Cognitive Computation (2019) 11:50–70
https://doi.org/10.1007/s12559-018-9593-6

A Line Feature Extraction Method for Finger-Knuckle-Print Verification

Jooyoung Kim[1] · Kangrok Oh[1] · Beom-Seok Oh[2] · Zhiping Lin[2] · Kar-Ann Toh[1]

Received: 27 October 2017 / Accepted: 4 September 2018 / Published online: 18 September 2018

Abstract

Due to its mobility and reliability, the outer finger-knuckle-print (FKP) possesses several advantages over other biometric traits of the hand. However, most existing state-of-the-art methods utilize either local features alone or together with global features for FKP verification. These methods often demand high computational cost despite their high verification accuracy. In this paper, we propose a novel and fast matrix projection method for extracting line features from the finger-knuckle-print for person verification. Essentially, both the horizontal and the vertical knuckle lines are extracted by projecting the knuckle print image onto a shift-and-difference matrix. Such a matrix enables directional image shifting and subtraction within a single matrix multiplication. The resultant difference image then goes through a sigmoidal activation for contrast enhancement. Subsequently, the Fourier spectrum of the contrast enhanced image is adopted as the holistic features of the given finger-knuckle-print image. The entire process of extracting the proposed features is expressed in an analytic form to facilitate a fast vectorized implementation. For cognition performance enhancement, the two directional line features are subsequently fused at the score level by minimizing the error counts of the extreme learning machine kernel. Extensive experiments are performed to compare the proposed method with competing methods using three public finger-knuckle-print databases. Our experimental results show encouraging performance in terms of verification accuracy and computational efficiency.

Keywords Finger-knuckle-print verification · Holistic line features · Extreme learning machine · Match scores fusion · Total error rate minimization

Introduction

Among the existing biometric traits such as fingerprint [1, 2], face [3, 4], iris, palmprint, and voice, considerable interest has been paid to hand-based biometrics due to their high user acceptance [5]. The palm print [6–9], hand geometry [10–13], palm texture [14], finger vein [15], and hand vein [16] have been relatively well studied. Recently, the inner and the outer knuckle surfaces [17, 18] have been proposed and investigated as a novel hand-based trait. Due to its mobility and reliability, in this work, we focus on the outer finger-knuckle-print (FKP), the skin pattern of the outer surface around the phalangeal joint of each finger. Essentially, those directional skin patterns around the phalangeal joint, which are nearly perpendicular to each finger, are our target line features.

Different from other hand-based biometrics, the outer finger-knuckle-print is at the dorsal side of the hand. Due to

✉ Kar-Ann Toh
katoh@yonsei.ac.kr

Jooyoung Kim
harrykim@yonsei.ac.kr

Kangrok Oh
kangrokoh@yonsei.ac.kr

Beom-Seok Oh
bsoh@ntu.edu.sg

Zhiping Lin
EZPLin@ntu.edu.sg

[1] School of Electrical and Electronic Engineering, Yonsei University, 50 Yonsei-ro Seodaemun-gu,
Seoul, 03722, Republic of Korea

[2] School of Electrical and Electronic Engineering, Nanyang Technological University, 50 Nanyang Avenue,
Singapore, 639798, Singapore

Table 1 Summary of existing works on finger-knuckle-print biometric

Category & index	Features	Information type	Database (#class[a] / #imgs.)	Best reported performance %	CPU time
Category I[b]					
Kumar09b [29]	KnuckleCode (Radon transform)	Local	IITD (158/790)	1.08 EER[c]	–
Zhang09a [22]	CompCode	Local	PolyU (120×4/5760)	1.09 EER	60.3ms[d]
Zhang10 [19]	ImCompCode & MagCode	Local	PolyU (165×4/7920)	1.48 EER	106.6ms[d]
Zhang11 [23]	Local-global information combination (LGIC)	Local+global	PolyU (165×4/7920)	0.402 EER	63.8ms[d]
Zhang12 [30]	Local feature integration (LFI)	Local	PolyU (165×4/7920)	1.27 EER	–
Zhang12 [30]	LGIC+LFI (denoted as LGIC_2 in [30])	Local+global	PolyU (165×4/7920)	0.358 EER	408ms[d]
Gao14 [31]	MoriCode&MtexCode	Local	PolyU (165×4/7920)	1.048 EER	249ms[e]
Nigam16 [24]	ITC (vcode&hcode)	Local	PolyU (165×4/7920)	0.892 EER	–
Jaswal17 [25]	DeepMatching (vcode&hcode)	Local	PolyU (165×4/7920)	0.98 EER	–
Category II[b]					
Woodard05 [18]	Curvature & Shape index	Global	Inhouse (177/1191)	94.5-98 CRR[c]	–
Kumar09a [32]	(PCA+ICA)*LDA[c] with finger geometry	Global	IITD (158/790)	1.39 EER	–
Tarhouni14 [27]	Multi-Scale LBP[c] & PCA	Local+global	PolyU (165×4/7920)	94.80 Acc	–
Category III[b]					
Zhang09b [33]	Band limited phase-only correlation (BLPOC)	Global	PolyU (165×4/7920)	1.68 EER	2.1ms[d]
Morales11 [21]	Orientation enhanced SIFT[c]	Local	PolyU (165×4/7920)	0.85 EER	–
Cheng12 [34]	Log-Gabor filter	Local	Mobile (109/561)	9 EER	–
Aoyama14 [28]	Phase-based local block matching	Local+global	PolyU (165×4/7920)	0.321 EER	–

[a] Class: subject (or) subject × number of fingers per subject

[b] Category I: coding-based approach, category II: subspace-based approach, category III: texture-based

[c] Abbreviations: equal error rate (EER), correct recognition rate (CRR), principal component analysis (PCA), independent component analysis (ICA), linear discriminant analysis (LDA), local binary pattern (LBP), scale invariant feature transform (SIFT)

[d] Taken from [19, 23, 30]. Measured using the Visual C#.Net 2005 on a PC with 2.33 GHz CPU

[e] Taken from [31]. Measured using the Visual C#.Net 2010 on a PC with 2.8 GHz CPU and 4 GBytes RAM

this location, it is less likely to be abraded [19]. Moreover, unlike other palm region biometrics such as the fingerprint and palm print, it is less likely to inadvertently leave its traits [20]. Also, the finger-knuckle-print can be readily acquired by low-resolution contactless camera [19]. Based on these characteristics, it can easily be applied to ATM machine or signature pad to assist user authentication. Such application can be conveniently realized because the finger-knuckle-print can be captured while the user is entering the password or holding a digital pen.

Background

Table 1 provides a brief summary of existing works on finger-knuckle-print biometric. According to Jaswal et al. [20], the existing finger-knuckle-print recognition algorithms can be categorized into three groups, namely, the coding-based approach, the subspace-based approach, and the texture-based approach[1].

The coding-based approach utilizes local features such as the blockwise orientation or the edge information for feature matching. These features are usually computed based on the local patch which encodes the detailed traits within a specific area. It is known that techniques belonging to this approach (see category I of Table 1) are less sensitive to illumination changes, but vulnerable to image rotation [20]. Methods belonging to the subspace-based approach project a raw image onto a low-dimensional subspace by learning the training data. The resulted subspace projected coefficients are then used as the feature vector for classification. Algorithms in this category mainly utilize the global information because the subspace features are extracted from the entire image where the holistic characteristic of the image is acquired. Finally, the texture-based approach include various texture analysis methods such as those using transform or local descriptors. For example, the transform-based texture analysis (see category III in Table 1) firstly converts the input image into the frequency domain using the Fourier transform or the discrete cosine transform. Then, the conventional texture features (e.g., the local binary patterns (LBP)) are extracted from the frequency domain. Another example is the local descriptor-based texture analysis which extracts local texture features using the scale invariant feature transform (SIFT) [21].

Among the three categories, the coding-based methods have received the most attention [20]. Zhang et al. [22] established among the largest finger-knuckle-print database and achieved promising verification performance using various coding methods. Subsequently, in 2011, Zhang et al. [23] improved the verification performance by combining both the local and global features. Recently, Nigam et al. [24] proposed the vcode and hcode schemes with incorrectly tracked corner (ITC) matching which can be considered as belonging to the coding-based methods. The vcode (hcode) stands for the vertical code (horizontal code) which consists of the vertical (horizontal) line features obtained by taking the image derivative along the x-axis (y-axis). Both the vcode and the hcode are then fed into the gradient ordinal relation pattern (GORP) or the star GORP (SGORP) for feature extraction. By combining the vcode and hcode with the two types of gradient ordinal relation patterns, four features, namely vcode$^{\mathrm{GORP}}$, hcode$^{\mathrm{GORP}}$, vcode$^{\mathrm{SGORP}}$, and hcode$^{\mathrm{SGORP}}$ (we shall call these four types of features vcode&hcode hereafter for convenience), are extracted. We shall abbreviate this method as ITC (vcode&hcode) hereafter for convenience. Following this approach, Jaswal et al. [25] proposed to adopt the DeepMatching [26] for matching these features. This method can also be considered as belonging to the coding-based methods.

The subspace-based methods are relatively less explored because the subspace projected coefficients do not possess high discrimination power for finger-knuckle-print verification [20] apparently. Tarhouni et al. [27] recently applied the multi-scale local binary pattern (LBP) for performance enhancement. This attempt can be considered as belonging to a combination of the local and global features. Among the texture-based methods, Zhang et al. [22] proposed a global texture method called band-limited phase only correlation (BLPOC) to determine the degree of similarity between the images. Aoyama et al. [28] modified the BLPOC into a local feature extraction method and achieved an improvedverification performance.

Motivation and Contributions

It is observed from Table 1 that most of the state-of-the-art methods extract and utilize the local features either alone or together with the global feature (denoted as "local+global" in the table) for FKP verification. This is mainly due to the highly discriminative information contained in the local features [23]. For example, as shown in the fifth column of Table 1, the local feature-based methods [21, 24, 25, 30] outperform the global feature-based methods [18, 32, 33] in terms of the best equal error rate (EER) performance. The local+global methods [23, 28, 30] show an even better verification performance (i.e., lower EER) than that of the local feature-based methods.

However, extracting the local knuckle features is often more computationally demanding than that of the global features [28]. According to Tables 3 and 4 of [28] (see the last column of Table 1), for example, the BLPOC which is the representative global method, is about 27–30 times faster than CompCode [22] and LGIC [23], and about 50 times faster than MoriCode&MagCode [31]. The LGIC$_2$ [30], which is among the best performing local+global methods, is about 194 times slower than the BLPOC. The information fusion (e.g., match scores fusion) imposes an additional computational cost on top of the time elapsed for extracting each local and global features. Such a high

[1] Although the authors of [20] have named a fourth group, namely, the "other image processing approach," no reference can be found to fall under such a group. Moreover, we found that the first three groups are enough to cover all the existing works in the literature. Based on these observations, the fourth group is excluded in our categorization.

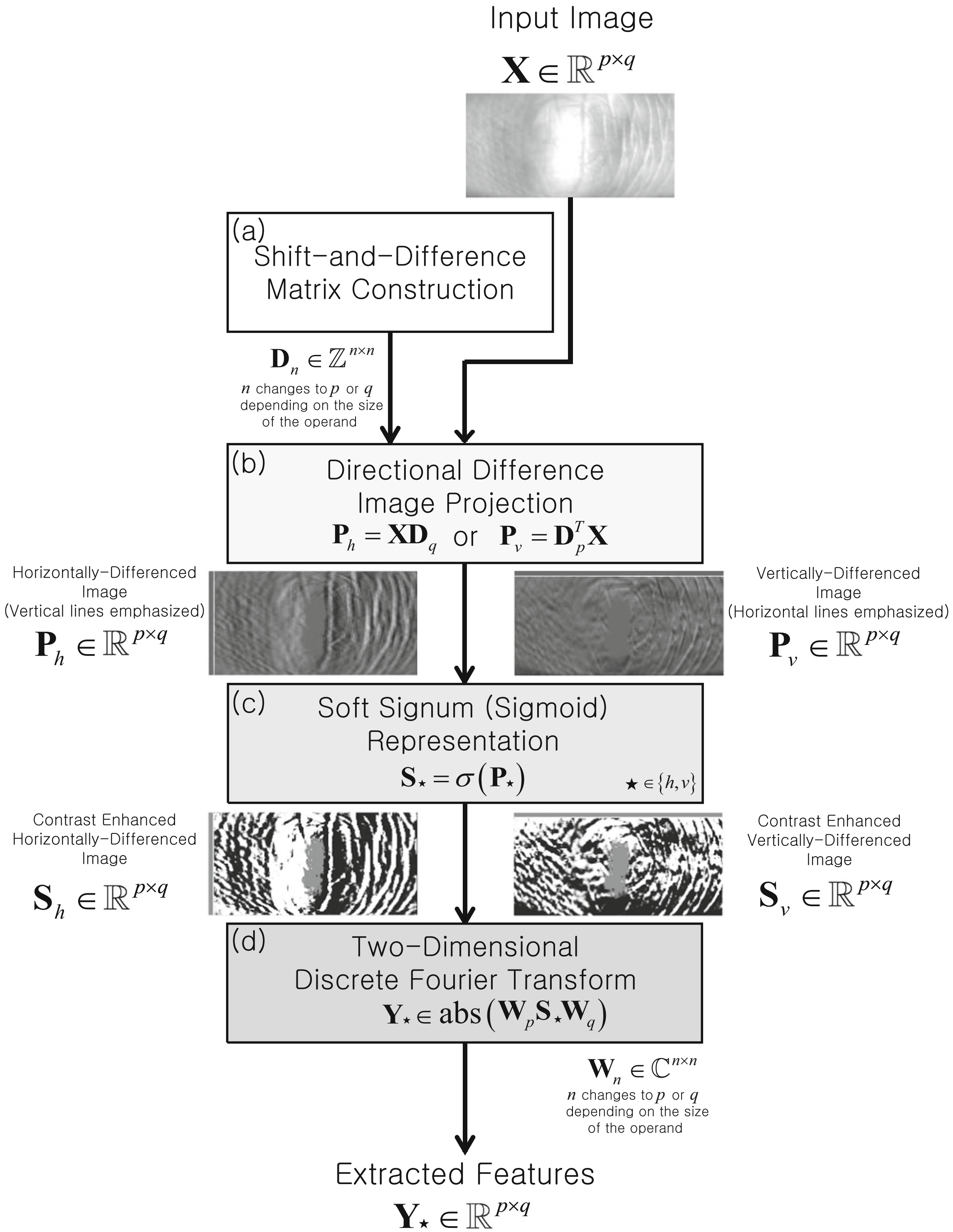

Input Image
$\mathbf{X} \in \mathbb{R}^{p \times q}$
(a) Shift–and–Difference Matrix Construction
$\mathbf{D}_n \in \mathbb{Z}^{n \times n}$
n changes to p or q depending on the size of the operand
(b) Directional Difference Image Projection
$\mathbf{P}_h = \mathbf{X}\mathbf{D}_q$ or $\mathbf{P}_v = \mathbf{D}_p^T\mathbf{X}$
Horizontally–Differenced Image (Vertical lines emphasized)
$\mathbf{P}_h \in \mathbb{R}^{p \times q}$
Vertically–Differenced Image (Horizontal lines emphasized)
$\mathbf{P}_v \in \mathbb{R}^{p \times q}$
(c) Soft Signum (Sigmoid) Representation
$\mathbf{S}_\star = \sigma(\mathbf{P}_\star)$
$\star \in \{h, v\}$
Contrast Enhanced Horizontally–Differenced Image
$\mathbf{S}_h \in \mathbb{R}^{p \times q}$
Contrast Enhanced Vertically–Differenced Image
$\mathbf{S}_v \in \mathbb{R}^{p \times q}$
(d) Two–Dimensional Discrete Fourier Transform
$\mathbf{Y}_\star \in \mathrm{abs}(\mathbf{W}_p \mathbf{S}_\star \mathbf{W}_q)$
$\mathbf{W}_n \in \mathbb{C}^{n \times n}$
n changes to p or q depending on the size of the operand
Extracted Features
$\mathbf{Y}_\star \in \mathbb{R}^{p \times q}$

Fig. 1 A graphical illustration of the proposed line feature projection (LFP) process

computational cost hiders the local+global methods to be deployed in real-time applications despite its outstanding verification capability. These observations motivate us to investigate into an effective and yet computationally efficient local+global feature extraction method for FKP verification.

In this work, we propose an analytic projection called the line feature projection (LFP) method. This method extracts the knuckle line features analytically by projecting the input image onto two predefined matrices for identity verification. The extracted features consist of a combination of the local and the global information. Particularly, each of the horizontal and the vertical knuckle lines is obtained within local patches by a single image shift-and-difference projection using the predefined matrix. After a contrast enhancement based on the sigmoidal activation [35] in which the difference values of range [−255, 255] are nonlinearly mapped to the range within [0, 1], the knuckle line patterns are transformed to a holistic set of line features by a two-dimensional discrete Fourier transform [36]. By making all these steps into a single projection form, we achieved computational efficiency as well as the desired verification accuracy performance. The extracted vertical and horizontal features are subsequently fused at match score level for performance enhancement. The total error rate (TER) minimization [37] which adopts the extreme learning machine (ELM) kernel [38–40] is utilized for the match scores fusion.

The main contributions of this work can be enumerated as follows:

- A system to utilize the finger-knuckle-print as the biometric trait for identity verification is proposed. Due to its mobility and reliability, the outer finger-knuckle-print possesses several advantages over other biometric traits of the hand. These advantages are attributed to its easily being captured by a low-resolution camera, less likely to be abraded and inadvertently leaving its traits.
- A projection matrix which extract the directional edges of an image is proposed. In contrast to those computationally intensive filtering-based methods which utilized image convolution, this projection matrix allows simple matrix multiplication with the target image for feature extraction.
- A fusion scheme to combine the two directional features is proposed. Different from the conventional regression distance error, the scheme minimizes the total classification error count.
- When the DeepMatching is incorporated with our extracted features, a competing verification accuracy with state-of-the-art deep learning method can be achieved.
- An extensive empirical evaluation of the proposed system with state-of-the-arts is provided. This would be useful for future benchmarking.

Organization

The paper is organized as follows: the background knowledge on the conventional circulant matrix and TER that adopts the ELM kernel is provided in "Preliminaries" for immediate reference. Section "Proposed Method And System" describes the proposed method which extracts horizontal and vertical knuckle line features. In the "Experiments" section, the experimental results, observations and discussions are provided. This is followed by some concluding remarks in the "Conclusion" section.

Preliminaries

In this section, we provide a brief description of several related techniques for immediate reference. These techniques include the circulant matrix [41] and the total error rate minimization classifier [37] that adopts the extreme learning machine kernel [38].

Circulant Matrix

According to [41], the k-vector circulant matrix $\mathbf{G} \in \mathbb{N}^{q \times q}$ is defined as follows:

$$\mathbf{G} = \begin{bmatrix} \mathbf{0}_{(q-k)\times k} & \mathbf{I}_{(q-k)\times(q-k)} \\ \mathbf{I}_{k\times k} & \mathbf{0}_{k\times(q-k)} \end{bmatrix}, \qquad (1)$$

where $\mathbf{I}$ and $\mathbf{0}$ respectively indicate the identity matrix and the zero matrix of size indicated in the subscripts. By projecting a given image $\mathbf{X} \in \mathbb{R}^{p\times q}$ onto this $\mathbf{G}$ matrix, an image with k-columns of shifts from the original $\mathbf{X}$ is resulted. Note that the right-most k-column vectors of $\mathbf{X}$ are overflowed and being moved to the first (left-most) k-columns of the shifted image by shifting projection.

Total Error Rate Minimization That Adopts the Extreme Learning Machine Kernel

Given m training samples $\mathbf{x}_i$, $i = 1, ..., m$, the conventional ELM learning algorithm [38] with $\tilde{N}$ hidden nodes can be written as

$$\tilde{\boldsymbol{\beta}} = \left(\mathbf{H}^T\mathbf{H}\right)^{-1}\mathbf{H}\mathbf{y}, \qquad (2)$$

where

$$\mathbf{H} = \begin{bmatrix} \phi\left(\mathbf{w}_1 \cdot \mathbf{x}_1 + b_1\right) & \cdots & \phi\left(\mathbf{w}_{\tilde{N}} \cdot \mathbf{x}_1 + b_{\tilde{N}}\right) \\ \vdots & \ddots & \vdots \\ \phi\left(\mathbf{w}_1 \cdot \mathbf{x}_m + b_1\right) & \cdots & \phi\left(\mathbf{w}_{\tilde{N}} \cdot \mathbf{x}_m + b_{\tilde{N}}\right) \end{bmatrix}_{m\times\tilde{N}},$$

$\mathbf{y} \in \mathbb{N}^m$ indicates the label of training data, $\phi(\cdot)$ denotes the sigmoid activation function, and $\mathbf{w}$ and b respectively denote the random weights and random bias. This solution minimizes the regression error in the least squares sense.

Instead of the least squares regression error, the classification error counting can be minimized. The classification error counting is different from the regression error in the sense that it is a discrete error count and is more suitable for classification problems. The classification error count can be expressed in terms of the classification total error rate which is defined as the sum of the false acceptance rate and the false rejection rate. Mathematically, it can be written as [37]

$$\text{TER} = \underbrace{\frac{1}{m^-}\sum_{j=1}^{m^-} L\left(g\left(\mathbf{x}_j^-\right) \geq \tau\right)}_{\text{False acceptance rate}} + \underbrace{\frac{1}{m^+}\sum_{j=1}^{m^+} L\left(g\left(\mathbf{x}_j^+\right) < \tau\right)}_{\text{False rejection rate}},$$

where m^+ and m^- ($m = m^+ + m^-$) respectively indicate the positive class and the negative class populations, $g(\cdot)$ denotes the predictor's output, τ is a predefined decision threshold, and $L(\cdot)$ is a step-loss function which returns a 1 whenever $(\cdot)$ is true, otherwise 0.

With an appropriate normalization plus inclusion of an offset term η, minimization of the TER with respective to a classifier linear in its parameters $\boldsymbol{\beta}$ turned out to have an analytic solution form [37]:

$$\hat{\boldsymbol{\beta}} = \left(\frac{1}{m^-}\mathbf{H}_-^T\mathbf{H}_- + \frac{1}{m^+}\mathbf{H}_+^T\mathbf{H}_+\right)^{-1} \times \left(\frac{(\tau-\eta)}{m^-}\mathbf{H}_-^T\mathbf{1}_- + \frac{(\tau+\eta)}{m^+}\mathbf{H}_+^T\mathbf{1}_+\right), \quad (3)$$

where

$$\mathbf{H}_- = \begin{bmatrix} \phi\left(\mathbf{w}_1 \cdot \mathbf{x}_1^- + b_1\right) & \cdots & \phi\left(\mathbf{w}_{\tilde{N}} \cdot \mathbf{x}_1^- + b_{\tilde{N}}\right) \\ \vdots & \ddots & \vdots \\ \phi\left(\mathbf{w}_1 \cdot \mathbf{x}_{m^-}^- + b_1\right) & \cdots & \phi\left(\mathbf{w}_{\tilde{N}} \cdot \mathbf{x}_{m^-}^- + b_{\tilde{N}}\right) \end{bmatrix}_{m^- \times \tilde{N}},$$

$$\mathbf{H}_+ = \begin{bmatrix} \phi\left(\mathbf{w}_1 \cdot \mathbf{x}_1^+ + b_1\right) & \cdots & \phi\left(\mathbf{w}_{\tilde{N}} \cdot \mathbf{x}_1^+ + b_{\tilde{N}}\right) \\ \vdots & \ddots & \vdots \\ \phi\left(\mathbf{w}_1 \cdot \mathbf{x}_{m^+}^+ + b_1\right) & \cdots & \phi\left(\mathbf{w}_{\tilde{N}} \cdot \mathbf{x}_{m^+}^+ + b_{\tilde{N}}\right) \end{bmatrix}_{m^+ \times \tilde{N}}$$

respectively indicate the ELM regression matrix constructed using only the negative samples $\mathbf{x}^-$ and positive samples $\mathbf{x}^+$, $\mathbf{1}_- = [1, \ldots, 1]^T \in \mathbb{N}^{m^-}$ and $\mathbf{1}_+ = [1, \ldots, 1]^T \in \mathbb{N}^{m^+}$. The learned parameter vector $\hat{\boldsymbol{\beta}}$ is then used to estimate the prediction output of an unseen test data $\mathbf{x}_t$ as follows:

$$\hat{y}_t = \mathbf{h}_t^T \hat{\boldsymbol{\beta}},$$

where $\mathbf{h}_t = [\phi(\mathbf{w}_1 \cdot \mathbf{x}_t + b_1), \cdots, \phi\left(\mathbf{w}_{\tilde{N}} \cdot \mathbf{x}_t + b_{\tilde{N}}\right)]^T$ indicates an ELM regressor vector constructed using $\mathbf{x}_t$.

The main difference between the ELM in Eq. 2 and the total error rate minimization that adopts the extreme learning machine kernel (TERELM) in Eq. 3 is that ELM adopts a regression-based solution whereas TERELM adopts a classification counting-based solution. When the class-specific weight is set at $M = (1, 1)$, the solution of TERELM is the same as that of ELM which uses the conventional least-squares learning. Such selection of the weight setting ($m^+ = m^- = 1$) is suitable for data sets with balanced categorical samples. However, for data sets with imbalanced samples between the two classes, other selections of the weight setting provide a mechanism to cater for the different importance of each class. Since the biometric samples often have a much larger impostor pool than the genuine-user pool, the TERELM is preferred over the ELM in our application.

Proposed Method And System

In this section, we propose a simple and yet effective projection-based feature extraction method called the line feature projection (LFP) for finger-knuckle-print verification. Figure 1 shows an overview of the proposed method for feature extraction which consists of four main processing steps. Essentially, the proposed method extracts both horizontal and vertical line features (see item (b) in Fig. 1) by projecting the input finger-knuckle image onto a shift-and-difference matrix (item (a) in Fig. 1). The extracted line features are subsequently transformed into a soft signum (sigmoid) representation (item (c) in Fig. 1) for contrast enhancement followed by the Fourier transform analysis (item (d) in Fig. 1). By integrating all the four steps into a single closed-form calculation step, we achieve computational efficiency as well as the desired accuracy performance. While the following subsections detail each of the four feature extraction steps, the "Proposed System: Fusion of H-LFP and V-LFP at Score Level" section provides a description of the overall proposed system and the "Empirical Analysis" section provides an empirical analysis of the feature extraction mechanism.

Shift-and-Difference Matrix

As aforementioned, the proposed feature extraction method starts from generating a horizontal and vertical difference image using the input image $\mathbf{X} \in \mathbb{R}^{p \times q}$. The conventional approach to generate the difference image consists of two steps: (i) shift the $\mathbf{X}$ by projecting it onto the circulant matrix $\mathbf{G}$ generated using Eq. 1 and (ii) subtract the shifted image from the input $\mathbf{X}$. With an aim of simplifying these two steps into a single matrix operation, we propose a shift-and-difference matrix $\mathbf{O}_n \in \mathbb{Z}^{n \times n}$ which is defined in our preliminary work [42]:

$$\mathbf{O}_n = \mathbf{I}_{n \times n} - \begin{bmatrix} \mathbf{0}_{(n-k) \times k} & \mathbf{I}_{(n-k) \times (n-k)} \\ \mathbf{I}_{k \times k} & \mathbf{0}_{k \times (n-k)} \end{bmatrix}_{n \times n} = \left[\, \mathbf{N}_{n \times k} \mid \mathbf{C}_{n \times (n-k)} \,\right]$$

$$
= \left[\begin{array}{cccc|ccccccc}
\multicolumn{4}{c|}{\overbrace{\hphantom{-1\quad 0\quad 0\quad 0}}^{k}} & \multicolumn{7}{c}{\overbrace{\hphantom{-1\quad 0\quad \cdots\quad \cdots\quad \cdots\quad \cdots\quad 0}}^{n-k}} \\
1 & 0 & \cdots & 0 & -1 & 0 & \cdots & \cdots & \cdots & \cdots & 0 \\
0 & 1 & & \vdots & 0 & -1 & & & & & \vdots \\
\vdots & & \ddots & 0 & \vdots & & \ddots & & & & \vdots \\
0 & \cdots & 0 & 1 & 0 & \cdots & 0 & -1 & 0 & 0 & 0 \\
0 & \cdots & & 0 & 1 & 0 & \cdots & 0 & -1 & 0 & 0 \\
\vdots & \ddots & \ddots & \vdots & 0 & \ddots & \ddots & & \ddots & \ddots & 0 \\
0 & & \cdots & 0 & 0 & 0 & 1 & 0 & \cdots & 0 & -1 \\
-1 & 0 & 0 & 0 & \vdots & 0 & 0 & 1 & 0 & \cdots & 0 \\
0 & -1 & & \vdots & \vdots & & & & \ddots & & \vdots \\
\vdots & & \ddots & 0 & 0 & & & & & 1 & 0 \\
0 & \cdots & 0 & -1 & 0 & \cdots & \cdots & \cdots & \cdots & 0 & 1
\end{array}\right]_{n\times n}
\begin{array}{l}
\\
\left.\vphantom{\begin{array}{c}1\\0\\ \vdots\\0\end{array}}\right\} k \\
\left.\vphantom{\begin{array}{c}0\\ \vdots\\0\end{array}}\right\} n-2k, \\
\left.\vphantom{\begin{array}{c}1\\0\\ \vdots\\0\end{array}}\right\} k
\end{array}
\qquad (4)
$$

where k indicates the number of image (row or column) vectors to be shifted, $\mathbf{I}_{n\times n}$ and $\mathbf{0}_{n\times n}$ respectively denotes the identity and zero matrices of size $n \times n$ pixels. Note that only the k is an adjustable parameter.

As shown in the second line of Eq. 4, the $\mathbf{O}_n$ matrix can be considered as a composition of two sub-matrices namely, $\mathbf{N}_{n\times k}$ and $\mathbf{C}_{n\times(n-k)}$. The former sub-matrix $\mathbf{N}_{n\times k}$ corresponds to the shifting and subtracting operation of those boundary image pixels which are overflowed by the shifting. On the other hand, the $\mathbf{C}_{n\times(n-k)}$ sub-matrix corresponds to the shifting and subtracting operation of the remaining image pixels.

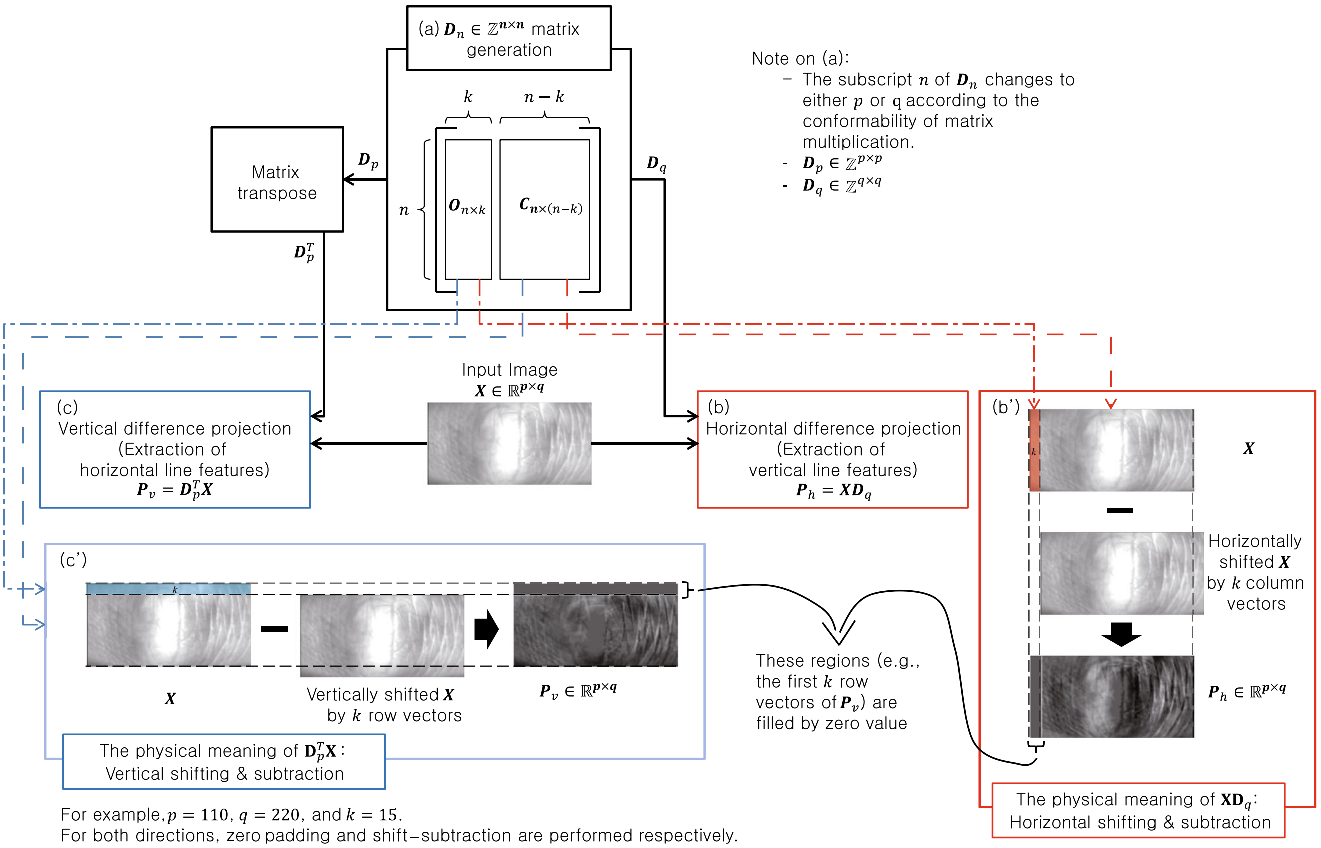
It is worth noting here that the operation involving $\mathbf{N}_{n\times k}$ triggers a noise block in the resulted difference image. From our empirical analysis (see "Empirical Analysis"), the effect of the noise block is obviously negative. The analysis also shows that the size of the noise block is proportional to k. To deal with this noise problem, in this work, an improved shift-and-difference matrix $\mathbf{D}_n \in \mathbb{Z}^{n\times n}$ is proposed:

$$
\mathbf{D}_n = \left[\, \mathbf{0}_{n\times k} \mid \mathbf{C}_{n\times(n-k)} \,\right], \qquad (5)
$$

where the sub-matrix $\mathbf{N}_{n\times k}$ of $\mathbf{O}_n$ is replaced by a zero matrix $\mathbf{0}_{n\times k}$. The core idea here is to eliminate the effect of the overflowed boundary noise pixels. The effect of the newly proposed $\mathbf{D}_n$ matrix is empirically analyzed in the "Empirical Analysis" section.

Directional Difference Image Projection

With the proposed shift-and-difference matrix $\mathbf{D}_n$ (see "Circulant Matrix"), a directional difference image can be obtained by a single projection operation. Even though deriving a difference image along any direction is possible in the image plane [43], we shall focus only on the horizontal


(a) $D_n \in \mathbb{Z}^{n\times n}$ matrix generation
k
$n-k$
n
$O_{n\times k}$
$C_{n\times(n-k)}$
Matrix transpose
D_p
D_p^T
D_q
Note on (a):
– The subscript n of D_n changes to either p or q according to the conformability of matrix multiplication.
- $D_p \in \mathbb{Z}^{p\times p}$
- $D_q \in \mathbb{Z}^{q\times q}$
Input Image
$X \in \mathbb{R}^{p\times q}$
(c)
Vertical difference projection
(Extraction of
horizontal line features)
$P_v = D_p^T X$
(b)
Horizontal difference projection
(Extraction of
vertical line features)
$P_h = XD_q$
(b')
X
Horizontally shifted X by k column vectors
$P_h \in \mathbb{R}^{p\times q}$
The physical meaning of $\mathbf{XD}_q$:
Horizontal shifting & subtraction
(c')
X
Vertically shifted X by k row vectors
$P_v \in \mathbb{R}^{p\times q}$
The physical meaning of $\mathbf{D}_p^T\mathbf{X}$:
Vertical shifting & subtraction
These regions (e.g., the first k row vectors of P_v) are filled by zero value

For example, $p = 110$, $q = 220$, and $k = 15$.
For both directions, zero padding and shift−subtraction are performed respectively.

Fig. 2 A graphical illustration of (a) shift-and-difference matrix generation, (b) horizontal and (c) vertical difference projection on a finger-knuckle-print image. Items labeled as (b') and (c') correspond to the physical meaning of (b) and (c) respectively

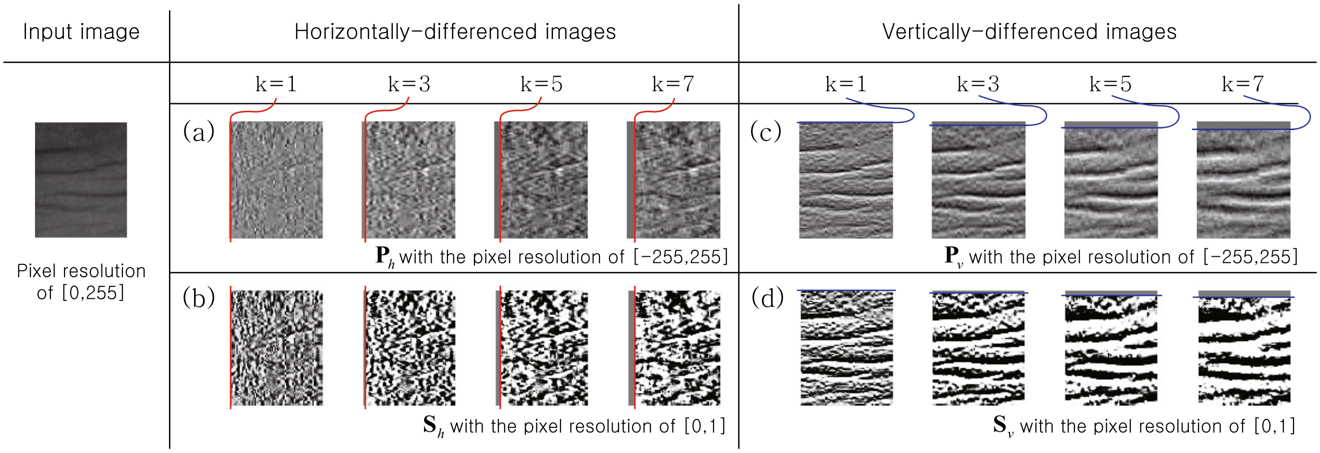

Input image
Horizontally–differenced images
Vertically–differenced images
k=1
k=3
k=5
k=7
(a)
(b)
(c)
(d)
Pixel resolution of [0,255]
P_h with the pixel resolution of [−255,255]
P_v with the pixel resolution of [−255,255]
S_h with the pixel resolution of [0,1]
S_v with the pixel resolution of [0,1]

Fig. 3 Some sample images of (a) horizontal and (c) vertical difference images shown with different shift count k. Images (b) and (d) correspond to the soft signum represented images of (a) and (c) respectively

and vertical directions in this work. This is because the finger-knuckle image is dominated by horizontal lines (or vertical lines if the image is rotated by 90°). An investigation into other directions will be the subject of our future study for more generic applications. The following subsections respectively describe the extraction of horizontal and vertical line features in detail.

Horizontal Difference Projection (Extraction of Vertical Line Features)

The horizontal difference image can be obtained by shifting the input image towards its left and its right. Since the leftward- and the rightward-shifting are symmetrical to each other, in this work, we shall use only the rightward-shifting which is defined as follows:

$$\mathbf{P}_h = \mathbf{X}\mathbf{D}_q, \tag{6}$$

where $\mathbf{P}_h \in \mathbb{R}^{p\times q}$ indicates the resulted horizontal difference image using k-columns of shift from the original $\mathbf{X}$, and $\mathbf{D}_q \in \mathbb{Z}^{q\times q}$ denotes the improved shift-and-difference matrix computed using Eq. 5. Note that the size of $\mathbf{D}$ is of $q \times q$ pixels due to the conformation of matrix-matrix multiplication.

Items (b) and (b’) in Fig. 2 respectively show the horizontal difference projection using $\mathbf{P}_h$ and its physical meaning applied on an exemplar finger knuckle image. As shown in the item (b’), those vertical lines are amplified by using $\mathbf{P}_h$. Moreover, the noise box caused by the overflowed boundary pixels are effectively eliminated.

Vertical Difference Projection (Extraction of Horizontal Line Features)

Similar to the horizontal difference projection described above, the vertical difference image can be obtained by shifting the input finger knuckle image to its upward and downward direction. In this work, we shall adopt the downward shifting which is defined as follows:

$$\mathbf{P}_v = \mathbf{D}_p^T\mathbf{X}, \tag{7}$$

where $\mathbf{P}_v$ indicates the obtained vertical difference image based on k row of shift vectors, and $\mathbf{D}_p \in \mathbb{Z}^{p\times p}$.

For the similar reason with that of $\mathbf{D}_q$ in Eq. 6, the size of shift-and-difference matrix $\mathbf{D}$ is $p \times p$ pixels. Items (c) and (c’) in Fig. 2 respectively show the vertical difference image projection using $\mathbf{P}_v$ and its physical meaning in which the horizontal lines are extracted. Similar

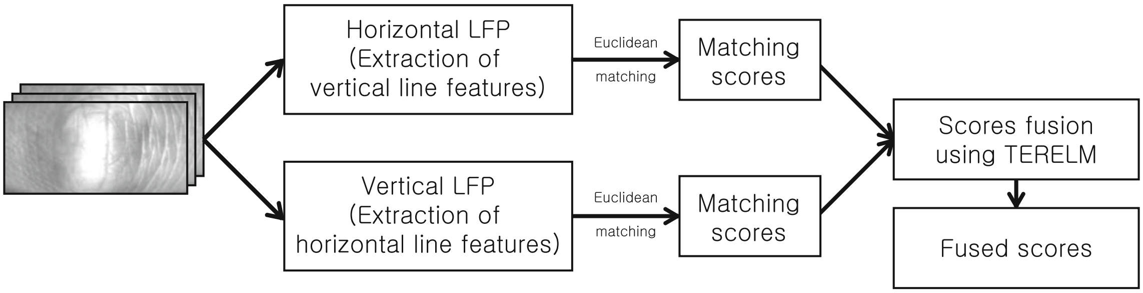

Horizontal LFP (Extraction of vertical line features)
Euclidean matching
Matching scores
Vertical LFP (Extraction of horizontal line features)
Euclidean matching
Matching scores
Scores fusion using TERELM
Fused scores

Fig. 4 The overview of the proposed system (combined LFP)

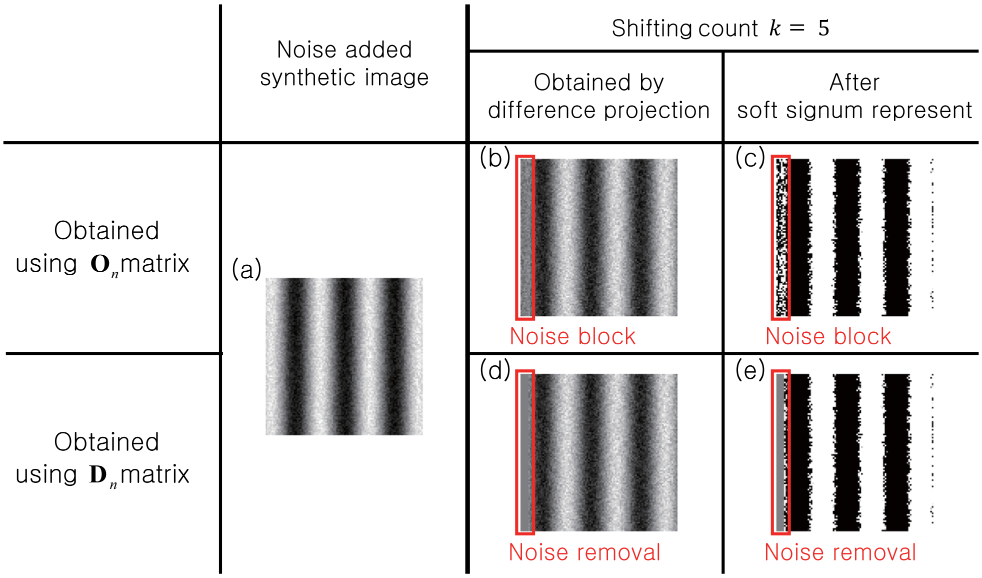

Noise added synthetic image
Shifting count $k = 5$
Obtained by difference projection
After soft signum represent
Obtained using $\mathbf{O}_n$matrix
Obtained using $\mathbf{D}_n$matrix
(a)
(b)
(c)
(d)
(e)
Noise block
Noise block
Noise removal
Noise removal

Fig. 5 Synthetic images which illustrate the effects of shifting count, soft signum representation, and boundary noise removal

to the results shown in item (b') in Fig. 2, the boundary noise effect is effectively eliminated.

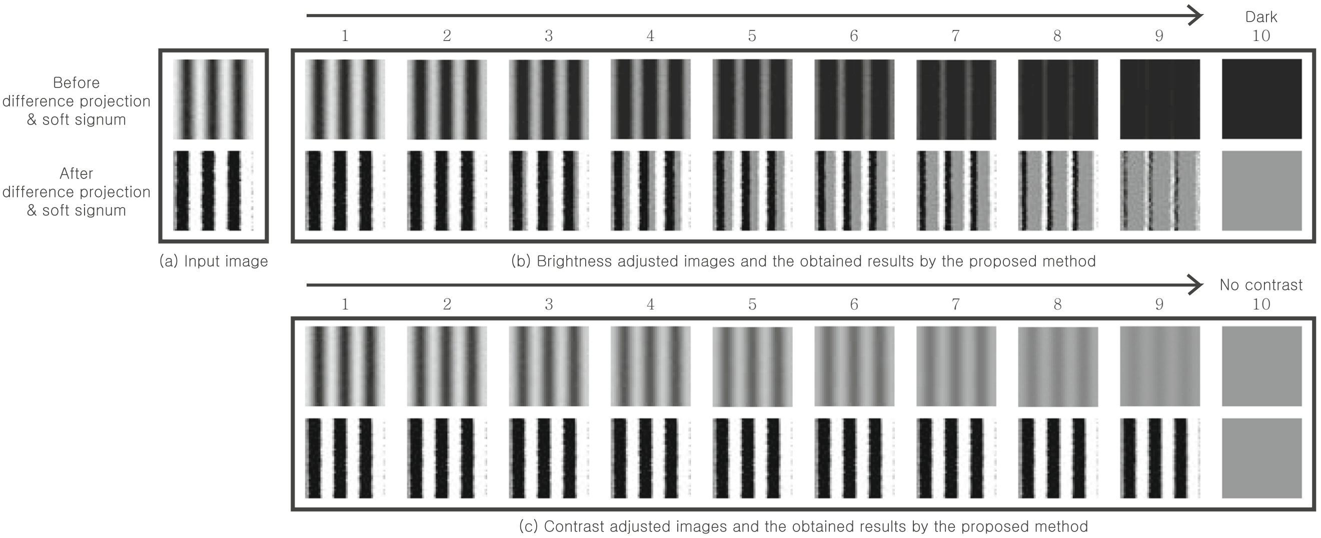
Soft Signum (Sigmoid) Representation

After the shift-and-difference projection operation, the resulted image contains only the difference values between k-pixels shifted row or column vectors. Since we are targeting on the presence of line features, a ternarization using Heaviside step function [44] was adopted in our previous work to represent image in unit sign information (i.e., $+1$, 0, -1) [42]. This process could also be considered as a de-noising process which removes non-line features. However, a hard signum representation for binary image is found to lost much of the geometric information [45]. Hence, in this work, we adopt a sigmoid function [46] for soft signum representation:

$$\sigma(x) = \frac{1}{1+e^{-x}}. \tag{8}$$

For a given projected matrix $\mathbf{P}_\star \in \mathbb{R}^{p\times q}$, where $\star \in \{h, v\}$, the soft signum represented image $\mathbf{S}_\star \in \mathbb{R}^{p\times q}$ is given by

$$\mathbf{S}_\star = \sigma(\mathbf{P}_\star). \tag{9}$$


Before difference projection & soft signum
After difference projection & soft signum
1
2
3
4
5
6
7
8
9
Dark 10
1
2
3
4
5
6
7
8
9
No contrast 10

(a) Input image

(b) Brightness adjusted images and the obtained results by the proposed method

(c) Contrast adjusted images and the obtained results by the proposed method

Fig. 6 Synthetic images which show varying amount of brightness and contrast at pixel level. Item (a) shows the input images and its 5-pixels shift-and-difference projection & soft signum represented images. Items (b) and (c) respectively show images with brightness and contrast changes on (a)

Sub-figures (b) and (d) of Fig. 3 respectively show the processed images of those from sub-figures (a) and (c) of Fig. 3 based on the soft signum representation.

Two-Dimensional Fourier Spectrum

After extracting the line features, the Fourier spectrum is adopted as a descriptor due to its shift-invariant characteristic. Based on the soft signum represented line images (see Fig. 3), we compute the magnitude, abs $(\cdot)$, of the two-dimensional discrete Fourier transform $\mathbf{F} \in \mathbb{C}^{p\times q}$ as follows:

$$\text{abs}\left(\mathbf{F}\left(u,v\right)\right) = \sqrt{\text{Re}\left(\mathbf{F}\left(u,v\right)\right)^2 + \text{Im}\left(\mathbf{F}\left(u,v\right)\right)^2}, \tag{10}$$

where

$$\mathbf{F}\left(u,v\right) = \sum_{n=1}^{q}\sum_{m=1}^{p} \mathbf{S}_\star\left(m,n\right)\cdot\exp\left[-j2\pi\left(\frac{(m-1)}{p}\left(u-1\right) + \frac{(n-1)}{q}\left(v-1\right)\right)\right],$$

$\text{Re}\left(\mathbf{F}\left(u,v\right)\right)$ and $\text{Im}\left(\mathbf{F}\left(u,v\right)\right)$ are respectively the real and imaginary parts of $\mathbf{F}\left(u,v\right)$, $j = \sqrt{-1}$ denotes the imaginary number, $u = 1,\ldots,p$ and $v = 1,\ldots,q$.

To facilitate a closed form representation, we apply the discrete Fourier transform (DFT) via matrix multiplication following [47]. Attributed to the mechanism of vectorization, this matrix multiplication shows fast computational speed under Matlab implementation. The DFT matrix $\mathbf{W}_n$ is a $n \times n$ square matrix defined as follows:

$$\mathbf{W}_n\left(a,b\right) = \left(\frac{\omega^{(a-1)(b-1)}}{\sqrt{n}}\right), \tag{11}$$

where $a, b = 1,\ldots,n$ and $\omega = e^{-\frac{2\pi i}{n}}$. Here, we note that $\mathbf{W}_n$ is a symmetric matrix in which the n becomes either p or q according to the conformation of the matrix multiplication. The Fourier spectrum based on the two-dimensional DFT on the soft signum represented image $\mathbf{S}_\star \in \mathbb{R}^{p\times q}$ can then be written as follows:

$$\mathbf{Y}_\star = \text{abs}\left(\mathbf{W}_p\mathbf{S}_\star\mathbf{W}_q\right), \tag{12}$$

where $\mathbf{Y}_\star \in \mathbb{R}^{p\times q}$, $\mathbf{W}_p \in \mathbb{C}^{p\times p}$ and $\mathbf{W}_q \in \mathbb{C}^{q\times q}$.

Summary of the Proposed Directional Line Features

As shown in Fig. 1 and discussed in the "Shift-and-Difference Matrix" to "Two-Dimensional Fourier Spectrum" sections, the four steps are performed in a sequential manner. In this subsection, we wrap up the four processing steps into a single closed form expression. The horizontal output feature matrix $\mathbf{Y}_h$ with input image matrix $\mathbf{X}$ is thus written as

$$\mathbf{Y}_h = \text{abs}\left(\mathbf{W}_p\sigma\left(\mathbf{X}\mathbf{D}_q\right)\mathbf{W}_q\right), \tag{13}$$

and the vertical output feature matrix $\mathbf{Y}_v$ is written as

$$\mathbf{Y}_v = \text{abs}\left(\mathbf{W}_p\sigma\left(\mathbf{D}_p^T\mathbf{X}\right)\mathbf{W}_q\right). \tag{14}$$

For convenience, we shall call the method to produce $\mathbf{Y}_h$ as horizontal LFP (H-LFP), the method to produce $\mathbf{Y}_v$ as vertical LFP (V-LFP). The procedure of the proposed method is summarized in Algorithm 1.

Algorithm 1 The proposed line feature projection method

```
Given: $m$ input samples, $\mathbf{X}_i \in \mathbb{R}^{p\times q}$, $i = 1, ..., m$, shift count $k$, sigmoid function $\sigma(\cdot)$, $\star \in \{h, v\}$
Output: $m$ output features, $\mathbf{Y}_i \in \mathbb{R}^{p\times q}$, $i = 1, ..., m$
$\mathbf{D}_p \in \mathbb{Z}^{p\times p}$, $\mathbf{D}_q \in \mathbb{Z}^{q\times q}$ $\leftarrow$ shift-and-difference matrix generated using Eq. 5
$\mathbf{W}_p \in \mathbb{C}^{p\times p}$, $\mathbf{W}_q \in \mathbb{C}^{q\times q}$ $\leftarrow$ DFT matrix generated using Eq. 11
for $i \leftarrow 1$ to $m$ do
    if $\star \leftarrow h$ then
        $\mathbf{Y}_i \leftarrow \text{abs}\left(\mathbf{W}_p\sigma\left(\mathbf{X}_i\mathbf{D}_q\right)\mathbf{W}_q\right)$
    else
        $\mathbf{Y}_i \leftarrow \text{abs}\left(\mathbf{W}_p\sigma\left(\mathbf{D}_p^T\mathbf{X}_i\right)\mathbf{W}_q\right)$
    end if
end for
```

Note: $'A \leftarrow B'$ indicates that B is assigned to A.

Proposed System: Fusion of H-LFP and V-LFP at Score Level

Items (a) and (c) in Fig. 3 respectively show the horizontal and vertical difference finger knuckle images obtained using $k \in \{1, 3, 5, 7\}$ (column and row) vectors. From the two items in the figure, it is observed that the horizontal difference projection extracts different information from that of the vertical difference projection. For example, the horizontal lines are amplified by the vertical difference projection (see "Horizontal Difference Projection (Extraction of Vertical Line Features)" for details). However, the details of the horizontal lines can be noised by the horizontal difference projection. Similarly, as discussed in the "Vertical Difference Projection (Extraction of Horizontal Line Features)" section, vertical line information can effectively be extracted by the horizontal image projection.

To take advantage of extracting direction-specific line information, we propose to fuse the two directional difference images at score level to improve the verification performance. According to [48], biometrics fusion can be performed at three different levels, namely, (i) feature level which uses a direct concatenation of multiple sets of extracted features for fusion, (ii) match score level which combines the output probability scores of different classifiers for fusion, and (iii) decision level which uses the discrete outputs of different classifiers for fusion. Among these levels, the match score level fusion is known to produce the best classification accuracy performance [48]. This is because the match score, which is a similarity measure between the input data and the system-stored template, possesses richer information than that of the decision level fusion and less redundant and noisy information than that of the feature level fusion.

The approaches to perform fusion at the match score level can be further divided into two categories, namely, (i) fusion by linear operation (e.g., SUM rule, MAX rule) and (ii) fusion by learning-based method (e.g., SVM, ELM, and TERELM). With an appropriate classifier learning, the latter approach can produce better classification accuracy than that of the former approach. Among the classifiers, we choose the TERELM because it provides the best verification accuracy and computational efficiency among the implemented classifiers (see "Experiments" for details). Note that the fusion of these two directional features shall be called the combined LFP (C-LFP) hereafter for simplicity. Figure 4 shows the processing steps of the C-LFP in the proposed system.

The matching scores, which consists of genuine-user scores and impostor scores, are respectively obtained based on the Euclidean distance between two templates of an identity (i.e., intra-matching) and of different identities (i.e., inter-matching). For instance, let $\mathbf{Y}_{hi} \in \mathbb{R}^{p\times q}$ and $\mathbf{Y}_{hj} \in \mathbb{R}^{p\times q}$ be the two H-LFP features extracted from the i-th and j-th subjects, respectively. The Euclidean distance between these two feature matrices, which is defined as follows:

$$s_{(i,j)} = \sqrt{\sum_{b=1}^{q}\sum_{a=1}^{p}\left(\mathbf{Y}_{hi}\left(a,b\right) - \mathbf{Y}_{hj}\left(a,b\right)\right)^2}, \tag{15}$$

is a genuine-user score if $i = j$, otherwise an impostor score (i.e., $i \neq j$).

The genuine-user scores (i.e., $\mathbf{s}_h^+$ and $\mathbf{s}_v^+$ where $\forall \mathbf{s}^+ = \left[s_1^+, \cdots, s_{m^+}^+\right]^T$) and impostor scores (i.e., $\mathbf{s}_h^-$ and $\mathbf{s}_v^-$ where $\forall \mathbf{s}^- = \left[s_1^-, \cdots, s_{m^-}^-\right]^T$) obtained respectively using the H-LFP and V-LFP features, are then used to construct the regressor matrix $\mathbf{H}$ in Eq. 3.

Empirical Analysis

In this section, we conduct two empirical analysis which investigate into (i) the effect of the zero padding in $\mathbf{D}_n$ and (ii) the sensitivity of the proposed method to illumination variation. Both the analyses are conducted using a synthetic image (see item (a) of Fig. 5), which consists of vertically stacked sinusoidal wave vectors (i.e., looks like a set of horizontal stripes). The Gaussian noise is subsequently injected to the synthetic image to simulate the noisy condition. The following subsections provide the detailed descriptions of the analysis, obtained results and our observations.

Analysis on the Effect of Zero Padding in $\mathbf{D}_n$ Recall that in "Shift-and-Difference Matrix," we proposed to substitute a zero matrix $\mathbf{0}_{n\times k}$ for the sub-matrix $\mathbf{N}_{n\times k}$ of the shift-and-difference matrix $\mathbf{O}_n$. This was to eliminate the noise block in the resulted difference image (see items (b) to (e) of Fig. 5, for example). In this study, the effect of the zero matrix substitution is qualitatively investigated. Particularly, the resulted difference image by using $\mathbf{D}_n$ is compared with that obtained using $\mathbf{O}_n$ [42]. Figure 5 shows the input synthetic image (item (a)), the obtained difference images by H-LFP using the previous shift-and-difference matrix $\mathbf{O}_n$ (items (b) and (c)) and the proposed shift-and-difference matrix $\mathbf{D}_n$ (items (d) and (e)), respectively. Note that the items (b) to (e) of Fig. 5 are obtained at a shifting count of $k = 5$ to show the noise removal effect clearly.

As shown in the items (b) and (c) of Fig. 5, the resulted information contained in the first $k = 5$ column vectors are corrupted by noise. We call the first k noisy column vectors as a noise block. The noise block is caused mainly by differences in pixel intensity values between the original k column vectors and the overflowed and shifted k columns of vectors. The size of the noise block is thus proportional to the value of k (see Fig. 3 for difference images obtained at several settings of $k \in \{1, 3, 5, 7\}$). Different from the results obtained using $\mathbf{O}_n$ (see items (b) and (c)), the noise block is replaced by a constant zero value using the proposed $\mathbf{D}_n$ matrix (see item (d)). Due to the sigmoidal mapping process which takes the pixel value range from [-255,255] to [0,1], the constant zero value in item (d) turns into 0.5 in item (e). In a nutshell, the proposed $\mathbf{D}_n$ matrix efficiently alleviates the effect of the noise block caused by overflow. Our empirical results as shown in Figs. 10 and 11 also reveal that the impact of the proposed $\mathbf{D}_n$ becomes more obvious at a higher value setting for k.

Analysis of the robustness to brightness and contrast variations Since the proposed method extracts the directional line information directly at image pixel level, observing its sensitivity to illumination variation is of a particular interest. The goal of this analysis study is to empirically observe the extent of variation of image brightness and contrast the proposed method can deal with [43].

The brightness and contrast of the input synthetic image (see Fig. 6a) is artificially adjusted to fit 10 levels. According to [49], the image brightness can be adjusted by adding or subtracting a constant to the image pixel intensities. Assuming the synthetic image is in gray scale with the pixel resolution of $[0, 255]$, the 10 levels of brightness (see Fig. 6b) can be achieved by setting the constant β at -25.5. After the intensity adjustment, those

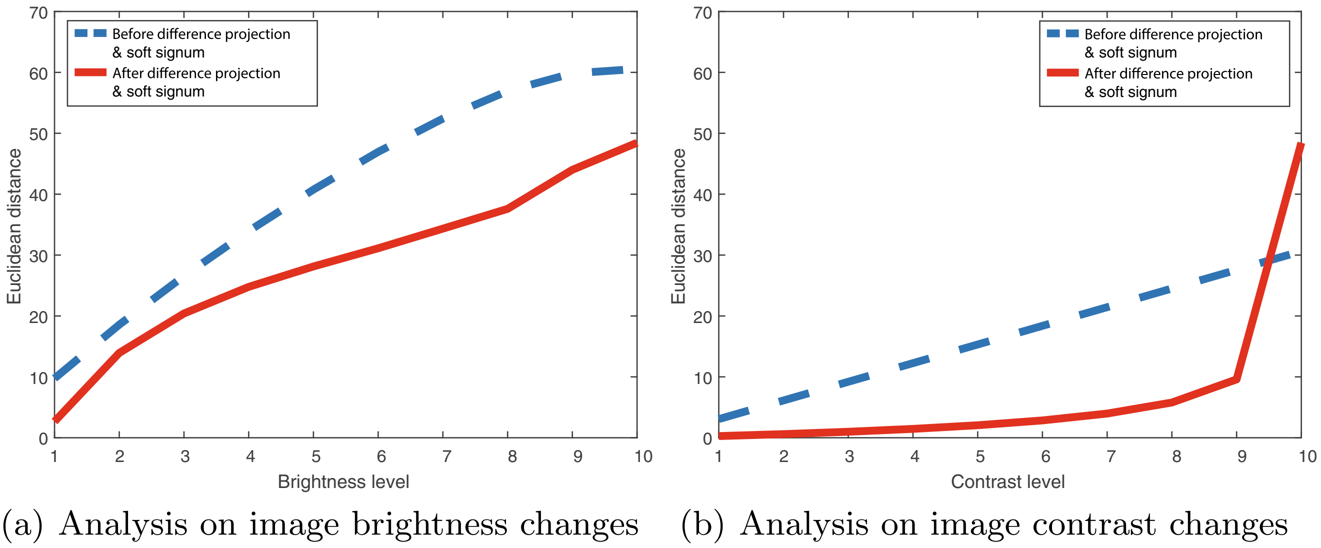


Fig. 7 Robustness of the images varying amount of brightness and contrast at pixel level. Graph labeled as **a** shows the Euclidean distance between Fig. 6**b** and **a**, and graph **b** shows the Euclidean distance between Fig. 6**c** and **a**. **a** Analysis on image brightness changes. **b** Analysis on image contrast changes

pixels with intensity values beyond the range [0, 255] are subsequently truncated and scaled [49]. Similar to the image brightness, we also quantify the image contrast into 10 levels (see Fig. 6c) using the Michelson contrast [50]:

$$\frac{I_{\max} - I_{\min}}{I_{\max} + I_{\min}}, \tag{16}$$

where $I_{\max}$ and $I_{\min}$ respectively denotes the highest and lowest image pixels in gray-scale.

Figure 6b and c respectively shows the brightness and contrast adjusted images obtained using the $k = 5$ columns shifted synthetic image (see Fig. 6a). As shown in Fig. 6b, the proposed method can extract the line information at brightness levels 1 to 3 without loss of much information.

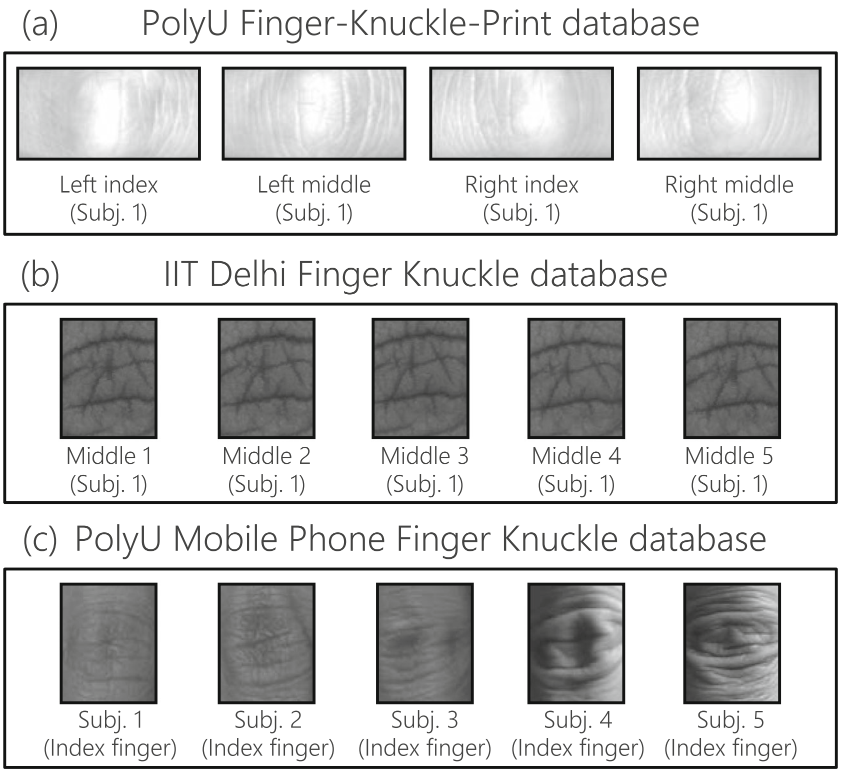


Fig. 8 Some samples images of the **a** PolyU, **b** IITD, and **c** Mobile databases

Table 2 Two experiments for performance evaluation of the proposed method

Experiments	Brief descriptions	Database
Experiment I	Evaluation of the FKP verification performance w.r.t.: (a) Comparison with the benchmarking methods (b) Comparison with other state-of-the-arts methods (non-deep learning-based) (c) CPU processing time performance	PolyU
Experiment II	Evaluation of the FKP verification performance under a less controlled (more realistic) environment	IITD, Mobile
Experiment III	Comparison with the DeepMatching method	PolyU, IITD, Mobile

However, the results obtained at a higher brightness level ($\geqslant 4$) are quite different from the results shown in Fig. 6a. This is due to the loss of texture information after the brightness adjustment. Different from the results obtained at different brightness levels (see Fig. 6b), the proposed method appear to be not affected by the contrast variation (see Fig. 6c). Except for the contrast level 10, our results remain unchanged over the evaluated contrast variation (levels 1 to 9).

In Fig. 7, we now quantify the above results (see Fig. 6) by computing the Euclidean distance to measure the similarity between the input image (see Fig. 6a) and the brightness (Fig. 6b) and contrast (Fig. 6c) adjusted images respectively. This is essentially a matching process between the original image and the adjusted images. Figure 7 a, b shows the two tracks of matching at various brightness and contrast levels based on the Euclidean similarity measure. In this quantitative evaluation, the lower the distance value, the less sensitive to illumination variation.

As shown in Fig. 7a, those images after the shift-and-difference projection followed by the soft signum step show a lower distance value than that before over the entire brightness levels. Although the processed images show a lower distance than that before, the entire distance remains to have a trend which increases with respect to the brightness level. On the other hand, Fig. 7b shows that the shift-and-difference projection followed by soft signum make the processed features less sensitive to contrast changes except at contrast level 10. It is worth noting here that since the texture information at contrast level 10 is completely lost (see the rightmost images of Fig. 6c), having such a high Euclidean distance value is not a surprise. Based on these results and observations, it can be concluded that the proposed features are not much affected by contrast variation while alleviating the effect of image brightness variation marginally.

Table 3 Class-specific weight settings for TER learning taken from [37]

$M : (w_{-}, w_{+})$	$M : (w_{-}, w_{+})$
$1 : (1, 1)$	$6 : (0.5m^{-}, m^{+})$
$2 : (m^{-}, m^{+})$	$7 : (m^{-}, 0.7m^{+})$
$3 : (m^{-}, 1)$	$8 : (0.7m^{-}, m^{+})$
$4 : (1, m^{+})$	$9 : (m^{-}, 0.3m^{+})$
$5 : (m^{-}, 0.5m^{+})$	$10 : (0.3m^{-}, m^{+})$

Experiments

This study aims to empirically verify the effectiveness and computational efficiency of the proposed LFP in finger-knuckle-print verification. To achieve this goal, the proposed method is evaluated on three publicly available FKP databases, and then compared with competing state-of-the-art methods in terms of the EER and CPU processing time. The following subsections provide details of the utilized databases, our experimental settings, results, and discussions.

Databases

Three FKP databases from the public domain, namely, PolyU [22], IIT Delhi [32], and PolyU mobile phone [34], are utilized in this experimental study. The PolyU database consists of 7,920 finger knuckle images (=4 fingers×12 samples×165 identities), which were acquired using a CCD camera with a LED light source. The images are of 110×220 pixel resolution. As shown in Fig. 8a, the 4 fingers include the left index finger, the left middle finger, the right index finger, and the right middle finger. Among the 165 subjects (125 males and 40 females), the 143 subjects were in the age group 20–30 years while the others were in the age group 30–50 years. A half of the 12 images was collected in Session 1 while the other half was in Session 2. Since the cropped images provided by the data publisher

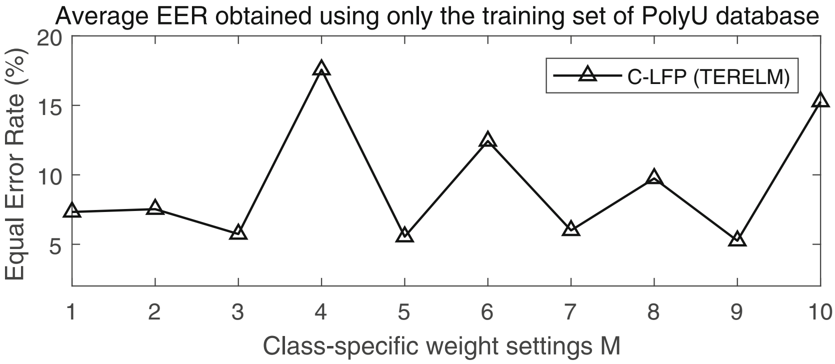


Fig. 9 The effect of TER class-specific weights tuning on the EER performance

are available only for 147 identities, we have performed the similar cropping process for the remaining 18 subjects as described in [19].

The IIT Delhi database (will be called IITD hereafter) consists of 158 identities wherein each subject contains 5 images taken only from the middle finger. Thus this database contains 790 images in total. The entire images were captured using a compact digital camera and are of 100×80 pixel resolution. All the subjects, which consist of student and staff at IIT Delhi, were in the age group 16-55 years. Some sample images of a subject are shown in Fig. 8b.

The last PolyU mobile phone database (will be called Mobile hereafter) [34] consists of 561 finger knuckle images taken from 187 fingers (i.e., 3 images per finger) of 109 subjects. The knuckle images were taken using a mobile phone camera. Different from the first two databases, the image pixel resolution of this database varies over images. To simplify the matter caused by the inconsistent image resolution, all the images of this database are resized to have a 100×80 pixel resolution as in [34]. Some of the processed images are shown in Fig. 8c.

Except for the manual cropping (i.e., for the 18 subjects of PolyU database) and resizing the Mobile knuckle images, no other image processing technique has been applied to the databases.

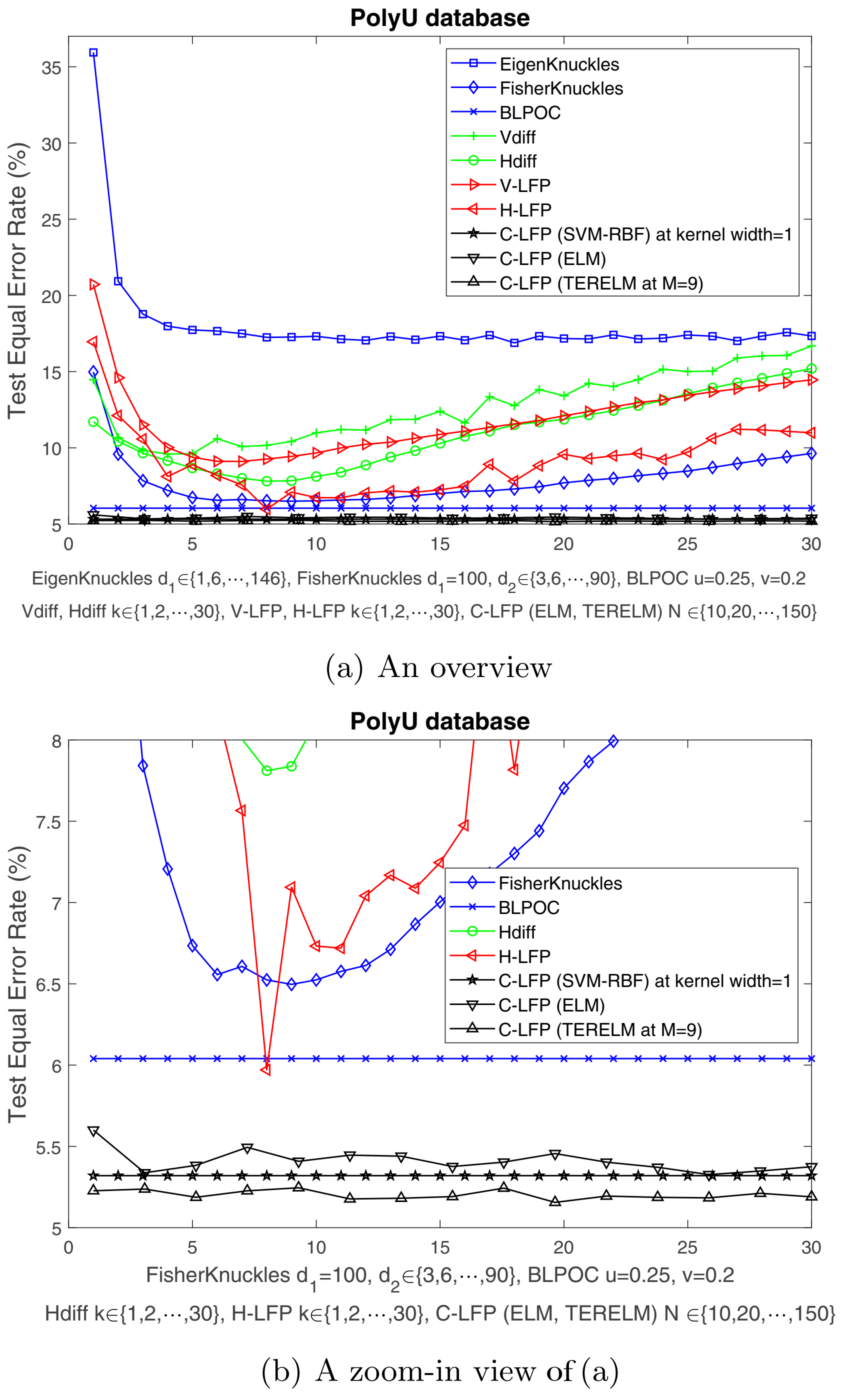
Experimental Setup

As shown in Table 2, three experiments are conducted to empirically investigate the FKP verification performance of the proposed method. Under the first experiment (denoted as Experiment I in Table 2), the proposed method is evaluated on the PolyU database in terms of the EER and CPU processing time performance. To position the proposed method among the related works, its performance is compared with that of the conventional FKP verification methods. Under the second experiment (Experiment II), the proposed method is evaluated under a more realistic scenario using the knuckle images of IITD and Mobile databases. The final Experiment III is conducted to make a comparison with a deep leaning based method utilizing all the three databases.

The methods included in our performance comparison are EigenKnuckles [32], FisherKnuckles [32], BLPOC [33], Vertical difference image (will be called Vdiff hereafter) [42], Horizontal difference image (Hdiff) [42], and DeepMatching [25]. Both EigenKnuckles and FisherKnuckles which extract the global FKP image features are chosen as the baseline methods for comparison. As for the representative global feature extraction method, the BLPOC is included in this comparison study. The two methods Vdiff and Hdiff are the authors' preliminary work on the FKP verification. Both methods are included in the comparison study to show the performance enhancement achieved in this work. Finally, the DeepMatching-based method is included in this study to make a comparison between the proposed method and the deep learning based method for FKP verification.

Following [19], the first half (Session 1) of the 7,920 PolyU images is used as the gallery set while the other half (Session 2) is used as the probe set. Since no standard data protocol is available for the IITD database, we perform a random 10 runs of twofold cross validation test in which the average values are recorded. Due to the limited number of samples (3 images/finger), a 10 runs of threefold test is conducted for the Mobile database. Due to the randomness in the ELM kernel, another 10 trails using different random hidden weights of ELM is performed for C-LFP where the average values are reported. All experiments were performed on a PC with 3.40GHz CPU and 16GBytes RAM under the Matlab R2017a platform. The Matlab code for the DeepMatching was obtained from [26].

Parameter settings for the proposed method The proposed V-LFP and H-LFP have an adjustable parameter, namely, the number of pixels k to be shifted. Since this parameter k is closely related to the line information being



(a) An overview

(b) A zoom-in view of (a)

Fig. 10 Experiment I-(a): The EER performance comparison among the implemented methods using the PolyU database. The k indicates the number of vectors to be shifted, d_1 and d_2 respectively denote the subspace dimension learnt by the PCA and the FLDA, u and v are the pass band ratios for image height and width, and $\tilde{N}$ is the number of hidden nodes. **a** An overview. **b** A zoom-in view of (a)

extracted (see Fig. 3a, c for example), the verification performance of the proposed method is evaluated over a grid set of $k \in \{1, 2, \ldots, 30\}$. On top of the parameter k, the C-LFP possesses two additional adjustable parameters namely, the number of hidden nodes $\tilde{N}$ in the ELM kernel and the class-specific weights setting $M \in \{1, 2, ..., 10\}$ (see Table 3 for details) in the TER learning [37]. Similar to k, our proposed method shall be evaluated over the grid settings of $\tilde{N} \in \{10, 20, \ldots, 150\}$ for the PolyU database and $\tilde{N} \in \{20, 40, \ldots, 600\}$ for both the IITD and the Mobile databases to observe the impact of $\tilde{N}$ on the classification accuracy. The limited search range of $\tilde{N}$ for the PolyU database was due to the memory constraint caused by the large data size. Since the optimal values of the utilized parameters in C-LFP can only be evaluated by the accuracy performance, the systematic grid search was performed based on the validation accuracy over the operating ranges of those parameters described above.

Table 4 Experiment I-(b): Comparison with competing state-of-the-art FKP verification methods in the literature in terms of the best EER performance using the PolyU database

Methods	Year	Best EER (%)
CompCode [22]	2009	1.72*
ImCompCode&MagCode [19]	2010	1.48*
LFI [30]	2012	1.27*
MoriCode&MtexCode [31]	2014	1.05*
ITC (vcode&hcode) [24]	2016	0.89*
EigenKnuckles [32] at $d_1 = 86$	2009	16.89
FisherKnuckles [32] at $d_1 = 100$, $d_2 = 27$	2009	6.50
BLPOC [33] at $u = 0.25$, $v = 0.2$	2009	6.04
Vdiff [42] at $k = 2$	2016	9.59
Hdiff [42] at $k = 8$	2016	7.81
V-LFP at $k = 7$		9.10
H-LFP at $k = 8$	Proposed	5.97
C-LFP (SVM-RBF) at kernel width=1	features and	5.32
C-LFP (ELM) at $\tilde{N} = 130$	fusion	5.33
C-LFP (TERELM) at $\tilde{N} = 100$, $M = 9$		5.16

*The EER value is taken from the original paper. The data protocol of this PolyU database is standardized [22]

Different from the parameter $\tilde{N}$, we locate the best combination of the TER weights among the ten M settings (see Table 3 which is taken from [37]) based on a twofold cross validation test using only the training set. Figure 9 shows the obtained classification accuracies over the ten M settings obtained using the training set of PolyU database. As shown in Fig. 9, the proposed C-LFP performs the best at $M = 9$. Note that both IITD and Mobile databases produce a similar accuracy trend comparing to that of the PolyU database, these results are not included here due to space constraint. Following [37], we set $\eta = \tau = 0.5$ in Eq. 3.

Recall that the C-LFP is a fusion of H-LFP and V-LFP at score level. The optimal value k for both H-LFP and V-LFP are chosen by a systematic grid search based on 10 runs of twofold cross validation tests using only the training set (gallery images). This is because the discriminative power of both LFPs over k can only be quantified by the classification accuracy performance. Due to the large number of impostor matchings on the PolyU database, training a classifier for match scores fusion often encounters an out-of-memory problem. To prevent such a problem, a random sub-sampling is performed on the impostor scores. Particularly, the number of impostor scores is reduced to be equal to the number of genuine scores according to [51]. The reduced impostor scores together with the entire genuine scores are then used for training the TER classifier for scores fusion (see "Proposed System: Fusion of H-LFP and V-LFP at Score Level" for details).

As described in the "Proposed System: Fusion of H-LFP and V-LFP at Score Level" section, one of our propositions is to fuse the match scores using the TERELM (denoted as C-LFP (TERELM) hereafter). To evaluate the performance standing of the TERELM model, fusions using the ELM model (denoted as C-LFP (ELM)) and the SVM adopting the RBF kernel (denoted as C-LFP (SVM-RBF)) are also included. The RBF kernel is selected for SVM due to its acclaimed accuracy performance over the linear and polynomial kernel.

Parameter settings for the compared methods Similar to the proposed V-LFP and H-LFP features, both Vdiff and Hdiff features [42] have the parameter k which controls the number of pixels to be shifted, as an adjustable parameter. The verification performance of Vdiff and Hdiff shall also be investigated over $k \in \{1, 2, \ldots, 30\}$ similar to that of the proposed LFPs.

For those subspace learning-based methods, EigenKnuckles and FisherKnuckles, we shall follow the conventional parameter settings. Particularly, the verification performance of EigenKnuckles shall be investigated over the reduced feature dimension $d_1 \in \{1, 6, \ldots, 146\}$. For FisherKnuckles, the input image is firstly projected onto a $d_1 = 100$ dimensional subspace learnt using the principal component analysis (PCA) [52], and then projected onto $d_2 \in \{3, 6, \ldots, 90\}$ dimensional subspace learnt using the linear discriminant analysis (LDA) [53].

The BLPOC has two adjustable parameters, namely, the pass band ratio u for the horizontal length and v for the vertical length. Since the data protocol for the PolyU database is standardized as mentioned, we adopt the settings

$u = 0.25$ and $v = 0.2$ which are taken from [23] for the PolyU database. For both the IITD and Mobile databases, different pass band ratios $u, v \in \{0.1, 0.2, \ldots, 1\}$ shall be experimented to locate its best verification accuracy performance.

Results and Discussion

Experiment I-(a). Comparison with those benchmark methods Figure 10 shows the test EER performances of the compared methods obtained using the PolyU database. Except for the BLPOC and C-LFP (SVM-RBF), all the obtained EER values are plotted over the running parameter (see the caption of Fig. 10) of each method. Since we have fixed the adjustable parameters for BLPOC, its EER value is repeated 30 times to show it as a line for easy comparison. In a similar manner, the best test accuracy performance of C-LFP (SVM-RBF) obtained at kernel width 1, which performs the best among $\{1, 2, 5\}$ settings, is also repeated 30 times in the figure. Note that the lower is the EER value, the better is the identity verification performance.

Figure 10a shows that among the compared non-fusion-based methods, H-LFP at $k = 8$ and BLPOC produce the lowest test EER performances. These two methods slightly outperform the FisherKnuckles at $d_1 = 100$, $d_2 = 27$ and Hdiff at $k = 8$ in terms of the best test EER performance. Among the compared methods, the EigenKnuckles yields the worst EER performance. Interestingly, both H-LFP and Hdiff, which extract the horizontal line features, produce slightly lower EER values than those of V-LFP and Vdiff which extract line features along the vertical direction. This indicates that the PolyU knuckle images contain more discriminative information in the horizontal direction than that in the vertical direction.

The figure also shows that the verification accuracy performance of the proposed features can be enhanced by match scores fusion using TERELM, C-LFP (TERELM). Particularly, the best EER of 5.97% for H-LFP at $k = 8$ and of 9.10% for V-LFP at $k = 7$ are reduced to 5.16% as a result of the score level fusion using TERELM. The C-LFP (ELM) and the C-LFP (SVM-RBF) respectively show 0.17% and 0.16% higher EER values (poorer accuracy performance) than that of C-LFP (TERELM) (see Fig. 10 (b)). As shown in Table 5, the computational speed of C-LFP (SVM-RBF) is much slower than that of C-LFP (ELM) and C-LFP (TERELM) in evaluating the test finger knuckle samples.

Experiment I-(b). Comparison with competing state-of-the-arts (non-deep learning based) Recall that the data protocol (e.g., image resolution, partitions) for the PolyU database is standardized [22]. This implies that the EER values obtained from our experiments can directly be compared with the reported ones in the literature. We take this opportunity to compare the proposed method with the competing state-of-the-art FKP verification methods in the literature in terms of the best test EER. The following methods are included in this comparison study: CompCode [22], ImCompCode&MagCode [19], LFI [30], MoriCode&MtexCode [31] and ITC (vcode&hcode) [24] (recall from the introduction section and see the first block of rows in Table 4). See the "Results and Discussion" section for comparisons between the proposed method and the DeepMatching [25] for FKP verification.

Among the compared state-of-the-art methods, ITC (vcode&hcode) shows the lowest EER value which is about 0.16% lower than that of MoriCode&MtexCode method. The EER of C-LFP (TERELM), which performs the best among the proposed methods, appears to be worse than these compared methods. It is worth noting here that, however, the ITC (vcode&hcode) is computationally demanding [24] because of the multiple times of information fusion. The LFI is also computationally heavy as it extracts the local feature from the entire image pixels [30]. The high efficiency of the proposed method is an advantage over the compared state-of-the-art methods (see "Results and Discussion" for our discussion on the computational efficiency). As described in the "Results and Discussion" section (see Table 6), the proposed features can produce significantly improved accuracy performance if incorporated with the DeepMatching.

Experiment I-(c). CPU processing time performance The compared methods are now evaluated in terms of their CPU processing times. As shown in Table 5, both the training time and test time are measured at specific parameter settings (see the last column of Table 5) in seconds. These parameter settings are the ones which produce the best test EER performance. For convenience, the best EER performances are also included in the table (see the fourth column of Table 5). For Vdiff, Hdiff, V-LFP, and H-LFP methods, the time elapsed for constructing the projection matrix and extracting features from the training set is reported as the training time. Since no training is required, we report only the test time for BLPOC. Note here that the test time indicates the average time (over 10 random trials) elapsed for evaluating one test sample.

In terms of the training time, Vdiff, Hdiff, V-LFP, and H-LFP are among the fastest methods. As shown in the third column of Table 5, both the V-LFP and H-LFP consume only about 0.5 ms to extract the feature from a sample. Due to the match score calculation step which is computationally heavy, training the C-LFP is much slower than that of the V-LFP and H-LFP. Among the three compared C-LFPs, the

Table 5 Experiment I-(c): The average CPU processing time taken over 10 random trials measured using the PolyU database. The test time means the average time elapsed for extracting the feature from a probe sample, and matching it against the features extracted from the entire gallery samples

Methods	Training time, sec	Test time, ms		Best EER, %	Parameter settings
		Feature	Matching/ estimation		
EigenKnuckles	12.34	2.6	6.6	16.89	$d_1 = 91$
FisherKnuckles	14.41	3.2	7.4	6.50	$d_1 = 100, d_2 = 27$
BLPOC†	–	–	4907.5	6.04	$u = 0.25, v = 0.2$
Vdiff	1.7310	0.489	145.7	9.59	$k = 4$
Hdiff	1.7786	0.497	152.4	7.81	$k = 8$
V-LFP	1.7294	0.483	149.3	9.10	$k = 7$
H-LFP	1.6778	0.482	145.2	5.97	$k = 8$
C-LFP (SVM-RBF)	312.01	308.7	92.7	5.32	Kernel width=1
C-LFP (ELM)	299.44	299.5	3.2	5.33	$\tilde{N} = 130$
C-LFP (TERELM)	299.28	299.5	4.1	5.16	$\tilde{N} = 100, M = 9$

†The BLPOC is a non-training-based matching method

C-LFP (SVM-RBF) appears to be much slower than that of C-LFP (ELM) and C-LFP (TERELM) in evaluating the test samples. The EigenKuckles and the FisherKnuckles are about 5–6 times slower than the proposed LFPs in terms of the feature extraction time. Despite the fact that C-LFP appears to be the slowest among the compared methods in feature extraction, it is still comparable with that of the LFIC$_2$ and MoriCode&MtexCode (note: measured under the C#.NET platform while our experiment is performed under the Matlab platform, see Table 1 for details). Among the compared methods, the BLPOC appears to be the most computationally expensive in terms of the test time even though it does not require the training stage.

Experiment II. FKP verification under less controlled (more realistic) environment As shown in Table 2, the compared methods are now evaluated under the less controlled environment using the IITD database (see Fig. 11a) and the Mobile database (see Fig. 11b). Similar to the plot shown in Fig. 10, both Fig. 11a and b show the test EER values of the compared methods, which are plotted over the experimented parameter settings.

Similar to the results shown in Fig. 10, the BLPOC and C-LFP with various classifiers are again the top four methods in terms of the verification accuracy. Among the three compared C-LFPs, the C-LFP (TERELM) shows slightly better test EER value in both databases, than those of C-LFP (ELM) and C-LFP (SVM-RBF). However, the BLPOC shows the among highest verification accuracy in both databases. The proposed V-LFP and H-LFP produce similar or slightly lower EER values than those of the Vdiff and H-diff after $k > 5$. The two subspace based methods namely, EigenKnuckles and FisherKnuckles produce the highest EER values on the two databases.

Experiment III. Comparison with the DeepMatching method In this experiment, we implement and investigate into the state-of-the-art DeepMatching method reported in [25]. With the four feature extraction methods (vcode&hcode as mentioned in the introduction section) which extract directional edge maps and using the DeepMatching, Jaswal et al. [25] achieved the state-of-the-art verification accuracy. We shall call this method as DeepMatching (vcode&hcode) for convenience. The DeepMatching is a CNN [54] inspired method to compute the dense correspondences between two images using a multi-layered convolutional architecture. It can handle non-rigid deformation and repetitive textures without any learning stage for its neural network model [26].

For the DeepMatching (vcode&hcode), we have an adjustable threshold T for the SGOP calculation. The value of T is experimentally fixed at $T = 0.2$ for the PolyU database and at $T = 0.05$ for the IITD and Mobile databases. For the non-training based score level fusion of the above four feature extractors, the best EER performance among the three fusion rules (SUM,MIN,MAX) will be reported.

Table 6 summarizes the verification performance of the DeepMatching method on all the three databases. Note here that the implemented performance and the reported EER

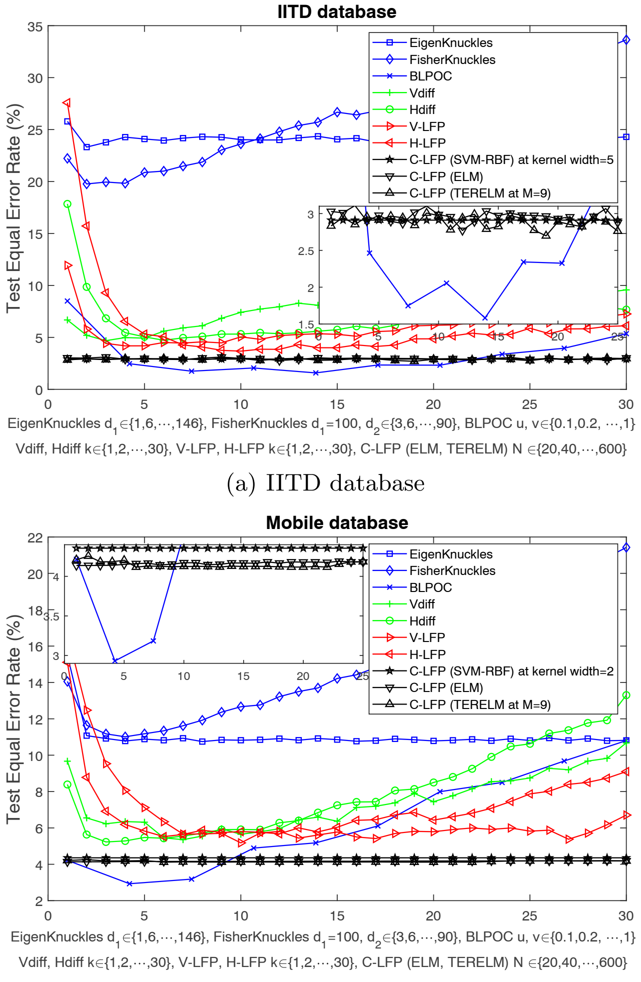
Table 6 Experiment III: Comparison with the DeepMatching method on all the three databases. The EER in the table represents the best test EER value and the CPU time provides the total elapsed time, which is the sum of the training and the test time

Methods	PolyU database			IITD database		Mobile database	
	EER, %	CPU time, sec	Parameter	EER, %	Parameter	EER, %	Parameter
DeepMatching (vcode^{GOP}) [25]	1.99	52.43	–	5.93	–	10.00	–
DeepMatching (hcode^{GOP}) [25]	5.82	52.22	–	6.94	–	6.53	–
DeepMatching (vcode^{SGOP}) [25]	3.83	51.77	$T = 0.2$	7.64	$T = 0.05$	9.38	$T = 0.05$
DeepMatching (hcode^{SGOP}) [25]	16.67	51.35	$T = 0.2$	6.44	$T = 0.05$	6.00	$T = 0.05$
DeepMatching (Fusion) [25]	1.71	119.16	MIN rule	0.13	MAX rule	0.31	MAX rule
V-LFP	9.10	1.88	$k = 7$	4.18	$k = 4$	5.18	$k = 10$
H-LFP	5.97	1.82	$k = 8$	3.68	$k = 10$	5.52	$k = 6$
DeepMatching (V-LFP)	0.20	22.53	$k = 2$	0.13	$k = 3$	0.09	$k = 2$
DeepMatching (H-LFP)	0.90	22.77	$k = 3$	0.00	$k = 4$	0.36	$k = 2$

in [25] differ slightly since the conventional ROI cropping protocol was adopted instead of the original one [25] proposed by the authors. As shown in the first block of rows of Table 6, the DeepMatching with a fusion of four extracted features shows outstanding verification accuracy compare to our V-LFP and H-LFP in all three databases (the second block of rows of Table 6). However, its computational cost is about 63 times heavier than that of the V-LFP. In order to make a comparison only within the feature extraction methods, we also implemented the DeepMatching on our extracted features by replacing our matching scheme by the DeepMatching scheme. Here, we exclude the two-dimensional Fourier spectrum step since it was originally adopted as a feature descriptor whose result (non-textured feature matrix) is not suitable for DeepMatching. As a result, our DeepMatching on H-LFP shows test EER performance of 0.9% at $k = 3$ and DeepMatching on V-LFP shows 0.2% test EER at $k = 2$ for the PolyU database (see the last block of rows of Table 6). Although it took much more CPU time (about 12 times more) than the conventional V-LFP and H-LFP, we can get similar or even better verification accuracy with less CPU time compare to the vcode&hcode in all three FKP databases. This result is encouraging in the sense that our feature extraction method can extract representative features and produces competitive verification accuracy when DeepMatching is incorporated.

Summary of results The results can be summarized as follows:

- From Figs. 10 and 11, the proposed features extracted by the V-LFP and H-LFP appear to possess similar or slightly more discriminative information than that of the Vdiff and Hdiff.
- The proposed V-LFP and H-LFP significantly outperform the compared subspace learning based methods (i.e., EigenKnuckles and FisherKnuckles) in terms of the test EER performance. The BLPOC showed even lower EER value than that of the V-LFP and H-LFP. However, as shown in Table 5, both the V-LFP and H-LFP significantly outperform the BLPOC in terms of the computational efficiency.
- The proposed C-LFP (TERELM) which fuses both the V-LFP and H-LFP show the competitive verification accuracy within the compared methods over the three databases with high computational efficiency. Although the BLPOC appear to be similar to the C-LFP (TERELM), their testing computational costs are respectively about 16 times heavier than that of the C-LFP (TERELM).
- Although the proposed LFPs have a slightly lower matching accuracy than that of the DeepMatching (vcode&hcode), the proposed feature extraction methods are shown to possess rich features with low computational cost. Moreover, when combing our features



(a) IITD database

(b) Mobile database

Fig. 11 Experiment II: The EER performance comparison among the implemented methods using the **a** IITD and **b** Mobile databases. The k indicates the number of vectors to be shifted, d_1 and d_2 respectively denote the subspace dimension learnt by the PCA and the FLDA, u and v are the pass band ratios for image height and width, and $\tilde{N}$ is the number of hidden nodes

with the DeepMatching method, a competing verification accuracy with the state-of-the-arts can be achieved.

Conclusion

In this paper, an analytic projection-based LFP method was proposed for finger-knuckle-print verification. Particularly, the vertically and horizontally differenced knuckle image obtained using the proposed shift-and-difference matrix was respectively activated using the sigmoid function for line contrast enhancement. The extracted line features were then extracted via a Fourier spectrum analysis. The two extracted directional line features were finally fused at the score level using the TERELM for verification performance enhancement. Our results on three public finger-knuckle-print databases showed that the proposed projection achieved competitive verification accuracy performance at lower computational cost than those of the compared FKP verification methods.

Acknowledgments The authors are thankful to the associate editor and the anonymous reviewers for their constructive comments to improve the paper.

Funding information This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (Grant number: NRF-2015R1D1A1A09061316).

Compliance with Ethical Standards

Conflict of interests The authors declare that they have no conflict of interest.

Informed consent Informed consent was not required as no human or animals were involved.

Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors.

References

1. Wang Y, Hu J, Phillips D. A fingerprint orientation model based on 2D Fourier expansion (FOMFE) and its application to singular-point detection and fingerprint indexing. IEEE Trans Pattern Anal Mach Intell. 2007;29(4):573–585.
2. Wang Y, Hu J. Global ridge orientation modeling for partial fingerprint identification. IEEE Trans Pattern Anal Mach Intell. 2011;33(1):72–87.
3. Hu H, Gu J. Multi-manifolds discriminative canonical correlation analysis for image set-based face recognition. Cogn Comput. 2016;8(5):900–909.
4. Mi JX, Li C, Li C, Liu T, Liu Y. A human visual experience-inspired similarity metric for face recognition under occlusion. Cogn Comput. 2016;8(5):818–827.
5. Jain AK, Ross A, Prabhakar S. An introduction to biometric recognition. IEEE Trans Circuits Syst Video Technol. 2004;14(1):4–20.
6. Zhang D, Kong WK, You J, Wong M. Online palmprint identification. IEEE Trans Pattern Anal Mach Intell. 2003;25(9):1041–1050.
7. Kong AWK, Zhang D. Competitive coding scheme for palmprint verification. In: Proceedings IEEE international conference on pattern recognition. Cambridge, UK; 2004. p. 520–523.
8. Zheng P. Gaussian shape descriptor for palmprint authentication. Cogn Comput. 2010;2(4):303–311.
9. Dai J, Feng J, Zhou J. Robust and efficient ridge-based palmprint matching. IEEE Trans Pattern Anal Mach Intell. 2012;34(8):1618–1632.
10. Ross A, Jain A, Pankati S. A prototype hand geometry-based verification system. In: Proceedings 2nd conference on audio and video based biometric person authentication; 1999. p. 166–171.
11. Fabregas J, Faundez-Zanuy M. Biometric recognition performing in a bioinspired system. Cogn Comput. 2009;1(3):257–267.
12. El-Alfy ESM, Abdel-Aal RE. Abductive learning ensembles for hand shape identification. Cogn Comput. 2014;6(3):321–330.

13. Faundez-Zanuy M, Mekyska J, Font-Aragonès X. A new hand image database simultaneously acquired in visible, near-infrared and thermal spectrums. Cogn Comput. 2014;6(2):230–240.
14. Kumar A, Zhang D. Personal recognition using hand shape and texture. IEEE Trans Image Process. 2006;15(8):2454–2461.
15. Xie SJ, Yoon S, Yang J, Lu Y, Park DS, Zhou B. Feature component-based extreme learning machines for finger vein recognition. Cogn Comput. 2014;6(3):446–461.
16. Kumar A, Prathyusha KV. Personal authentication using hand vein triangulation and knuckle shape. IEEE Trans Image Process. 2009;18(9):2127–2136.
17. Li Q, Qiu Z, Sun D, Wu J. Personal identification using knuckleprint. In: Advances in biometric person authentication. Guangzhou, China; 2005. p. 680-689.
18. Woodard DL, Flynn PJ. Finger surface as a biometric identifier. Comput Vis Image Underst. 2005;100(3):357–384.
19. Zhang L, Zhang L, Zhang D, Zhu H. Online finger-knuckle-print verification for personal authentication. Pattern Recogn. 2010;43(7):2560–2571.
20. Jaswal G, Kaul A, Nath R. Knuckle print biometrics and fusion schemes–overview, challenges, and solutions. ACM Computing Surveys (CSUR). 2016, Vol. 49.
21. Morales A, Travieso CM, Ferrer MA, Alonso JB. Improved finger-knuckle-print authentication based on orientation enhancement. IEEE Trans Image Process. 2006;15(8):2454–2461.
22. Zhang L, Zhang L, Zhang D. Finger-knuckle-print: a new biometric identifier. In: Proceedings IEEE international conference on image processing. Cairo, Egypt; 2009. p. 1981–1984.
23. Zhang L, Zhang L, Zhang D, Zhu H. Ensemble of local and global information for finger-knuckle-print recognition. Pattern Recogn. 2011;44(9):1990–1998.
24. Nigam A, Tiwari K, Gupta P. Multiple texture information fusion for finger-knuckle-print authentication system. Neurocomputing. 2016;188:190–205.
25. Jaswal G, Nigam A, Nath R. Finger knuckle image based personal authentication using deepmatching. In: Proceedings International conference on identity, security and behavior analysis. Jeju, Korea; 2017. p. 1-8.
26. Revaud J, Weinzaepfel P, Harchaoui Z, Schmid C. Deepmatching: Hierarchical deformable dense matching. Int J Comput Vis. 2016;120(3):300–323.
27. El-Tarhouni W, Shaikh MK, Boubchir L, Bouridane A. Multi-scale shift local binary pattern based-descriptor for finger-knuckle-print recognition. In: Proceedings IEEE conference on microelectronics. Doha, Qatar; 2014. p. 184-187.
28. Aoyama S, Ito K, Aoki T. A finger-knuckle-print recognition algorithm using phase-based local block matching. Inf Sci. 2014;268:53–64.
29. Kumar A, Zhou Y. Human identification using knucklecodes. In: Proceedings IEEE Conference on Biometrics: Theory, Applications, and Systems. Washington, USA; 2009. p. 1–6.
30. Zhang L, Zhang L, Zhang D, Guo Z. Phase congruency induced local features for finger-knuckle-print recognition. Pattern Recogn. 2012;45(7):2522–2531.
31. Gao G, Yang J, Qian J, Zhang L. Integration of multiple orientation and texture information for finger-knuckle-print verification. Neurocomputing. 2014;135:180–191.
32. Kumar A, Ravikanth C. Personal authentication using finger knuckle surface. IEEE Trans Inf Forensics Secur. 2009;4(1):98–110.
33. Zhang L, Zhang L, Zhang D. Finger-knuckle-print verification based on band-limited phase-only correlation. In: Proceedings International conference on computer analysis of images and patterns. Münster, Germany; 2009. p. 141-148.
34. Cheng K, Kumar A. Contactless finger knuckle identification using smartphones. In: Proceedings IEEE Conference on Biometrics Special Interest Group. Darmstadt, Germany; 2012. p. 1–6.
35. Hassan N, Akamatsu N. A new approach for contrast enhancement using sigmoid function. The Intern Arab J Inf Tech. 2004; 1(2):221–226.
36. Miyazawa K, Ito K, Aoki T, Kobayashi K, Nakajima H. An effective approach for iris recognition using phase-based image matching. IEEE Trans Pattern Anal Mach Intell. 2008;30 (10):1741–1756.
37. Toh KA, Eng HL. Between classification-error approximation and weighted least-squares learning. IEEE Trans Pattern Anal Mach Intell. 2008;30(4):658–669.
38. Huang GB, Zhu QY, Siew CK. Extreme learning machine: theory and applications. Neurocomputing. 2006;70(1):489–501.
39. Wang S, Deng C, Lin W, Huang GB, Zhao B. NMF-Based image quality assessment using extreme learning machine. IEEE Trans Cybern. 2017;47(1):232–243.
40. Deng C, Wang S, Li Z, Huang GB, Lin W. Content-insensitive blind image blurriness assessment using weibull statistics and sparse extreme learning machine. IEEE Trans Syst Man Cybern Syst Hum. 2017.
41. Davis PJ. Circulant matrices, 2nd. New York: American Mathematical Soc.; 2012.
42. Kim J, Oh K, Teoh ABJ, Toh KA. Finger-knuckle-print for identity verification based on difference images. In: Proceedings IEEE conference on industrial electronics and applications. Hefei, China; 2016. p. 1073–1077.
43. Gonzalez RC. Digital image processing Pearson Education India. 2009.
44. Abramowitz M, Stegun IA. Handbook of mathematical functions with formulas, graphs, and mathematical tables (9th Printing). New York: Courier Corporation; 1972.
45. Hong L, Wan Y, Jain A. Fingerprint image enhancement: algorithm and performance evaluation. IEEE Trans Pattern Anal Mach Intell. 1998;20(8):777–789.
46. Verhulst PF. Mathematical researches into the law of population growth increase. Nouveaux mémoires de l' Académie Royale des Sciences et Belles-Lettres de Bruxelles. 1845;18(1):1–42.
47. Rao KR, Yip PC. The transform and data compression handbook. CRC Press. 2000.
48. Ross A, Jain A. Information fusion in biometrics. Pattern Recogn Lett. 2003;24(13):2115–2125.
49. Szeliski R. Computer vision: algorithms and applications. Springer Science & Business Media. 2010.
50. Michelson AA. Studies in optics. Courier Corporation. 1995.
51. Oh BS, Oh K, Toh KA, Teoh ABJ. A single layer feedforward fusion network for face verification. In: Proceedings 13th International Conference on Control Automation Robotics & Vision; 2014. p. 944–948.
52. Turk M, Pentland A. Eigenfaces for recognition. J Cogn Neurosci. 1991;3(1):71–86.
53. Belhumeur PN, Hespanha JP, Kriegman DJ. Eigenfaces vs fisherfaces: recognition using class specific linear projection. IEEE Trans Pattern Anal Mach Intell. 1997;19(7):711–720.
54. LeCun Y, Bottou L, Bengio Y, Haffner P. Gradient-based learning applied to document recognition. Proc IEEE. 1998;86(11): 2278–2324.