



A robust multispectral palmprint matching algorithm and its evaluation for FPGA applications

Chao Li^{*,a}, Yannick Benezeth^b, Keisuke Nakamura^c, Randy Gomez^c, Fan Yang^b

^a State Key Laboratory of Acoustics, Institute of Acoustics, Chinese Academy of Sciences, Beijing, China

^b LE21 FRE2005 CNRS, Arts et Métiers, Univ. Bourgogne Franche-Comté, Dijon F-21000, France

^c Honda Research Institute Japan Co., Ltd., 8-1 Honcho, Wako-shi, Saitama, Japan

ARTICLE INFO

Keywords:

Multispectral biometric authentication
Real-time image processing
Partial least square regression
Embedded system
High-level synthesis

ABSTRACT

Multispectral image modalities can offer high accuracy performance to the biometric systems by improving the discrimination along the spectral dimension. However, its adoptions are usually challenged by low signal to noise ratio, inter band misalignment, high data volume and computational efficiency. This paper presents a fast multispectral palmprint biometric technique using partial least square regression model and score-level fusion, with which an embedded recognition and verification system is implemented for evaluation. Our experiments are conducted using the PolyU palmprint database, and the results demonstrate that the proposed method can achieve a higher accuracy and a lower running cost compared to the reference implementations dedicated to the real-time and/or embedded applications.

1. Introduction

In the past decades, biometric technologies have been widely used in personal authentication applications, such as access control [1,2], e-banking [3], vehicle personalization [4], etc. Up to present, various reliable biometric traits have been utilized and evaluated, including fingerprint, iris, palmprint, finger-knuckle-print, hand geometry. Comparing to ID card or password, these biometric characteristics are not easy to lose and some of them are hard to be copied, which makes the authentication systems more convenient, effective and secure.

As one of the frequently used biometric features, palmprint has several advantages [5,6]: low-resolution imaging can be employed; low cost capture devices can be used; it is very difficult, if not impossible, to fake a palmprint; the line features of the palmprints are stable, etc. It is for these reasons that palmprint recognition has attracted an increasing amount of attention from researchers. Meanwhile, compared to regular color images, multispectral modalities can provide higher accuracy performance. This is because a palm has different absorption capacity for different wavelengths of light and multispectral images capture more precisely these information, so each band of multispectral images represents particular features of a palm. That allows to obtain multifarious information to improve the distinguishability of palmprint image features.

Recently, many palmprint based multispectral biometric solutions have been developed. For example, Xu and Guo [7] represents the

multispectral palmprint images as quaternion features extracted through the quaternion principal components analysis, and achieve better performance in recognition applications. Xu et al. [8] improve the multispectral palmprint recognition method by using digital shearlet transform and multiclass projection extreme learning machine. Hong et al. [9] develop a hierarchical approach for multispectral palmprint recognition by fusing the block dominant orientation code and block-based histogram of oriented gradient features extracted from different light bands. According to the reported experiment data obtained within the laboratory environment, today's multispectral palmprint biometric algorithms have been able to provide high accuracy performance: average recognition rates of 99.9% for the IITD palm database [10] and 99.56% for the PolyU multispectral palmprint database [8].

In practical, finding a person of interest from a large candidate database is far from easy. It usually requires a high performance hardware platform to manage the candidate database and/or ensure the execution speed of the system. Furthermore, the efficiency performance of the systems reduce sharply with the raising of the candidate database size, until it cannot satisfy the application requirements. For these potential issues, Zhang et al. [6] propose a low-cost multispectral palmprint system that can operate in real time and acquire high-quality images. It provides a high recognition accuracy by fusing the multispectral information at score level. However, the hardware device of this system is based on the CPUs without embeddability. Pudzs et al.

* Correspondence author.

E-mail address: chao.li.1986@ieee.org (C. Li).

[11] prototype a multimodal palm biometric system into an Altera DE2-115 board from Terasic, but it achieves only an EER of 16.65% and a verification speed of 0.8 s per image, which can hardly satisfy the accuracy performance or real-time requirement of the real-life applications. Nikisins et al. [12] recently proposed a touch-less palmprint biometric system capable of real-time accurate person identification. The algorithm of this system is robust and well implemented with a series of hardware optimizations. It is stated that its equal error rate with automatic alignment is 1.82%. Yet, it fuses the palm crease and vein features on a single RGB image in the sensor level in order to keep palm illumination constant and to stream images for the processing at the sensors framerate, potentially preventing the system from benefiting from the multispectral imaging techniques. Consequently, many opportunities still exist to find a greater embedded solution for high accuracy multispectral palmprint biometric.

In this paper, we focus the work on the real-time embedded multispectral palmprint biometric applications. Our work is conducted by the Algorithm-Architecture Adequation (AAA) methodology, introduced by the AOSTE team of INRIA (French National Institute for computer science and applied mathematics) [13]. The key feature of AAA is to rapidly prototype complex real-time embedded applications based on automatic code generation. The concerned algorithm and its hardware architecture are studied simultaneous within a Software/Hardware co-design framework, which allows an embedded implementation optimized both in algorithm and hardware level.

Firstly, we applied this AAA methodology on the algorithm aspect. With multispectral imaging, the information presented at multiple wavelengths is usually consolidated to perform the final decision by using the data fusion technique [14]. Generally, there are four levels of fusion in the multispectral biometric methods: image/pixel level, feature level, matching score level and decision level. For example, Wang et al. [15] fused palmprint and palm vein images by using a novel edge-preserving and contrast-enhancing wavelet fusion method in their personal recognition system. This method provides a high accuracy performance, but the image registration procedure takes 9 s, which hinders it from real-time implementation. Hao et al. [16] evaluated several well-known image-level fusion schemes for multispectral palm images with a data set of 84 images, and then extended their work to a larger database and proposed a new feature-level registration method for image fusion [17]. The results by image- and feature-level fusions are improved, but the required registration procedure is time-consuming [15]. For score-level fusion and decision-level fusion, Ross et al. [18] found that the former works better than the later because match scores contain more pattern information. Furthermore, in the point of view algorithm-architecture adequation, fusing data in score-level often has smaller data intensity, which may potentially improve the performance of the system. For the reasons above, the score level fusion technique is used in our work. That is, the dissimilarities of the band maps of the input multispectral images are measured first within their own bands, and then the results are fused in the score level.

Meanwhile, since the biometric information at different light band contribute differently to the palm recognition or verification, we base the dissimilarity measurement on the partial least squares regression. Partial least squares has received a great amount of attention in the field of chemometrics. The algorithm has become a standard tool for processing a wide spectrum of chemical data problems. The success of partial least squares in chemometrics resulted in a lot of applications in other scientific areas including bioinformatics [19], food research [20], medicine [21], pharmacology [22], social sciences [23], etc. Recently, the regression method is successfully used in a hyperspectral face recognition application in the biometric field [24], which effectively improve the test accuracy by modeling the relations between training and prediction matrices. The use of partial least squares regression in palmprint biometric authentication allows us to weight the components of sample vectors through a statistical analysis, further raising the accuracy of the system.

Multispectral images acquire images at different wavelengths, allowing more color channels than regular color images, resulting a high computation intensity and low speed efficiency. In recent years, embedded systems have made great progress. Many highly effective embedded devices have been made available to engineers at a very convenient price and widely used in various signal processing and communication systems for their significant advantages in terms of running-cost, embeddability, power consumption or flexibility [25–31]. Moreover, many comparative studies indicate that Field Programmable Gate Arrays (FPGAs) can often achieve better comprehensive properties than other platforms in most cases. For example, in the work of Zou et al. [32], the running speed of the FPGA implementation of the Smith–Waterman Algorithm is $3.4 \times$ compared to the Graphics Processing Unit (GPU) and over $40 \times$ compared to the Central Processing Unit (CPU), while Kestur et al. [33] demonstrated that FPGA has similar performance at higher energy efficiency compared to the CPU and GPU platforms. In order to satisfy the processing speed and embeddability constraints of real-life palmprint authentication applications, we select FPGA as the target device for its benefits of high efficiency-cost ratio.

Since conventional FPGA design flow cannot provide a software-friendly environment for the algorithm development and verification, we implement the design through a High-Level Synthesis (HLS) technique [34]. It can effectively accelerate the design cycles by automating the C-to-RTL synthesis, even for the users with fewer register-transfer language programming experiences, as well as improve the maintainability of the design by facilitating the algorithm description. Furthermore, a series of optimizations are made in the C code level to improve the design performances for the purpose of high running speed.

This paper presents our study related to the design, implementation and evaluation of an embedded real-time multispectral palmprint biometric system. Its accuracy performance for recognition and verification is measured using the Region Of Interest (ROI) version of the PolyU multispectral palmprint database [35]. The experiment results demonstrate a very high recognition rate, nearly 100%, as well as a very low average Equal Error Rate (EER), lower than 0.1%. Meanwhile, its running efficiency performance is also estimated through a FPGA based palmprint authentication implementation. The final implementation achieves a recognition speed of 1.37 frames per millisecond (with a multispectral image resized into 13-by-13 and including normalization, feature matching, score fusion and decision), which allows to perform more complex pre- and post- processing in real-time, such as hand image segmentation, gesture analysis, hand language interpretation, etc.

The remainder of this paper is organized as follows: [Section 2](#) describes the proposed palmprint authentication algorithm; [Section 3](#) presents the hardware design and optimization cycles; [Section 4](#) analyzes the experiment results of its hardware embedded design; finally, a conclusion is given in [Section 5](#).

2. Algorithm description

Partial least square (PLS) is a wide class of methods for modeling relations between sets of observed variables by means of latent variables. It comprises of regression and classification tasks as well as dimension reduction techniques and modeling tools. Projections of the observed data to its latent structure by means of PLS was developed by Herman Wold and coworkers [36–38]. The underlying assumption of all PLS methods is that the observed data is generated by a system or process which is driven by a small number of latent (not directly observed or measured) variables. Its goal is to maximize the covariance between the two parts of a paired data set even though those two parts are in different spaces.

This work pre-process the original multispectral images through a 2-D average filter:

$$I_{out}(i, j) = \frac{1}{L^2} \sum_{ii=i-L/2}^{i+L/2} \sum_{jj=j-L/2}^{j+L/2} I_{in}(ii, jj) \quad (1)$$

where I_{in} is one of the frames of the input multispectral image, I_{out} is the output of the filter, and L is the filter size. Next, the filtered images are assigned to the proposed algorithm.

Fig. 1 shows the overall flowchart of the implemented palmprint recognition/verification algorithm using PLS regression. In this case, the input images are organized as N multispectral H -by- W -by- B cube arrays, where H and W refer to the height and width of the images respectively, B is the band number and N is the number of multispectral palmprint images (known also as observed samples). Next, the set of multispectral image arrays are reshaped into the form of $X \in \mathbb{R}^{N \times D \times B}$ with $D = H \times W$. Let $X_b \in \mathbb{R}^{N \times D}$ be the b th frame of X in the B direction, so the rows and columns of X_b respectively correspond to the samples and vectorized reflectance map (feature variable vector) within the b th band.

In our case, every band of the input multispectral images has an independent matching process and scores are fused in order to perform the final decision. The matching process is based on the Partial Least Squares (PLS) regression, which necessitates a testing and a training datasets, we therefore treat every sub matrix X_b as a testing matrix assigned to the concerned matching channel. We define the combination of each regression model and its pre- and post-process cycles as a matching channel. Within each channel, the input matrix X_b is first normalised:

$$\bar{X}_b = \frac{X_b - \mu_b}{\sigma_b} \quad (2)$$

where $\bar{X}_b \in \mathbb{R}^{N \times D}$ is the normalized input matrix in the b th band, μ_b and σ_b are the mean and standard deviation of the training matrix $X_{train,b} \in \mathbb{R}^{N_t \times D}$, where N_t is the number of training samples. Finally, different regression coefficients and decision strategies are used depending on the intended applications (recognition or verification).

2.1. Training process of PLS regression

In the case of this paper, we project the two normalisation matrices ($\bar{X}_{train,b}$ and \bar{Y}_r) of $X_{train,b}$ and its response variables Y_r onto two separate directions specified by unit vectors w_x and w_y , to obtain two random variables $\bar{X}_{train,b}w_x$ and \bar{Y}_rw_y that are again univariate and hence whose covariance can be computed. In this way we can assess the relation assuming between $\bar{X}_{train,b}$ and \bar{Y}_r . Given two directions w_x and w_y , their covariance can be measured as:

$$\begin{aligned} \mathbb{E}[\bar{X}_{train,b}w_x \bar{Y}_rw_y] &= \mathbb{E}[w_x^T \bar{X}_{train,b}^T \bar{Y}_r w_y] \\ &= w_x^T \mathbb{E}[\bar{X}_{train,b}^T \bar{Y}_r] w_y \end{aligned} \quad (3)$$

Let C_{xy} be the sample covariance matrix $\mathbb{E}[\bar{X}_{train,b}^T \bar{Y}_r]$ between X and Y , where X and Y are two feature matrices whose i th rows corresponds to the sample $\bar{X}_{train,b}(i)$ and $Y_r(i)$, we can write:

$$\begin{aligned} C_{xy} &= \mathbb{E}[\bar{X}_{train,b}^T \bar{Y}_r] \\ &= \frac{1}{N_t} \sum_{i=1}^{N_t} \bar{X}_{train,b}(i) Y_r^T(i) \\ &= \frac{1}{N_t} X^T Y \end{aligned} \quad (4)$$

The directions w_x and w_y can therefore be found as follows:

$$\max_{w_x, w_y} C(w_x, w_y) = w_x^T C_{xy} w_y = \frac{1}{N_t} w_x^T X^T Y w_y \quad (5)$$

$$\text{subject to } \|w_x\| = \|w_y\| = 1 \quad (6)$$

The directions that solve the maximal covariance optimization are the first singular vectors $w_x = u_1$ and $w_y = v_1$ of the singular value decomposition of C_{xy}

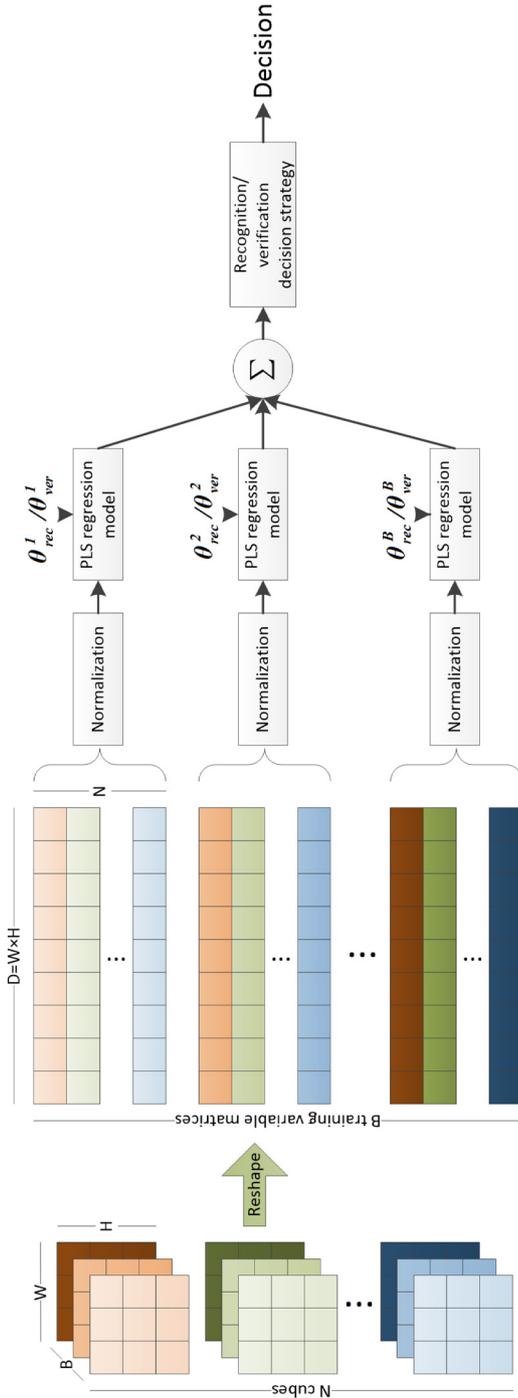


Fig. 1. Overall flowchart of the proposed algorithm.

Input: training matrix $\bar{X}_{train.b}$, response variables \bar{Y}_r , principal component number k

Output: regression coefficients θ_b

- 1: initialization
- 2: **for** $i = 1, 2, \dots, k$ **do**
- 3: $u_i \leftarrow$ first column of $\bar{X}_{train.b}^T \bar{Y}_r$
- 4: $u_i \leftarrow u_i / \|u_i\|$
- 5: **repeat**
- 6: $u_i \leftarrow \bar{X}_{train.b-i}^T \bar{Y}_r \bar{Y}_r^T \bar{X}_{train.b-i} u_i$
- 7: $u_i \leftarrow u_i / \|u_i\|$
- 8: **until** convergence
- 9: $p_i \leftarrow \bar{X}_{train.b-i}^T \bar{X}_{train.b-i} u_i / (u_i^T \bar{X}_{train.b-i} \bar{X}_{train.b-i} u_i)$
- 10: $c_i \leftarrow \bar{Y}_r^T \bar{X}_{train.b-i} u_i / (u_i^T \bar{X}_{train.b-i} \bar{X}_{train.b-i} u_i)$
- 11: $\bar{X}_{train.b-i+1} \leftarrow \bar{X}_{train.b-i} (I - u_i p_i^T)$
- 12: **end for**
- 13: $\theta_b = u(p^T u)^{-1} c^T$

Algorithm 1. Pseudocode of PLS regression algorithm.

$$C_{xy} = U\Sigma V^T \quad (7)$$

where the value of the covariance is given by the corresponding singular value σ_1 . In this paper, more than one direction is wanted. Let the direction number be k , we deflate X_k by projecting its columns into the space orthogonal to $X_{k-1}u_{k-1}$, where X_k is the feature matrix of the k th direction and u_k is its first singular vector. The deflation of X can be written as:

$$X_j = \left(I - \frac{X_{j-1}u_{j-1}u_{j-1}^T X_{j-1}^T}{u_{j-1}^T X_{j-1}^T X_{j-1} u_{j-1}} \right) X_{j-1} \quad (8)$$

The PLS regression algorithm is shown in Algorithm 1. The repeat loop computes the first singular value by the iterative method. This results in u_i converging to the first right singular vector $Y^T X_j$. Next, the deflation of X_j is computed. Finally, the regression coefficients θ_b is given by [36]:

$$\theta_b = u(p^T u)^{-1} c^T \quad (9)$$

where c is a matrix with columns

$$c_j = \frac{Y^T X_j u_j}{u_j^T X_j^T X_j u_j} \quad (10)$$

2.2. Recognition

Palmprint recognition identifies the candidate palmprint having the highest matching score with a given palmprint image. For a palmprint recognition system having N_r palm candidates, its matching score row vector of the b th band, $\hat{Y}_{rec}^b \in \mathbb{R}^{N_r}$, is obtained by multiplying the normalized b th band of the input image, \bar{X}_b , by its PLS regression coefficients $\theta_{rec}^b \in \mathbb{R}^{D \times N_r}$:

$$\hat{Y}_{rec}^b = \bar{X}_b \times \theta_{rec}^b \quad (11)$$

We denote the prediction matrix for the training process of palmprint recognition by $Y_{rec}^p \in \mathbb{R}^{N_i \times N_r}$. $Y_{rec}^p(m, n)$, with $m = 1, 2, \dots, N_i$ and $n = 1, 2, \dots, N_r$, is 1 if the m th training sample is acquired from the n th palm candidate and 0 otherwise. For a training matrix having $N_r = 4$ and two training samples per class, Y_{rec}^p is as follows:

$$Y_{rec}^p = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix}^T \quad (12)$$

The regression coefficients of the b -channel, θ_{rec}^b , is computed by assigning Y_{rec}^p to the input argument \bar{Y}_r of Algorithm 1. The matching scores of each channel \hat{Y}_{rec}^b are then fused:

$$\hat{Y}_{rec} = \sum_{b=1}^B \hat{Y}_{rec}^b \quad (13)$$

We estimated the weight of each band using the Mont Carol method, and find that there is no any weight vectors can provide a recognition rate greater than equal weights. All the spectral bands are therefore considered contribute equally to the matching process.

In the recognition matching score vector, its n th element, $\hat{Y}_{rec}^b(n)$, is expected to be maximal when the input sample matches with the n th palm candidate and close to 0 otherwise. Finally, the index of the maximum of \hat{Y}_{rec} , noted as d_{rec} , gives the estimated class of the input sample.

2.3. Verification

Palmprint verification is used to determine whether the input image matches with a given candidate palmprint image ($N_r = 1$). Its regression coefficient matrix within the b th channel is defined as $\theta_{ver}^b \in \mathbb{R}^{D \times 1}$, and its matching score \hat{Y}_{ver}^b can be computed by replacing θ_{rec}^b in Eq. (11) using θ_{ver}^b .

θ_{ver}^b is a coefficient column, representing the regression coefficient vector of the given palm candidate. Its prediction element Y_{ver}^p is a binary vector whose element is 1 when the training sample matches with the palmprint candidate and otherwise 0. For example, for a training matrix having $N_i = 6$, if two of the training samples are acquired from the first palm candidate, the prediction vector Y_{ver}^p will be $[1, 1, 0, 0, 0, 0]^T$. The training of θ_{ver}^b is realized by assigning Y_{ver}^p to the input argument \bar{Y}_r of Algorithm 1.

After the fusion of matching scores, the elements of verification decision d_{ver} is computed using a threshold value ϑ :

$$d_{ver} = \begin{cases} 1 & \text{if } \hat{Y}_{ver} \geq \vartheta \\ 0 & \text{else} \end{cases} \quad (14)$$

d_{ver} is the verification result between the input sample and the palm candidate, in which one for acceptance and zero for rejection. The threshold value ϑ is selected when the equal error rate of the verification system is obtained.

3. Embedded hardware design and optimization

This section presents our study on real-time embedded palmprint recognition. We first describe the behavior of the algorithm implemented into the register-transfer level through the high-level synthesis process, then a series of optimizations are made for efficiency improvement.

3.1. Original FPGA design

High-Level Synthesis (HLS) is a promising technique to improve FPGA development productivity and the maintainability of the products by automating the C-to-RTL synthesis [39–42]. We use hereby a world's leading HLS tool Vivado_HLS (formerly AutoPilot from AutoESL) [43]. This tool is selected for its significant ability to generate high quality RTL implementations [44].

We base the desired FPGA design on a classical master-slave architecture, which consists of an external memory (slave) and a processor (master) customized depending on the given algorithm. The two blocks are typically interconnected through a memory interface, which allows random accesses to the memory address locations of X and θ_{rec} . Since this implementation has only a single input sample ($N = 1$), the output argument (decision) is an integer number representing the label of the matched palm candidate. Its corresponding output interface $drec_vld$ is therefore performed by using a data bus with a valid signal. Additionally, in order to raise the throughput of the design, two memory port interfaces are implemented for each input argument. For example, both of the ports X_mem1 and X_mem2 are for the argument X . Within Vivado_HLS, the synthesis of different interfaces can be easily controlled by the INTERFACE directive or by using a configuration setting [43].

Algorithm 2 shows the C pseudocode of the algorithm to be synthesized, with which a 4-band framework is implemented, including red, green, blue and NIR for instance. Its inputs include the filtered multispectral palmprint cube X and the regression coefficients θ_{rec} , and the return is the decision variable d_{rec} . As shown in Fig. 1, we first reshape the input multispectral cube during the initialization process, then normalize the test samples. The recognition matching score matrix of each channel is computed by the PLS regression model with the given coefficient matrix, and fused by using equal weight sum. Finally the decision is made. In this original version, PLS regression model and decision function (Line 14–17 and 21) are packaged into the sub functions to add to its readability.

3.2. Design optimization

Despite of many benefits in terms of complexity, maintainability, development productivity, etc., it exists still a significant performance gap between HLS-based and manual register-transfer level implementations for some applications in terms of time control, execution speed, consumption, etc.[40,45]. Consequently, we made a series of code-level optimizations to improve the performance of the original implementation.

Generally speaking, the quality of the HLS based FPGA implementations are impacted by the following three factors: high-level description of language, optimization forms and applying orders of optimization forms [46,47]. As shown in Fig. 2, in this case, multiple optimization forms are made successively in different hierarchies,

include function inline, loop manipulation, pipeline and symbol expression manipulation.

In order to manipulate the loops in different function levels, the function hierarchy is first flattened by function inline. This transformation enables logic optimization across function boundaries and improve latency/interval by reducing function call overhead. Next, we manipulate the loops of the source code by using loop fusion and unrolling. HLS abstracts the input source code as a control and datapath flow graph, in which a sequence of successive operations is processed as a control step. Fig. 3(a) shows the diagram of the control flow extracted from the function inline version of the proposed algorithm. In order to reduce the state and transit number, the loops of normalization computations (Lines 2, 5, 8 and 11) and loops containing in the PLS regression models (Line 14,15,16 and 17) are fused into a single one respectively due to the same loop boundary and independent bodies (Fig. 3(b)). Furthermore, the initialization operations in S_{11} is moved to the beginning of the input code and fused into the initialization step S_0 . Finally, the loop of S_2 in Fig. 3(b) is unrolled completely to parallelize its iterations (see Fig. 3(c)). In our case, the transformation of loop manipulation can reduce the hardware consumption of logical control and add to the instruction level parallelism by centering the operations into a single control step.

The pseudocode of the optimized implementation is shown in Algorithm 3, in which two optimization directives (`#pragma AP pipeline`) are placed under the loops to perform iteration pipeline optimization. Since the loop control, iteration count and dependent operations will result in some delays, a initiation interval is required between the iterations. This value can be estimated through the operation scheduling generated by using HLS tools automatically. With the pipeline directives, the factor II is used to specify the desired initiation interval for the pipeline. Additionally, the expression of score level fusion (Line 19 in Algorithm 2) is segmented into short expressions (Line 12 and 13 in Algorithm 3). This transformation can enhance the detection ability of HLS tools in terms of Instruction-Level Parallelism.

4. Algorithm implementations and experiments

This section presents proposed algorithm implementations using multiples configurations and analyzes the evaluation experiments. We first determine its parameters, then estimate the accuracy performance of the palmprint recognition and verification algorithms respectively. All experiments have been achieved in the environment of MATLAB.

4.1. Palmprint database description

We conduct all of the experiments by using the ROI version of the PolyU multispectral palmprint database provided by Hong Kong Polytechnic University [6]. This database is captured with NIR and visible light (red, green and blue color). Fig. 4 shows a typical multispectral palmprint sample in the a) Blue, b) Green, c) Red and d) NIR bands. It can be observed that the line features are clearer in the blue and green bands than in the red and NIR bands. While the Red band can reveal some vein structure, the NIR band can show the palm vein structures as well as partial line information [6].

All the images are divided into two sessions, whose average time interval was about 9 days. For each session, 6 samples are acquired, so we have 6000 samples (6 samples \times 2 sessions \times 500 persons) of 128×128 pixels in total.

4.2. Parameter configuration

The subject of this experiment is to determine the following two parameters of the proposed algorithm: the dimension of the average-filtered multispectral frames W/H ($W = H$) and the number of principal components of PLS regression k . To do this, the images of the first session are used to perform the test sample matrix, and one of the six

Input: multispectral cube X , regression coefficients θ_{rec}

Output: decision d_{rec}

- 1: initialization
- 2: **for all** the feature variables of the red band **do**
- 3: $\bar{X}_{red} \leftarrow \text{normalize } X_{red}$
- 4: **end for**
- 5: **for all** the feature variables of the green band **do**
- 6: $\bar{X}_{green} \leftarrow \text{normalize } X_{green}$
- 7: **end for**
- 8: **for all** the feature variables of the blue band **do**
- 9: $\bar{X}_{blue} \leftarrow \text{normalize } X_{blue}$
- 10: **end for**
- 11: **for all** the feature variables of the NIR band **do**
- 12: $\bar{X}_{NIR} \leftarrow \text{normalize } X_{NIR}$
- 13: **end for**
- 14: $\hat{Y}_{rec}^{red} \leftarrow \text{pls_regress_model}(\bar{X}_{red}, \theta_{rec}^{red})$
- 15: $\hat{Y}_{rec}^{green} \leftarrow \text{pls_regress_model}(\bar{X}_{green}, \theta_{rec}^{green})$
- 16: $\hat{Y}_{rec}^{blue} \leftarrow \text{pls_regress_model}(\bar{X}_{blue}, \theta_{rec}^{blue})$
- 17: $\hat{Y}_{rec}^{NIR} \leftarrow \text{pls_regress_model}(\bar{X}_{NIR}, \theta_{rec}^{NIR})$
- 18: **for all** the elements of \hat{Y}_{rec} **do**
- 19: $\hat{Y}_{rec}(i) \leftarrow \sum_{b=1}^B \hat{Y}_{rec}^b(i)$
- 20: **end for**
- 21: $d_{rec} \leftarrow \text{decision_func}(\hat{Y})$

Algorithm 2. Pseudocode of the original palmprint recognition algorithm behavior.

images of the second session is selected at random to train the regression coefficients. Next, the given multispectral images are fused into 2D images by using the spatio-spectral covariance based band fusion method recently proposed by Uzair et al. [24]. This band fusion method incorporates local spatial information as well as efficiently removes noise by averaging both the spectral and spatial dimensions. Moreover, it enables to handle small misalignment (see Section 3A of Ref. [24] for more details). Finally, the fused images are assigned to a regression channel for accuracy and efficiency estimations. All the experiment results are calculated with the average value of ten independent measurements.

Fig. 5 shows the 3-D plot of recognition rate over the image size and number of principal components of PLS regression. Fig. 6 plots the recognition rates and the running time of the MATLAB recognition implementation over the dimension of input multispectral cubes. We can see that the curve of recognition rate shows a flattening circa $W = 13$, but the running time raises linearly. In order to obtain a high accuracy-cost ratio, the dimension of the input multispectral images is defined as $W = H = 13$, which provides an average recognition ratio of 92.4% with a running speed of 1.26 images per second in this test.

Fig. 7 shows the measurement results of recognition rate and running time over the principal component number of PLS regression. Similarly, it is found that the accuracy curve starts to trend to flattening at $k = 44$ and the running time raises permanently. Consequently, k is set as 44, which results in an average recognition rate of 93.2% with a running speed of 2.84 images per second in this test.

4.3. Recognition accuracy

The recognition accuracy of the proposed algorithm is evaluated by using recognition rate with the smallest training set. The training set is created by choosing a single sample from the six samples of one session at random, then the other session or all of the rest samples for testing.

First of all, the average recognition rates of the proposed algorithm is measured. The experiment is repeated 10 times in order to obtain an unbiased result, in which the two sessions are used for training and testing alternately. Table 1 compares the recognition accuracy of the proposed method with other two reference implementations, which are based on the Multiclass Projection Extreme Learning Machine (MPELM) [8] and Quaternion Principal Component Analysis (QPCA) [7].

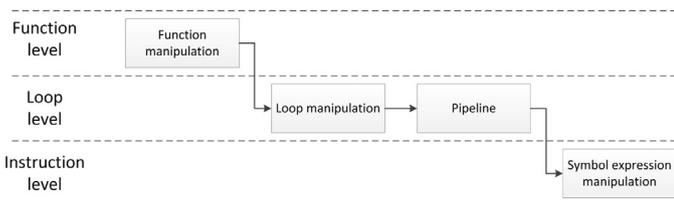


Fig. 2. Optimization strategy of the palmprint recognition design.

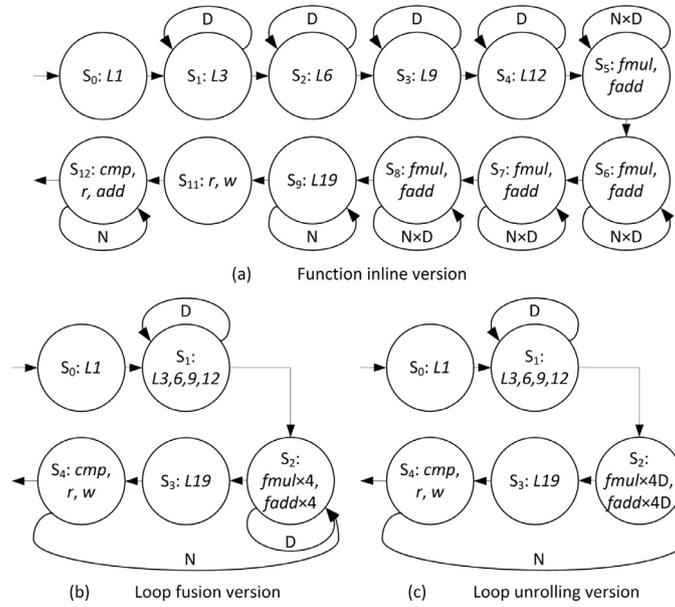


Fig. 3. Diagram of the finite state machine extracted from the proposed algorithm: S_i is the state identification, L^* is the line number of the operations in Algorithm 2, and $fmul$, $fadd$, cmp , r and w are the multiplication, addition, comparison, reading and writing operators.

We can see that our method provides a similar accuracy performance compared to the other methods by using only two bands and a single sample for training. When all the four bands are applied, a very high recognition rate, nearly 100%, is achieved.

What is interesting is that when a larger test sample set is used, the recognition rate raises instead of decreasing, which seems abnormal. In our opinion, this is caused by the following two reasons:

- During each session, the palms are will fixed for all of the six captures, the ROI misalignment between the samples from the same

session is therefore lower than the ones between the different sessions;

- There is an interval between the two capturing sessions (around 9 days in average), the palms texture and veins may vary physiologically more or less, even if they are considered highly stable.

Consequently, using the samples from different session for training and test respectively is a more objective method to evaluate the biometric systems. The experiments demonstrate that our algorithm provides the best accuracy performance within the most challenging

Input: multispectral cube X , regression coefficients θ_{rec}

Output: decision d_{rec}

```

1: initialization
2: for all the feature variables of all the bands do
3:   #pragma AP pipeline II=5
4:   normalize  $X_{rec}^{red}$ ,  $X_{rec}^{green}$ ,  $X_{rec}^{blue}$  and  $X_{rec}^{NIR}$ 
5: end for
6: for the  $i$ -th element of  $\hat{Y}_{rec}$  do
7:   #pragma AP pipeline II=86
8:    $\hat{Y}_{rec}^{red}(i) \leftarrow \bar{X}_{red} * \theta_{rec}^{red}(i)$ 
9:    $\hat{Y}_{rec}^{green}(i) \leftarrow \bar{X}_{green} * \theta_{rec}^{green}(i)$ 
10:   $\hat{Y}_{rec}^{blue}(i) \leftarrow \bar{X}_{blue} * \theta_{rec}^{blue}(i)$ 
11:   $\hat{Y}_{rec}^{NIR}(i) \leftarrow \bar{X}_{NIR} * \theta_{rec}^{NIR}(i)$ 
12:   $tmp1 \leftarrow \hat{Y}_{rec}^{red} + \hat{Y}_{rec}^{green}$ ,  $tmp2 \leftarrow \hat{Y}_{rec}^{blue} + \hat{Y}_{rec}^{NIR}$ 
13:   $\hat{Y}_{rec}(i) \leftarrow tmp1 + tmp2$ 
14:  if  $\hat{Y}_{rec}(i-1) < \hat{Y}_{rec}(i)$  then
15:     $d_{rec} \leftarrow i$ 
16:  end if
17: end for

```

Algorithm 3. Pseudocode of the optimized palmprint recognition algorithm behavior.

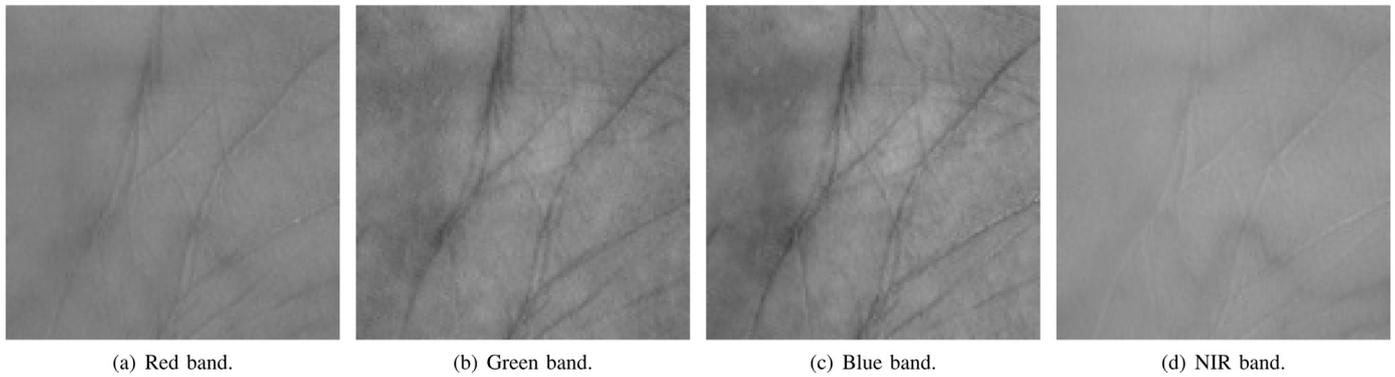


Fig. 4. Example of the PolyU multispectral palmprint database.

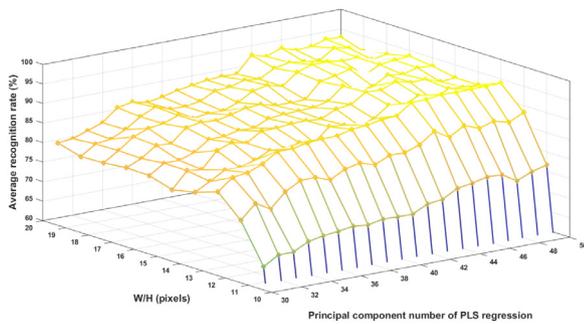


Fig. 5. 3-D plot of recognition rate over the image size and number of principal components of PLS regression.

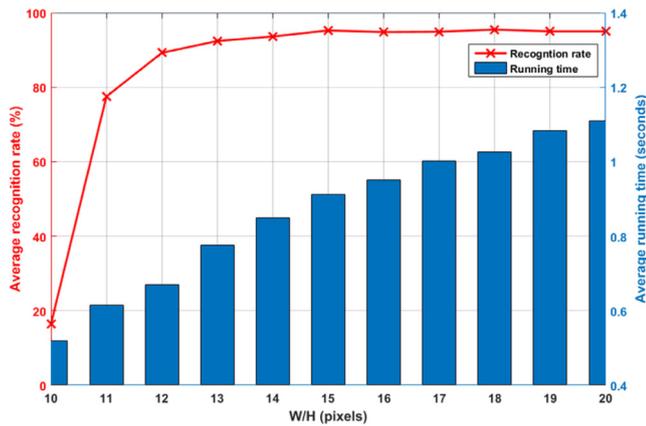


Fig. 6. Plot of recognition rate and running time over multispectral cube dimension: $W = H$.

conditions among all the references.

4.4. Verification accuracy

The EER of the proposed verification algorithm is analyzed qualitatively for accuracy estimation. Fig. 8 plots the DET (Detection Error Tradeoff) curves of the palmprint verification implementations with different band channel schemes, including single-, double-, triple- and overall-bands. As expected, the more bands the scheme covers, the lower EER is achieved. Eventually, as shown in Fig. 8(d), a very low EER is obtained with only fewer images accepted/rejected falsely. The orders of magnitude of the EERs of the four schemes vary from 10 down to 10^{-4} with the increasing of the number of channels. Meanwhile, it should be noted that the DET curves of Fig. 8(d) is the result of one measurement. When repeating the measurement experiment with the overall-band schemes, there is often no any falsely accepted/

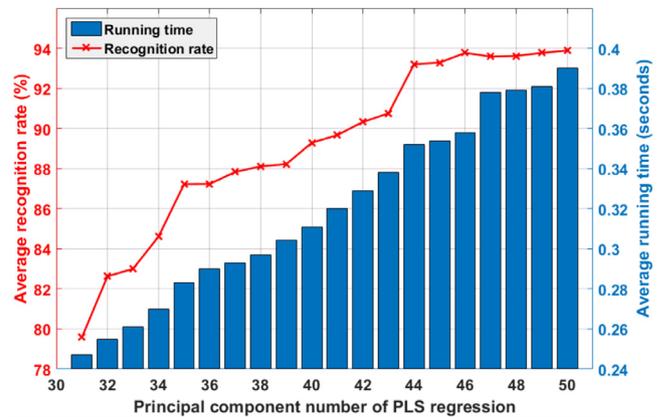


Fig. 7. Plot of recognition rate and running time over the principal component number of PLS regression: $W = H = 13$.

Table 1
Comparison of recognition rate.

Methods	Bands	Sample number (train vs. test)	Average recognition rate
MPELM [8]	NIR + red	1 vs. 11	97.33%
	NIR + red	3 vs. 9	99.56%
QPCA [7]	All	6 vs. 6	98.13%
Proposed	NIR + green	1 vs. 6	98.23%
	NIR + green	1 vs. 11	99.98%
	All	1 vs. 6	99.96%
	All	1 vs. 11	99.99%

rejected samples. That is, the matching score matrix obtained by it demonstrates a high dissimilarity, allowing a threshold value that can classify all the input samples perfectly.

Finally, the verification accuracy of the different palmprint verification schemes are quantitatively analyzed by comparing to the other high accuracy palmprint biometric designs evaluated by using the same database. The average EERs calculated from 6 independent repetitive experiments are used for the proposed algorithm. Tab. 2 shows the EERs of the proposed algorithm and the other three palmprint verification designs. We can see that for the double- and triple-band schemes, it is the method of Hong et al. [9] that achieves the lowest EER, whereas our method is the worst. However, when using full-band scheme, our method provides a very low EER, and the other methods do not effectively reduce their EERs comparing to the double- and triple-band schemes.

4.5. Hardware evaluation

We first functionally verified the embedded implementation of the

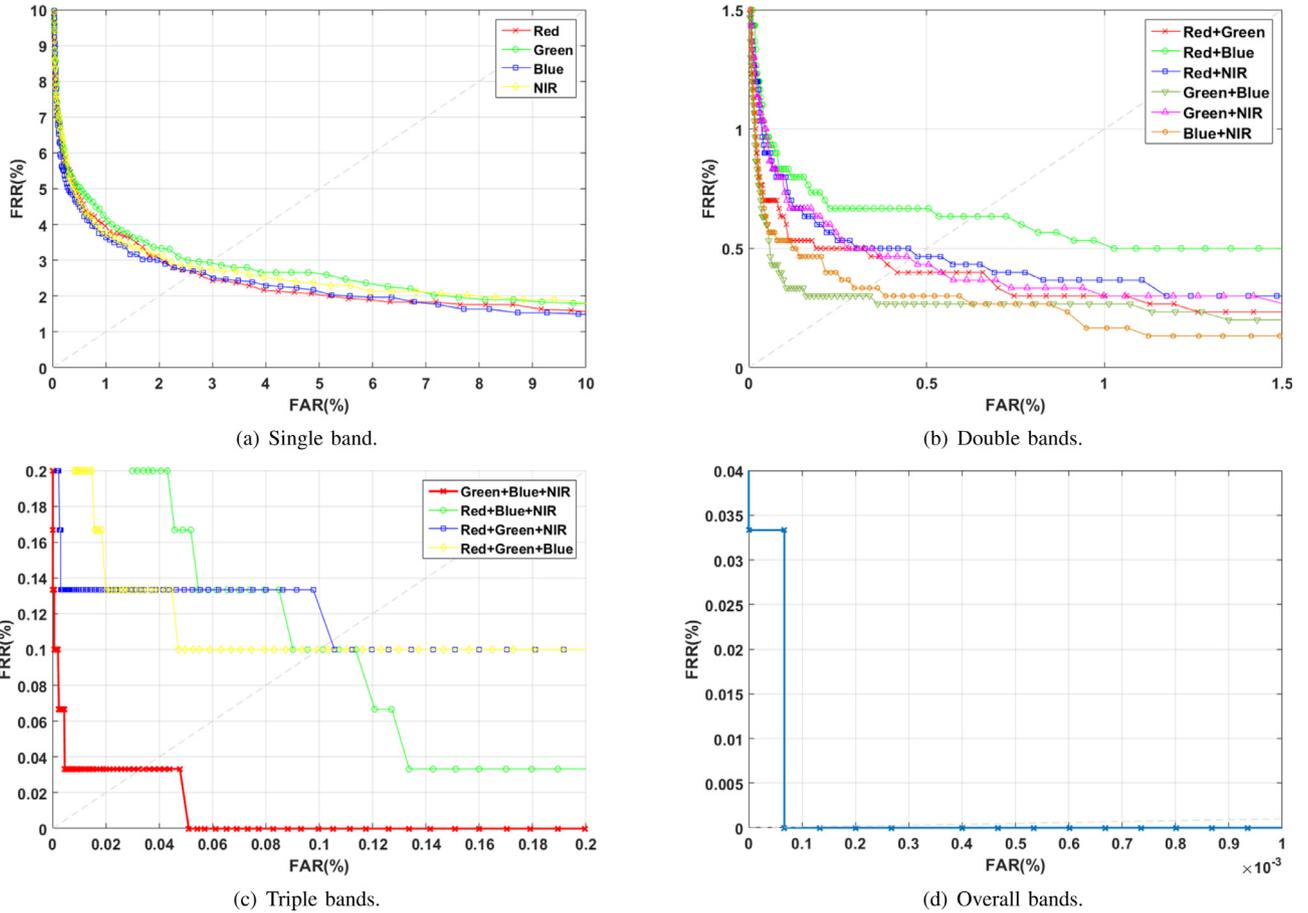


Fig. 8. DET curve of the palmprint verification implementation with different band channel schemes.

Table 2
Comparison of EER.

Methods	Band schemes	EERs (%)
Hong et al. [9]	Green + red	0.0074
	All	0.0079
Zhang et al. [6]	Blue + red	0.0121
	Blue + red + NIR	0.0121
	All	0.0121
Han et al. [48]	All	0.0396
Proposed	Green + red	0.41
	Blue + red	0.63
	Blue + red + NIR	0.096
	All	7.33×10^{-4}

palmprint recognition system. The testbench is established by using System Generator of Xilinx. The Matlab prototype of the same recognition algorithm is used as the reference. Thanks to the powerful IP core library of Xilinx, the high-precision and complex mathematical operations in Matlab are also available in FPGAs. Experiments demonstrate that the two implementations have the same classification results.

The running speed performance of the hardware design is evaluated using the device *xc5vfx70tff1136-1* of Xilinx with a clock cycle period of 8.34 ns, which is estimated by Vivado_HLS. Its measurements cover the initializing, matching, fusion and decision cycles. Table 3 compares our methods to other four reference implementations based on the same database.

We can see that all of the reference implementations require a feature extraction or band fusion cycle before feature matching, which results in a high running time cost. Our method is based directly on the

pixel values of the input multispectral images, so either the original or the optimized version of our implementation provides a much higher efficiency performance. Meanwhile, Table 3 demonstrates also that the optimizations made accelerates the implementation of the proposed algorithm from 1.58×10^{-2} seconds upto 7.3×10^{-4} s, achieving a speedup around $21.67 \times$.

Table 4 presents the consumption of different components of the optimized implementation. Comparing to the original version, its average hardware utilization rate increases by around $5.88 \times$, which is much lower than the acceleration ratio. That demonstrates that the applied optimization methods can effectively improve the implementation performance by using the additional area of the target device, and provide a high efficiency-area ratio. In additional, our implementation requires an external memory to save the test data and regression coefficients. For a N -to- N_r system, its size can be estimated as follow: $(1 + N_r) \times D \times S$, where D is the number of the feature variables, and S is the size of data type.

5. Conclusion

This paper presents an embedded system for fast multispectral palmprint biometric applications. Comparing to the reference designs, experiments demonstrate that the implementation with the proposed approach has a higher recognition/verification accuracy (99.96% vs. 99.56%), and a lower running time (0.73 ms vs. 66.62 ms). Its advantages include:

- It does not require high resolution input samples. As demonstrated in Section 4.2, the dimension of the input multispectral images used in this work is 13-by-13, which is much lower that the resolutions

Table 3
Efficiency comparison.

Implementations	Feature extraction or fusion (ms)	Feature matching (ms)	Environments	Devices
Hong et al. [9]	66	0.62	–	CPU, 2.90 GHz, 2 GB RAM
MPELM [8]	115.8	0.1	MATLAB 7.12, Windows 7-64bit	Intel Core i7, 3.4 GHz, 12 GB RAM
QPCA [7]	94	0.435	MATLAB 7.0, Windows XP (x64)	Xeon 5160 CPU, 3.0 GHz, 4 GB RAM
Zhang et al. [6]	144	0.24	Visual C++ 6.0, Windows XP	T6400 CPU, 2.13 GHz, 2 GB RAM
Original proposed	Overall running time: 15.8 ms		Vivado_HLS	xc5vfx70ff1136-1 of Xilinx, 120 MHz
Optimized proposed	Overall running time: 0.73 ms		Vivado_HLS	xc5vfx70ff1136-1 of Xilinx, 120 MHz

Table 4
Hardware consumption estimation.

Components	BRAM	DSP	FF	LUT
Expression	–	–	0	2877
Instance	–	40	3132	3638
Memory	16	–	0	0
Multiplexer	–	–	–	13,300
Register	–	–	26,197	–
ShiftMemory	–	–	0	16,846
Total	16	40	29,329	36,661
Utilization(%)	5	31	65	81

provided by the database, 128-by-128.

- It is very appropriate for the multispectral image based biometric applications. We base the proposed algorithm on the PLS regression and score fusion framework. Table 2 shows that the reference designs can hardly further improve the accuracy performance when the number of light bands increases, whereas the accuracy of our method raises sharply with it, so the proposed method can better benefit the palmprint biometric applications from the multispectral modalities.
- It has the ability of real-time processing. Comparing to the solutions based on the platforms of other types, it achieves a much higher running speed. Unlike the other methods, the proposed algorithm does not need feature extraction or band fusion before matching cycle, so it greatly simplifies the processing framework, resulting in a much lower running cost potentially. According to the efficiency estimation, the running speed is around 1.37 samples per millisecond, satisfying the requirements of real-time processing.
- It is very easy to be transplanted to the other hardware platforms for different using purposes. The matching process of the proposed algorithm is actually an operation of matrix production, which can be easily implemented and optimized by using any currently-available computing platforms.

Meanwhile, some issues exist still. Firstly, it is well known that ROI misalignment impact considerably to the precision of recognition, which is not discussed in this paper. Secondly, the training process of the proposed algorithm is implemented in a personal computer, making the update of system and data set inconvenient. How to transplant the training algorithm into FPGA is challenged by the complex operations involved in it, such as singular value decomposition, etc. For the first issue, many robust palmprint ROI extraction and alignment methods have been proposed, allowing us to further complete our system. In the future work, we will further improve the methods presented in this paper by transplanting the training cycle of the proposed algorithm into FPGAs and making more simulations and evaluations in deep in order to realize an autonomous, adaptive and portable biometric system.

Acknowledgment

The authors would like to thank the CAS Pioneer Hundred Talents Program for their funding of our studies.

Supplementary material

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.sysarc.2018.05.008.

References

- [1] R.S. Wyant, N. Nedjah, L. de Macedo Mourelle, Efficient Biometric Palm-Print Matching on Smart-Cards, Springer International Publishing, Cham, pp. 236–247. doi:10.1007/978-3-319-09153-2_18.
- [2] G. Hachez, J.-J. Quisquater, F. Koeune, Biometrics, Access Control, Smart Cards: A Not So Simple Combination, Springer US, Boston, MA, pp. 273–288. doi:10.1007/978-0-387-35528-3_16.
- [3] F.J. Zareen, S. Jabin, Authentic mobile-biometric signature verification system, IET Biom. 5 (1) (2016) 13–19, <http://dx.doi.org/10.1049/iet-bmt.2015.0017>.
- [4] J. Kang, D.V. Anderson, M.H. Hayes, Face recognition for vehicle personalization with near infrared frame differencing, IEEE Trans. Consum. Electron. 62 (3) (2016) 316–324, <http://dx.doi.org/10.1109/TCE.2016.7613199>.
- [5] X. Wu, D. Zhang, K. Wang, Fisherpalms based palmprint recognition, Pattern Recognit. Lett. 24 (15) (2003) 2829–2838, [http://dx.doi.org/10.1016/S0167-8655\(03\)00141-7](http://dx.doi.org/10.1016/S0167-8655(03)00141-7).
- [6] D. Zhang, Z. Guo, G. Lu, L. Zhang, W. Zuo, An online system of multispectral palmprint verification, IEEE Trans. Instrum. Measur. 59 (2010) 480–490.
- [7] X. Xu, Z. Guo, Multispectral palmprint recognition using quaternion principal component analysis, 2010 International Workshop on Emerging Techniques and Challenges for Hand-Based Biometrics (ETCHB), (2010), pp. 1–5, <http://dx.doi.org/10.1109/ETCHB.2010.5559287>.
- [8] X. Xu, L. Lu, X. Zhang, H. Lu, W. Deng, Multispectral palmprint recognition using multiclass projection extreme learning machine and digital shearlet transform, Neural Comput. Appl. 27 (1) (2016) 143–153, <http://dx.doi.org/10.1007/s00521-014-1570-8>.
- [9] D. Hong, W. Liu, J. Su, Z. Pan, G. Wang, A novel hierarchical approach for multispectral palmprint recognition, Neurocomputing 151, Part 1 (2015) 511–521, <http://dx.doi.org/10.1016/j.neucom.2014.09.013>.
- [10] A. Kumar, S. Shekhar, Personal identification using multibiometrics rank-level fusion, IEEE Trans. Systems, Man Cybernet. Part C 41 (5) (2011) 743–752, <http://dx.doi.org/10.1109/TSMCC.2010.2089516>.
- [11] M. Pudzs, R. Fuksis, R. Ruskuls, T. Eglitis, A. Kadikis, M. Greitans, Fpga based palmprint and palm vein biometric system, 2013 International Conference of the BIOSIG Special Interest Group (BIOSIG), (2013), pp. 1–4.
- [12] O. Nikisins, T. Eglitis, M. Pudzs, M. Greitans, Algorithms for a novel touchless bi-modal palm biometric system, International Conference on Biometrics, (2015), pp. 436–443.
- [13] A. team, The AAA Methodology and SynDEX, Technical Report, INRIA Paris-Rocquencourt Research Center France, 2017.
- [14] A.K. Jain, P. Flynn, A.A. Ross, Handbook of Biometrics, Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2007.
- [15] W.-Y. Yau, J.-G. Wang, E. Sung, A. Suwandy, Fusion of palmprint and palm vein images for person recognition based on “laplacianpalm” feature, 2007 IEEE Conference on Computer Vision and Pattern Recognition, 00 (2007), pp. 1–8, <http://dx.doi.org/10.1109/CVPR.2007.383386>.
- [16] Y. Hao, Z. Sun, T. Tan, Comparative Studies on Multispectral Palm Image Fusion for Biometrics, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 12–21. doi:10.1007/978-3-540-76390-1_2.
- [17] Y. Hao, Z. Sun, T. Tan, C. Ren, Multispectral palm image fusion for accurate contact-free palmprint recognition, 2008 15th IEEE International Conference on Image Processing, (2008), pp. 281–284, <http://dx.doi.org/10.1109/ICIP.2008.4711746>.
- [18] A.A. Ross, A.K. Jain, K. Nandakumar, Handbook of Multibiometrics, Vol. 6 Springer US, 2006, <http://dx.doi.org/10.1007/0-387-33123-9>.
- [19] D.V. Nguyen, D.M. Rocke, Tumor classification by partial least squares using microarray gene expression data. Bioinformatics 18 (1) (2002) 39.
- [20] M. Martens, H. Martens, Statistical Procedures in Food Research, Elsevier Applied Science, p. 293359.
- [21] K. Worsley, An overview and some new developments in the statistical analysis of pet and fmri data, Hum. Brain Mapp. (1997) 5:254258.
- [22] J. Nilsson, S.D. Jong, A.K. Smilde, Multiway calibration in 3d qsar, J. Chemom. 11 (6) (1997) 511–524.
- [23] J. Hulland, Use of partial least squares (pls) in strategic management research: a review of four recent studies, Strategic Manag. J. (1999) 20:195204.
- [24] M. Uzair, A. Mahmood, A. Mian, Hyperspectral face recognition with spatiospectral

- information fusion and pls regression, *IEEE Trans. Image Process.* 24 (3) (2015) 1127–1137, <http://dx.doi.org/10.1109/TIP.2015.2393057>.
- [25] D.-S. Kim, J. Kwon, Moving object detection on a vehicle mounted back-up camera, *Sensors* 16 (1) (2015) 23, <http://dx.doi.org/10.3390/s16010023>.
- [26] Z. Yonghong, An nprod algorithm ip design for real-time image matching application onto fpga, *Electrical and Control Engineering (ICECE), 2010 International Conference on*, (2010), pp. 404–409, <http://dx.doi.org/10.1109/iCECE.2010.105>.
- [27] S.-W. Chen, Y.-H. Chen, Hardware design and implementation of a wavelet denoising procedure for medical signal preprocessing, *Sensors* 15 (10) (2015) 26396, <http://dx.doi.org/10.3390/s151026396>.
- [28] M. Chiesi, L. Vanzolini, C. Mucci, E. Scarselli, R. Guerrieri, Power-aware job scheduling on heterogeneous multicore architectures, *Parallel Distrib. Syst. IEEE Trans.* 26 (3) (2015) 868–877, <http://dx.doi.org/10.1109/TPDS.2014.2315203>.
- [29] O. Oballe-Peinado, F. Vidal-Verdú, J.A. Sánchez-Durán, J. Castellanos-Ramos, J.A. Hidalgo-López, Smart capture modules for direct sensor-to-fpga interfaces, *Sensors* 15 (12) (2015) 29878, <http://dx.doi.org/10.3390/s151229878>.
- [30] F. Nombela, E. García, R. Mateos, A. Hernández, Efficient implementation of a symbol timing estimator for broadband plc, *Sensors* 15 (8) (2015) 20825, <http://dx.doi.org/10.3390/s150820825>.
- [31] C. Li, V. Brost, Y. Benezeth, F. Marzani, F. Yang, Design and evaluation of a parallel and optimized light-tissue interaction-based method for fast skin lesion assessment, *J. Real-Time Image Process.* (2015) 1–14, <http://dx.doi.org/10.1007/s11554-015-0494-6>.
- [32] D. Zou, Y. Dou, F. Xia, Optimization schemes and performance evaluation of smith-waterman algorithm on cpu, gpu and fpga, *Concurr. Comput.* 24 (14) (2012) 1625–1644, <http://dx.doi.org/10.1002/cpe.1913>.
- [33] S. Kestur, J. Davis, O. Williams, Blas comparison on fpga, cpu and gpu, *VLSI (ISVLSI), 2010 IEEE Computer Society Annual Symposium on*, (2010), pp. 288–293, <http://dx.doi.org/10.1109/ISVLSI.2010.84>.
- [34] P. Coussy, A. Morawiec, *High-Level Synthesis: From Algorithm to Digital Circuit, first ed.*, Springer Publishing Company, Incorporated, 2008.
- [35] Z. Guo, D. Zhang, L. Zhang, W. Liu, Feature band selection for online multispectral palmprint recognition, *IEEE Trans. Inf. Forensics Secur.* 7 (3) (2012) 1094–1099, <http://dx.doi.org/10.1109/TIFS.2012.2189206>.
- [36] H. Wold, Soft modelling: the basic design and some extensions, *Systems Under Indirect Obs. Part II* (1982) 36–37 <http://ci.nii.ac.jp/naid/10006132197/en/>.
- [37] H. Wold, *Partial Least Squares*, John Wiley & Sons, Inc. doi:10.1002/0471667196.ess1914.pub2.
- [38] S. Wold, H. Ruhe, H. Wold, D. III. W.J., *The collinearity problem in linear regression. the partial least squares (pls) approach to generalized inverse.* *J. Sci. Stat. Comput.* 5 (1984). 745–743.
- [39] J. Cong, B. Liu, S. Neuendorffer, J. Noguera, K. Vissers, Z. Zhang, High-level synthesis for fpgas: from prototyping to deployment, *Comput.-Aided Des. Integr. Circ. Syst. IEEE Trans.* 30 (4) (2011) 473–491, <http://dx.doi.org/10.1109/TCAD.2011.2110592>.
- [40] Y. Liang, K. Rupnow, Y. Li, D. Min, M.N. Do, D. Chen, High-level synthesis: productivity, performance, and software constraints, *J. Electric. Comput. Eng.* 2012 (2012) 14. Article ID 649057. doi:10.1155/2012/649057.
- [41] C. Li, S. Balla-Arabé, F. Yang, Embedded multi-spectral image processing for real-time medical application, *J. Syst. Archit.* (2015), <http://dx.doi.org/10.1016/j.sysarc.2015.12.002>.
- [42] C. Li, S. Balla-Arabé, D. Ginhac, F. Yang, Embedded implementation of vhr satellite image segmentation, *Sensors* 16 (6) (2016) 771, <http://dx.doi.org/10.3390/s16060771>.
- [43] *Vivado Design Suite User Guide, XILINX, ug902(2012.2) ed.*, 2012.
- [44] W. Meeus, K. Van Beeck, T. Goedemé, J. Meel, D. Stroobandt, An overview of today's high-level synthesis tools, *Des. Autom. Embedded Syst.* 16 (3) (2012) 31–51, <http://dx.doi.org/10.1007/s10617-012-9096-8>.
- [45] K. Rupnow, Y. Liang, Y. Li, D. Min, M. Do, D. Chen, High level synthesis of stereo matching: Productivity, performance, and software constraints, *2011 International Conference on Field-Programmable Technology (FPT)*, IEEE, 2011.
- [46] J. Cong, B. Liu, R. Prabhakar, P. Zhang, A study on the impact of compiler optimizations on high-level synthesis, in: H. Kasahara, K. Kimura (Eds.), *Languages and Compilers for Parallel Computing, Lecture Notes in Computer Science*, vol. 7760, Springer Berlin Heidelberg, 2013, pp. 143–157, http://dx.doi.org/10.1007/978-3-642-37658-0_10.
- [47] Q. Huang, R. Lian, A. Canis, J. Choi, R. Xi, N. Calagar, S. Brown, J. Anderson, The effect of compiler optimizations on high-level synthesis-generated hardware, *ACM Trans. Reconfigurable Technol. Syst.* 8 (3) (2015) 14:1–14:26, <http://dx.doi.org/10.1145/2629547>.
- [48] D. Han, Z. Guo, D. Zhang, Multispectral palmprint recognition using wavelet-based image fusion, *2008 9th International Conference on Signal Processing*, (2008), pp. 2074–2077, <http://dx.doi.org/10.1109/ICOSP.2008.4697553>.



Chao Li received his M.S. in Electronic Engineering and Ph.D. in Image Processing from the University of Burgundy in 2012 and 2016, respectively. He also obtained the M.S. in Computer Science from the University of Paris-Sud in 2013. He is currently an associate professor in the State Key Laboratory of Acoustics of IACAS. His research interests are in the areas of SW/HW Co-design, High Performance Computer, Image Processing and Acoustic Detection.



Yannick Benezeth is associate professor at the University of Burgundy (France). He obtained his Ph.D. in computer science from the University of Orléans in 2009. He also received the engineer degree from the ENSI de Bourges and the MS degree from the University of Versailles-Saint-Quentin-en-Yvelines in 2006. His research interests include biomedical engineering, image processing and video analytics.



Keisuke Nakamura received a B.E. in 2007 from the Department of Control and System Engineering, Tokyo Institute of Technology, Japan. He also studied in the Department of Electronic and Electrical Engineering at the University of Strathclyde, United Kingdom from 2007 to 2008. He received an M.E. from the Department of Mechanical and Control Engineering at Tokyo Institute of Technology in 2010, and Ph.D. in Informatics in 2013 from Kyoto University, Japan. He is currently a senior scientist for Honda Research Institute Japan, Co., Ltd. His research interests are in the field of robotics, control systems, and signal processing. He is currently a member of IEEE.



Randy Gomez received M.Eng.Sci. in Electrical Engineering at the University of New South Wales (UNSW), Australia in 2002. He obtained his Ph.D. in 2006 from the Graduate School of Information Science, Nara Institute of Science and Technology, Japan. He was a JSPS fellow at Kyoto University, Japan. His research interests include robust speech recognition, acoustic modeling and adaptation, human-robot interaction, signal processing among others. Currently, he is a senior scientist at Honda Research Institute Japan.



Fan Yang received the B.S. degree in electrical engineering from the University of Lanzhou, China, in 1982 and the M.S. (computer science) and Ph.D. degrees (image processing) from the University of Burgundy, France, in 1994 and 1998, respectively. She is currently a full professor and member of LE2I UMR CNRS, Laboratory of Electronic, Computing, and Imaging Sciences at the University of Burgundy, France. Her research interests are in the areas of pattern recognition, neural network, parallelism and real time implementation, and, more specifically, automatic face image processing algorithms and architectures. Pr. Yang is member of the French research group ISIS (Information, Signal, Images and Vision), she livens up the

theme C: Algorithm Architecture Mapping.