on the $16 \times 16$ blocks of the Gabor filtered images (there are eight filters, corresponding to eight different angles used to build the Gabor filter, and eight filtered images per one fingerprint).

In the second case, script `match.m` is used. The ridge skeleton, and the coordinates of the minutiae are considered for comparison. For this, image alignment is required (script `align2.m` is used for that). Fingerprint alignment is executed as follows: once a fingerprint is pre-processed, it is necessary to find its corresponding position in the database image in order to make a comparison. The solution consists in calculating the 2-dimensional correlation for every possible location of the fingerprint within the database image. The highest correlation factor determines the part of the reference image that matches the probed fingerprint the best.

Matching score determines the similarity of two fingerprint features Fp1 and Fp2. The minutiae features are compared as follows: a minutia $m_i$ in Fp1 and a minutia $m_j$ in Fp2 are matched if the Euclidian distance between them is smaller than some pre-determined threshold.

## 3.1 Matching Exercise 1

Run the demo file Lab5Fingerprint2.m to read, process and match as follows: first, try one against 10 different impressions of the same finger, then try one impression of your finger against 10 different finger(s). Note that calculations may take several minutes in Matlab.

For each comparison,

- Record the scores (use a table form) for the matching method based on Gabor filter (it works based on the minimum score); estimate the mean and deviation of the score within the class of your fingerprints. This can be used to predetermine the threshold in the identification using script `match.m` in Exercise 2.
- Record the scores for the minutiae based matching (it works based on the maximum score; note the difference compared to the Gabor score), estimate the mean and deviation.
- Calculate FRR using the mean and deviation of the authentic scores. Estimate FNMR as a ratio of the number of authentic score that lie outside of the critical interval and the number of scores (30). Assume 0.05 level of significance (i.e. 95% of confidence).
- calculate FAR using the mean and deviation of the non-authentic scores: FAR occurs when a data point from non-authentic score falls within the critical interval. Estimate FMR as the ratio of the number of data points from the non-authentic scores that lie within the critical interval and the total number of scores (30).

Comment on:

- How the choice of the threshold affects the FRR and FAR of the matching algorithms.

### 3.2 Matching Exercise 2

Develop the simple identification procedure. Choose one of your fingerprints (e.g. left thumb) to be used as the probed one. Create the database, using 1–2 different impressions of the same finger and 10 fingerprints of your other fingers.

A sample algorithm for identification (comparing 1 to many) can be as follows:

- Submit the probed fingerprint to the sensor 30 times
- Analise each probed fingerprint against all the fingerprints in your database (ideally, we should have them processed and stored as feature vectors already), and record the matching scores for the Gabor filter method and the Minutiae matching method. Note: choose the threshold vector for the Minutiae-Skeleton algorithm based on the value calculated in Exercise 1. No threshold is required for the Gabor based method. For each fingerprint comparison, store the Gabor filter scores and the Minutiae scores in individual arrays, with the index corresponding to the fingerprint number.
- Determine the minimum score for the Gabor filter, and the maximum score for the Minutia scores. Choose the best match.

Comment on:

- What can possibly be improved in the identification procedure?

## 4 Laboratory Report

In your report, include:

- General description of your lab exercise flow, illustrated with graphs and the maps of the minutiae point for selected images resulted from running the Demo (you can compare good and bad quality ones) (10%).
- Your calculations and responses to the questions in the Matching exercise 1 (30%)
- Your calculations and responses to the questions in the Matching exercise 2 (30%)
- Your comparative analysis of two matching algorithms: Minutiae based and Gabor filter based (10%).
- The fragments of your modified Matlab code (in the report text) or Matlab files (.m files) and illustration were appropriate (20%).

# 5 Acknowledgments

We wish to thank research students who contributed to the projects implemented in the Biometric Technologies Laboratory (BT Lab). The acquisition executable `digitalPersona.exe` was created in the BT Lab based on the Digital Persona SDK. The implementation of some fingerprint processing protocols according to NIST (National Institute of Standardization, USA) are credited to K. Kryszczak, EPFL. We also wish to thank the department IT staff for setting up the software in the UofC labs, and the TA S. Samoil for verifying the lab exercise and codes.

*Svetlana Yanushkevich*
October 2, 2015