

# **LBP Extensions used in Finger Vein Spoofing Detection**

Daniel Kocher

Stefan Schwarz

Andreas Uhl

Technical Report 2016-04

August 2016

**Department of Computer Sciences**

Jakob-Haringer-Straße 2  
5020 Salzburg  
Austria  
[www.cosy.sbg.ac.at](http://www.cosy.sbg.ac.at)

**Technical Report Series**

## LBP Extensions used in Finger Vein Spoofing Detection

Daniel Kocher<sup>1</sup> Stefan Schwarz<sup>2</sup> Andreas Uhl<sup>3</sup>

**Abstract:** Biometric systems based upon finger vein images have been shown to be vulnerable to presentation attacks. In this supplement, we detail a variety of methods extending local binary patterns (LBP) which can be used to distinguish between fake and real finger vein images in terms of description and implementation / experimentation parameters used.

**Keywords:** Finger Vein Spoofing, Spoofing Detection, Biometrics, Texture-based, Local descriptor, Local Binary Pattern (LBP), Local Radius Index (LRI), Local Derivative Pattern (LDP), Local Graph Structure (LGS), Symmetric Local Graph Structure (SLGS)

### 1 Introduction

Biometric traits have emerged to replace or at least complement the traditional authentication methods (e.g. passwords). One biometric trait enjoying more and more popularity are veins. One advantage of veins over other biometric traits is the fact that they are embedded *inside* the human body, as opposed to traits like fingerprints or faces. Moreover, vein images can be acquired in an unintrusive manner which is not the case for other biometric traits, such as iris acquisition. However, despite being resistant to tampering, vein-based authentication is vulnerable to presentation attacks [TVM14]. In this paper, we focus on finger veins (FVs) as biometric traits.

In general, counter-measures to presentation (or spoofing) attacks in biometrics can be categorised in (1) liveness-based, (2) motion-based and (3) texture-based methods. Liveness-based methods, e.g., [Ra15], use signs of vitality to ensure that the image is captured from a living human being. In contrast, motion-based methods utilise unnatural movements on scenes as indication of spoofing, e.g. caused by hand motion when presenting a photo or a display to the sensor. Texture-based methods aim to explore textural artifacts in the images captured by the sensor (e.g. caused by recapturing artifacts). Texture-based techniques have been proven to be applicable to the imagery in the FV-Spoofing-Attack database [To15] for evaluation, in particular baseline LBP [RB15].

In 2015, the first competition on counter-measures to finger vein spoofing attacks took place [To15]. The competition baseline algorithm looks at the frequency domain of vein images, exploiting the bandwidth of vertical energy signal on real finger vein images, which is different for fakes ones. Three teams participated in this competition. The first

---

<sup>1</sup> University of Salzburg, Department of Computer Sciences

<sup>2</sup> University of Salzburg, Department of Computer Sciences

<sup>3</sup> University of Salzburg, Department of Computer Sciences, Jakob Haringer Strasse 2, 5020 Salzburg, uhl@cosy.sbg.ac.at

team (GUC) uses binarised statistical images features (BSIF). They represent each pixel as a binary code. This code is obtained by computing the pixel's response to a filter that are learnt using statistical properties of natural images [To15]. The second team (B-Lab) uses monogenic scale space based global descriptors employing the Riesz transform. This is motivated by the fact that local object appearance & shape within an image can be represented as a distribution of local energy and local orientation information. The best approach (team GRIP-PRIAMUS) utilises local descriptors, i.e., local binary patterns (LBP), and local phase quantisation (LPQ) and Weber local descriptors (WLD). They distinguish between full and cropped images. LBPs and LPQ/WLD are used to classify full and cropped images, respectively.

However, counter-measures to finger vein spoofing attacks were/are already developed prior or independent to this competition. In 2013, the authors of [Ng13] introduced a fake finger vein image detection based upon Fourier, and Haar and Daubechies wavelet transforms. For each of these features, the score of spoofing detection was computed. To decide whether a given finger vein image is fake or real, an SVM was used to combine the three features.

The authors of [Ti15] propose windowed dynamic mode decomposition (W-DMD) to be used to identify spoofed finger vein images. DMD is a mathematical method to extract the relevant modes from empirical data generated by non-linear complex fluid flows. While DMD is classically used to analyse a set of image sequences, the W-DMD method extracts local variations as low rank representation inside a single still image. It is able to identify spoofed images by capturing light reflections, illuminations and planar effects.

A detection framework based on singular value decomposition (SVD) is proposed in a rather confused paper [MS15]. Finger vein images are classified based on image quality assessment (IQA) without giving any clear indication about the actual IQA and any experimental results.

Finally, [RB15] proposes a scheme using steerable pyramid is used to extract features. Steerable pyramids are a set of filters in which a filter of arbitrary orientation is synthesised as a linear combination of a set of basis functions. This enables the steerable pyramids scheme to compute the filter response at different orientations. This scheme shows consistent high performance for the finger vein spoofing detection problem and outperforms many other texture-classification-based techniques. It is compared to techniques from [To15], including two LBP variants, and to quality-based approaches computing block-wise entropy, sharpness, and standard deviation.

In a recent paper [KSU16], inspired by the success of basic LBP techniques [MS15, To15] in finger vein spoofing detection and the availability of a wide variety of LBP extensions and generalisations in literature, we have empirically evaluated different features obtained by using these more recent LBP-related feature extraction techniques for finger vein spoofing detection. Due to space restriction in that paper [KSU16], we were not able to provide references and detailed descriptions for the original LBP technique as well as its more recent variants. This is provided in this manuscript as supplementary material.

The remainder of this manuscript is organised as follows. The LBP features evaluated in [KSU16] are described in Section 2.

## 2 Local Binary Pattern Extensions

For our experiments ([KSU16]), the LBP-based features described in the following are applied to all pixels of the image except for those which have not enough neighboring pixels available. Pixels are traversed line-wise and for each feature, a histogram has been constructed.

### 2.1 Local Binary Pattern (BaseLBP)

The traditional *Local Binary Pattern* operator was originally introduced by [OPH94] in 1994. They proposed the operator as a nonparametric  $3 \times 3$  kernel. However, LBP can be parameterized in two ways, i.e. the number of neighboring pixels  $P$  and the radius  $R$  from the center pixel. The  $P$  neighboring pixels are distributed evenly spaced on the circle of radius  $R$  with respect to a given center pixel. Using these parameters, the  $3 \times 3$  kernel has  $P = 8$  neighbors distributed evenly spaced on a circle of radius  $R = 1$ .

An LBP is defined as an ordered set of binary values determined by comparing the values of the center pixel to the values of each neighboring pixels. Equation 1 transforms the ordered set of binary values to a decimal value, where  $i_c$  denotes the center pixel, located at position  $(x_c, y_c)$ , and  $i_n$  denotes the  $n^{\text{th}}$  neighboring pixel [MD11].

$$LBP_{R,P}(x_c, y_c) = \sum_{n=0}^{P-1} s(i_n - i_c) \cdot 2^n \quad (1)$$

The function  $s(x)$  is defined in Equation 2.

$$s(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{if } x < 0 \end{cases} \quad (2)$$

105	88	91	$\xrightarrow[\text{with the center}]{\text{Intensity comparison}}$	1	0	0
85	92	100		0		1
121	130	111		1	1	1

Fig. 1: Example: LBP with  $P = 8$  and  $R = 1$

The binary sequence of the example is defined clockwise from the top-left as 10011110. In our experiments, the parameters  $R$  and  $P$  were altered as follows:

$$(R, P) \in \{(1, 8), (2, 16), (3, 24)\} \quad (3)$$

## 2.2 Local Line Binary Pattern (LLBP)

The *Local Line Binary Pattern (LLBP)* operator was proposed by [PS09] for face recognition in 2009. The benefit of this pattern is that it can emphasize the change in image intensity such as vertices, edges and corners [RSS11]. The neighborhood shape is a straight line, instead of a circle shape. The operator consists of two components: a horizontal and a vertical component. Hence, the template resulting from this pattern has the shape of a cross. The magnitude of the LLBP is calculated by calculating the line binary code for both, the horizontal and the vertical component.

Equation 4 to 6 provide the mathematical definition of the LLBP, where  $LLBP_h$ ,  $LLBP_v$ ,  $LLBP_m$  denote the LLBP in horizontal direction, in vertical direction and its magnitude, respectively.  $h_n$  is the  $n^{\text{th}}$  pixel on the horizontal line,  $v_n$  is the  $n^{\text{th}}$  pixel on the vertical line and  $N$  is the length of the line. The center pixel is located at  $(x_c, y_c)$ , where  $c = \frac{N}{2}$ . Hence, the center pixel is the intersection pixel of the vertical and the horizontal line. The function  $s$  is defined as usual (see Section 2.1) [RSS11].

$$LLBP_{hN,c}(x_c, y_c) = \sum_{n=1}^{c-1} s(h_n - h_c) \cdot 2^{c-n-1} + \sum_{n=c+1}^N s(h_n - h_c) \cdot 2^{n-c-1} \quad (4)$$

$$LLBP_{vN,c}(x_c, y_c) = \sum_{n=1}^{c-1} s(v_n - v_c) \cdot 2^{c-n-1} + \sum_{n=c+1}^N s(v_n - v_c) \cdot 2^{n-c-1} \quad (5)$$

$$LLBP_m = \sqrt{LLBP_h^2 + LLBP_v^2} \quad (6)$$

Both, the horizontal as well as the vertical component, extract a binary code of  $N - 1$  bits for each pixel. Thus, by concatenating the two binary codes the total binary code of the *LLBP* for each pixel is obtained by  $2 \cdot (N - 1)$ . We altered  $N$  in our experiments:

$$N \in \{8, 16, 24\} \quad (7)$$

## 2.3 Completed Local Binary Pattern (CLBP)

Guo et al. [GZZ10] proposed a method to complete and generalize the LBP, namely the *completed LBP (CLBP)*. In this LBP variant, a region is represented by its center pixel and a so-called *local difference sign-magnitude transform (LDSMT)*, which decomposes the local structure into two components, i.e. a *difference sign* and a *difference magnitude* component, denoted CLBP\_S and CLBP\_M, respectively. In essence, CLBP\_S is equal to the standard LBP (using -1 instead of 0 to encode a negative difference). The center pixel component (CLBP\_C) is represented by thresholding the local gray level against the average gray level of the whole image. The binary values of these three components are then combined to generate a final CLBP histogram.

The steps to compute the three CLBP components, given a center pixel  $i_c$  and  $P$  circularly and evenly spaced neighbors  $i_n$  ( $n = 0, 1, \dots, P-1$ ), are rather simple, i.e.  $d_n = i_n - i_c$ , where  $d_n$  describes the local structure at a given center pixel  $i_c$ . The LDSMT then is performed by decomposing  $d_n$  as follows:

$$d_n = s_n * m_n \quad \begin{cases} s_n = \text{sign}(d_n) \\ m_n = |d_n| \end{cases} \quad \text{sign}(d_n) = \begin{cases} 1, & d_n \geq 0 \\ -1, & \text{otherwise} \end{cases} \quad (8)$$

The CLBP\_S operator is computed just as the standard LBP operator defined in Section 2.1. Let  $c_i$  denote an adaptive threshold. Furthermore, let  $c_{mean}$  be the mean value of  $m_n$  and  $c_{avg}$  be the average gray level, both with respect to the whole image. Then the remaining two CLBP operators are formally defined by

$$\text{CLBP\_M}_{P,R} = \sum_{n=0}^{P-1} t(m_n, c_{mean}) \cdot 2^n \quad \text{CLBP\_C}_{P,R} = t(i_c, c_{avg}) \quad t(x, c) = \begin{cases} 1, & x \geq c \\ 0, & x < c \end{cases} \quad (9)$$

The histogram is then created by a simple concatenation of the histograms of the three CLBP components. For the evaluation,  $R$  and  $P$  were chosen as described in Section 2.1 (Equation 3).

#### 2.4 Median-Robust Extended Local Binary Pattern (MRELBP)

A robust LBP variant was proposed by [Li15]. The *Median Robust Extended LBP (MRELBP)* increases the tolerance to image blur and noise corruption. It is based upon the Extended LBP, which provides four descriptors. The MRELBP is referred to as a *very simple, high-quality, yet efficient multiresolution approach* [Li15]. This approach has some favorable properties, i.e. gray-scale and rotation invariance as well as strong discriminativeness and noise robustness. Moreover it can be implemented efficiently.

To increase the robustness, the MRELBP descriptor uses median values instead of single pixel intensities. Before the three components of the MRELBP descriptor, namely MRELBP\_CI, MRELBP\_NI and MRELBP\_RD, are defined, some notations are introduced.  $\mathbf{X}_{c,\omega}$  denotes a patch of size  $\omega \times \omega$  centered at a pixel  $i_c$  and  $\mathbf{X}_{r,p,\omega_r,n}$  denotes a patch of size  $\omega_r \times \omega_r$  centered at a neighboring pixel  $i_{r,p,n}$ . The neighboring pixels are circularly and evenly spaced neighbors of a center pixel  $i_c$  on radius  $r$  and  $i_{r,p,n}$  denotes the  $n^{\text{th}}$  of the  $p$  neighbors on radius  $r$ . The function  $\phi(\mathbf{X}_i)$  denotes the median value of the pixels of patch  $\mathbf{X}_i$ . The mean of  $\phi(\mathbf{X}_{c,\omega})$  over the whole image is denoted by  $\mu_\omega$ . Based on these notations, the MRELBP is formally defined as follows:

$$\text{MRELBP\_CI}(i_c) = s(\phi(\mathbf{X}_{c,\omega}) - \mu_\omega) \quad (10)$$

$$\text{MRELBP\_NI}_{r,p}(i_c) = \sum_{n=0}^{p-1} s(\phi(\mathbf{X}_{r,p,\omega_r,n}) - \mu_{r,p,\omega_r}) \cdot 2^n \quad \mu_{r,p,\omega_r} = \frac{1}{p} \cdot \sum_{n=0}^{p-1} \phi(\mathbf{X}_{r,p,\omega_r,n}) \quad (11)$$

$$\text{MRELBP\_RD}_{r,r-1,p,\omega_r,\omega_{r-1}} = \sum_{n=0}^{p-1} s(\phi(\mathbf{X}_{r,p,\omega_r,n}) - \phi(\mathbf{X}_{r-1,p,\omega_{r-1},n})) \cdot 2^n \quad (12)$$

Again, the respective histograms of these three components are concatenated to form a single histogram representing the whole image. For the experiments, we only considered MRELBP with two radii, i.e.  $r$  and  $r-1$  in Equation 12 (with the corresponding  $(\omega_r \times \omega_r)$ - and  $(\omega_{r-1} \times \omega_{r-1})$ -patches, respectively). For the constantly chosen parameters  $\omega = 3$ ,  $\omega_{r-1} = 3$  and  $\omega_r = 5$ , the parameters  $r$ ,  $r-1$  and  $p$  were then altered as follows:

$$(r-1, p, r) \in \{(1, 8, 4), (2, 16, 6), (3, 24, 8)\} \quad (13)$$

## 2.5 Local Derivative Pattern (LDP)

The *Local Derivative Pattern (LDP)* was proposed by [Zh10] in 2010. LBP encodes the binary result of the first-order derivative among local neighbors [Zh10], while LDP is a high-order local pattern which contains more detailed discriminative features.

The first-order derivative is denoted by  $I_\alpha^1(Z)$ , where  $\alpha = 0^\circ, 45^\circ, 90^\circ$  or  $135^\circ$  and  $I(Z)$  denotes the image. Let  $Z_0$  be the center pixel and  $Z_i, i = 1, \dots, 8$  be the neighboring pixels around  $Z_0$ . Equation 14 computes the four first-order derivatives at  $Z_0$ . Figure 2 shows an example of the neighborhood for pixel  $Z_0$  [Zh10].

$$\begin{aligned} I_{0^\circ}^1 &= I(Z_0) - I(Z_4) & I_{45^\circ}^1 &= I(Z_0) - I(Z_3) \\ I_{90^\circ}^1 &= I(Z_0) - I(Z_2) & I_{135^\circ}^1 &= I(Z_0) - I(Z_1) \end{aligned} \quad (14)$$

$Z_1$	$Z_2$	$Z_3$
$Z_8$	$Z_0$	$Z_4$
$Z_7$	$Z_6$	$Z_5$

Fig. 2: Neighborhood of  $Z_0$  (LDP)

The  $n^{\text{th}}$ -order LDP describes gradient trend changes in a local region of directional  $(n-1)^{\text{th}}$  order derivative images. Equation 15 shows the computation of the  $n^{\text{th}}$  order for a direction  $\alpha$ , where  $I_\alpha^{n-1}(V)$  denotes the  $(n-1)^{\text{th}}$  order derivative in direction  $\alpha$  at  $V_0$ . The function  $s$  is defined in Equation 16.

$$LDP_\alpha^n(V_0) = \{s(I_\alpha^{n-1}(V_0), I_\alpha^{n-1}(V_1)), s(I_\alpha^{n-1}(V_0), I_\alpha^{n-1}(V_2)), \dots, s(I_\alpha^{n-1}(V_0), I_\alpha^{n-1}(V_8))\} \quad (15)$$

$$s(I_\alpha^{n-1}(V_0), I_\alpha^{n-1}(V_i)) = \begin{cases} 0 & \text{if } I_\alpha^{n-1}(V_0) \cdot I_\alpha^{n-1}(V_i) > 0 \\ 1 & \text{if } I_\alpha^{n-1}(V_0) \cdot I_\alpha^{n-1}(V_i) \leq 0 \end{cases} \quad (16)$$

where  $i = 1, \dots, 8$  are the indices of the neighbors

The LDP of order  $n$  ( $LDP^n(V)$ ) is obtained by concatenating the bit-strings of each direction (see Equation 17).

$$LDP^n(V) = \{LDP_\alpha^n(V) | \alpha = 0^\circ, 45^\circ, 90^\circ, 135^\circ\} \quad (17)$$

In the experiments, we evaluated the LDP of order  $n = 2$  with radius  $R = 1$ . Moreover, we evaluated two parameters / versions of the LDP: (1) the four 8-Bit integers (one for

each direction) were concatenated and the resulting 32-Bit integer was inserted into the histogram, and (2) each of the four 8-Bit integers was inserted into the histogram separately (no concatenation at all).

## 2.6 Local Radius Index (LRI)

[ZNP13] proposed a different type of statistical texture image feature, i.e. the *Local Radius Index (LRI)*, which is computationally simple compared to other metrics. It is based upon the fact that textures typically contain repetitive smooth regions and transitions between these regions. Such regions are identified by the edges around them. The authors introduce an inter-edge distance, which is the distance between two adjacent edges. The distribution of the inter-edge distance is then used to characterize the texture of an image. Another LRI operator is designed to capture the distributions of distances-to-nearest-edges. As a consequence, the authors propose two LRI operators, each of which results in eight integer directional indices for a given pixel. Each index represents a direction (in steps of  $45^\circ$ ). This results in a total number of eight histograms per LRI operator.

Both LRI operators depend on a threshold  $T$  and a size limit  $K$ . The directional index  $I_{i,d}$  for both LRI operators, i.e. LRI-A and LRI-D, can then be computed, for each pixel  $i$  with respect to the eight directions  $d = 1, \dots, 8$ , as follows.

### LRI-A

1.  $I_{i,d} = 0$  if the absolute difference between the current & the adjacent pixel in direction  $d$  is less than  $T$ .
2.  $I_{i,d} = \min(j, K)$  if  $j > 0$  successive pixels in direction  $d$  are greater than the current one by at least  $T$ , and the  $(j+1)^{\text{th}}$  is not.
3.  $I_{i,d} = \max(-j, -K)$  if  $j > 0$  successive pixels in direction  $d$  are smaller than the current one by at least  $T$ , and the  $(j+1)^{\text{th}}$  is not.

### LRI-D

1.  $I_{i,d} = \min(j, K) \bmod K$ , if for some  $j \geq 1$ ,  $j-1$  successive pixels in direction  $d$  have absolute difference with the current one less than  $T$ , and the  $j^{\text{th}}$  is greater than the current pixel by at least  $T$ .
2.  $I_{i,d} = -(\min(j, K) \bmod K)$ , if for some  $j \geq 1$ ,  $j-1$  successive pixels in direction  $d$  have absolute difference with the current one less than  $T$ , and the  $j^{\text{th}}$  is smaller than the current pixel by at least  $T$ .

For both LRI operators,  $T$  thresholds the value of what is considered an edge and  $K$  represents the maximum size of texture elements captured by the operators. In essence, LRI-A represents the size information of texture elements and LRI-D measures the distance to adjacent texture elements.

We chose  $K = 4$  to be constant for our experiments while altering  $T \in \{4, 8, 16\}$ .

## 2.7 Local Graph Structure (LGS)

Another approach, based on *Local Graph Structure (LGS)*, is proposed by [AB11]. The idea of this method is to represent each pixel of an image with a graph structure. This graph structure then captures the spatial information with respect to the neighboring pixels. It has been successfully utilized in face recognition. Moreover, there exists a refined version of it, called *Symmetric Local Graph Structure (SLGS)* [Ab14]. Since the two versions differ only in the graph structure in use, we will describe them simultaneously.

Both versions of the LGS result in an eight bit integer for a given pixel and they are applied to all pixels of the image. A given pixel is the starting point of the graph structure. The graph structure of both, LGS and SLGS, can be seen in Fig. 3. The gray dot represents the starting pixel and the arrows show how the neighboring pixels are traversed. All neighboring pixels are thresholded against the source pixel based upon the traversal of the graph structure. An edge is labeled 1 if the source pixel is smaller or equal to the target pixel. In contrast, an edge is labeled 0 if the source pixel is larger than the target pixel. The eight bit value is then obtained by reading the edge labels accordingly (counterclockwise).

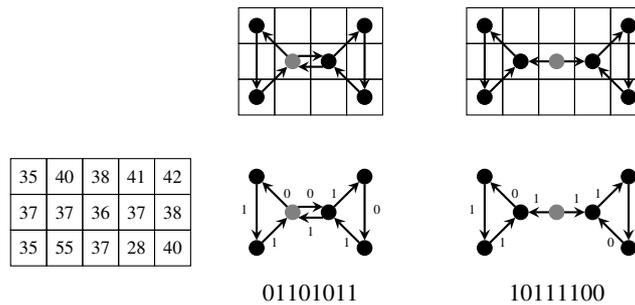


Fig. 3: Graph structure of LGS (top left) and SLGS (top right), and an example (below).

For the (S)LGS experiments, we did not alter any parameters since this would essentially result in a different graph structure.

## 3 Conclusion

In this manuscript we have described in detail various LBP-extension features which have been successfully used in [KSU16] for detecting spoofed finger vein images.

## References

- [AB11] Abusham, Eimad E. A.; Bashir, Housam K.: Face Recognition Using Local Graph Structure (LGS). In (Jacko, Julie A., ed.): Human-Computer Interaction. Interaction Techniques and Environments: 14th International Conference, HCI International 2011, Orlando, FL, USA, July 9-14, 2011, Proceedings, Part II. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 169–175, 2011.
- [Ab14] Abdullah, Mohd Fikri Azli; Sayeed, Md Shohel; Muthu, Kalaiarasi Sonai; Bashier, Housam Khalifa; Azman, Afizan; Ibrahim, Siti Zainab: Face recognition with Symmetric Local Graph Structure (SLGS). *Expert Systems with Applications*, 41(14):6131 – 6137, 2014.

- [GZZ10] Guo, Z.; Zhang, L.; Zhang, D.: A Completed Modeling of Local Binary Pattern Operator for Texture Classification. *IEEE Transactions on Image Processing*, 19(6):1657–1663, June 2010.
- [KSU16] Kocher, Daniel; Schwarz, Stefan; Uhl, Andreas: Empirical Evaluation of LBP-Extension Features for Finger Vein Spoofing Detection. In: *Proceedings of the International Conference of the Biometrics Special Interest Group (BIOSIG'16)*. Darmstadt, Germany, p. 8, 2016.
- [Li15] Liu, Li; Fieguth, P.; Pietikainen, M.; Lao, Songyang: Median Robust Extended Local Binary Pattern for Texture Classification. In: *Image Processing (ICIP), 2015 IEEE International Conference on*. pp. 2319–2323, Sept 2015.
- [MD11] Mirmohamadsadeghi, L.; Drygajlo, A.: Palm Vein Recognition with Local Binary Patterns and Local Derivative Patterns. In: *Biometrics (IJCB), 2011 International Joint Conference on*. pp. 1–6, Oct 2011.
- [MS15] Mythily, B.; Sathyaseelan, K.: Measuring the Quality of Image for Fake Biometric Detection: Application to Finger Vein. In: *National Conference on Research Advances in Communication, Computation, Electrical Science and Structures (NCRACCESS)*. pp. 6–11, 2015.
- [Ng13] Nguyen, Dat Tien; Park, Young Ho; Shin, Kwang Yong; Kwon, Seung Yong; Lee, Hyeon Chang; Park, Kang Ryoung: Fake finger-vein image detection based on Fourier and wavelet transforms. *Digital Signal Processing*, 23(5):1401–1413, 2013.
- [OPH94] Ojala, T.; Pietikainen, M.; Harwood, D.: Performance evaluation of texture measures with classification based on Kullback discrimination of distributions. In: *Pattern Recognition, 1994. Vol. 1 - Conference A: Computer Vision and Image Processing, Proceedings of the 12th IAPR International Conference on*. volume 1, pp. 582–585 vol.1, Oct 1994.
- [PS09] Petpon, Amnat; Srisuk, Sanun: Face Recognition with Local Line Binary Pattern. *Image and Graphics, International Conference on*, 0:533–539, 2009.
- [Ra15] Raghavendra, R.; Avinash, M.; Marcel, S.; Busch, C.: Finger vein liveness detection using motion magnification. In: *Biometrics Theory, Applications and Systems (BTAS), 2015 IEEE 7th International Conference on*. pp. 1–7, Sept 2015.
- [RB15] Raghavendra, R.; Busch, C.: Presentation Attack Detection Algorithms for Finger Vein Biometrics: A Comprehensive Study. In: *2015 11th International Conference on Signal-Image Technology Internet-Based Systems (SITIS)*. pp. 628–632, Nov 2015.
- [RSS11] Rosdi, Bakhtiar Affendi; Shing, Chai Wuh; Suandi, Shahrel Azmin: Finger Vein Recognition Using Local Line Binary Pattern. *Sensors*, 11(12):11357, 2011.
- [Ti15] Tirunagari, S.; Poh, N.; Bober, M.; Windridge, D.: Windowed DMD as a microtexture descriptor for finger vein counter-spoofing in biometrics. In: *Information Forensics and Security (WIFS), 2015 IEEE International Workshop on*. pp. 1–6, Nov 2015.
- [To15] Tome, P.; Raghavendra, R.; Busch, C.; Tirunagari, S.; Poh, N.; Shekar, B. H.; Groganiello, D.; Sansone, C.; Verdoliva, L.; Marcel, S.: The 1st Competition on Counter Measures to Finger Vein Spoofing Attacks. In: *Biometrics (ICB), 2015 International Conference on*. pp. 513–518, May 2015.
- [TVM14] Tome, P.; Vanoni, M.; Marcel, S.: On the Vulnerability of Finger Vein Recognition to Spoofing. In: *Biometrics Special Interest Group (BIOSIG), 2014 International Conference of the*. pp. 1–10, Sept 2014.

- [Zh10] Zhang, Baochang; Gao, Yongsheng; Zhao, Sanqiang; Liu, Jianzhuang: Local Derivative Pattern Versus Local Binary Pattern: Face Recognition With High-Order Local Pattern Descriptor. *IEEE Transactions on Image Processing*, 19(2):533–544, Feb 2010.
- [ZNP13] Zhai, Y.; Neuhoff, D. L.; Pappas, T. N.: Local Radius Index - A New Texture Similarity Feature. In: *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. pp. 1434–1438, May 2013.