# Phase-Based Palmprint Identification With Convolutional Sparse Coding

Luis Rafael Marval-Pérez, Koichi Ito, *Member, IEEE*, and Takafumi Aoki, *Senior Member, IEEE*

***Abstract*—Phase-based image matching has shown high recognition accuracy in palmprint verification. The algorithm compares a pair of palmprint images by extracting local phase features from the images and computing local correlation functions between them. A major drawback of this algorithm is its high computational cost associated with the evaluation of local correlation functions. This needs to be addressed, especially in the case of one-to-many comparisons required for palmprint identification. The problem becomes increasingly severe as the number of enrolled images increases. In this paper, we propose a novel palmprint identification algorithm with low computational complexity, which employs a sparse representation of enrolled phase features (i.e., phase templates) to evaluate local correlation functions. For this purpose, we also develop an efficient Convolutional Sparse Coding (CSC) algorithm that can derive a compact representation of phase templates. The proposed method reduces the computational cost of phase-based palmprint identification without significant degradation of recognition performance. Our experiments using public databases clearly demonstrate the advantage of the proposed method over conventional methods.**

***Index Terms*—Convolutional sparse coding, palmprint identification, phase-based image matching, phase correlation, phase features, biometrics.**

## I. Introduction

BIOMETRIC recognition is currently available in a wide range of applications including border controls, financial transactions and authentication in personal devices. There are many biometric traits that can be used for person recognition; examples include face, iris, fingerprint, palmprint, hand geometry, vascular patterns, DNA, voice, gait, and signature [1]. Among these traits, a palmprint is a very appealing trait for biometric recognition due to its balance between person recognition capability and simplicity of image data collection. The palmprint comprises a wide area, and has stable and discriminative features including principal lines, wrinkles, and ridges. A palmprint image can be captured by using a camera under an unconstrained environment while causing little to no discomfort to users [2]. In this paper, we focus on one-to-many comparisons required in the *identification* task instead of one-to-one comparisons required in the *verification* task, and investigate a fast and accurate palmprint identification method.

The approaches used in palmprint recognition are classified into four types [3], [4], [5]: (i) sub-space-based methods, (ii) coding-based methods, (iii) Convolutional Neural Network (CNN)-based methods, and (iv) correlation-based methods. An overview of each approach is given below.

*(i) Sub-space-based methods* — Sub-space-based methods employ Principal Component Analysis (PCA) [6], Independent Component Analysis (ICA) [7], their derivatives [8], [9], Linear Discriminant Analysis (LDA) [10], and Random Projection (RP) [11]. Dual Power Analysis (DPA) [12] has been proposed, which uses feature representation of 2D DCT and feature selection based on discrimination power, as a similar approach to LDA. These methods enable fast recognition by reducing the dimension of the features, while the problem is that the recognition accuracy is not very high.

*(ii) Coding-based methods* — The major approach of palmprint recognition is to use Gabor-coding-based methods inspired by the success in iris recognition [13]. Typical methods describe the palm texture by encoding the orientation of line features or ordinal features across the palm surface to have compact codes for identifying individuals such as Palmcode [14], Competitive Code [15], Double-Orientation Code [16], and Ordinal Code [17]. These Gabor-coding-based methods exhibit short computation times and low storage requirements [4]. PalmHash and PalmPhasor [18] have been proposed as another approach. These methods are intended for cancelable biometrics and to reduce the computational and storage cost. Other methods using Extreme Downsampling Method (EDM) [19] and Multiple-order Texture Co-occurrence Code (MTCC) [20] have also been proposed recently. These characteristics make them suitable for one-to-many comparisons required in the *identification* task. However, for advanced palmprint recognition systems with contactless image acquisition, Gabor-coding-based methods are not necessarily adequate since they lack the ability to cope with nonlinear palmprint deformations and imperfect preprocessing resulting from hand pose variations.

*(iii) CNN-based methods* — With the recent progress of CNN in face recognition, CNN-based methods are also considered in palmprint recognition. Dong *et al.* [21] proposed a modified version of MobileNet [22] for palmprint recognition. Zhao *et al.* [23] proposed a method using a modified

Manuscript received 25 January 2022; revised 30 April 2022; accepted 12 June 2022. Date of publication 17 June 2022; date of current version 15 July 2022. This work was supported in part by the Japan Society for the Promotion of Science (JSPS) KAKENHI under Grant 19H04106 and Grant 21H03457. This article was recommended for publication by Associate Editor Y. Wang upon evaluation of the reviewers' comments. *(Corresponding author: Koichi Ito.)*

Luis Rafael Marval-Pérez was with the Graduate School of Information Sciences, Tohoku University, Sendai 9808579, Miyagi, Japan. He is now with LIONS Data, Inc., Tokyo 1060032, Japan.

Koichi Ito and Takafumi Aoki are with the Graduate School of Information Sciences, Tohoku University, Sendai 9808579, Miyagi, Japan (e-mail: ito@aoki.ecei.tohoku.ac.jp).

Digital Object Identifier 10.1109/TBIOM.2022.3183568

version of AlexNet [24] for hyperspectral palmprint images. Genovese *et al.* [25] proposed PalmNet combining CNN and Gabor-based filter tuning. Liu and Kumar [26] proposed a deep learning-based contactless palmprint identification framework consisting of Residual Feature Network (RFN) and Soft-Shifted Triplet Loss (SSTL). Zhao and Zhang [27] convert CNN features into the label matrix by Joint Constrained Least-Square Regression (JCLSR) to improve their interpretability and performance. They also proposed a Learning Complete and Discriminative Direction Representation (LCDDR) method [28]. Liang *et al.* [29] proposed CompNet consisting of Learnable Gabor Convolution (LGC) layer, PostProcessing Unit (PPU), and multisize Competitive Blocks (CBs). Liu *et al.* [18] proposed Similarity Metric Hashing Network (SMHNet), which extracts plamprint features using a simple CNN, calculates scores by Structural Similarity Index Measure (SSIM), and compresses features by hashing. Shao *et al.* [30] considered the combination of Deep Distillation Hashing (DDH) and knowledge distillation to build a compact CNN model for palmprint recognition. Wu *et al.* [31] considered extracting the binary template from palmprint and palmvein images using deep hashing network. Xu *et al.* [32] proposed an accurate method for palmprint recognition by transfer learning of CNN pre-trained with soft biometrics for palmprints. These CNN-based methods achieve higher recognition accuracy than the coding-based methods, while they have the problem of requiring a considerable amount of training data.

*(iv) Correlation-based methods* — The most successful correlation-based methods for biometric recognition are those using phase-based matching. Phase-based matching techniques [33], [34] have demonstrated efficient verification performance for palmprint images by combining hierarchical correspondence search [35] with Band-Limited Phase-Only Correlation (BLPOC) [36]. Phase-based matching utilizes phase features extracted from local block images through 2D Discrete Fourier Transform (DFT). These local phase features are used to compute BLPOC functions that provide us accurate and robust similarity evaluation between images. In a real system, local phase features of users are to be enrolled in the system's gallery (which we call *phase templates*). However, these phase templates are large in size and the BLPOC functions are computationally expensive, especially for one-to-many comparisons required in the identification task. This results in a long computation time which is prohibitive to identification task with a large number of users. We can conclude that the phase-based matching approaches have excellent palmprint recognition accuracy at the cost of computational complexity and template size.

In this paper, we propose a compact representation of phase features that achieves identification time comparable to that of coding-based methods while keeping the discriminative capability of phase-based matching better than that of CNN-based methods. We can significantly reduce the computational complexity of BLPOC with compact phase templates obtained using Convolutional Sparse Coding (CSC) [37], which exploits the sparsity of phase features. We also propose a Phase-Based Hierarchical Block Matching algorithm with CSC (PB-HBM-CSC) for palmprint identification, which reduces the computation time without searching the corresponding blocks unlike the phase-based correspondence matching algorithm [33], [34]. Through the experiments on three public databases: PolyU Palmprint Image Database,[1] CASIA Palmprint Database,[2] and Tongji Database [38], we demonstrate the effectiveness of the proposed method and its advantages over conventional methods.

This study comprises two major contributions:

- a compact feature representation of phase features using CSC to reduce the computational complexity of BLPOC and
- PB-HBM-CSC for palmprint identification, which achieves identification time comparable to that of coding-based methods while keeping the discriminative capability of phase-based matching better than that of CNN-based methods.

## II. Phase-Based Image Matching for Biometric Recognition

In this section, we give an overview of the palmprint identification system using phase-based matching. We revisit the mathematical notation of phase-based matching [33], [34], [36]. The original notation for one-to-one matching is extended to that for one-to-many matching.

### A. General Considerations of Palmprint Identification System

A palmprint identification system using phase-based image matching consists of preprocessing, phase template extraction, and matching. The preprocessing method proposed in [14] is generally used to extract a palm region from a hand image. Since this method cannot handle actual situations such as finger opening and closing, it is necessary to use a modified version [39] or a CNN-based method [40]. The palm region is extracted as a region whose centroid is located on the perpendicular bisector of the line segment between keypoints. The extracted central region of the palm is normalized to have a fixed size, scale, rotation, and translation. We refer to the preprocessed palmprint images enrolled in the system's database (i.e., *gallery*) as *reference images*, and refer to the preprocessed input image to be identified as a *probe image*. Regarding the feature extraction process, local phase features are extracted from reference images and are stored in the form of *phase templates* inside the gallery during enrollment. Local phase features extracted from the probe image are compared with the enrolled phase templates during identification. The system conveys a similarity score (based on POC) for every reference image enrolled in the gallery. As a result, the highest score is expected to correspond to the specific palmprint image. Palmprint identification requires one-to-many matching between a probe image and a set of reference images.

[1] http://www4.comp.polyu.edu.hk/ biometrics/

[2] http://www.cbsr.ia.ac.cn/english/Palmprint Databases.asp

### B. Image Matching Using POC

For mathematical definition, we use the following symbols: $\lfloor . \rfloor$ means rounding towards negative infinity, $\lceil . \rceil$ means rounding towards positive infinity, $\otimes$ means circular cross-correlation and $\circledast$ means circular convolution. Consider two palmprint images, a reference image $I$ and a probe image $J$. We extract two block images of $L_1 \times L_2$ pixels from $I$ and $J$, respectively. Denote $f(n_1, n_2)$ (called a *reference block*) as a block centered at a point $\boldsymbol{p}$ on $I$, and denote $g(n_1, n_2)$ (called a *probe block*) as a block centered at a point $\boldsymbol{q}$ on the reference image. The ranges of image coordinates are given by $n_1 = -L_1^-, \ldots, L_1^+$ and $n_2 = -L_2^-, \ldots, L_2^+$ where $L_1^- = \lceil (L_1 - 1)/2 \rceil$, $L_1^+ = \lfloor (L_1 - 1)/2 \rfloor$, $L_2^- = \lceil (L_2 - 1)/2 \rceil$ and $L_2^+ = \lfloor (L_2 - 1)/2 \rfloor$.

The level of brightness, or DC component of a block, does not contribute to the description of the palm's texture and thus it should be subtracted before windowing. Let the DC components of the reference block $f(n_1, n_2)$ and probe block $g(n_1, n_2)$ be $f^{DC}$ and $g^{DC}$, respectively. Then, suitable frequency representations $F(k_1, k_2)$ and $G(k_1, k_2)$ of the blocks are given by the following expressions:

$$F(k_1, k_2) = \sum_{n_1, n_2} \left\{ f(n_1, n_2) - f^{DC} \right\} w(n_1, n_2) \times W_{L_1}^{k_1 n_1} W_{L_2}^{k_2 n_2}, \quad (1)$$

$$G(k_1, k_2) = \sum_{n_1, n_2} \left\{ g(n_1, n_2) - g^{DC} \right\} w(n_1, n_2) \times W_{L_1}^{k_1 n_1} W_{L_2}^{k_2 n_2}, \quad (2)$$

where $k_1 = -L_1^-, \ldots, L_1^+$, $k_2 = -L_2^-, \ldots, L_2^+$, $W_{L_1} = e^{-j\frac{2\pi}{L_1}}$, $W_{L_2} = e^{-j\frac{2\pi}{L_2}}$, and $\sum_{n_1, n_2}$ denotes $\sum_{n_1=-L_1^-}^{L_1^+} \sum_{n_2=-L_2^-}^{L_2^+}$. The function $w(n_1, n_2)$ is the Hanning window function. We define the *phase template* $X(k_1, k_2)$ of the reference block $f(n_1, n_2)$ as

$$X(k_1, k_2) = H(k_1, k_2) \frac{F(k_1, k_2)}{|F(k_1, k_2)|}, \quad (3)$$

where $H(k_1, k_2)$ is a spectral weighting function [41]. On the other hand, we define the *phase feature* $Y(k_1, k_2)$ of the probe block $g(n_1, n_2)$ as

$$Y(k_1, k_2) = \frac{G(k_1, k_2)}{|G(k_1, k_2)|}. \quad (4)$$

In order to evaluate the similarity between the reference block $f(n_1, n_2)$ and the probe block $g(n_1, n_2)$, we first compute the weighted cross-power spectrum $R(k_1, k_2)$ from $X(k_1, k_2)$ and $Y(k_1, k_2)$ as

$$R(k_1, k_2) = \overline{X(k_1, k_2)} Y(k_1, k_2). \quad (5)$$

The POC function $r^{\mathrm{POC}}(n_1, n_2)$ is obtained by the 2D IDFT of $R(k_1, k_2)$ as

$$r^{\mathrm{POC}}(n_1, n_2) = \frac{1}{L_1 L_2} \sum_{k_1, k_2} R(k_1, k_2) W_{L_1}^{-k_1 n_1} W_{L_2}^{-k_2 n_2}, \quad (6)$$

where $\sum_{k_1, k_2}$ denotes $\sum_{k_1=-L_1^-}^{L_1^+} \sum_{k_2=-L_2^-}^{L_2^+}$. The POC function is a correlation function useful for biometric recognition. It indicates a distinctive sharp peak when the image blocks $f(n_1, n_2)$ and $g(n_1, n_2)$ are similar. When they are dissimilar, the peak height, denoted as $\alpha$, drops significantly. The peak location indicates the translational displacement $\boldsymbol{\delta} = [\delta_1, \delta_2]$ between image blocks [41], [42].

The use of BLPOC [36] makes it possible to focus only on the essential frequency band in calculating the correlation function, which is particularly useful for biometrics. Let the band size of BLPOC be given by $B_1 \times B_2$, where $B_1 < L_1$ and $B_2 < L_2$. The BLPOC function uses limited frequency components $(k_1, k_2)$ as $k_1 = -B_1^-, \ldots, B_1^+$ and $k_2 = -B_2^-, \ldots, B_2^+$, where $B_1^- = \lceil (B_1 - 1)/2 \rceil$, $B_1^+ = \lfloor (B_1 - 1)/2 \rfloor$, $B_2^- = \lceil (B_2 - 1)/2 \rceil$ and $B_2^+ = \lfloor (B_2 - 1)/2 \rfloor$. Thus, the BLPOC function is the band-limited 2D IDFT of $R(k_1, k_2)$ as

$$r(n_1, n_2) = \frac{1}{B_1 B_2} {\sum_{k_1, k_2}}' R(k_1, k_2) W_{B_1}^{-k_1 n_1} W_{B_2}^{-k_2 n_2}, \quad (7)$$

where $\sum'_{k_1, k_2}$ denotes $\sum_{k_1=-B_1^-}^{B_1^+} \sum_{k_2=-B_2^-}^{B_2^+}$. The resulting BLPOC function $r(n_1, n_2)$ is defined for $n_1 = -B_1^-, \ldots, B_1^+$ and $n_2 = -B_2^-, \ldots, B_2^+$. In the same way as with POC, the location of the correlation peak $[\delta_1^{\mathrm{BL}}, \delta_2^{\mathrm{BL}}]$ indicates the translational shift between block images but with a factor, $\delta_1 = \delta_1^{\mathrm{BL}} L_1 / B_1$ and $\delta_2 = \delta_2^{\mathrm{BL}} L_2 / B_2$.

Let $x(n_1, n_2)$ and $y(n_1, n_2)$ be the band-limited 2D IDFTs of $X(k_1, k_2)$ and $Y(k_1, k_2)$, respectively. Unlike $f(n_1, n_2)$ and $g(n_1, n_2)$, these spatial representations have size $B_1 \times B_2$, thus, for $x(n_1, n_2)$ and $y(n_1, n_2)$, $n_1 = -B_1^-, \ldots, B_1^+$ and $n_2 = -B_2^-, \ldots, B_2^+$. Then, note that the BLPOC function $r(n_1, n_2)$ can be expressed in the spatial domain as the following circular cross-correlation as

$$r(n_1, n_2) = x(n_1, n_2) \otimes y(n_1, n_2), \quad (8)$$

where $\otimes$ indicates the operator of circular cross-correlation. In general, however, it is unnecessary to calculate $x(n_1, n_2)$ and $y(n_1, n_2)$ explicitly since the BLPOC function $r(n_1, n_2)$ is efficiently calculated by the band-limited 2D IDFT of $R(k_1, k_2)$ as shown in Eq. (7). In the rest of this paper, we will only use the band-limited versions of phase templates, phase features and cross-phase spectrum, i.e., $X(k_1, k_2)$, $Y(k_1, k_2)$ and $R(k_1, k_2)$, with $k_1 = -B_1^+, \ldots, B_1^+$ and $k_2 = -B_2^-, \ldots, B_2^+$.

## III. Phase-Based Image Matching With CSC for One-to-Many Comparisons

We define the CSC representation of the phase template and describe the calculation of the BLPOC function based on the CSC representation. The identification task requires one-to-many comparisons, in which the same probe block must be compared with a set of reference blocks from different reference images stored in the gallery database. Let $Y(k_1, k_2)$ denote a phase feature extracted from the probe image $J$, and let $\{X_1(k_1, k_2), \ldots, X_{N_u}(k_1, k_2)\}$ denote a collection of $N_u$ phase templates in the frequency domain extracted from the reference images $\{I_1, \ldots, I_{N_u}\}$ registered into the gallery database, where $N_u$ indicates the number of users registered in the system. The problem considered here is to find a computationally efficient way of calculating the $N_u$

BLPOC functions $\{r_1(n_1, n_2), \ldots, r_{N_u}(n_1, n_2)\}$ for evaluating the similarities between the probe's phase feature $Y$ and the reference's phase templates $\{X_1(k_1, k_2), \ldots, X_{N_u}(k_1, k_2)\}$. This section describes a basic idea of reducing the computational complexity of BLPOC functions as well as the data amount to be stored into the gallery database by introducing a sparse representation of phase templates.

### A. CSC Representation of Phase Templates

Consider here the spatial-domain representations $\{x_1(n_1, n_2), , \ldots, x_{N_u}(n_1, n_2)\}$ of phase templates $\{X_1(k_1, k_2), , \ldots, X_{N_u}(k_1, k_2)\}$. They contain enhanced palmprint textures in a normalized form, which are useful for highly accurate biometric recognition. Our experimental observation shows that these enhanced textures can be decomposed into a fewer number of primitive patterns, to be later used as atoms for sparse representation [37]. These primitive patterns repeat across enrolled phase templates at different locations. As a result, the $i$-th phase template $x_i(n_1, n_2)$ can be approximated by combining atoms with specific positions and intensities in the form of a sum of circular convolutions between atoms and sparse codes:

$$x_i(n_1, n_2) \approx \sum_{j=1}^{N_d} z_{i,j}(n_1, n_2) \circledast d_j(n_1, n_2), \tag{9}$$

where $\circledast$ indicates the operator of circular convolution, $n_1 = -B_1^-, \ldots, B_1^+$, and $n_2 = -B_2^-, \ldots, B_2^+$. Note that we use $i = 1, \ldots, N_u$ and $j = 1, \ldots, N_d$ if not otherwise specified. Being $d_j(n_1, n_2)$ the impulse response of the $j$-th convolution filter (or the $j$-th *atom*), the set of filters $\{d_1(n_1, n_2), \ldots, d_{N_d}(n_1, n_2)\}$ is called the *convolutional dictionary* for the sparse representation, where $N_d$ indicates the number of filters. The function $z_{i,j}(n_1, n_2)$ is the $j$-th code for the $i$-th phase template. In our design, the codes are sparse so that a collection of code functions $\{z_{i,1}(n_1, n_2), \ldots, z_{i,N_d}(n_1, n_2)\}$ has only $N_c$ non-zero coefficients as

$$\sum_{j=1}^{N_d} {\sum_{n_1,n_2}}' \lim_{p\to 0} \left|z_{i,j}(n_1, n_2)\right|^p = \sum_{j=1}^{N_d} \left\|\boldsymbol{z}_{i,j}\right\|_0 = N_c, \tag{10}$$

where $\sum'_{n_1,n_2}$ denotes $\sum_{n_1=-B_1^-}^{B_1^+} \sum_{n_2=-B_2^-}^{B_2^+}$. In Eq. (10), $\boldsymbol{z}_{i,j}$ is a vector whose elements are the values $z_{i,j}(n_1, n_2)$ and $\|\boldsymbol{z}_{i,j}\|_0$ is the $L^0$-norm of the vector, i.e., the number of non-zero elements of the vector [43]. The synthesis of phase templates in Eq. (9) can be rewritten in the frequency domain as

$$X_i(k_1, k_2) \approx \sum_{j=1}^{N_d} Z_{i,j}(k_1, k_2) D_j(k_1, k_2), \tag{11}$$

where $D_j(k_1, k_2)$ and $Z_{i,j}(k_1, k_2)$ are the 2D DFTs of $d_j(n_1, n_2)$ and $z_{i,j}(n_1, n_2)$, respectively.

### B. CSC-Based Approximation of BLPOC Functions

We describe how BLPOC functions can be represented using CSC formulas (Eqs. (9) and (11)), where we call this CSC-based approximation as CSC-BLPOC. In the following, we write equations in both domains: spatial domain and frequency domain for clarity purpose. Substituting the approximation of $x_i(n_1, n_2)$ in Eq. (9) into the definition of BLPOC function Eq. (8) (or equivalently, Eq. (11) into Eq. (5)), we have

$$r_i(n_1, n_2) \approx \left\{ \sum_{j=1}^{N_d} z_{i,j}(n_1, n_2) \circledast d_j(n_1, n_2) \right\} \otimes y(n_1, n_2) \tag{12}$$

in the spatial domain and

$$R_i(k_1, k_2) \approx \left\{ \sum_{j=1}^{N_d} \overline{Z_{i,j}(k_1, k_2) D_j(k_1, k_2)} \right\} Y(k_1, k_2) \tag{13}$$

in the frequency domain, respectively. These equations can be rewritten as

$$r_i(n_1, n_2) \approx \hat{r}_i(n_1, n_2) = \sum_{j=1}^{N_d} z_{i,j}(n_1, n_2) \otimes c_j(n_1, n_2) \tag{14}$$

in the spatial domain and

$$R_i(k_1, k_2) \approx \hat{R}_i(k_1, k_2) = \sum_{j=1}^{N_d} \overline{Z_{i,j}(k_1, k_2)} C_j(k_1, k_2) \tag{15}$$

in the frequency domain, respectively, where $c_j(n_1, n_2)$ and $C_j(k_1, k_2)$ are defined by

$$c_j(n_1, n_2) = d_j(n_1, n_2) \otimes y(n_1, n_2), \tag{16}$$

$$C_j(k_1, k_2) = \overline{D_j(k_1, k_2)}\, Y(k_1, k_2). \tag{17}$$

The function $\hat{r}_i(n_1, n_2)$ in Eq. (14) is an approximation of $r_i(n_1, n_2)$ using CSC, which we call CSC-BLPOC function. We use the symbol $\hat{\alpha}$ and $[\hat{\delta}_1^{\mathrm{BL}}, \hat{\delta}_2^{\mathrm{BL}}]$ to denote the height and location of the correlation peak for the CSC-BLPOC function. Since $c_j(n_1, n_2)$ is the cross-correlation function between the $j$-th atom $d_j(n_1, n_2)$ and $y(n_1, n_2)$, we call $c_j(n_1, n_2)$ *atomic correlation function*. As a result, we can understand Eq. (14) as a decomposition of a BLPOC function into a set of $N_d$ atomic correlation functions $\{c_1(n_1, n_2), \ldots, c_{N_d}(n_1, n_2)\}$.

### C. Computational Advantage of CSC-BLPOC

The purpose of using CSC-BLPOC is to reduce the computational cost of evaluating $N_u$ BLPOC functions $\{r_1(n_1, n_2), \ldots, r_{N_u}(n_1, n_2)\}$ corresponding to $N_u$ phase templates $\{X_1(k_1, k_2), \ldots, X_{N_u}(k_1, k_2)\}$. CSC-BLPOC functions can be computed from only $N_d$ ($\ll N_u$) atomic correlations regardless of the number of phase templates $N_u$. Furthermore, such computation is very fast by taking advantage of sparse codes. To confirm the above, we rewrite Eq. (14) as

$$\hat{r}_i(n_1, n_2) = \sum_{(j,m_1,m_2)\in\zeta_i} z_{i,j}(m_1, m_2) \tilde{c}_j(n_1 + m_1, n_2 + m_1), \tag{18}$$

where $\tilde{c}_j(n_1, n_2)$ is a 2D periodic extension of $c_j(n_1, n_2)$ (centered at $(0, 0)$ and with the 2D period of $(B_1, B_2)$). Note that the set $\zeta_i = \{(j, m_1, m_2) | z_{i,j}(m_1, m_2) \neq 0\}$ contains only $N_c$ triplets having nonzero values of $z_{i,j}(n_1, n_2)$. For better understanding of the computational advantage of CSC-BLPOC, we summarize the computational flow for both BLPOC and CSC-BLPOC cases. Given a phase feature

**Algorithm 1** Computation of BLPOC Functions

**for** $i \in \{1, \cdots, N_u\}$ **do**
(i) Compute $R_i(k_1, k_2)$ by Eq. (5)
(ii) Compute $r_i(n_1, n_2)$ by Eq. (7)
**end for**

**Algorithm 2** Computation of CSC-BLPOC Functions

**for** $j \in \{1, \cdots, N_d\}$ **do**
(i) Compute $C_j(k_1, k_2)$ by Eq. (17)
(ii) Compute $c_j(n_1, n_2)$ in the same way as Eq. (7)
**end for**
**for** $i \in \{1, \cdots, N_u\}$ **do**
(iii) Compute $\hat{r}_i(n_1, n_2)$ by Eq. (18)
**end for**

TABLE I
COMPUTATIONAL COST OF BLPOC FUNCTIONS AND CSC-BLPOC FUNCTIONS

| Method | # of operations |
|---|---|
| BLPOC | (i) $4N_u B^2$ real multiplications |
| | (ii) $8N_u B^2 \log_2 B$ real multiplications |
| CSC-BLPOC | (i) $4N_d B^2$ real multiplications |
| | (ii) $8N_d B^2 \log_2 B$ real multiplications |
| | (iii) $N_u N_c (2M+1)^2$ real multiplications |

$Y(k_1, k_2)$ extracted from a probe image and a set of phase templates $\{X_1(k_1, k_2), \ldots, X_{N_u}(k_1, k_2)\}$ extracted from $N_u$ reference images in the gallery, BLPOC and CSC-BLPOC are computed as shown in Algorithm 1 and 2.

In the case of CSC-BLPOC, we assume that the filters and the sparse codes are computed in advance through the CSC optimization algorithm described in the next section. Thus, the filters in frequency domain $\{D_1(k_1, k_2), \ldots, D_{N_d}(k_1, k_2)\}$ are given, and $N_u$ sets of sparse codes $\{z_{i,1}(n_1, n_2), \ldots, z_{i,N_d}(n_1, n_2)\}$ are given as shown in Algorithm 2. Additionally, if we assume $|\hat{\delta}_1^{\mathrm{BL}}| < M_1$ and $|\hat{\delta}_2^{\mathrm{BL}}| < M_2$ for the BLPOC peak positions, we can further reduce the computation of $\hat{r}_i(n_1, n_2)$ to a limited search range: $n_1 = -M_1, \ldots, M_1$ and $n_2 = -M_2, \ldots, M_2$.

In order to clarify the computational savings by CSC-BLPOC, we give a rough comparison of the computation cost between BLPOC and CSC-BLPOC as shown in Table I. For simplicity, we use a square image block size and a square search range, i.e., $B = B_1 = B_2$ and $M = M_1 = M_2$. Moreover, we assume that one complex multiplication corresponds to four real multiplications. We can see a clear similarity between the computation of BLPOC functions and the computation of atomic correlation functions (i.e., the step (i) and (ii) in CSC-BLPOC). If $N_d \ll N_u$, the computation of atomic correlation functions is much faster than the computation of BLPOC functions, where $N_d$ is the number of atoms, $N_u$ is the number of phase templates and the speed-up factor is $N_u/N_d$. The remaining step (iii) in CSC-BLPOC can be, in a sense, regarded as an overhead of CSC-based computation. For instance, assuming $N_u = 360$, $B = 32$, $M = 8$, $N_d = 16$, and $N_c = 32$, the number of real multiplications for computing the BLPOC functions is 16,220,160, while that for computing CSC-BLPOC functions is 4,050,176, and hence CSC-BLPOC is more than 4 times faster than BLPOC.

The concept of CSC-BLPOC computation can be extended to other correlation methods such as [44], [45] in the field of correlation pattern recognition [46].

## IV. CSC Optimization Algorithm of Phase Templates

In this section, we propose an efficient optimization algorithm for deriving a compact CSC representation of phase templates to be enrolled in the gallery.

### A. Problem Statement

Let $\hat{x}_i(n_1, n_2)$ denote the CSC approximation of the spatial-domain phase template $x_i(n_1, n_2)$, which is given by

$$\hat{x}_i(n_1, n_2) = \sum_{j=1}^{N_d} z_{i,j}(n_1, n_2) \circledast d_j(n_1, n_2). \tag{19}$$

The optimization problem considered in this section is to find adequate codes and filters to minimize the reconstruction error with a sparsity constraint. Formally, the problem is described as follows:

$$\begin{aligned} \underset{d_j(n_1,n_2), z_{i,j}(n_1,n_2)}{\operatorname{argmin}} \quad & \sum_{i=1}^{N_u} {\sum_{n_1,n_2}}' \left| x_i(n_1, n_2) - \hat{x}_i(n_1, n_2) \right|^2 \\ \text{subject to} \quad & \sum_{j=1}^{N_d} \left\| \boldsymbol{z}_{i,j} \right\|_0 = N_c \text{ for } i = 1, \ldots, N_u, \\ & {\sum_{n_1,n_2}}' \left| d_j(n_1, n_2) \right|^2 = 1 \text{ for } j = 1, \ldots, N_d. \end{aligned} \tag{20}$$

The sparsity constraint, i.e., $\sum_{j=1}^{N_d} \|\boldsymbol{z}_{i,j}\|_0 = N_c$, is a distinct characteristic in our biometric problem. To achieve a regular data structure for phase templates in the gallery, we force the code to have the same size $N_c$ for every template. Since this kind of $L^0$-norm constraint is difficult to solve, most of the researches on CSC relax this constraint to $L^1$-norm constraint [47], [48], [49], [50], [51]. However, we found that, for a small $N_c$ as is required here, the ideal $L^0$-norm constraint can be efficiently addressed through the use of a specially designed CSC optimization algorithm based on Matching Pursuit (MP) [52] technique, which is described below.

### B. Basic Strategy for Optimization

The filter optimization is a convex problem if we fix the codes in Eq. (20), while the code optimization is a combinatorial NP-hard problem due to the use of $L^0$-norm if we fix the filters. This code optimization is usually relaxed into a convex problem by changing the $L^0$-norm constraint to the $L^1$-norm constraint such as $\sum_{j=1}^{N_d} \|\boldsymbol{z}_{i,j}\|_1 \leq N_c$. The relaxed code optimization is known in the field of statistics as Least Absolute Shrinkage and Selection Operator (LASSO) regression. Taking advantage of the biconvex nature of the relaxed problem, a common practice in CSC optimization is to alternate between code optimization for fixed filters

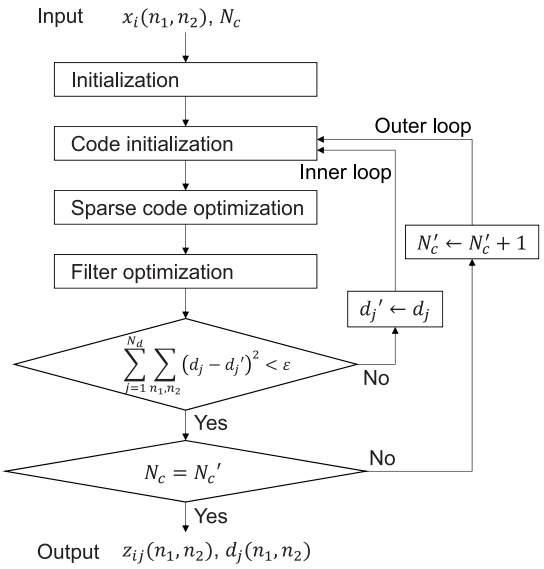


Fig. 1. Flow diagram of the CSC optimization algorithm.

and filter optimization for fixed codes. Following this strategy, [47], [48], [49] proposed the application of Alternating Direction Method of Multipliers (ADMM) [53] to CSC. Furthermore, [50], [51] presented extended frameworks that make them possible to employ not only ADMM but also proximal gradient methods such as Fast Iterative Shinkage-Thresholding (FISTA) [54].

However, these algorithms can not be applied to our problem defined by Eq. (20), where strict $L^0$-norm constraint with small $N_c$ (i.e., highly sparse condition) must be addressed. Clearly, it is quite difficult to address the general class of $L^0$-norm problems directly. But, for problems with small $N_c$, we found that MP technique can be applied efficiently to code optimization. For smaller $N_c$, MP ensures better reconstruction accuracy close to the global optimum, as is theoretically analyzed in [55].

From the above-mentioned point of view, we propose a new optimization method that alternately performs MP for code optimization and Ordinary Least Squares (OLS) for filter optimization. Note that the proposed approach can be regarded, in a sense, as an extension of the Method of Optimal Directions (MOD) [56] to convolutional versions of dictionary learning.

### C. Incremental CSC Optimization Algorithm

The overall structure of our proposed CSC optimization algorithm is depicted in Fig. 1. The basic procedures for the inner loop are code optimization based on MP and filter optimization using OLS as described above. Although relatively good performance is expected by alternating between MP and OLS, the quality of the solution significantly depends on the initial setting of filter $d_j(n_1, n_2)$. We address this drawback by adding an outer loop to start the optimization with the simplified sparsity constraint and to repeat the optimization by gradually increasing the number of non-zero elements until it reaches $N_c$. For this purpose, we introduce a new counter

**Algorithm 3** Initialization

Set $N_c'$ to 2
**for** $j \in \{1, \cdots, N_d\}$ **do**
  Initialize $d_j(n_1, n_2)$ by random values generated from the normal distribution with $\mu = 0$ and $\sigma^2 = 1$
  Apply a Hanning window to $d_j(n_1, n_2)$
  Compute the filter energy $e_j$ by

$$e_j = \sum_{n_1,n_2}{}' \left| d_j(n_1, n_2) \right|^2$$

  Normalize $d_j(n_1, n_2)$ by $d_j(n_1, n_2)/\sqrt{e_j}$
**end for**

**Algorithm 4** Code Initialization

**Require:** $N_c'$ and $z_{i,j}(n_1, n_2)$
**Ensure:** $z_{i,j}(n_1, n_2)$
**if** $N_c' \leq 10$ and # of inner-loop iterations $\leq 8$ **then**
  **for** $i \in \{1, \cdots, N_u\}$ and $j \in \{1, \cdots, N_d\}$ **do**
    $z_{i,j}(n_1, n_2) \leftarrow 0$
  **end for**
**end if**

$N_c'$ for controlling the number of nonzero elements in the inner loop. We found that the result of optimization with the $L^0$-norm sparsity $N_c'$ can be used as a good initial condition for the case of $N_c' + 1$. This incremental approach drastically improves the quality of the final solution. This algorithm consists of 4 steps as follows.

*(i) Initialization:* Algorithm 3 describes the initialization of the $L^0$-norm sparsity $N_c'$ and filters $d_j(n_1, n_2)$.

*(ii) Code Initialization:* The code initialization is shown in Algorithm 4. For every iteration the codes $z_{i,j}(n_1, n_2)$ are to be initialized to 0 in the ordinary MP algorithm. However, we observe through a set of preliminary experiments that the reconstruction accuracy can be improved by skipping the code initialization and preserving the code values for the next iteration, after the counter $N_c'$ reaches a certain value. We determine such value of $N_c'$ by empirical observations.

*(iii) Sparse Code Optimization:* Let us define the residual atomic correlation $c_{i,j}^{res}(n_1, n_2)$ as follows:

$$c_{i,j}^{res}(n_1, n_2) = d_j(n_1, n_2) \otimes \left\{ x_i(n_1, n_2) - \hat{x}_i(n_1, n_2) \right\}, \quad (21)$$

where $\hat{x}_i(n_1, n_2)$ is computed by Eq. (19). Then, "Sparse code optimization" in Fig. 1 is shown in Algorithm 5.

*(iv) Filter Optimization:* The filter optimization problem in Fig. 1 (derived from Eq. (20)) is solved easily in the frequency domain, similar to the recent CSC studies [47], [48], [49], [51]:

$$\underset{D_j(k_1,k_2)}{\operatorname{argmin}} \sum_{i=1}^{N_u} \left| X_i(k_1, k_2) - \sum_{j=1}^{N_d} D_j(k_1, k_2) Z_{i,j}(k_1, k_2) \right|^2. \quad (22)$$

This is an OLS formulation independent of each frequency component, which produces the optimized filter $d_j(n_1, n_2)$ and scaled codes $z_{i,j}$. Then, the filter optimization is shown in Algorithm 6. Note that the outputs of the CSC optimization

**Algorithm 5** Sparse Code Optimization

**Require:** $N_c'$, $x_i(n_1, n_2)$, $z_{i,j}(n_1, n_2)$ and $d_j(n_1, n_2)$
**Ensure:** $z_{i,j}(n_1, n_2)$
  **for** $i \in \{1, \cdots, N_u\}$ **do**
    Count the number of non-zero elements $N_c''$ by $\sum_{j=1}^{N_d} \|\boldsymbol{z}_{i,j}\|_0$
    **if** $N_c'' < N_c'$ **then**
      **while** $N_c'' < N_c'$ **do**
        Compute $c_{i,j}^{res}(n_1, n_2)$ by Eq. (21)
        Find $\{j', n_1', n_2'\}$ maximizing the absolute value of the residual atomic correlation functions by

$$\{j', n_1', n_2'\} \leftarrow \underset{j,n_1,n_2}{\operatorname{argmax}} \left| c_{i,j}^{res}(n_1, n_2) \right|$$

        Update $z_{i,j'}(n_1', n_2')$ by

$$z_{i,j'}(n_1', n_2') \leftarrow c_{i,j'}^{res}(n_1', n_2')$$

        $N_c'' \leftarrow N_c'' + 1$
      **end while**
    **else**
      Compute $c_{i,j}^{res}(n_1, n_2)$ by Eq. (21)
      Update all the nonzero elements in $z_{i,j}(n_1, n_2)$ by

$$z_{i,j}(n_1, n_2) \leftarrow z_{i,j}(n_1, n_2) + \eta c_{i,j}^{res}(n_1, n_2) \quad (0 < \eta < 1)$$

    **end if**
  **end for**

**Algorithm 6** Filter Optimization

**Require:** $x_i(n_1, n_2)$ and $z_{i,j}(n_1, n_2)$
**Ensure:** $d_j(n_1, n_2)$ and $z_{i,j}(n_1, n_2)$
  Compute 2D DFT of $x_i(n_1, n_2)$ and $z_{i,j}(n_1, n_2)$
  Obtain $D_j(k_1, k_2)$ by solving Eq. (22)
  **for** $j \in \{1, \cdots, N_d\}$ **do**
    Compute the filter energy $e_j$ by

$$e_j = \sum_{k_1,k_2}{}' \left| D_j(k_1, k_2) \right|^2 / (B_1 B_2)$$

    Normalize $D_j(k_1, k_2)$ by $D_j(k_1, k_2)/\sqrt{e_j}$
    Scale $z_{i,j}(n_1, n_2)$ by $z_{i,j}(n_1, n_2)\sqrt{e_j}$
  **end for**
  Compute 2D DFT of $D_j(k_1, k_2)$

algorithm are available both in the spatial and frequency domain depending on applications, although the outputs are in the frequency domain in Algorithm 6. In practice, we use a subset of phase templates, for example, which are extracted from the center of the palm, to reduce the computation time required for filter optimization. In the experiments, this approach did not result in any loss of reconstruction accuracy.

## V. Palmprint Identification With Phase-Based Hierarchical Block Matching

In the real situation, palmprint images usually contain global translations, rotations and nonlinear deformations. Hence, given a small local block on a full-size palmprint image, it is not an easy task to find the corresponding block location on another palmprint image. In our previous paper [33], we applied a phase-based correspondence matching technique using a coarse-to-fine search strategy [35] to palmprint recognition, which we later extended to other biometric traits [34]. The use of phase-based correspondence matching makes it possible to achieve very high recognition accuracy in palmprint [33], finger-knuckle-print [57], [58], [59], and face [60]. The goal of this section is to modify the correspondence matching algorithm to perform one-to-many comparison with reduced computational complexity using the proposed CSC-based correlation technique proposed in Section III. In the following, we present three palmprint identification algorithms.

### A. Phase-Based Correspondence Matching

The first is Phase-Based Correspondence Matching (PB-CM) [33], which is the baseline. Given a reference image $I$ (registered in the gallery) and a probe image $J$, the correspondence matching algorithm (i) extracts a set of block images from $I$ at predetermined locations, (ii) searches the corresponding block locations on $J$ with a coarse-to-fine strategy, (iii) extracts a set of corresponding blocks from $J$, and (iv) evaluates the block-wise similarities to have an overall matching score between $I$ and $J$. By using phase-based image matching in the steps (ii) and (iv), this algorithm achieves high accuracy in one-to-one biometric recognition. However, a major problem regarding the computational efficiency arises when we apply this correspondence matching algorithm to one-to-many comparisons required for identification. We must repeat the steps (i)–(iv) $N_u$ times to compare a single probe image $J$ with a set $N_u$ reference images $\{I_1, \ldots, I_{N_u}\}$. The locations of the reference blocks on the reference images $I_1, \ldots, I_{N_u}$ are predetermined when we enroll the reference blocks (or more precisely the phase templates Eq. (3) of the blocks) in the gallery. Hence, for every reference block to be enrolled, it is sufficient to compute 2D DFT only once for generating its phase template and this computation can be done off-line. On the other hand, the locations of the probe blocks on $J$ vary depending on the reference images $I_1, \ldots, I_{N_u}$ to be compared. This means the total number of different probe blocks could become very large, as much as the total number of reference blocks enrolled in the gallery. As a consequence, the number of 2D DFT computations to be performed on-line for generating the phase features of probe blocks in Eq. (4) is significantly large in the identification task.

### B. Phase-Based Hierarchical Block Matching

The second is Phase-Based Hierarchical Block Matching (PB-HBM), which uses blocks at fixed locations in an image pyramid for matching to reduce the computation complexity

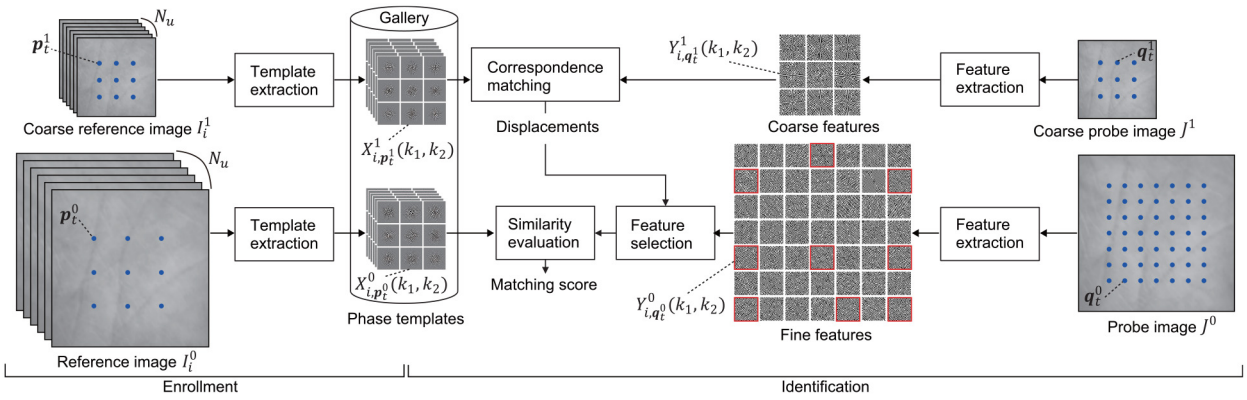


Fig. 2. Overall palmprint identification algorithm using two-layer Phase-Based Hierarchical Block Matching (PB-HBM). Red boxes on fine features indicate the selected features for similarity evaluation.

without searching the corresponding blocks unlike [33]. PB-HBM starts with defining $N_b$ reference blocks on $I$, finds their best match (corresponding) blocks on $J$, and evaluates their block-wise similarity. The corresponding blocks are selected from predetermined candidate blocks, i.e., fixed locations, on $J$.

Let $P = \{\boldsymbol{p}_1, \ldots, \boldsymbol{p}_{N_b}\} \subseteq \mathbb{Z}^2$ denote the set of locations (i.e., coordinates) for the $N_b$ reference blocks on $I$, and let $Q_{\text{cand}} \subseteq \mathbb{Z}^2$ denote the set of locations for candidate blocks on $J$, where we assume the number of candidate locations in $Q_{cand}$ is sufficiently large, i.e., $|Q_{cand}| \geq N_b$. Given the set $P$ on $I$, the problem is to find a set of corresponding block locations $Q(\subseteq Q_{\text{cand}})$ on $J$. To do this, we shall adopt a coarse-to-fine search strategy on a multi-scale image pyramid. We use the superscript $l$, such as $I^l$ and $J^l$, to indicate the layer $l = 0, \ldots, l_{max}$, in the image pyramid. By starting at the original image layer $I^0 = I$ and $J^0 = J$, the lower resolution images, $I^l$ and $J^l$ for $(l \geq 1)$, are created as follows:

$$I^l(n_1, n_2) = \frac{1}{4} \sum_{j_1=0}^{1} \sum_{j_2=0}^{1} I^{l-1}(2n_1, +j_1, 2n_2 + j_2), \quad (23)$$

$$J^l(n_1, n_2) = \frac{1}{4} \sum_{j_1=0}^{1} \sum_{j_2=0}^{1} J^{l-1}(2n_1, +j_1, 2n_2 + j_2). \quad (24)$$

The reference block locations $P^l = \{\boldsymbol{p}_1^l, \ldots, \boldsymbol{p}_{N_b}^l\}$ on the $l$-th layer of the reference image $I^l$ are determined in advance from $P$ as follows:

$$\boldsymbol{p}_t^l = \left\lfloor \frac{1}{2^l} \boldsymbol{p}_t \right\rfloor, \quad (25)$$

where $t = 1, \ldots, N_b$ and $l = 0, \ldots, l_{max}$. Thus, for original image layer $l = 0$, we have $\boldsymbol{p}_t^0 = \boldsymbol{p}_t$. Given the reference block $\boldsymbol{p}_t^l$ on $I^l$, the problem considered here is to find the corresponding block $\boldsymbol{q}_t^l$ on $J^l$, for all $t = 1, \ldots, N_b$, with coarse-to-fine strategy starting from $l = l_{max}$ to $l = 0$. We reduce the computational complexity of the coarse-to-fine search by restricting the corresponding blocks to a set of predetermined locations $Q_{\text{cand}}^l$ on $J^l$.

The procedure to recursively find the corresponding block pairs $(\boldsymbol{p}_t^l, \boldsymbol{q}_t^l)$ from the upper layer pairs $(\boldsymbol{p}_t^{l+1}, \boldsymbol{q}_t^{l+1})$ consists of the following steps: (i) compute BLPOC function between the blocks at $\boldsymbol{p}_t^{l+1}$ and $\boldsymbol{q}_t^{l+1}$, (ii) derive their displacement $\boldsymbol{\delta}_t^{l+1}$, and (iii) update their positions to have $(\boldsymbol{p}_t^l, \boldsymbol{q}_t^l)$ with higher resolution, where $\boldsymbol{p}_t^l$ is given by Eq. (25) and $\boldsymbol{q}_t^l$ is derived as

$$\boldsymbol{q}_t^l = \underset{\boldsymbol{q} \in Q_{\text{cand}}^l}{\text{argmin}} \left\| \boldsymbol{q} - 2\left(\boldsymbol{q}_t^{l+1} + \boldsymbol{\delta}_t^{l+1}\right) \right\|_2. \quad (26)$$

For simplicity, we start the above recursion with the initial setting:

$$\boldsymbol{q}_t^{l_{max}} = \boldsymbol{p}_t^{l_{max}}. \quad (27)$$

As a result of the above procedure, we can obtain the set of $N_b$ pairs $(\boldsymbol{p}_t^0, \boldsymbol{q}_t^0)$ of corresponding blocks at the original image layer. For all the block pairs $(\boldsymbol{p}_t^0, \boldsymbol{q}_t^0)$, we compute BLPOC functions and derive their block-wise similarities (i.e., BLPOC peak values $\alpha_t^0$). By taking an average of these block-wise similarities, we have an overall matching score between the image pairs $I$ and $J$.

Fig. 2 illustrates the overall flow of palmprint identification using two-layer PB-HBM. Note that we empirically found that two resolution layers (coarse $l = 1$ and original $l = 0$) are sufficient, and hence $l_{max} = 1$. The enrolment process is depicted at the left side of the gallery database while the identification process depicted at the right side. In the enrolment process, we construct an image pyramid for each reference image $I_i$, extract the phase templates $X_{i,\boldsymbol{p}_t^l}^l(k_1, k_2)$ from $I_i^l$ at fixed locations $\boldsymbol{p}_t^l$ and store them into the gallery, where $i = 1, \ldots, N_u$, $l = 0, 1$ and $t = 1, \ldots, N_b$. A detailed description of the enrolment procedure is shown in Algorithm 7. In the identification process, we construct an image pyramid of $J$, extract the phase features $Y_{\boldsymbol{q}}^l(k_1, k_2)$ from $J^l$ at fixed locations $q$ in $Q_{\text{cand}}^l$ and compare them with the phase templates stored into the gallery. These are a total of $2N_uN_b$ comparisons which are carried out by computing BLPOC functions between phase features and phase templates. A detailed description of the identification procedure is shown in Algorithm 8.

### *C. Phase-Based Hierarchical Block Matching With Convolutional Sparse Coding*

The third is the palmprint identification algorithm proposed in this paper, PB-HBM-CSC, which introduces CSC to

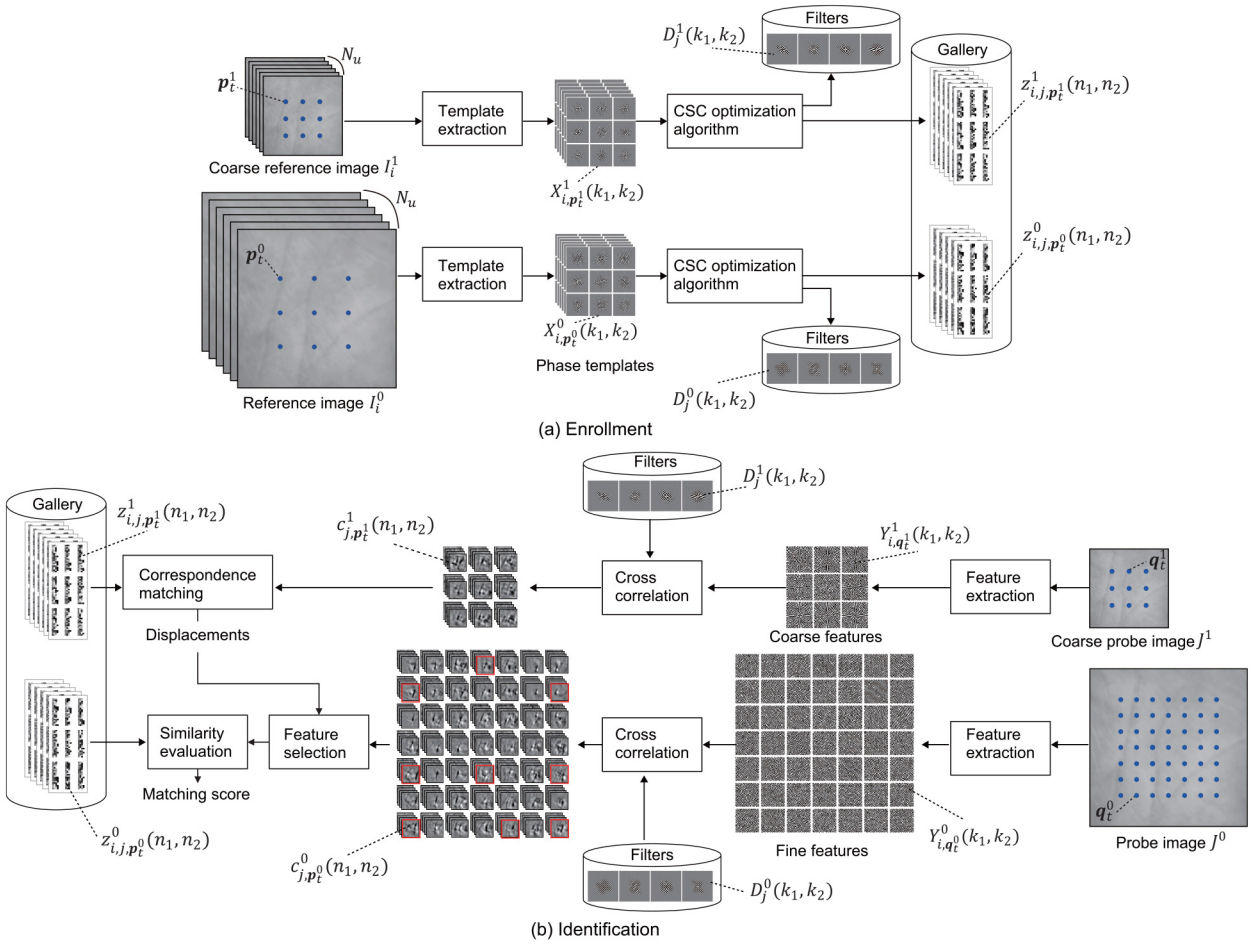


Fig. 3. Overall palmprint identification using two-layer Phase-Based Hierarchical Block Matching with Convolutional Sparse Coding (PB-HBM-CSC): (a) Enrolment for PB-HBM-CSC and (b) Palmprint identification using PB-HBM-CSC.

**Algorithm 7** Enrolment Procedure for PB-HBM

**Require:** $N_u$ reference images $I_i$
**Ensure:** $2N_uN_b$ phase templates $X^l_{i,\boldsymbol{p}^l_t}(k_1,k_2)$
  **for** $i \in \{1,\cdots,N_u\}$) **do**
    Generate an image pyramid, $I^1_i$ and $I^0_i(=I_i)$, by Eq. (23)
    **for** $l \in \{0,1\}$ **do**
      **for** $t \in \{1,\cdots,N_b\}$ **do**
        Obtain a reference block location $\boldsymbol{p}^l_t$ on $I^l_i$by Eq. (25)
        Compute the local phase template $X^l_{i,\boldsymbol{p}^l_t}(k_1,k_2)$ by Eq. (3)
        Store the phase template into the gallery
      **end for**
    **end for**
  **end for**

reduce the computation time without degrading the identification accuracy of PB-HBM in the identification task. Fig. 3 illustrates the overall process of PB-HBM-CSC. Fig. 3 (a) illustrates the enrolment process where we store the sparse representation of phase templates into the gallery. During enrolment, first, we extract all the phase templates $X^l_{i,\boldsymbol{p}^l_t}(k_1,k_2)$. Next, for each layer, we apply the CSC optimization algorithm described in Sect. IV to the phase templates of that layer. Then, we store the obtained convolution filters $D^l_j(k_1,k_2)$ and sparse codes $z^l_{i,j,\boldsymbol{p}^l_t}(n_1,n_2)$ into the gallery. A detailed description of the enrolment process is shown in Algorithm 9. Fig. 3 (b) illustrates the identification process. At the beginning of this process, we extract the phase features $Y^l_{\boldsymbol{q}}(k_1,k_2)$ and compute their atomic correlation functions $c^l_{j,\boldsymbol{q}}(k_1,k_2)$ for all $\boldsymbol{q}$ in $Q^l_{\text{cand}}$. Then, we compute the CSC-BLPOC functions between atomic correlation functions and sparse codes. A detailed description of the identification process is shown in Algorithm 10. By applying CSC-BLPOC to PB-HBM, we reduce the number of 2D-IDFTs required for identification. While PB-HBM requires $2N_uN_b$ 2D-IDFTs for the computation of BLPOC functions, PB-HBM-CSC only requires $N_d(N_b + |Q_{\text{cand}}|)$ (where $N_d \ll N_u$) for the computation of atomic correlation functions.

## VI. Experiments and Discussion

This section presents an experimental evaluation of the proposed and conventional methods of palmprint identification and clarifies the major advantages of the proposed method.

**Algorithm 8** Identification Procedure for PB-HBM

**Require:** A probe image $J$
**Ensure:** Matching scores for reference images $I_i$
  Generate an image pyramid, $J^1$ and $J^0(=J)$, by Eq. (24)
  **for** $l \in \{0, 1\}$ **do**
    **for** each candidate block location $\boldsymbol{q}(\in Q^l_{\text{cand}})$ on $J^l$ **do**
      Compute the phase feature $Y^l_{\boldsymbol{q}}(k_1, k_2)$ by Eq. (4)
    **end for**
  **end for**
  **for** $i \in \{1, \cdots, N_u\}$ **do**
    **for** $t \in \{1, \cdots, N_b\}$ **do**
      Obtain $\boldsymbol{p}^0_t$ and $\boldsymbol{p}^1_t$ by Eq. (25)
      Find the corresponding block location $\boldsymbol{q}^1_t$ on $J^1$ be equal to $\boldsymbol{p}^1_t$ on $I^1_i$ by Eq. (27)
      Compute the BLPOC function for the block pair $(\boldsymbol{p}^1_t, \boldsymbol{q}^1_t)$ to derive $\boldsymbol{\delta}^1_t$, where we use Eq. (5) and (7) with $X^l_{i,\boldsymbol{p}^1_t}(k_1, k_2)$ stored in the gallery and the $Y^l_{\boldsymbol{q}^1_t}(k_1, k_2)$ prepared above
      Determine $\boldsymbol{q}^0_t$ from $\boldsymbol{q}^1_t$ and $\boldsymbol{\delta}^1_t$ using Eq. (26)
      Compute the BLPOC function for the block pair $(\boldsymbol{p}^0_t, \boldsymbol{q}^0_t)$ to derive the peak value $\alpha^0_t$, where we use Eq. (5) and (7) with $X^0_{i,\boldsymbol{p}^0_t}(k_1, k_2)$ and $Y^0_{\boldsymbol{q}^0_t}(k_1, k_2)$
    **end for**
    Compute the matching score between $I_i$ and $J$ as an average of $\alpha^0_t$ for $t = 1, \cdots, N_b$
  **end for**

**Algorithm 9** Enrolment Procedure for PB-HBM-CSC

**Require:** $N_u$ reference images $I_i$
**Ensure:** $D^l_j(k_1, k_2)$ and $z^l_{i,j,\boldsymbol{p}^l_t}(n_1, n_2)$
  Compute the local phase templates $X^l_{i,\boldsymbol{p}^l_t}(k_1, k_2)$ following the enrolment procedure for PB-HBM
  **for** $l \in \{0, 1\}$ **do**
    Compute sparse code $z^l_{i,j,\boldsymbol{p}^l_t}(n_1, n_2)$ and convolution filters $D^l_j(k_1, k_2)$ by applying the CSC optimzation algorithm in Sect. IV to the phase templates $X^l_{i,\boldsymbol{p}^l_t}(k_1, k_2)$
    Store convolution filters and sparse codes into the gallery
  **end for**

**Algorithm 10** Identification Procedure for PB-HBM-CSC

**Require:** A probe image $J$
**Ensure:** Matching scores for reference images $I_i$
  Generate an image pyramid, $J^1$ and $J^0(=J)$, by Eq. (24)
  **for** $l \in \{0, 1\}$ **do**
    **for** each candidate block location $\boldsymbol{q}(\in Q^l_{cand})$ on $J^l$ **do**
      Compute the phase feature $Y_{\boldsymbol{q}}(k_1, k_2)$ by Eq. (4)
      **for** each convolution filter $D^l_j$ **do**
        Compute the atomic correlation functions $c_{j,\boldsymbol{q}}(k_1, k_2)$ using $Y_{\boldsymbol{q}}(k_1, k_2)$ by Eq. (17)
      **end for**
    **end for**
  **end for**
  **for** $i \in \{1, \cdots, N_u\}$ **do**
    **for** $t \in \{1, \cdots, N_b\}$ **do**
      Obtain $\boldsymbol{p}^0_t$ and $\boldsymbol{p}^1_t$ by Eq. (25)
      Find the corresponding block location $\boldsymbol{q}^1_t$ on $J^1$ be equal to $\boldsymbol{p}^1_t$ on $I^1_i$ by Eq. (27)
      Compute the CSC-BLPOC function for the block pair $(\boldsymbol{p}^1_t, \boldsymbol{q}^1_t)$ to derive $\boldsymbol{\delta}^1_t$, where we use Eq. (18) with $z^1_{i,j,\boldsymbol{p}^1_t}(n_1, n_2)$ stored in the gallery and the $c^1_{j,\boldsymbol{q}^1_t(k_1,k_2)}$ prepared above
      Determine $\boldsymbol{q}^0_t$ from $\boldsymbol{q}^1_t$ and $\boldsymbol{\delta}^1_t$ by Eq. (26)
      Compute the CSC-BLPOC function for the block pair $(\boldsymbol{p}^0_t, \boldsymbol{q}^0_t)$ to derive the peak value $\hat{\alpha}^0_t$, where we use Eq. (18) with $z^0_{i,j,\boldsymbol{p}^0_t}(n_1, n_2)$ and $c^0_{j,\boldsymbol{q}^0_t(k_1,k_2)}$
    **end for**
    Compute the matching score between $I_i$ and $J$ as an average of $\hat{\alpha}^0_t$ for $t = 1, \cdots, N_b$
  **end for**

### A. Methods

In the experiments, we implement the following 14 methods and evaluate their performance. As sub-space-based methods, we selected (i) *LDA* [10], (ii) *RP* [11], and DPA [12]. Note that we implemented (iii) Single-source DPA (*SDPA*) since only the left palmprints are used in the experiments. As conventional coding-based methods, we selected (iv) Competitive Code (*CompCode*) [15], (v) Ordinal Code (*OrdiCode*) [17], (vi) Discriminative and Robust Competitive Code (*DR_CC*) [61], (vii) Collaborative Representation Competitive Code (*CR_CC*) [38], (viii) *PalmHash* [62], and (ix) *PalmPhasor* [62], which have good recognition accuracy and high speed, essential for the identification task. Note that the method (vii) requires to register multiple samples for each enrolled user, i.e., 4 samples. As for CNN-based methods, there are few methods whose implementations are publicly available, or even if they are, the training process does not proceed well. We selected (x) Modified AlexNet (*mAlexNet*) [23] and (xi) Modified MobileNet (*mMobileNet*) [21], which employ the famous CNN model for palmprint recognition. Zao *et al.* [23] employs a modified version of AlexNet [24] trained to extract features that are suitable for recognition in a single spectral band. Dong *et al.* [21] proposed a modified version of MobileNet [22] for palmprint recognition. These CNNs can be trained with a limited number of training images. In addition to the above CNN-based methods, we refer to the results of 6 state-of-the-art methods: PalmNet [25], JCLSR [27], CompNet [29], SMHNet [18], DDH [30], and LCDDR [28]. Note that we refer only to results for which the experimental conditions are almost the same. (xii) Phase-Based Correspondence Matching (*PB-CM*) [33] can be regarded as a baseline for the methods (xiii) and (xiv). (xiii) Phase-Based Hierarchical Block Matching (*PB-HBM*) described in Sect. V-B is a modified version of method (xii), where correspondence matching is carried out on a predetermined set of candidate blocks with a fixed location. (xiv) Phase-Based Hierarchical Block Matching with Convolutional Sparse Coding (*PB-HBM-CSC*) described in Sect. V-C reduces the computational complexity of method (xiii) by introducing CSC for phase template representation.

### B. Palmprint Databases

We employ three public databases for our experiments: (A) the PolyU palmprint database,[1] (B) the CASIA palmprint database,[2] and (C) the Tongji palmprint database [38].

*(A) PoyU database:* The images in PolyU contain minute displacements and deformations due to the fixed hand in image acquisition. This database consists of 385 palms acquired in two sessions, with at least 8 images acquired from each palm in each session. For all the methods except the CNN methods, the first image of the first session (385 images) is used for enrollment and the rest of the images except for the first image of the second session are used for test. For the CNN methods, the first images of the first and second sessions are used for validation, the rest of the images from the first session are used for training, and the rest of the images from the second session are used for test.

*(B) CASIA database:* The images in CASIA contain various types of deformations due to the contactless image acquisition. This database contains 600 palms, with a maximum of 10 images per palm. For all the methods except the CNN methods, the first image (600 images) is used for enrollment and the second through fifth images are used for test. For the CNN methods, the first and sixth images are used for validation, the second through fifth images are used for test, and the rest of the images are used for training. Note that the weights of CNNs pretrained using PolyU are used as initial values for training on CASIA.

*(C) Tongji database:* The images in Tongji contain various types of deformations due to the contactless image acquisition as in the CASIA database. This database consists of 600 palms acquired in two sessions, with 10 images acquired from each palm in each session. The first image of the second session is used for enrollment and the rest for the test of all the methods. The methods (i), (iii), (vii) cannot be evaluated on Tongji, since these methods require multiple training samples per user. The methods (x) and (xi) are pretrained on PolyU as well as CASIA.

As mentioned in Section II-A, we extract and normalize the regions of interest from these palmprint images using the preprocessing method described in [14], where the image size after preprocessing is $160 \times 160$ pixels. In Tongji, the preprocessed images of $128 \times 128$ pixels are available. We use the preprocessed images in this experiment to guarantee reproducibility. Therefore, the number of reference and candidate blocks in Tongji differs from other databases due to the small image size. In all three databases, the right palmprint image is flipped in order to consider all images as left palmprint images.

### C. Experimental Conditions

In (i) LDA, (ii) RP, and (iii) SDPA, we use feature vectors with 40 elements. Table II summarizes the parameters for the method (iv)–(ix). (x) mAlexNet is trained using the SGD algorithm as in [23]: 100 epochs and a batch size of 50 images. The learning rate, weight decay, and momentum are set to 0.015, 1e-1, and 0.5, respectively. (xi) mMobileNet is also trained using the SGD algorithm with variable learning rate as in [21]: 100 epochs and a batch size of 30 images. The initial learning rate, weight decay, and momentum are set to 0.001, 0.0005, and 0.9, respectively. Both CNNs use $128 \times 128$-pixel images as input, and random cropping is used for data augmentation during training. Instead of the softmax layer, we use features from the last dense layer with Collaborative Representation Classification (CRC) [63] in recognition. This modification improves the recognition performance and does not significantly affect the identification time of CNNs. Table III summarizes the parameters for the phase-based methods (vii)–(ix), which are obtained empirically. We also improve the recognition accuracy of methods (xii)–(xiv) by removing blocks without texture in the matching process. For this purpose, we use the average amplitude of the image blocks, $\nu_F$, to determine the blocks without texture, which is calculated by $\nu_F = \frac{1}{B_1 B_2} \sum_{k_1,k_2}{}' |F(k_1, k_2)|^2$. If $\nu_F$ is

TABLE II
PARAMETERS FOR GABOR-CODING BASED METHODS

(iv) CompCode, (vi) DR_CC, (viii) PalmHash, (ix) PalmPhasor

| Parameter | Value |
|---|---|
| Number of orientations | 6 |
| Radial frequency $\mu$ | 0.0916 radians per pixel |
| Elliptical Gaussian envelope with standard deviations $(\sigma, \beta)$ | $(2.8090, 5.6179)$ |

(v) OrdiCode

| Parameter | Value |
|---|---|
| Number of orientations | 3 |
| Filter's horizontal scale $\delta_x$ | 0.9363 |
| Filter's vertical scale $\delta_y$ | 2.8090 |

(vii) CR_CC

| Parameter | Value |
|---|---|
| Patch size | 14 |
| $(\lambda, \sigma_{\text{Gabor}}, \mu)$ | $(1.35, 4.85, 14.1)$ |

TABLE III
PARAMETERS FOR PHASE-BASED METHODS

(xii) PB-CM, (xiii) PB-HBM and (xiv) PB-HBM-CSC

| Parameter | Value |
|---|---|
| Number of reference blocks $N_b$ | 25 for CASIA and PolyU databases<br>16 for Tongji database |
| Spacing between adjacent reference blocks | 11 pixels |
| Block image size $L_1 \times L_2$ | $48 \times 48$ pixels |
| Phase template/feature size $B_1 \times B_2$ | $32 \times 32$ coefficients |
| Spectral weighting function $H(k_1, k_2)$ is a Gauss function with variance $\sigma^2$ | $\sigma^2 = 0.7$ |
| Search-range for the BLPOC function $(M_1, M_2)$ | $(8, 8)$ for the coarse resolution<br>$(5, 5)$ for the original resolution |

(xiii) PB-HBM and (xiv) PB-HBM-CSC

| Parameter | Value |
|---|---|
| Number of candidate blocks $\lvert Q_{\text{cand}} \rvert$ | 196 for PolyU and CASIA databases<br>144 for Tongji database |
| Spacing between adjacent candidate blocks | 8 pixels |

(xiv) PB-HBM-CSC

| Parameter | Value |
|---|---|
| Number of filters $N_d$ for CSC | 16 |
| Number of coefficients $N_c$ for CSC | 32 |
| $\epsilon$ | $1/(N_c' + 1)$ |

lower than $thr_{AC} = 0.04$, we do not use the image block in correspondence matching as it has no texture.

We evaluate the performance of the above methods using the average computation time excluding preprocessing time and the rank-1 identification accuracy. We implemented all the methods using Python 3 and performed the experiments on a computer with an Intel Xeon E5450 3.00 GHz, 32 GB and Ubuntu 14.04.5. Methods (x) and (xi) are implemented using TensorFlow with Keras.

### *D. Results and Discussion*

Table IV summarizes the experimental results of identification accuracy and computation time. First, we discuss the experimental results of identification accuracy. In PolyU, most methods, except for sub-space-based methods, achieved identification accuracy of over 99% since palm images are taken with the hand fixed. In particular, (vi) DR_CC and phase-based matching (xii)–(xiv) exhibit 100% identification accuracy. (v) mAlexNet and (vi) mMobileNet have low identification accuracy due to lack of training data. In CASIA, the identification accuracy of coding-based methods (iv)–(vii) is lower than that of PolyU since these methods could not deal with displacements and deformations. CNN-based methods such as PalmNet [25], JCLSR [27], and LCDDR [28] can handle such displacements and deformations, and thus achieve high identification accuracy, while (v) mAlexNet and (vi) mMobileNet have low identification accuracy due to lack of training data. The phase-based matching methods (xii)–(xiv) have higher identification accuracy than the latest CNN-based methods. In Tongji, the images are taken in a contactless situation as in CASIA, however, most methods achieve high identification accuracy due to the small tilt and rotation of the hand. Among all the methods, (x) mAlexNet and (xi) mMobileNet show low identification accuracy since the training and test images are from different databases. The identification accuracy of CompNet [29], (xii) PB-CM, and (xiii) PB-HBM is 100%, and (xiv) PB-HBM-CSC achieves the next highest accuracy. These results demonstrate the effectiveness of phase-based matching (xii)–(xiv) in palmprint identification in terms of identification accuracy. Next, we discuss the experimental results of computation time. (iii) SDPA is the shortest identification time because of its quite short one comparison. The coding-based methods also have short identification time since both feature extraction and one comparison are fast. The CNN-based methods have short identification time since they evaluate the computation time on GPU. (vii) PB-CM has the highest identification accuracy but the longest identification time due to the time required for feature extraction. By introducing HBM, a significant reduction in computation time is achieved, although the overall identification time is longer than that of coding-based methods. With further introduction of CSC, the overall identification time of PB-HBM-CSC (xiv) is comparable to that of the coding-based methods since one comparison is significantly faster. Our future work is to implement PB-HBM-CSC on GPUs to achieve speedups comparable to CNN-based methods.

TABLE IV
EXPERIMENTAL RESULTS OF IDENTIFICATION ACCURACY (ACC.), THE COMPUTATION TIME OF FEATURE EXTRACTION OF ONE SAMPLE (FEAT. EXT.) AND ONE COMPARISON (ONE COMP.), AND THE TOTAL IDENTIFICATION TIME (TOTAL). "*" INDICATES THAT THE EXPERIMENTAL RESULTS WERE REFERRED FROM THE PAPER. NOTE THAT "—" INDICATES THAT THE COMPUTATION TIME IS NOT GIVEN

(A) PolyU database (385 enrolled palmprint images)

| Method | Acc. [%] | Feart.Ext. | One Comp. | Total |
|---|---|---|---|---|
| (i) LDA [10] | 97.542 | 13.942ms | 0.268ms | 0.117s |
| (ii) RP [11] | 96.831 | 35.407ms | 0.550ms | 0.247s |
| (iii) SDPA [12] | 93.726 | 3.076ms | 0.003ms | 0.004s |
| (iv) CompCode [15] | 99.935 | 6.79ms | 1.42ms | 0.553s |
| (v) OrdiCode [17] | 99.224 | 1.98ms | 0.85ms | 0.329s |
| (vi) DR_CC [61] | 100 | 462ms | 0.55ms | 0.674s |
| (vii) CR_CC [38] | 99.418 | 17.24ms | 22.40$\mu$s | 0.026s |
| (viii) PalmHash [62] | 99.192 | 24.697ms | 0.912ms | 0.376s |
| (ix) PalmPhasor [62] | 98.221 | 134.179ms | 0.938ms | 0.496s |
| (x) mAlexNet [23] | 87.904 | 140ms | 0.09ms | 0.175s |
| (xi) mMobileNet [21] | 95.149 | 198ms | 0.09ms | 0.233s |
| *LCDDR [28] | 99.890 | — | — | — |
| (xii) PB-CM [33] | 100 | 6,594ms | 3.57ms | 7.968s |
| (Xiii) PB-HBM | 100 | 165ms | 3.89ms | 1.663s |
| (xiv) PB-HBM-CSC | 100 | 228ms | 1.87ms | 0.948s |

(B) CASIA database (600 enrolled palmprint images)

| Method | Acc. [%] | Feart.Ext. | One Comp. | Total |
|---|---|---|---|---|
| (i) LDA [10] | 85.708 | 13.104ms | 0.268ms | 0.174s |
| (ii) RP [11] | 86.583 | 28.779ms | 0.549ms | 0.358s |
| (iii) SDPA [12] | 91.542 | 3.044ms | 0.003ms | 0.005s |
| (iv) CompCode [15] | 96.458 | 7.02ms | 1.42ms | 0.859s |
| (v) OrdiCode [17] | 94.583 | 2.01ms | 1.41ms | 0.848s |
| (vi) DR_CC [61] | 97.208 | 470ms | 0.60ms | 0.830s |
| (vii) CR_CC [38] | 97.083 | 17.81ms | 23.50$\mu$s | 0.032s |
| (viii) PalmHash [62] | 91.333 | 25.565ms | 0.912ms | 0.573s |
| (ix) PalmPhasor [62] | 85.292 | 137.667ms | 0.939ms | 0.701s |
| (x) mAlexNet [23] | 84.375 | 140ms | 0.09ms | 0.194s |
| (xi) mMobileNet [21] | 87.459 | 198ms | 0.09ms | 0.252s |
| *PalmNet-G [25] | 99.680 | 160ms | 10ms | 6.160s |
| *PalmNet-PCA [25] | 99.680 | 160ms | 10ms | 6.160s |
| *JCLSR [27] | 98.940 | 33.055ms | 0.131ms | 0.112s |
| *LCDDR [28] | 99.180 | — | — | — |
| (xii) PB-CM [33] | 99.958 | 10,143ms | 3.78ms | 12.411s |
| (xiii) PB-HBM | 99.917 | 167ms | 4.27ms | 2.729s |
| (xiv) PB-HBM-CSC | 99.917 | 227ms | 1.59ms | 1.181s |

(C) Tongji database (600 enrolled palmprint images)

| Method | Acc. [%] | Feart.Ext. | One Comp. | Total |
|---|---|---|---|---|
| (ii) RP [11] | 72.895 | 28.524ms | 0.547ms | 0.357s |
| (iv) CompCode [15] | 99.816 | 7.27ms | 1.42ms | 0.859s |
| (v) OrdiCode [17] | 99.596 | 2.10ms | 0.86ms | 0.518s |
| (vi) DR_CC [61] | 99.842 | 459ms | 0.34ms | 0.663s |
| (viii) PalmHash [62] | 96.175 | 51.443ms | 1.081ms | 0.700s |
| (ix) PalmPhasor [62] | 91.193 | 132.685ms | 0.922ms | 0.686s |
| (x) mAlexNet [23] | 49.825 | 140ms | 0.09ms | 0.194s |
| (xi) mMobileNet [21] | 68.229 | 199ms | 0.09ms | 0.253s |
| *PalmNet-G [25] | 99.800 | 160ms | 10ms | 6.160s |
| *PalmNet-PCA [25] | 99.830 | 160ms | 10ms | 6.160s |
| *CompNet [29] | 100 | — | — | — |
| *SMHNet [18] | 91.330 | — | — | — |
| *DDH [30] w/o KD | 97.850 | 3.33ms | 0.012ms | 0.011s |
| *DDH [30] w/ KD | 98.920 | 3.33ms | 0.012ms | 0.011s |
| (xii) PB-CM [33] | 100 | 6,579ms | 2.50ms | 8.079s |
| (xiii) PB-HBM | 100 | 118ms | 2.61ms | 1.684s |
| (xiv) PB-HBM-CSC | 99.991 | 163ms | 0.98ms | 0.751s |

Fig. 4 shows the Cumulative Match Characteristic (CMC) curve obtained from the experiment using CASIA, where the CMC curve indicates the probability of observing the correct ID within the top-$n_{\text{rank}}$ ranks. Among the conventional methods, DR_CC has the highest rank-$n$ identification accuracy, followed by CR_CC. The CNN-based methods, mALexNet and mMobileNet, achieved considerably lower identification accuracy than the code-based methods. None of the conventional methods reached the identification accuracy of the

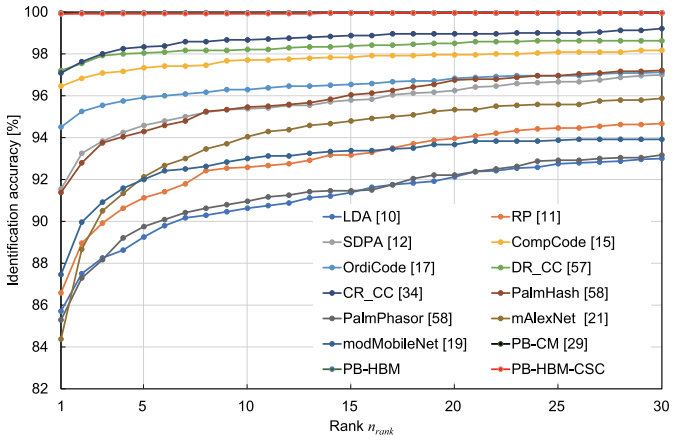


Fig. 4. CMC curve evaluated using CASIA for all methods.

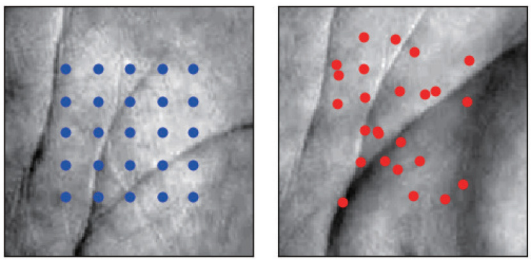

Fig. 5. Failure case of PB-HBM-CSC for the genuine pair in CASIA. Blue dots indicate the center of the reference blocks and red dots indicate the estimated center of the corresponding blocks.

TABLE V
COMPARISON OF ENROLLED DATA SIZE PER IMAGE FOR EACH METHOD

| Method | Bit length of elements | # of elements | Total size [bytes] |
|---|---|---|---|
| (i) LDA [10] | 32 | 40 | 160 |
| (ii) RP [11] | 32 | 40 | 160 |
| (iii) SDPA [12] | 32 | 40 | 160 |
| (iv) CompCode [15] | 3 | $32 \times 32$ | 384 |
| (v) OrdiCode [17] | 3 | $32 \times 32$ | 384 |
| (vi) DR_CC [61] | $3+1$ | $32 \times 32$ | 512 |
| (vii) CR_CC [38] | 32 | $9 \times 9 \times 6$ | 1,944 |
| (viii) PalmHash [62] | 6 | 1,024 | 6,144 |
| (ix) PalmPhasor [62] | 6 | 1,024 | 6,144 |
| (x) mAlexNet [23] | 32 | 1,000 | 4,000 |
| (xi) mMobileNet [21] | 32 | 1,024 | 4,096 |
| (xii) PB-CM [33] | 4 | $32 \times 16 \times 25 \times 2$ | 12,800 |
| (xiii) PB-HBM | 4 | $32 \times 16 \times 25 \times 2$ | 12,800 |
| (xiv) PB-HBM-CSC | $32+16$ | $32 \times 25 \times 2$ | 9,600 |

phase-based methods within the top 30. The accuracy of PB-HBM-CSC is comparable to that of other phase-based methods, and the genuine pair is among the top five in all the phase-based methods. Fig. 5 shows a failure case of PB-HBM-CSC for the genuine pair in CASIA. The reason is that PB-HBM-CSC could not find the corresponding points due to the significant deformation and large global displacement between the two images.

Table V compares the data size required to enroll the reference image in the gallery for each method. In the phase-based methods (xii)–(xii), the reference image is enrolled as $N_b \times$ (# of layers) $= 25 \times 2$ templates. In the case of PB-HBM and PB-CM, we employ 4-bit quantization of the phase component to improve the storage efficiency [34]. In the case of PB-HBM-CSC, the non-zero elements of the sparse code are stored as 6 bytes of data consisting of 32 bits of the coefficient value and 16 bits of its position and filter index. The data size required to enroll a reference image for PB-HBM-CSC is smaller than that required for PB-HBM and PB-CM. Although this data size is still large compared to conventional methods, an embedded system that can accept a data size of 10 Mbytes can enroll 1,000 users.

## VII. CONCLUSION

In this paper, we focused on the high discriminative capability of palmprint recognition by phase-based matching and reduced the computation time required for the identification task without sacrificing its discriminative capability. We proposed a compact representation of phase features using CSC and constructed a palmprint identification algorithm that provides both high identification accuracy and high identification speed by combining hierarchical block matching with its representation. The proposed method achieves the same level of identification time as the coding-based methods and a higher identification accuracy than the state-of-the-art CNN-based methods. The reduction of computation time using CSC is not limited to phase-based methods, but can be easily extended to correlation pattern recognition [44], [45], [46], [64], [65]. Another interesting point is that the use of CSC results in a multi-layer architecture like CNNs, however, it does not require a large number of training samples like CNNs. In the future, we plan to extend the identification framework of the proposed method to other biometric traits, such as face, finger-knuckle-print periocular, and iris.

## ACKNOWLEDGMENT

The authors would like to thank Dr. Shoichiro Aoyama for his contributions to this research.

## REFERENCES

[1] A. Jain, P. Flynn, and A. Ross, *Handbook of Biometrics*. New York, NY, USA: Springer, 2008. [Online]. Available: https://link.springer.com/book/10.1007/978-0-387-71041-9
[2] D. Zhang, *Palmprint Authentication*. Norwell, MA, USA: Kluwer, 2004.
[3] A. Kong, D. Zhang, and M. Kamel, "A survey of palmprint recognition," *Pattern Recognit.*, vol. 42, no. 7, pp. 1408–1418, Jan. 2009.
[4] D. Zhang, W. Zuo, and F. Yue, "A comparative study of palmprint recognition algorithms," *ACM Comput. Surveys*, vol. 44, no. 1, pp. 1–2, Jan. 2012.
[5] D. Zhong, X. Du, and K. Zhong, "Decade progress of palmprint recognition: A brief survey," *Neurocomputing*, vol. 328, pp. 16–28, Feb. 2019.
[6] G. Lu, D. Zhang, and K. Wang, "Palmprint recognition using eigenpalms features," *Pattern Recognit. Lett.*, vol. 24, nos. 9–10, pp. 1463–1467, Jun. 2003.
[7] T. Connie, A. Teoh, M. Goh, and D. Ngo, "Palmprint recognition with PCA and ICA," in *Proc. Image Vis. Comput.*, Nov. 2003, pp. 227–232.
[8] M. Wang and Q. Ruan, "Palmprint recognition based on two-dimensional methods," in *Proc. Int. Conf. Signal Process.*, Nov. 2006, pp. 427–432.
[9] D.-S. Huang, K.-H. Jo, Y.-Q. Zhou, and K. Han, *Intelligent Computing Theories and Technology*. Berlin, Germany: Springer, 2013. [Online]. Available: https://link.springer.com/book/10.1007/978-3-642-39482-9

[10] W.-S. Zheng, J. H. Lai, and S. Z. Li, "1D-LDA vs. 2D-LDA: When is vector-based linear discriminant analysis better than matrix-based?" *Pattern Recognit.*, vol. 41, no. 7, pp. 2156–2172, Jul. 2008.

[11] L. Leng, J. Zhang, G. Chen, M. Khan, and K. Alghathbar, "Two-directional two-dimensional random projection and its variations for face and palmprint recognition," in *Proc. ICCSA*, vol. 6786, Jun. 2011, pp. 458–470. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-642-21934-4_37

[12] L. Leng and J. Zhang, "PalmHash code vs. PalmPhasor code," *Neurocomputing*, vol. 108, pp. 1–12, May 2013.

[13] J. G. Daugman, "High confidence visual recognition of persons by a test of statistical independence," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 15, no. 11, pp. 1148–1161, Nov. 1993.

[14] D. Zhang, W.-K. Kong, J. You, and M. Wong, "Online palmprint identification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 9, pp. 1041–1050, Sep. 2003.

[15] A. Kong and D. Zhang, "Competitive coding scheme for palmprint verification," in *Proc. Int. Conf. Pattern Recognit.*, vol. 1, Dec. 2004, pp. 520–523.

[16] L. Fei, Y. Xu, W. Tang, and D. Zhang, "Double-orientation code and nonlinear matching scheme for palmprint recognition," *Pattern Recognit.*, vol. 49, pp. 89–101, Jan. 2016.

[17] Z. Sun, T. Tan, Y. Wang, and S. Z. Li, "Ordinal palmprint represention for personal identification," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 1, Jun. 2005, pp. 279–284.

[18] C. Liu, D. Zhong, and H. Shao, "Few-shot palmprint recognition based on similarity metric hashing network," *Neurocomputing*, vol. 456, pp. 540–549, Oct. 2021.

[19] Z. Yang, L. Leng, and W. Min, "Extreme downsampling and joint feature for coding-based palmprint recognition," *IEEE Trans. Instrum. Meas.*, vol. 70, Nov. 2020, Art. no. 5005112.

[20] Z. Yang, L. Leng, T. Wu, M. Li, and J. Chu, "Multi-order texture features for palmprint recognition," *Artif. Intell. Rev.*, vol. 70, pp. 1–17, May 2022.

[21] X. Dong, L. Mei, and J. Zhang, "Palmprint recognition based on deep convolutional neural networks," in *Proc. Int. Conf. Comput. Sci. Intell. Commun.*, Aug. 2018, pp. 82–88.

[22] A. G. Howard *et al.*, "MobileNets: Efficient convolutional neural networks for mobile vision applications," 2017, *arxiv.abs/1704.04861*.

[23] S. Zhao, B. Zhang, and C. P. Chen], "Joint deep convolutional feature representation for hyperspectral palmprint recognition," *Inf. Sci.*, vol. 489, pp. 167–181, Jul. 2019.

[24] A. Krizhevsky, I. Sutskever, and G. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Annu. Conf. Neural Inf. Process. Syst.*, 2012, pp. 1–9.

[25] A. Genovese, V. Piuri, K. N. Plataniotis, and F. Scotti, "PalmNet: Gabor-PCA convolutional networks for touchless palmprint recognition," *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 12, pp. 3160–3174, Dec. 2019.

[26] Y. Liu and A. Kumar, "Contactless palmprint identification using deeply learned residual features," *IEEE Trans. Biometrics, Behav., Identity Sci.*, vol. 2, no. 2, pp. 172–181, Apr. 2020.

[27] S. Zhao and B. Zhang, "Joint constrained least-square regression with deep convolutional feature for palmprint recognition," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 52, no. 1, pp. 511–522, Jan. 2022.

[28] S. Zhao and B. Zhang, "Learning complete and discriminative direction pattern for robust palmprint recognition," *IEEE Trans. Image Process.*, vol. 30, pp. 1001–1014, 2021.

[29] X. Liang, J. Yang, G. Lu, and D. Zhang, "CompNet: Competitive neural network for palmprint recognition using learnable Gabor kernels," *IEEE Signal Process. Lett.*, vol. 28, pp. 1739–1743, 2021.

[30] H. Shao, D. Zhong, and X. Du, "Deep distillation hashing for unconstrained palmprint recognition," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–13, Jan. 2021. [Online]. Available: https://ieeexplore.ieee.org/document/9335244

[31] T. Wu, L. Leng, M. K. Khan, and F. A. Khan, "Palmprint-palmvein fusion recognition based on deep hashing network," *IEEE Access*, vol. 9, pp. 135816–135827, 2021.

[32] H. Xu, L. Leng, Z. Yang, A. B. J. Teoh, and Z. Jin, "Multi-task pre-training with soft biometrics for transfer-learning Palmprint recognition," *Neural Process. Lett.*, to be published.

[33] K. Ito, S. Iitsuka, and T. Aoki, "A palmprint recognition algorithm using phase-based correspondence matching," in *Proc. Int. Conf. Image Process.*, Nov. 2009, pp. 1977–1980.

[34] S. Aoyama, K. Ito, and T. Aoki, "Similarity measure using local phase feature and its application to biometric recognition," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2013, pp. 180–187.

[35] K. Takita, M. A. Muquit, T. Aoki, and T. Higuchi, "A sub-pixel correspondence search technique for computer vision applications," *IEICE Trans. Fundam.*, vol. E87-A, no. 8, pp. 1913–1923, Aug. 2004.

[36] K. Ito, H. Nakajima, K. Kobayashi, T. Aoki, and T. Higuchi, "A fingerprint matching algorithm using phase-only correlation," *IEICE Trans. Fundam.*, vol. E87-A, no. 3, pp. 682–691, Mar. 2004.

[37] J. Mairal, F. Bach, and J. Ponce, *Sparse Modeling for Image and Vision Processing*. New York, NY, USA: Now, 2014.

[38] L. Zhang, L. Li, A. Yang, Y. Shen, and M. Yang, "Towards contactless palmprint recognition: A novel device, a new benchmark, and a collaborative representation based identification approach," *Pattern Recognit.*, vol. 69, pp. 199–212, Sep. 2017.

[39] K. Ito, T. Sato, S. Aoyama, S. Sakai, S. Yusa, and T. Aoki, "Palm region extraction for contactless palmprint recognition," in *Proc. Int. Conf. Biometrics*, May 2015, pp. 334–340.

[40] K. Ito *et al.*, "HandSegNet: Hand segmentation using convolutional neural network for contactless palmprint recognition," *IET Biometrics*, vol. 11, no. 2, pp. 109–123, 2022.

[41] K. Takita, T. Aoki, Y. Sasaki, T. Higuchi, and K. Kobayashi, "High-accuracy subpixel image registration based on phase-only correlation," *IEICE Trans. Fundam.*, vol. E86-A, no. 8, pp. 1925–1934, Aug. 2003.

[42] C. D. Kuglin and D. C. Hines, "The phase correlation image alignment method," in *Proc. Int. Conf. Cybern. Soc.*, 1975, pp. 163–165.

[43] I. Rish and G. Grabarnik, *Sparse Modeling: Theory, Algorithms, and Applications*. Hoboken, NJ, USA: CRC Press, 2014.

[44] M. Savvides, B. Kumar, and P. Khosla, "'Corefaces'—Robust shift invariant PCA based correlation filter for illumination tolerant face recognition," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jul. 2004, pp. 834–841.

[45] A. Rodriguez, V. Boddeti, B. Kumar, and A. Mahalanobis, "Maximum margin correlation filter: A new approach for localization and classification," *IEEE Trans. Image Process.*, vol. 22, no. 2, pp. 631–643, Feb. 2013.

[46] B. Kumar, A. Mahalanobis, and R. Juday, *Correlation Pattern Recognition*. Cambridge, U.K.: Cambridge Univ. Press, 2005.

[47] H. Bristow, A. Eriksson, and S. Lucey, "Fast convolutional sparse coding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 391–398.

[48] B. Kong and C. Fowlkes, "Fast convolutional sparse coding (FCSC)," Dept. Comput. Sci., UC Irvine, Irvine, CA, USA, Rep. TR-2014, 2014.

[49] B. Wohlberg, "Efficient convolutional sparse coding," in *Proc. IEEE Conf. Acoust. Speech Signal Process.*, May 2014, pp. 7173–7177.

[50] H. Bristow and S. Lucey, "Optimization methods for convolutional sparse coding," Jun. 2014, *arxiv.abs/1406.2407*.

[51] F. Heide, W. Heidrich, and G. Wetzstein, "Fast and flexible convolutional sparse coding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 5135–5143.

[52] S. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Trans. Signal Process.*, vol. 41, no. 12, pp. 3397–3415, Dec. 1993.

[53] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, 2010.

[54] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM J. Imag. Sci.*, vol. 2, no. 1, pp. 183–202, 2009.

[55] J. Tropp, "Greed is good: Algorithmic results for sparse approximation," *IEEE Trans. Inf. Theory*, vol. 50, no. 10, pp. 2231–2242, Oct. 2004.

[56] M. Elad, *Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing*. New York, NY, USA: Springer, 2010. [Online]. Available: https://link.springer.com/book/10.1007/978-1-4419-7011-4

[57] S. Aoyama, K. Ito, and T. Aoki, "A finger-knuckle-print recognition algorithm using phase-based local block matching," *Inf. Sci.*, vol. 268, pp. 53–64, Jun. 2014.

[58] D. Kusanagi, S. Aoyama, K. Ito, and T. Aoki, "Multi-finger knuckle recognition from video sequence: Extracting accurate multiple finger knuckle regions," in *Proc. Int. Joint Conf. Biometrics*, Sep. 2014, pp. 1–8.

[59] D. Kusanagi, S. Aoyama, K. Ito, and T. Aoki, "A practical person authentication system using second minor finger knuckles for door security," *IPSJ Trans. Comput. Vis. Appl.*, vol. 9, no. 8, pp. 1–13, Mar. 2017.

[60] K. Ito, T. Aoki, T. Hosoi, and K. Kobayashi, "Face recognition using phase-based correspondence matching," in *Proc. IEEE Conf. Autom. Face Gesture Recognit.*, Mar. 2011, pp. 173–178.

[61] Y. Xu, L. Fei, J. Wen, and D. Zhang, "Discriminative and robust competitive code for palmprint recognition," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 48, no. 2, pp. 232–241, Feb. 2018.
[62] L. Leng, M. Li, C. Kim, and X. Bi, "Dual-source discrimination power analysis for multi-instance contactless palmprint recognition," *Multimedia Tools Appl.*, vol. 76, no. 1, pp. 333–354, 2017.
[63] L. Zhang, M. Yang, and X. Feng, "Sparse representation or collaborative representation: Which helps face recognition?" in *Proc. Int. Conf. Comput. Vis.*, Nov. 2011, pp. 471–478.
[64] C. Xie, M. Savvides, and B. VijayaKumar, "Kernel correlation filter based redundant class-dependence feature analysis (KCFA) on FRGC2.0 data," in *Proc. AMFG*, vol. 3723, 2005, pp. 32–43. [Online]. Available: https://link.springer.com/chapter/10.1007/11564386_4
[65] F. Juefei-Xu and M. Savvides, "Subspace-based discrete transform encoded local binary patterns representations for robust periocular matching on NIST's face recognition grand challenge," *IEEE Trans. Image Process.*, vol. 23, no. 8, pp. 3490–3505, Aug. 2014.

**Luis Rafael Marval-Pérez** received the B.E. degree in electronic engineering from Simon Bolivar University, Venezuela, in 2006, and the M.S. degree and the Ph.D. degree in information sciences from Tohoku University, Sendai, Japan, in 2012 and 2019, respectively. He is currently working as a Machine Learning Engineer with LIONS Data, Inc. His research interests include pattern recognition, sparse modeling, machine learning, and computer vision.

**Koichi Ito** (Member, IEEE) received the B.E. degree in electronic engineering and the M.S. and Ph.D. degrees in information sciences from Tohoku University, Sendai, Japan, in 2000, 2002, and 2005, respectively, where he is currently an Associate Professor with the Graduate School of Information Sciences. From 2004 to 2005, he was a Research Fellow with the Japan Society for the Promotion of Science. His research interests include signal and image processing, computer vision, and biometric authentication.

**Takafumi Aoki** (Senior Member, IEEE) received the B.E., M.E., and D.E. degrees in electronic engineering from Tohoku University, Sendai, Japan, in 1988, 1990, and 1992, respectively, where he is currently a Professor with the Graduate School of Information Sciences. Since April 2018, he has also been serving as the Executive Vice President of Tohoku University. His research interests include theoretical aspects of computation, computer design and organization, LSI systems for embedded applications, digital signal processing, computer vision, image processing, biometric authentication, and security issues in computer systems. He received more than 20 academic awards as well as distinguished service awards for his contributions to victim identification in the 2011 Great East Japan Disaster.