UTP University of Science and Technology

**Faculty of Telecommunications, Computer Science and Electrical Engineering**

PhD Thesis abstract

Agata Giełczyk, MSc. Eng.

# Palmprint-based human identification in a mobile scenario

Supervisors:

Michał Choraś, DSc. PhD Eng.,
Rafał Kozik, DSc. PhD Eng.

Bydgoszcz 2020

# Contents

**1. Introduction** . . . 3

1.1. Author's publication . . . 3

**2. Biometrics** . . . 5

2.1. Overview of biometric techniques . . . 5

2.2. Multi-factor authentication systems . . . 5

2.3. Biometric traits . . . 6

2.4. Evaluation of biometric systems . . . 7

**3. Palmprint** . . . 9

3.1. Literature survey . . . 9

3.2. Mobile scenario . . . 9

3.3. Benchmark datasets . . . 11

**4. Proposed methods** . . . 13

4.1. ROI extraction algorithm . . . 13

4.2. Novel method using Histogram of Oriented Gradients . . . 13

4.3. Novel hybrid Colour-Texture method . . . 14

4.4. Novel hybrid Geometric-Texture method . . . 16

4.5. Novel method based on 3-value masks . . . 17

4.6. Novel method based on binary code . . . 19

4.7. Novel method based on Texture Energy Measure . . . 20

**5. Results** . . . 22

5.1. Experiments background . . . 22

5.2. Results of method using Histogram of Oriented Gradients . . . 22

5.3. Results for hybrid Colour-Texture method . . . 22

5.4. Results for hybrid Geometric-Texture method . . . 23

5.5. Results for method based on 3-value masks . . . 23

5.6. Results for method based on binary code . . . 23

5.7. Results for method based on Texture Energy Measure . . . 26

**6. Conclusions and future work** . . . 31

**Bibliography** . . . 33

# 1. Introduction

Nowadays users have a multitude of IT systems at their disposal. Each of those systems requires appropriate security methods. Due to this fact users are overloaded with the number of logins, passwords and codes that they have to remember and change on their daily basis. Biometrics was introduced to overcome this problem. Biometrics can be understood as human identification using a part of the body or human behaviour. This technique has recently been growing in popularity, and it is one of the most crucial issues in computer science now. Among its most challenging problems the following can be listed: pattern protection, liveliness detection, insufficiently high accuracy of identification and the development of new biometric traits.

This thesis concerns palmprint-based user identification in the mobile scenario. Even though this scenario introduces some important limitations, like limited computation power, it is a significant research problem in the biometrics domain. Palmprints have rich structure with numerous wrinkles, ridges and minutiae. As presented in the literature, palmprints may be used in some systems providing high accuracy and efficient computation time, though they are not yet widely adapted in many daily life applications.

The main objective of this work is to propose and evaluate new methods of palmprint-based user verification in the mobile scenario. In order to meet the objective, six brand new methods of palmprint-based user authentication were introduced in this work. Additionally, tests of some pre-processing methods were conducted, and the concept of new biometric dataset was proposed. Four different smartphones were used during the research. The obtained results were evaluated both in terms of accuracy and computation time.

## 1.1. Author's publication

Partially, the results presented in this thesis have been already published. They were described in some papers in journals and at international conferences (Italy, France, Ireland). Some methods were introduced as a result of the 3-month length internship form Erasmus+ program. Author visited the *Pattern Recognition and Applications LAB*. PRALab is one of leading biometric research group in Europe and works at University of Cagliari (Italy, Sardinia).

In the biometrics domain the following author's publication can be listed:

1. **Giełczyk A.** 2018. Bezpieczeństwo wybranych systemów biometrycznych. Nauka niejedno ma imię VI. Wydawnictwa Uczelniane UTP, pp. 37–44.
2. **Giełczyk A.**, Marcialis G.L., Choraś M. 2019. Binary Code for the Compact Palmprint Representation Using Texture Features. [In:] International Conference on Computer Analysis of Images and Patterns (CAIP), Springer, pp. 132–142 (CORE B).
3. **Giełczyk A.** 2017. Biometria mobilna. Perspektywy i wyzwania. Nauka niejedno ma imię V. Wydawnictwa Uczelniane UTP, pp. 21–28.
4. **Giełczyk A.**, Choraś M., Kozik R. 2018. Biometria obrazu dłoni jako część systemu wielopoziomowego uwierzytelniania użytkownika. Przegląd Telekomunikacyjny i Wiadomości Telekomunikacyjne 8–9, pp. 593–596.
5. **Giełczyk A.**, Choraś M., Kozik R. 2018. Hybrid Feature Extraction for Palmprint-Based User Authentication. [In:] International Conference on High Performance Computing & Simulation (HPCS), IEEE, pp. 629–633 (CORE B).
6. **Giełczyk A.**, Choraś M., Kozik R. 2019. Lightweight Verification Schema for Image-Based Palmprint Biometric Systems. Mobile Information Systems, Hindawi (IF=1,635).
7. **Giełczyk A.**, Choraś M., Kozik R. 2019. The mobile palmprint-based verification based on three-value masks. [In:] International Conference on High Performance Computing & Simulation (HPCS), IEEE, pp. 909–914 (CORE B).
8. **Giełczyk A.**, Dembińska K., Choraś M., Kozik R. 2019. Towards Mobile Palmprint Biometric System with the New Palmprint Database. [In:] International Conference on Image Processing and Communications (IP&C), Springer, pp. 149–157.
9. **Wojciechowska A.**, Choraś M., Kozik R. 2018. Evaluation of the pre-processing methods in image-based palmprint biometrics. [In:] International Conference on Image Processing and Communications (IP&C), Springer, pp. 43–48.
10. **Wojciechowska A.**, Choraś M., Kozik R. 2017. The method and an exemplary biometric system to verify users. Journal of Machine Construction and Maintenance. Problemy Eksploatacji 106(3), pp. 97–101.
11. **Wojciechowska A.**, Choraś M., Kozik R. 2017. The overview of trends and challenges in mobile biometrics. Journal of Applied Mathematics and Computational Mechanics 16(2), pp. 173–185.
12. **Wojciechowska A.**, Choraś M., Kozik R. 2018. Recent Advances in Image Pre-processing Methods for Palmprint Biometrics. [In:] International Conference on Computer Recognition Systems (CORES), Springer, pp. 268–275.

# 2. Biometrics

## 2.1. Overview of biometric techniques

Identity verification has become an emerging and important challenge for the digital market and overall society recently. Establishing identity in an efficient manner is necessary in numerous applications including, but not limited to, access control (soft and hard targets, applications), aviation transport, e-banking and mobile devices. Thanks to the raising computing power of electronic devices, the traditional methods of identity verification (cards, passwords and PIN numbers) can be replaced by or coupled with biometrics. Several examples of successful, large scale implementations of biometric systems can be listed in sectors such as, among others: biometric passports (including fingerprints and face images), electronic and mobile banking for authentication and confirmation of transactions, criminal investigation (e.g. AFIS and police fingerprints databases in many countries) and in systems such as EURODAC.

Biometric solutions make human life easier; they make the verification more convenient and faster. What is more, biometrics increases the security of the stored data. As presented in [1], biometrics may also be used as a part of the multi-factor authentication system (MFA). This approach may lead to the creation of the most secured system possible, because it considerably increases the spoofing effort for an attacker.

## 2.2. Multi-factor authentication systems

Using MFA could prevent many companied from data leaks. The 2014 JP-Morgan Chase[1] data breach is one the most well-known cyber-attack in Unites States. Even though one of the biggest banks in US spend $250 million to improve the security of the information system, the system was defrauded. The hackers gained access to log-in credentials by stealing them from the bank's employee. The attack began in early June and was not detected and stopped until mid-August. During this time the hackers compromised the data associated with over 83 million accounts (76 million households and 7 million small businesses).

[1] www.nytimes.com/2014/10/04/your-money/jpmorgan-chase-hack-ways-to-protect-yourself.html

Bank maintained that the attackers had not stolen any money, though it admitted that they had harvested name, passwords, phone numbers and home addresses. After months of investigation done by FBI and U.S. Secret Service the reason of the attack was discovered - failure to upgrade one of the bank's server with a double authentication system.

The Uber data breach took place in late 2016[2]. Two hackers stole private data about company's riders and drivers: names, phone numbers, email addresses and driver's license numbers. They were able to get access to backup files on the third-party server finding the credentials to access it inside code posted to a GitHub repository. The brute-force attack was successful due to the fact that repository was not secured with multifactor authentication, even though it is offered by GitHub. The hackers got private data of 57 million customers but they did not publish data anywhere. Uber paid them $100.000 ransom.

Deloitte's systems were attacked in October of November 2016 but the breach was not discovered until March 2017[3]. The email server was compromised though the admin account giving the hackers privileged, unrestricted access to all areas. In this server there were data from FIFA (football's world governing body), four banks, three airlines, car manufacturers, energy giants and pharmaceutical companies. However, it is said that very few clients had been affected and the company had contacted them. The hackers got information about user names, passwords and IP addresses. Even though Deloitte is one of the world's big four accountancy firms, they do not use multi-factor authentication on the email server.

## 2.3. Biometric traits

On the other hand, in some cases the biometric artefact used in such a system may be attacked. Loss of biometric identity threatens a person's security: it may cause social or financial losses and invades privacy, causing negative emotional impact on victims. Protecting biometric identities is particularly important in the today's digital world because of the limited number of biometrics. Thus, some techniques for protecting the template are implemented, as presented in [2].

Biometrics refers to the measurement and statistical analysis of people's biological (e.g., fingerprint) and behavioural (e.g., gait) characteristics, which can be used to recognize the identity of individuals [3]. The variety of biometrics is presented in Figure 1. Despite of the fact that fingerprint, face and iris recognition are now widespread [4], many other biometric features exist and can provide promising results: hand geometry, ear or palmprint, for example. The part of

[2] www.norton.com/internetsecurity-emerging-threats-uber-breach-57-million.html

[3] www.theguardian.com/business/2017/sep/25/deloitte-hit-by-cyber-attack-revealing-clients-secret-emails

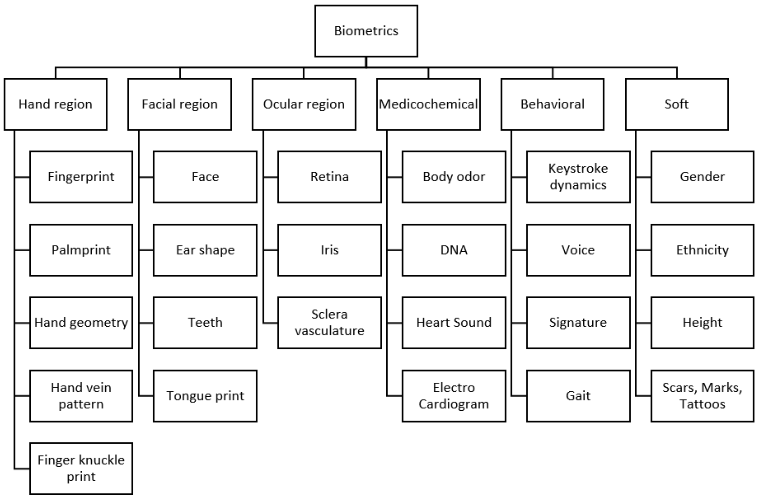

Figure 1. The variety of biometrics [4]

body or the behaviour of a person has to meet some requirements in order to become a biometric trait: universality, distinctiveness, permanence, collectability, performance, acceptability and circumvention.

## 2.4. Evaluation of biometric systems

There are plenty way of system evaluation. First of all, the system need to be accepted by its potential users. Although biometrics is increasingly popular, the users‘ acceptance seems to be discussed less often than it should be. A consumer perspective of a biometric system are presented, for instance in [5] or [6]. In state-of-art articles, the following factors may be important in studying the users‘ perception:

- Sociodemography - depends on age, gender, religion, abilities and personal experiences of users;
- Confidence - depends on users’ feedback and if they trust the system;
- Ease of use - depends on processing time and a sensor quality;
- Privacy issues - depends on potential risk, if the system is easy to defraud, if the template is secured;
- Physical invasiveness - depends on a biometric sensor, if the contact is needed or the sample acquisition is contactless;
- Cultural issues - depends on the user culture.

Moreover, there are some numerical metrics that may show whether the system works fine. Among them Accuracy can be enumerated. Accuracy is expressed with the Eq. 1.

$$Acc = \frac{TP + TN)}{(TP + FP + TN + FN)} \cdot 100\% \tag{1}$$

where:

- TP – true positives – true samples verified as true (correctly);
- FP – false positives – true samples verified as false (incorrectly);
- TN – true negatives – false samples verified as false (correctly);
- FN – false negatives – false samples verified as true (incorrectly).

On the other hand, errors may be evaluated. In the thesis mainly 3 kind of errors are used: FAR (False Acceptance Error) – Eq. 2, FRR (False Rejection Error) – Eq. 3 and EER (Equal Error Rate). EER can be expressed, if values of FAR and FRR are equal.

$$FAR = \frac{FN}{(FN + TN)} \tag{2}$$

$$FRR = \frac{FP}{(FP + FN)} \tag{3}$$

# 3. Palmprint

## 3.1. Literature survey

The palmprint can be recognized as one of the most promising biometric modalities. In a few words, it is the inner surface of the hand. The palmprint provides advantages such as: easy capturing process, relatively big surface, cost effectiveness, non-intrusive nature and rich texture [7]. Moreover, it is similar to the fingerprint, because it is made up of ridge and valleys of the skin. Since the palm's surface is larger than that of a finger, it is arguable that it contains more individual information.

Palmprint recognition has been discussed for more than 15 years now. There are several promising approaches proposed. They differ from each other in each step of identity recognition system (image acquisition, pre-processing, feature extraction and classification). The review of the state-of-the-art methods is presented in Table 1.

## 3.2. Mobile scenario

Originally, palmprints were acquired by direct contact with the scanner. However, physical contact is not desirable by users. It may also lead to a palmprint distortion that can be different depending on the amount of pressure exerted by the user on the scanner. Furthermore, the time of scanning is not sufficiently short for implementation in real-time applications. Thus, the traditional way of acquiring palmprint samples had to be improved [8]. Lately, the mobile user verification methods using biometrics research has also been on the rise. Among these implementations some main groups can be mentioned:

- a solely mobile system - the enrolment and verification processes are performed on the mobile device - in [9] the truly mobile solution based on palmprints was proposed and tested on Apple iPhone 4;
- a partially mobile system or a hybrid system – some steps of the processing pipeline are performed by mobile devices, the rest by the server - in [10] the image acquisition and the features extraction step are provided by mobile device, while the classification is moved to the server;

Table 1. The comparison of the state-of-the-art methods

| Publication | Scenario | Dataset | Pre-processing | Feature extraction | Classification | Accuracy |
|---|---|---|---|---|---|---|
| Zhang et al. [16] | PC | PolyU | Gaussian blur | Gabor filter | Hamminga distance | EER = 0.6% |
| Imitaz et al. [17] | PC | PolyU, IITD | illumination correction | 2D-DWT | Least squares method | up to 99% |
| Ray et al. [18] | PC | PolyU | operator Sobel<br>thresholding | Hough transform | Manhattan distance | up to 90% |
| El-Tarhouni et al. [19] | PC | PolyU | thresholding | LBP<br>Gabor filter | kNN | up to 98% |
| Mokni et al. [20] | PC | PolyU<br>IITD<br>CASIA | steerable filter<br>thresholding | fractal-based features | Random Forest | 98% |
| Tabejamaat et al. [7] | PC | PolyU | Gaussian blur | Gabor filter | angular distance | up to 99% |
| Dubey et al. [21] | PC | PolyU, IITD | Gaussian blur | bank of Gabor filter responses | angular distance | 99% |
| Kumar [22] | PC | CASIA | normalization | DWT | Hamminga distance | EER=1.17% |
| Ahmadi et al. [23] | PC | THUPALMLAB | CNN | Hough transform | MMC-matching algorithm | EER=0.04% |
| Matkowski et al. [24] | PC | CASIA, IITD | CNN | FERnet | SVM, kNN<br>Softmax | up to 99% |
| Choraś et al. [25] | mobile | own dataset | skin colour verification | PCA | Euclidean distance | EER=1.7% |
| Moco et al. [26] | mobile | IT dataset | color normalization | OLOF - Orthogonal Line Ordinal Features | Hamming distance | FRR=9.27%<br>FAR=0.03% |
| Kim et al. [14] | mobile | own dataset | YCbCr colour space | Gabor filter | chi-square distance | EER=2.88% |
| Fang [27] | mobile | own dataset | thresholding, thining | LEM - Line Edge Map | Hamming distance | EER=4.5% |
| Ungureanu et al. [28] | mobile | own dataset | LBP | SIFT | kNN | 90% |
| Tiwari et al. [15] | mobile | own dataset | normalization | SIFT, ORB | own similarity measure | EER=5.55% |
| Leng et al. [13] | mobile | own dataset | hand position verification | Gabor filter | Hamming distance | EER > 2% |
| Zhang et al. [29] | mobile | own dataset | own detector D | SiameseMobileNetwork | own classifier C | 90% |

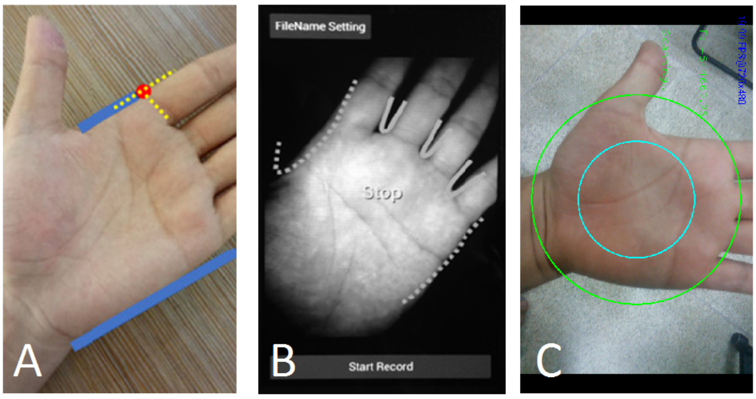

Figure 2. Graphical users assistants proposed by: A: Leng et al. [13], B: Kim et al. [14] and C: Tiwari et al. [15]

- a mobile application with a PC/cloud support - the mobile device performs only image acquisition step, sends the image to the PC or a cloud where the verification is processed and the result is sent back to the mobile device - in [11] the system was based on the face and hand movement, the verification was performed continuously in the cloud. Another example of such an approach was presented in [12], where the term 'Biometrics as a service' was mentioned. Even though in this kind of a system we gain more computing power, we take a risk of losing or repossessing the data during transmission.

Since using biometrics in mobile devices in not supervised, the sample acquisition step is crucial and catch particular researchers' attention. Thus, in numerous works graphical assistants are proposed. The examples of such systems are presented in Figure 2.

### 3.3. Benchmark datasets

In order to perform research and palmprint-based application development, benchmark dataset are often involved: PolyU, IITD and CASIA. They are available online for researchers and provide images of palmprints. The examples of samples from each benchmark dataset are presented in Figure 3, 4 and 5.

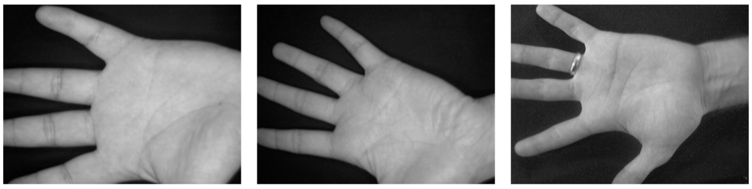

Figure 3. CASIA samples [30]

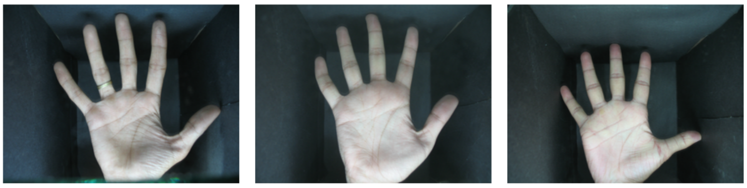

Figure 4. IITD samples [31]

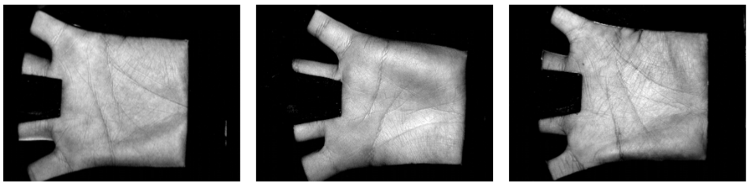

Figure 5. PolyU samples [32]

# 4. Proposed methods

## 4.1. ROI extraction algorithm

All steps of the algorithm are presented in Figure 6. First of all, the contours were found on the source image (A). Then the key points between fingers were extracted. Point A was set between the index finger and the middle finger, while point B between the ring finger and the little finger (B). Then, using the trigonometry function, the angle between points was calculated and the whole image was rotated by this angle (C). The last step was setting the ROI size to $128 \times 128$ (D, E).

## 4.2. Novel method using Histogram of Oriented Gradients

The main idea of this method is to use Histogram of Oriented Gradients for features extraction and Euclidean distance for matching. In pre-processing step 4 different methods of enhancing image are used: Gaussian blur, median filter, bilateral filter and sharpening with various parameters.

1. **Gaussian blur** is a low pass filter and it is implemented palmprint recognition. It is calculated separately for each pixel in the image and uses the Eq. 4., where $x$, $y$ are the distances to the original X and Y axis and $\sigma$ is the standard deviation. Each pixel gets the value equal to the weighted average of its neighbourhood. The size of neighbourhood (called kernel) may be modified. The bigger size is set, the more blurred the image is.

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{4}$$

2. **Median blur** is a filter which reduces effectively the impulsive noises (like salt and pepper noise). It is also widely implemented in image processing systems. It is calculated for each pixel of the image and also depends on the chosen neighbourhood size (kernel). Each pixel gets the value equal to the median value of pixels in the neighbourhood. Unfortunately, using this filter may affect the edges.
3. **Bilateral filter** uses weighted average as well, but introduces the second parameter, which modifies the Gaussian kernel shape. Although the time of

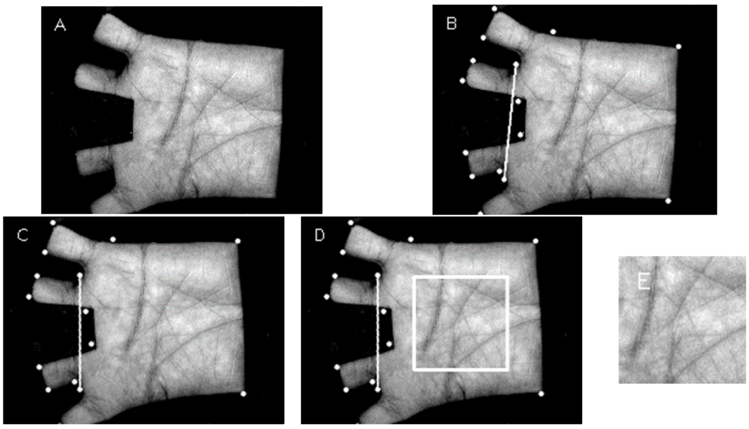

Figure 6. Steps of the ROI extraction algorithm

the computing for bilateral filter is higher than for other filters, it preserves better edges.

4. **Sharpening** may be implemented as subtracting the blurred image from the original one. For blurring it is possible to use one of above mentioned filters, but in the research the Gaussian blurred was used. It is visible, that the bigger is Gaussian kernel, the sharper the image is.

### 4.3. Novel hybrid Colour-Texture method

In the proposed algorithm there are three kind of features extracted. Firstly, we calculate the image moments (called also raw moments). Those are the texture-based features. However, before the moments are calculated, the pre-processing step is essential. First, the ROI is slightly blurred with the Gaussian blur to get rid of the noise and unwanted details. Then, Canny operator is performed. The edges detection is needed because moments have to be calculated from edges. Moments may be expressed with Eq. 5, where $x$, $y$ – distance from the origin to the horizontal and vertical axis, $i$, $j$ – the number of moments and $I$ - the intensity of pixel. In our method three moments-based are used. Those are area ($M_{00}$) and coordinates of the mass centre point ($\overline{x}$, $\overline{y}$), which are expressed with Eq. 6. and Eq. 7, respectively.

$$M_{ji} = \sum_{x}\sum_{y} I(x,y)x^{i}y^{j} \tag{5}$$

$$\overline{x} = \frac{M_{10}}{M_{00}} \tag{6}$$

$$\overline{y} = \frac{M_{01}}{M_{00}} \tag{7}$$

The next subset of features comes from the Discrete Fourier Transform. DFT for images (two dimensional signals) decomposes the image into its sinus and cosines components. It means that DFT transforms the image from the spatial domain to the frequency domain. It is possible to get two images: real (Re) and complex (Im) as products of the DFT. DFT is calculated also for the source image from the database. To compare the palmprints, the cross correlation is performed. Each element of the cross correlation's result (R) matrix is expressed with Eq. 8, where $x$, $y$ and $i$, $j$ - distances from the origin to the horizontal and vertical axis in images A and B (image from user and from the database).

$$R(x, y) = \sum_{ij} (A(i, j) \cdot B(x + i, y + j)) \tag{8}$$

For two images having the same size, cross correlation result matrix has a size $1 \times 1$. Value of the only one element of the matrix (C) may be expressed with Eq. 9

$$C = \sum_{ij} (A(i, j) \cdot B(i, j)) \tag{9}$$

To obtain more objective result the value of C should be normalized. After normalization it will have a value in range $< 0; 1 >$ and may be expressed with Eq. 10.

$$C_{NORM} = \frac{\sum_{ij} (A(i, j) \cdot B(i, j))}{\sqrt{\sum_{ij} A(i, j)^2 \cdot \sum_{ij} B(i, j)^2}} \tag{10}$$

The normalized cross correlation is performed for three images: the whole ROI (CC), the real part of ROI (Re) and the imaginary part of ROI (Im). The last feature is a color-based one. To extract the feature the histogram has to be calculated. However, before the histogram calculation, the normalization is essential. Thanks to the normalization step, the histogram uses the whole range of values (for grayscale images the range is $< 0; 255 >$) and gives more information about the image. Histograms may be simply compared to each other. The same histograms will give the result equal to one. The metric from histograms comparison (histogram of ROI from the user and from the database) is stored in the features vector as CH. Thanks to the normalization performed for each feature in the vector, no sophisticated measure is needed in the matching step. Values of features are add together and based on experimentally set threshold classified as positive or negative.

### 4.4. Novel hybrid Geometric-Texture method

The proposed algorithm uses the hand shape and the palmprint texture. First, normalization and thresholding were performed. Due to the variety of samples, the threshold was based on the average calculated from the whole sample. Then, the hand contour was detected and convex hull was found around the contour. From the convex hull, convexity defects were extracted. The set of 9 key points was found from contours:

0. top of the little finger,
1. valley between little and ring fingers,
2. of the ring finger,
3. valley between ring and middle fingers;
4. top of the middle finger,
5. valley between middle and index fingers,
6. top of the index finger,
7. mass center of contour;
8. mass center of convex hull.

Then, the features extraction part is executed. Due to the future implementation in a mobile scenario, we decided to use a short feature vector. The short vectors should not be excessively challenging for mobile devices. The elements of the feature vector are presented in equation 11, where dist is the distance between the key points. The distances are calculated using equation 12, where A and B are points between which the distance is estimated, $A_x$, $A_y$, $B_x$, and $B_y$, in which x and y are the coordinates of the points.

$$\begin{bmatrix} G_1 \\ G_2 \\ G_3 \\ G_4 \\ G_5 \end{bmatrix} = \begin{bmatrix} dist(0,1)/dist(1,5) \\ dist(2,3)/dist(1,5) \\ dist(4,5)/dist(1,5) \\ dist(5,6)/dist(1,5) \\ dist(7,8)/dist(1,5) \end{bmatrix} \tag{11}$$

$$dist(A,B) = \sqrt{(A_x - B_x)^2 + (A_y - B_y)^2} \tag{12}$$

The next step is matching.First, texture based template matching is used. There are multiple methods available. We decided to use three of them: CCOEFF, CCORR and SQDIFF in their normalized versions and compare the obtained results. Before the equality is measured between two ROI images, they need to be resized to an equal size. To calculate the similarity Eq. 13, 16, and 17 are used, where $x$, $y$ and $i$, $j$ - coordinates of points, $x, i = [0; w-1]$, $y, j = [0; h-1]$, $w$, $h$ - width and height of the ROI, $I$, $T$ - base image ROI and test image ROI. Normalization ensures that the optimal result is equal to 1. Values of features are add together and based on experimentally set threshold classified as positive or negative.

$$TM_{CCOEFF} = \frac{\sum_{x,y}(T'(x,y) \cdot I'(x,y))}{\sqrt{\sum_{x,y} T'(x,y)^2 \cdot \sum_{x,y} I'(x,y)^2}} \tag{13}$$

where:

$$T'(x,y) = T(x,y) - \frac{1}{w \cdot h} \sum_{i,j} T(i,j) \tag{14}$$

$$I'(x,y) = I(x,y) - \frac{1}{w \cdot h} \sum_{i,j} I(i,j) \tag{15}$$

$$TM_{CCORR} = \frac{\sum_{x,y}(T(x,y) \cdot I(x,y))}{\sqrt{\sum_{x,y} T(x,y)^2 \cdot \sum_{x,y} I(x,y)^2}} \tag{16}$$

$$TM_{SQDIFF} = 1 - \frac{\sum_{x,y}(T(x,y) - I(x,y))^2}{\sqrt{\sum_{x,y} T(x,y)^2 \cdot \sum_{x,y} I(x,y)^2}} \tag{17}$$

## 4.5. Novel method based on 3-value masks

In this approach the ROI is extracted using the previously presented algorithm. Then, the feature extraction is performed. In order to use the mask, it is essential to calculate the average value of the pixels $\overline{x}$ from the whole image and the standard deviation $\delta$, which are expressed with the Eq. 18 and Eq. 19, where $N$ - total number of pixels, $x_i$ - intensity of the $i$-pixel. Using the average and the standard deviation, two variables are introduced: $min$ and $max$, which are expressed with the Eq. 20 and Eq. 21.

$$\overline{x} = \frac{\sum_{i=1}^{N} x_i}{N} \tag{18}$$

$$\delta = \sqrt{\frac{\sum_{i=1}^{N} (x_i - \overline{x})^2}{N}} \tag{19}$$

$$min = \overline{x} - \frac{\delta}{2} \tag{20}$$

$$max = \overline{x} + \frac{\delta}{2} \tag{21}$$

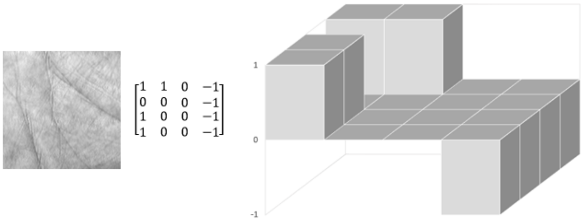

Figure 7. Example of a ROI with the corresponding mask presented as a vector and in a graphical form

Then the ROI image is divided into 16 non-overlapping blocks and the average value $\overline{x_a}$ of the pixel intensity for each block is calculated. Then the 16-element vector of features $vec$ is created using the Eq. 22.

$$vec[a] = \begin{cases} -1 & \text{if } \overline{x_a} \leq min \\ 0 & \text{if } min < \overline{x_a} \leq max \\ 1 & \text{if } \overline{x_a} > max \end{cases} \tag{22}$$

Thus, the vector of features is created. It may be visualized as the three-value matrix or as a 3D chart, which are presented in Figure 7.

The next step of the user authentication system is the classification. We decided to use the following methods: MD (*Manhattan distance*), SVM (*Support Vector Machine*), DT (*Decision Tree*). The Manhattan distance ($MD$) is a measure that can be very useful for the comparison of vectors. It is expressed with the Eq. 23, where $x$, $y$ - compared vectors of features, $n$ - the total number of elements in vectors $x$, $y$ and $k$ - the consecutive number of elements in the vector. The SVM is an example of a supervised learning algorithm and it needs to be trained in the enrollment stage of the processing system. In this case the most useful was the SVM kernel called RBF (*Radial Basis Function*). It is a method for clustering 2-class data. For optimal examples it is possible to find the hyper-plane that divides the higher-dimensional space into two classes. A Decision Tree is a binary tree that can be used either for classification or for regression. It minimizes the sum of differences between the input vector and the trained data in each node of the tree. Decision Trees are the basic algorithms for some other algorithms such as Boosting and Random Trees.

$$MD(x,y) = \sum_{k=1}^{n} |x_k - y_k| \tag{23}$$

## 4.6. Novel method based on binary code

For the feature extraction we used two measurements: the Haralick Sum Average (HSAvg) and Haralick Sum Variance. They are calculated using the Eq. 24 and Eq. 25 respectively, where: $N_g$ - number of gray levels (here: $N_g = 255$), $p(i,j)$ - entry in GLCM spatial-dependency matrix, $HSEnt$ - Haralick Sun Entropy expressed with the Eq. 27, $p_{x+y}(i)$ - entry in the marginal-probability matrix obtained by summing the rows of $p(i,j)$ and expressed using Eq. 26

$$HSAvg = \sum_{i=2}^{2N_g} i \cdot p_{x+y}(i) \tag{24}$$

$$HSVar = \sum_{i=2}^{2N_g} (i - HSEnt)^2 \cdot p_{x+y}(i) \tag{25}$$

$$p_{x+y}(i) = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} p(i,j) \tag{26}$$

$$HSEnt = \sum_{i=2}^{2N_g} p_{x+y}(i) \cdot \log \{p_{x+y}(i)\} \tag{27}$$

The above-mentioned metrics are calculated first for the whole ROI, thus obtaining a global measurements of the palmprint texture, and then, for the 16 non-overlapping blocks (size 32 × 32), thus obtaining a local measurements of the palmprint texture.

Eq. 28 and Eq. 29 were then used to compute the comparison between the above global and local metrics. The obtained bit value is included in a binary vector. In such a way, we obtain a 32 bit-lenght code. To the best of our knowledge, this is the smallest bit-code used for biometric applications.

In the matching step, two bit-codes are compared by the City Block Distance (CBD) which is expressed as Eq. 30, where $A$, $B$ - compared vectors and $n$ - vectors length.

The final decision is done by setting an appropriate acceptance threshold to the computed CBD, thus obtaining the standard classification in Genuine user and Impostors classes.

$$vec[a] = \begin{cases} 0 & \text{if } HSAvg_{local} \leq HSAvg_{global} \\ 1 & \text{if } HSAvg_{local} > HSAvg_{global} \end{cases} \tag{28}$$

$$vec[a] = \begin{cases} 0 & \text{if } HSVar_{local} \leq HSVar_{global} \\ 1 & \text{if } HSVar_{local} > HSVar_{global} \end{cases} \tag{29}$$

$$CBD = \sum_{k=1}^{n} |A_k - B_k| \tag{30}$$

## 4.7. Novel method based on Texture Energy Measure

In this approach we adopted the LAWS, widely used texture analysis vectors. Using a pair of vectors, the two-dimensional kernel is created. Convolution of kernel and the source image may emphasize some specific characteristic (like spots, ripple or edges). In our research, we decided to use the vectors called L5 (Eq. 31) and S5 (Eq. 32). This pair of vectors gives one of two kernels (depending on the vector order - Eq. 33 and Eq. 34). In order to make the result rotational invariant, it is recommended to combine the symmetric pair of kernels.

$$L5 = [1, 4, 6, 4, 1] \tag{31}$$

$$S5 = [-1, 0, 2, 0, -1] \tag{32}$$

$$kernel_{LS} = \begin{bmatrix} -1, & 0, & 2, & 0, & -1 \\ -4, & 0, & 8, & 0, & -4 \\ -6, & 0, & 12, & 0, & -6 \\ -4, & 0, & 8, & 0, & -4 \\ -1, & 0, & 2, & 0, & -1 \end{bmatrix} \tag{33}$$

$$kernel_{SL} = \begin{bmatrix} -1, & -4, & -6, & -4, & -1 \\ 0, & 0, & 0, & 0, & 0 \\ 2, & 8, & 12, & 8, & 2 \\ 0, & 0, & 0, & 0, & 0 \\ -1, & -4, & -6, & -4, & -1 \end{bmatrix} \tag{34}$$

In our proposed method, we used the convolution operations. Firstly, the ROI need to be extracted. After the ROI extraction, the convolution operations are performed. After applying kernel on the image, the texture energy measure (TEM) is calculated. It is computed by summing the absolute values in a local neighborhood using Eq. 35, where $C(i, j)$ is the illumination of the pixel and $i$, $j$ - x and y axis coordinates. During the research, we used the block sizes $m = 15 \times n = 15$. It means that $TEM$ is calculated 64 times for the source image, since the image size is $120px \times 120px$. After the $TEM$ measure is known for each block, the average value of $TEM$ is calculated ($TEM_{avg}$). Then, the 64-bit length binary vector of features is created using Eq. 36.

$$TEM = \sum_{i=1}^{m} \sum_{j=1}^{n} |C(i,j)| \tag{35}$$

$$vec[a] = \begin{cases} 0 & \text{if } TEM < TEM_{avg} \\ 1 & \text{if } otherwise \end{cases} \tag{36}$$

After implementing this kind of approach, we decided to run some experiments. The experiments are described in detail in the next section. Nevertheless, the obtained results were not satisfying. In order to improve the accuracy of the proposed system, we decided to add more features to the vector.

As the convolution operation using kernels $SL_{kernel}$ and $LS_{kernel}$ emphasize the horizontal and vertical lines, we decided to extend the feature vector by adding those obtained from runlengths. Runlength is the feature that can provide multiple pieces of information about the analysed texture. Basically, in a course texture it is expected that the long runs will occur relatively often, whereas a fine texture will contain a higher proportion of short runs. Runlengths are analysed both horizontally and vertically, and they are the amount of consecutive primitives (pixels) having the intensity equal to 255 (white pixels in thresholded image). Six features were added to the vector: 1) number of runs having $length \leq 3$, 2) number of $runlengths > 60$ and 3) the maximum runlength of image - analysed both horizontally and vertically.

# 5. Results

## 5.1. Experiments background

For running experiments we used 4 popular smartphones. The detailed specification of the involved devices is presented in table 2. Devices were bought in internal university projects: BSM 81/2017 and BN 43/2019.

Experiments were conducted using 3-fold methodology. The results of each method test with the average accuracy and time of computing for each devices are presented in nest sections.

Due to the limited computation power provided by the mobile devices, we decided to move the learning part of the system to the PC. Thus, the training is performed using the computer and as a result an XML file is created. Then, the file is moved to the mobile phone and the experiments are performed. For each testing sample, the system gives the answer: genuine or impostor.

## 5.2. Results of method using Histogram of Oriented Gradients

In order to test this method the whole system was moved to the mobile phones. Then, different pre-processing methods was used: raw samples (without pre-processing), Gaussian blur $(3 \times 3)$, Gaussian blur $(5 \times 5)$, median filter $(3 \times 3)$, median filter $(5 \times 5)$, bilateral filter and sharpening.

## 5.3. Results for hybrid Colour-Texture method

Table 5 presents the accuracy reached, the time of processing for each compared method and the difference between system using only geometric features

Table 2. Specification of mobile devices used for the research

| Device | Operating system | Processor | RAM |
|---|---|---|---|
| Samsung A5 2017 | Android 8.0 Oreo | $8 \times 1.90$GHz | 3GB |
| Xiaomi Mi6 | Android 8.0 Oreo | $8 \times 2.45$GHz | 4GB |
| Huawei P10 Lite | Android 7.0 Nougat | $8 \times 2.10$GHz | 3GB |
| Samsung Galaxy S5 | Android 6.0 Marshmallow | $4 \times 2.50$GHz | 2GB |

Table 3. Accuracy of HOG-based method for different pre-processing methods

| Pre-processing method | Accuracy |
|---|---|
| raw samples | 94,0% |
| Gaussian blur (3 × 3) 3 × 3 | 95,1% |
| Gaussian blur (5 × 5) | 96,1% |
| median filter (3 × 3) | 96,2% |
| median filter (5 × 5) | 95,8% |
| bilateral filter | **97,1%** |
| sharpening | 93,3% |

(row 1.) and system using fusion of features (rows 2. - 4.). It is visible that each texture-based method was able to improve the accuracy without time of computing increased significantly. In Table 6 time for one authorisation is presented.

## 5.4. Results for hybrid Geometric-Texture method

Table 7 presents the accuracy reached, the time of processing for each tested fusion. It is visible that each method was able to improve the accuracy but the highest result is not impressive (only 83%). In Table 6 time for one authorisation is presented.

## 5.5. Results for method based on 3-value masks

Using the above-mentioned ROI detection algorithm, features extraction by 3-values masks and three different methods of classification we were able to obtained the accuracy results up to 95.5%. The more detailed results (for 3-fold methodology) are presented in Table 9.

Besides the system accuracy, the times of training and the times of prediction were observed. The details of the average computational time are presented in Table 10. The obtained results are comparable for each method: Decision Tree (DT), City Block Distance (CBD) and SVM with different code length - 16, 25 and 36 elements.

## 5.6. Results for method based on binary code

Experiments were performed on the PolyU database. The following experimental results are subdivided in three groups: the first one is aimed to show that using the *Haralick Sum Average* and *Haralick Sum Variance* together is more efficient than using one of them. This show that the bit-lenght is someway constrained to be not less than 32 bits, due to the fact that HSA and HSV encodes probably low correlated characteristics of the palmprint. That is why, we perform

Table 4. Computational time comparison for HOG method

| **Xiaomi Mi6** | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Fold** | **Gaussian blur** $3 \times 3$ | **Gaussian blur** $5 \times 5$ | **Median filter 3** | **Median filter 5** | **Bilateral filter** | **Sharpening** | **F. ex. + classification** |
| **Fold 1** | 9.91 ms | 9.26 ms | 9.13 ms | 11.12 ms | 32.59 ms | 11.06 ms | 7.50 ms |
| **Fold 2** | 9.71 ms | 9.26 ms | 9.10 ms | 11.51 ms | 34.37 ms | 11.47 ms | 6.79 ms |
| **Fold 3** | 9.40 ms | 8.85 ms | 8.94 ms | 10.79 ms | 32.31 ms | 10.35 ms | 6.75 ms |
| **Average** | 9.67 ms | 9.13 ms | 9.06 ms | 11.14 ms | 33.09 ms | 10.96 ms | 7.01 ms |
| **Huawei P10 Lite** | | | | | | | |
| **Fold** | **Gaussian blur** $3 \times 3$ | **Gaussian blur** $5 \times 5$ | **Median filter 3** | **Median filter 5** | **Bilateral filter** | **Sharpening** | **F. ex. + classification** |
| **Fold 1** | 10.52 ms | 11.07 ms | 10.11 ms | 12.92 ms | 35.81 ms | 11.98 ms | 7.21 ms |
| **Fold 2** | 11.23 ms | 10.88 ms | 10.96 ms | 13.07 ms | 34.98 ms | 12.62 ms | 8.11 ms |
| **Fold 3** | 10.87 ms | 10.91 ms | 9.91 ms | 13.18 ms | 34.92 ms | 12.78 ms | 7.43 ms |
| **Average** | 10.87 ms | 10.95 ms | 10.33 ms | 13.06 ms | 35.24 ms | 12.46 ms | 7.58 ms |
| **Samsung Galaxy A5 2017** | | | | | | | |
| **Fold** | **Gaussian blur** $3 \times 3$ | **Gaussian blur** $5 \times 5$ | **Median filter 3** | **Median filter 5** | **Bilateral filter** | **Sharpening** | **F. ex. + classification** |
| **Fold 1** | 12.50 ms | 12.15 ms | 12.35 ms | 14.61 ms | 37.80 ms | 13.88 ms | 7.41 ms |
| **Fold 2** | 12.10 ms | 11.96 ms | 11.69 ms | 14.00 ms | 36.36 ms | 13.20 ms | 6.54 ms |
| **Fold 3** | 12.35 ms | 12.09 ms | 11.89 ms | 14.28 ms | 37.13 ms | 13.45 ms | 6.58 ms |
| **Average** | 12.32 ms | 12.07 ms | 11.98 ms | 14.30 ms | 37.10 ms | 13.51 ms | 6.84 ms |
| **Samsung Galaxy S5** | | | | | | | |
| **Fold** | **Gaussian blur** $3 \times 3$ | **Gaussian blur** $5 \times 5$ | **Median filter 3** | **Median filter 5** | **Bilateral filter** | **Sharpening** | **F. ex. + classification** |
| **Fold 1** | 12.85 ms | 12.18 ms | 11.67 ms | 15.42 ms | 42.88 ms | 14.40 ms | 13.90 ms |
| **Fold 2** | 12.05 ms | 11.83 ms | 10.90 ms | 14.05 ms | 38.92 ms | 14.03 ms | 11.13 ms |
| **Fold 3** | 12.45 ms | 12.30 ms | 11.42 ms | 14.33 ms | 38.98 ms | 14.70 ms | 11.97 ms |
| **Fold** | 12.45 ms | 12.11 ms | 11.33 ms | 14.60 ms | 40.26 ms | 14.38 ms | 12.33 ms |

Table 5. Obtained results: accuracy, time, accuracy improvement and time delay evaluated with experimentally set threshold

| Method | Acc. | Acc. imp. |
|---|---|---|
| Geometric | 74% | - |
| Geometric + CCOEFF | 83% | 9% |
| Geometric + CCORR | 77% | 3% |
| Geometric + SQDIFF | 78% | 4% |

Table 6. Computational time comparison for CT method

| **Xiaomi Mi6** | | | |
|---|---|---|---|
| **Fold** | **Pre-processing** | **F. ex. + Classification** | **Sum** |
| **Fold 1** | 14.9 ms | 22.4 ms | 37.3 ms |
| **Fold 2** | 16.3 ms | 19.9 ms | 36.2 ms |
| **Fold 3** | 16.4 ms | 19.1 ms | 35.5 ms |
| **Average** | 15.8 ms | 20.5 ms | 36.3 ms |
| **Huawei P10 Lite** | | | |
| **Fold** | **Pre-processing** | **F. ex. + Classification** | **Sum** |
| **Fold 1** | 13.5 ms | 22.1 ms | 35.6 ms |
| **Fold 2** | 13.6 ms | 21.9 ms | 35.5 ms |
| **Fold 3** | 14.7 ms | 23.4 ms | 38.1 ms |
| **Average** | 13.9 ms | 22.5 ms | 36.4 ms |
| **Samsung Galaxy A5 2017** | | | |
| **Fold** | **Pre-processing** | **F. ex. + Classification** | **Sum** |
| **Fold 1** | 12.5 ms | 26.4 ms | 38.9 ms |
| **Fold 2** | 12.5 ms | 25.2 ms | 37.7 ms |
| **Fold 3** | 11.5 ms | 25.2 ms | 36.7 ms |
| **Average** | 12.2 ms | 25.6 ms | 37.7 ms |
| **Samsung Galaxy S5** | | | |
| **Fold** | **Pre-processing** | **F. ex. + Classification** | **Sum** |
| **Fold 1** | 14.7 ms | 27.5 ms | 42.2 ms |
| **Fold 2** | 12.5 ms | 28.7 ms | 41.2 ms |
| **Fold 3** | 13.2 ms | 34.0 ms | 47.2 ms |
| **Average** | 13.5 ms | 30.1 ms | 43.5 ms |

Table 7. GT method accuracy

| Method | Accuracy |
|---|---|
| Geometric | 74% |
| Geometric + CCOEFF | **83%** |
| Geometric + CCORR | 77% |
| Geometric + SQDIFF | 78% |
| Geometric + CCOEFF + CCORR + SQDIFF | 83% |
| Geometric + 2 · CCOEFF | 83% |
| Geometric + 3 · CCOEFF | 83% |
| Geometric + 4 · CCOEFF | 83% |

our algorithm for each two possible sample pair three times: 1) using only *HSAvg* and 16-bits length vector 2) using only *HSVar* and 16-bits length vector 3) using both *HSAvg* and *HSVar* and the whole 32-bits length vector. The results of this experiment are presented in Table 11.

The second group of experiments was aimed to investigate the dependence on the number of templates. As a matter of fact, it is quite acknowledged that the accuracy may be increased by storing more templates into the system. Therefore, we used 3 and 6 templates per user. During testing the vector of features is compared to each enrollment sample and the average value is calculated. This average value is compared to an experimentally fixed threshold. Comparing the testing sample to the set of enrollment samples is essential also in the real live applications. The results of all experiments are provided in Table 12. The enlargement of the enrollment set size causes a slight increase of the accuracy (from 89.15% to 92.16%), however it may also due to some statistical fluctuation because of the random selection of the templates. The obtained results demonstrate that the high accuracy of verification is weakly dependent on the number of templates. The computation time comparison is presented in Table 13.

## 5.7. Results for method based on Texture Energy Measure

For evaluating our methods, we chose to use the PolyU palmprint database. For the first experiment, we used the 64-length vector of features (coming only from the TEM and convolution operations). In the experiments, we used a 10-fold classification - we run the experiment 10 times using a different set of training sets in order to provide lack of dependency on data. We decided to use 10 positive and 10 negative samples in the training step. The average accuracy (understood as an number of well classified samples divided by the total number of samples) of the method is 84.4%, so we did not find it satisfying. Therefore, we extended the vector and ran the experiment again. The average accuracy reached 94.5%, which gave over 10% of accuracy increase comparing to the shortest 64-bit vector. The more detailed results of both experiments are presented in Table 14.

Table 8. Computational time comparison for GT method

| Xiaomi Mi6 | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Fold** | **Geom.** | **Geometric<br>+CCOEFF** | **Geometric<br>+CCORR** | **Geometric<br>+SQDIFF** | **Geom.+CCOEFF<br>+CCORR+SQDIFF** | **Geometric<br>+2 x CCOEFF** | **Geometric<br>+3 x CCOEFF** | **Geometric<br>+4 x CCOEFF** |
| **Fold 1** | 32.2 ms | 32.3 ms | 31.9 ms | 32.5 ms | 37.2 ms | 32.1 ms | 32.8 ms | 32.1 ms |
| **Fold 2** | 34.0 ms | 33.3 ms | 32.3 ms | 32.3 ms | 37.6 ms | 32.7 ms | 32.1 ms | 31.7 ms |
| **Fold 3** | 32.4 ms | 32.4 ms | 32.5 ms | 31.8 ms | 37.7 ms | 31.8 ms | 31.8 ms | 32.4 ms |
| **Average** | 32.9 ms | 32.7 ms | 32.3 ms | 32.2 ms | 37.5 ms | 32.2 ms | 32.2 ms | 32.1 ms |

| Huawei P10 Lite | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Fold** | **Geom.** | **Geometric<br>+CCOEFF** | **Geometric<br>+CCORR** | **Geometric<br>+SQDIFF** | **Geom.+CCOEFF<br>+CCORR+SQDIFF** | **Geometric<br>+2 x CCOEFF** | **Geometric<br>+3 x CCOEFF** | **Geometric<br>+4 x CCOEFF** |
| **Fold 1** | 58.6 ms | 50.1 ms | 51.3 ms | 54.9 ms | 58.1 ms | 55.1 ms | 56.2 ms | 50.0 ms |
| **Fold 2** | 53.9 ms | 52.0 ms | 51.8 ms | 56.4 ms | 56.7 ms | 52.6 ms | 58.8 ms | 55.2 ms |
| **Fold 3** | 56.0 ms | 54.7 ms | 55.3 ms | 50.8 ms | 59.3 ms | 52.5 ms | 53.1 ms | 51.4 ms |
| **Average** | 56.2 ms | 52.3 ms | 52.8 ms | 54.0 ms | 58.0 ms | 53.4 ms | 56.0 ms | 52.2 ms |

| Samsung Galaxy A5 2017 | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Fold** | **Geom.** | **Geometric<br>+CCOEFF** | **Geometric<br>+CCORR** | **Geometric<br>+SQDIFF** | **Geom.+CCOEFF<br>+CCORR+SQDIFF** | **Geometric<br>+2 x CCOEFF** | **Geometric<br>+3 x CCOEFF** | **Geometric<br>+4 x CCOEFF** |
| **Fold 1** | 80.8 ms | 89.3 ms | 96.8 ms | 83.5 ms | 101.9 ms | 839. ms | 91.5 ms | 84.0 ms |
| **Fold 2** | 81.6 ms | 80.5 ms | 83.4 ms | 81.8 ms | 95.4 ms | 80.3 ms | 82.7 ms | 81.1 ms |
| **Fold 3** | 75.8 ms | 81.5 ms | 79.9 ms | 79.4 ms | 94.9 ms | 80.0 ms | 79.8 ms | 80.8 ms |
| **Average** | 79.4 ms | 83.8 ms | 86.7 ms | 81.6 ms | 97.4 ms | 81.4 ms | 84.7 ms | 82.0 ms |

| Samsung Galaxy S5 | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Fold** | **Geom.** | **Geometric<br>+CCOEFF** | **Geometric<br>+CCORR** | **Geometric<br>+SQDIFF** | **Geom.+CCOEFF<br>+CCORR+SQDIFF** | **Geometric<br>+2 x CCOEFF** | **Geometric<br>+3 x CCOEFF** | **Geometric<br>+4 x CCOEFF** |
| **Fold 1** | 86.2 ms | 79.5 ms | 86.3 ms | 82.3 ms | 81.7 ms | 90.3 ms | 85.2 ms | 87.9 ms |
| **Fold 2** | 66.7 ms | 72.7 ms | 74.2 ms | 76.2 ms | 82.4 ms | 80.0 ms | 76.4 ms | 72.8 ms |
| **Fold 3** | 72.1 ms | 72.8 ms | 73.9 ms | 78.3 ms | 89.0 ms | 78.0 ms | 80.9 ms | 77.8 ms |
| **Average** | 75.0 ms | 75.0 ms | 78.1 ms | 78.9 ms | 84.4 ms | 82.8 ms | 80.8 ms | 79.5 ms |

Table 9. Accuracy of 3-value mask method with different classification step

| Method | Fold 1 | Fold 2 | Fold 3 | Average |
|---|---|---|---|---|
| Manhattan distance | 93.0% | 93.5% | 92.2% | 92.9% |
| SVM | 95.7% | 95.2% | 95.6% | **95.5%** |
| Decision tree | 85.5% | 85.6% | 85.5% | 85.5% |

Table 10. Computational time comparison for 3-value mask method

| **Xiaomi Mi6** | | | | | | |
|---|---|---|---|---|---|---|
| **Fold** | **Pre-processing** | **DT** | **SVM-16** | **SVM-25** | **SVM-36** | **CBD** |
| **Fold 1** | 14.7 ms | 63.9 ms | 64.1 ms | 69.9 ms | 73.3 ms | 65.8 ms |
| **Fold 2** | 14.8 ms | 64.2 ms | 64.3 ms | 73.3 ms | 70.2 ms | 64.3 ms |
| **Fold 3** | 13.9 ms | 64.2 ms | 65.3 ms | 71.8 ms | 71.9 ms | 64.2 ms |
| **Average** | 14.5 ms | 64.1 ms | 64.5 ms | 71.8 ms | 71.6 ms | 64.8 ms |
| **Huawei P10 Lite** | | | | | | |
| **Fold** | **Pre-processing** | **DT** | **SVM-16** | **SVM-25** | **SVM-36** | **CBD** |
| **Fold 1** | 14.2 ms | 86.4 ms | 89.3 ms | 88.7 ms | 90.0 ms | 89.0 ms |
| **Fold 2** | 13.8 ms | 85.0 ms | 84.1 ms | 86.4 ms | 89.5 ms | 83.9 ms |
| **Fold 3** | 14.3 ms | 84.7 ms | 83.8 ms | 85.3 ms | 86.5 ms | 83.7 ms |
| **Average** | 14.1 ms | 85.4 ms | 85.7 ms | 86.8 ms | 88.7 ms | 85.5 ms |
| **Samsung Galaxy A5 2017** | | | | | | |
| **Fold** | **Pre-processing** | **DT** | **SVM-16** | **SVM-25** | **SVM-36** | **CBD** |
| **Fold 1** | 10.3 ms | 195.8 ms | 194.3 ms | 183.8 ms | 190.8 ms | 194.1 ms |
| **Fold 2** | 8.3 ms | 180.4 ms | 178.9 ms | 181.7 ms | 187.4 ms | 183.2 ms |
| **Fold 3** | 8.1 ms | 177.7 ms | 180.0 ms | 183.1 ms | 188.8 ms | 179.3 ms |
| **Average** | 8.9 ms | 184.6 ms | 184.4 ms | 182.9 ms | 189.0 ms | 185.5 ms |
| **Samsung Galaxy S5** | | | | | | |
| **Fold** | **Pre-processing** | **DT** | **SVM-16** | **SVM-25** | **SVM-36** | **CBD** |
| **Fold 1** | 13.4 ms | 162.8 ms | 169.2 ms | 173.3 ms | 187.2 ms | 164.8 ms |
| **Fold 2** | 10.3 ms | 176.3 ms | 178.4 ms | 150.9 ms | 185.3 ms | 165.3 ms |
| **Fold 3** | 10.0 ms | 168.6 ms | 172.1 ms | 190.4 ms | 193.5 ms | 177.3 ms |
| **Average** | 11.2 ms | 169.9 ms | 172.5 ms | 171.5 ms | 188.7 ms | 169.1 ms |

Table 11. Comparison of accuracy for *Haralick Sum Average* and *Haralick Sum Variance* separately and together

| Name | HSAvg | HSVar | HSAvg + HSVar |
|---|---|---|---|
| **Threshold** | 3 | 3 | 6 |
| **Genuine** | 83.9% | 83.6% | 83.5% |
| **Impostor** | 10.0% | 9.0% | 7.0% |
| **Accuracy** | 89.6% | 90.8% | **92.0%** |

Table 12. Obtained results: accuracy for each fold, standard deviation of the accuracy, average of the accuracy for each experiment - with 1, 3 and 6 enrollment samples

| Experiment | Fold 01 | Fold 02 | Fold 03 | Standard deviation | Accuracy |
|---|---|---|---|---|---|
| **1 sample** | – | – | – | – | 91.96% |
| **3 samples** | 87.50% | 92.51% | 87.45% | 2.91% | 89.15% |
| **6 samples** | 90.95% | 94.56% | 90.97% | 2.08% | 92.16% |

Table 13. Computational time comparison for binary code method

| **Xiaomi Mi6** | | | |
|---|---|---|---|
| **Fold** | **Pre-processing** | **F. ex. + classification** | **Sum** |
| **Fold 1** | 15.0 ms | 14.6 s | 14.6 s |
| **Fold 2** | 14.9 ms | 14.5 s | 14.5 s |
| **Fold 3** | 14.8 ms | 14.4 s | 14.4 s |
| **Average** | 14.9 ms | 14.5 s | 14.5 s |
| **Huawei P10 Lite** | | | |
| **Fold** | **Pre-processing** | **F. ex. + classification** | **Sum** |
| **Fold 1** | 13.9 ms | 13.9 s | 13.9 s |
| **Fold 2** | 14.3 ms | 13.8 s | 13.8 s |
| **Fold 3** | 13.8 ms | 14.5 s | 14.5 s |
| **Average** | 14.0 ms | 13.8 s | 13.8 s |
| **Samsung Galaxy A5 2017** | | | |
| **Fold** | **Pre-processing** | **F. ex. + classification** | **Sum** |
| **Fold 1** | 13.4 ms | 13.8 s | 13.8 s |
| **Fold 2** | 11.8 ms | 13.7 s | 13.7 s |
| **Fold 3** | 10.6 ms | 13.7 s | 13.7 s |
| **Average** | 12.1 ms | 13.7 s | 13.7 s |
| **Samsung Galaxy S5** | | | |
| **Fold** | **Pre-processing** | **F. ex. + classification** | **Sum** |
| **Fold 1** | 15.1 ms | 15.3 s | 15.3 s |
| **Fold 2** | 12.8 ms | 15.5 s | 15.5 s |
| **Fold 3** | 12.8 ms | 15.6 s | 15.6 s |
| **Average** | 13.5 ms | 15.5 s | 15.5 s |

Table 14. Obtained accuracy results for both feature vectors (TEM method)

| **Fold** | **Short code (64 elements)** | **Long code (70 elements)** |
|---|---|---|
| **Fold 1** | 84.78% | 94.80% |
| **Fold 2** | 84.73% | 95.01% |
| **Fold 3** | 83.53% | 94.05% |
| **Average** | 84.35% | **94.62%** |

Table 15. Computational time comparison for TEM-based method

| **Xiaomi Mi6** | | | |
|---|---|---|---|
| **Fold** | **Pre-processing** | **F. ex. + classification** | **Sum** |
| **Fold 1** | 10.42 ms | 67.71 ms | 76.13 ms |
| **Fold 2** | 11.08 ms | 66.50 ms | 77.58 ms |
| **Fold 3** | 13.26 ms | 65.72 ms | 78.98 ms |
| **Average** | 11.59 ms | 65.98 ms | 77.57 ms |
| **Huawei P10 Lite** | | | |
| **Fold** | **Pre-processing** | **F. ex. + classification** | **Sum** |
| **Fold 1** | 12.17 ms | 93.20 ms | 105.37 ms |
| **Fold 2** | 13.02 ms | 95.11 ms | 108.13 ms |
| **Fold 3** | 12.87 ms | 93.58 ms | 106.45 ms |
| **Average** | 12.69 ms | 93.96 ms | 106.65 ms |
| **Samsung Galaxy A5 2017** | | | |
| **Fold** | **Pre-processing** | **F. ex. + classification** | **Sum** |
| **Fold 1** | 11.72 ms | 169.19 ms | 180.91 ms |
| **Fold 2** | 10.68 ms | 170.00 ms | 180.68 ms |
| **Fold 3** | 11.23 ms | 173.50 ms | 184.73 ms |
| **Average** | 11.21 ms | 170.90 ms | 182.11 ms |
| **Samsung Galaxy S5** | | | |
| **Fold** | **Pre-processing** | **F. ex. + classification** | **Sum** |
| **Fold 1** | 13.10 ms | 161.94 ms | 175.04 ms |
| **Fold 2** | 12.78 ms | 165.89 ms | 178.67 ms |
| **Fold 3** | 13.20 ms | 156.19 ms | 169.39 ms |
| **Average** | 13.03 ms | 161.34 ms | 174.37 ms |

Apart from the accuracy, the operation time was assessed. Table 15 presents the time of processing one single authentication. For each fold, the average pre-processing time and the average feature extraction with classification time were calculated.

# 6. Conclusions and future work

In this thesis 6 different methods of palmprint user identification were proposed. They were implemented on 4 mobile devices. During the experiments both time and accuracy of the methods were observed. In the accuracy domain the obtained results were in range $< 83\%, 98\% >$, which are results comparable to the state-of-the-art methods. In computational time almost each proposed method gave the result (single sample verification) in less than 0.2 s. One method, namely method based on binary code, was much slower and gave the result in almost 14 s. The detailed results of all proposed methods and the comparison to some state-of-the-art methods is presented in Table 16. When the information of computational time is missing in selected article, the 'no data' comment is placed in the table.

Nonetheless in palmprint biometrics there are still some emerging issues that can be treated as a plan for future work. First of all, the sample aging can be enumerated. Even though palmprint images are said to remain unchanged while the time is passing, some differences may appear, eg. caused by scars. Thus, the livelong learning techniques have gain more researchers attention recently.

Then, template protection has to be mentioned. Since the biometrics is said to be a special kind of personal data, the template of biometric sample needs to be protected very carefully. The most common approach is *cancelable biometrics*. This concept was proposed to alleviate the problem of impossibility to change the

Table 16. The comparison of the proposed methods with some state-of-the-art methods

| Name | Max. accuracy | Comp. time |
|---|---|---|
| Moco et al. [26] | FRR=9.27%, FAR = 0.03% | 466 ms |
| Kim et al. [14] | EER = 2.88% | 685 ms |
| Fang [27] | EER = 4.5% | no data |
| Ungureanu et al. [28] | 90% | no data |
| Tiwari et al. [14] | EER = 5.55% | 889.2 ms |
| Leng et al. [13] | EER more than 2% | no data |
| Zhang et al. [29] | 90% | 162 ms |
| HOG-based method | 97.1% | 33.1 ms |
| Hybrid CT method | 92.0% | 36.3 ms |
| Hybrid GT method | 83.0% | 32.7 ms |
| Method based on 3-value mask | **98.3%** | 86.1 ms |
| Method based on binary code | 92.2% | 13700 ms |
| Method based on TEM | 94.5% | 77.6 ms |

biometric trait after loosing some private data. Its main idea is to intentionally distort the image using various irreversible transformations in the signal domain or in the feature domain. An example of such an approach may be to divide the whole ROI image into non-overlapping blocks and mix the positions of the blocks. Even though the image can be stolen, no private information may be extracted from this kind of a distorted image.

All in all, there are still numerous issues to focus on in the future.

# Bibliography

[1] Abuarqoub, A.: D-FAP: Dual-Factor Authentication Protocol for Mobile Cloud Connected Devices. J. Sens. Actuator Netw. 9, 1 (2019).

[2] Ali, S.S., Ganapathi, I.I., Prakash, S., Consul, P., Mahyo, S.: Securing biometric user template using modified minutiae attributes. Pattern Recognit. Lett. 129, pp. 263–270 (2020).

[3] Unar, J.A., Seng, W.C., Abbasi, A.: A review of biometric technology along with trends and prospects. Pattern Recognit. 47, pp. 2673–2688 (2014).

[4] Jain, A.K., Nandakumar, K., Ross, A.: 50 years of biometric research: Accomplishments, challenges, and opportunities. Pattern Recognit. Lett. 79, pp. 80–105 (2016).

[5] Lancelot Miltgen, C., Popovič, A., Oliveira, T.: Determinants of end-user acceptance of biometrics: Integrating the "Big 3" of technology acceptance with privacy context. Decis. Support Syst. 56, pp. 103–114 (2013).

[6] El-Abed, M., Giot, R., Hemery, B., Rosenberger, C.: A study of users' acceptance and satisfaction of biometric systems. In: Security Technology (ICCST), 2010 IEEE International Carnahan Conference on. pp. 170–178. IEEE (2010).

[7] Tabejamaat, M., Mousavi, A.: Generalized Gabor filters for palmprint recognition. Pattern Anal. Appl. 21, pp. 261–275 (2018).

[8] Barra, S., De Marsico, M., Nappi, M., Narducci, F., Riccio, D.: A hand-based biometric system in visible light for mobile environments. Inf.

[9] Franzgrote, M., Borg, C., Ries, B.J.T., Bussemaker, S., Jiang, X., Fieleser, M., Zhang, L.: Palmprint verification on mobile phones using accelerated competitive code. In: Hand-Based Biometrics (ICHB), 2011 International Conference on. pp. 1–6. IEEE (2011).

[10] Yang, X., Pang, S., Cheng, K.T.: Mobile image search with multimodal context-aware queries. In: Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on. pp. 25–32. IEEE (2010).

[11] Fenu, G., Marras, M.: Controlling User Access to Cloud-Connected Mobile Applications by Means of Biometrics. IEEE Cloud Comput. 5, pp. 47–57 (2018).

[12] Talreja, V., Ferrett, T., Valenti, M.C., Ross, A.: Biometrics-as-a-service: A framework to promote innovative biometric recognition in the cloud. In: 2018 IEEE International Conference on Consumer Electronics (ICCE). pp. 1–6. IEEE (2018).

[13] Leng, L., Gao, F., Chen, Q., Kim, C.: Palmprint recognition system on mobile devices with double-line-single-point assistance. Pers. Ubiquitous Comput. 22, pp. 93–104 (2018).

[14] Kim, J.S., Li, G., Son, B., Kim, J.: An empirical study of palmprint recognition for mobile phones. IEEE Trans. Consum. Electron. 61, pp. 311–319 (2015).

[15] Tiwari, K., Hwang, C.J., Gupta, P.: A palmprint based recognition system for smartphone. In: 2016 Future Technologies Conference (FTC). pp. 577–586. IEEE, San Francisco, CA, USA (2016).

[16] Zhang, D., Kong, W.-K., You, J., Wong, M.: Online palmprint identification. IEEE Trans. Pattern Anal. Mach. Intell. 25, pp. 1041–1050 (2003).

[17] Imtiaz, H., Fattah, S.A.: A histogram-based dominant wavelet domain feature selection algorithm for palm-print recognition. Comput. Electr. Eng. 39, pp. 1114–1128 (2013).

[18] Ray, K.B., Misra, R.: Palm Print Recognition Using Hough Transforms. Presented at the 2015 International Conference on Computational Intelligence and Communication Networks (2015).

[19] El-Tarhouni, W., Boubchir, L., Al-Maadeed, N., Elbendak, M., Bouridane, A.: Multispectral palmprint recognition based on local binary pattern histogram fourier features and gabor filter. In: 2016 6th European Workshop on Visual Information Processing (EUVIP). pp. 1–6. IEEE, Marseille, France (2016).

[20] Mokni, R., Mezghani, A., Drira, H., Kherallah, M.: Multiset Canonical Correlation Analysis: Texture Feature Level Fusion of Multiple Descriptors for Intra-modal Palmprint Biometric Recognition. In: Paul, M., Hitoshi, C., and Huang, Q. (eds.) Image and Video Technology. pp. 3–16. Springer International Publishing, Cham (2018).

[21] Dubey, P., Kanumuri, T., Vyas, R.: Sequency codes for palmprint recognition. Signal Image Video Process. 12, pp. 677–684 (2018).

[22] Kumar, A.: Toward More Accurate Matching of Contactless Palmprint Images Under Less Constrained Environments. IEEE Trans. Inf. Forensics Secur. 14, pp. 34–47 (2019).

[23] Ahmadi, M., Soleimani, H.: Palmprint image registration using convolutional neural networks and Hough transform. ArXiv190400579 Cs. (2019).

[24] Matkowski, W.M., Chai, T., Kong, A.W.K.: Palmprint Recognition in Uncontrolled and Uncooperative Environment. IEEE Trans. Inf. Forensics Secur. 15, pp. 1601–1615 (2020).

[25] Choraś, M., Kozik, R.: Contactless palmprint and knuckle biometrics for mobile devices. Pattern Anal. Appl. 15, pp. 73–85 (2012).

[26] Moco, N.F., Lobato Correia, P.: Smartphone-based palmprint recognition system. Presented at the 21st International Conference on Telecommunications (ICT), Lisbon, Portugal (2014).

[27] Fang, L.: Mobile based palmprint recognition system. In: Control, Automation and Robotics (ICCAR), 2015 International Conference on. pp. 233–237. IEEE (2015).

[28] Ungureanu, A., Costache, C.: Palm Print as a Smartphone Biometric: Another option for digital privacy and security. IEEE Consum. Electron. Mag. 5, pp. 71–78 (2016).

[29] Zhang, Y., Zhang, L., Liu, X., Zhao, S., Shen, Y., Yang, Y.: Pay By Showing Your Palm: A Study of Palmprint Verification on Mobile Platforms. In: 2019 IEEE International Conference on Multimedia and Expo (ICME). pp. 862–867. IEEE, Shanghai, China (2019).

[30] Palmprint benchmark dataset CASIA, `http://biometrics.idealtest.org/index.jsp` [accessed: 06-03-2017].

[31] Palmprint benchmark dataset IITD, http://www4.comp.polyu.edu.hk/~csajaykr/database.php [accessed: 06-03-2017].

[32] Palmprint benchmark dataset PolyU, http://www4.comp.polyu.edu.hk/~biometrics/ [accessed: 06-03-2017].