Computer Vision and Image Understanding 113 (2009) 477–501

Contents lists available at ScienceDirect

Computer Vision and Image Understanding

journal homepage: www.elsevier.com/locate/cviu

Hand-based verification and identification using palm–finger segmentation and fusion

Gholamreza Amayeh, George Bebis *, Ali Erol, Mircea Nicolescu

Computer Vision Laboratory, University of Nevada, Reno 89557, United States

ARTICLE INFO

Article history:
Received 30 January 2008
Accepted 28 November 2008
Available online 24 December 2008

Keywords:
Biometrics
Hand-based verification
Hand-based recognition
Fusion
PCA
Majority voting
SVMs

ABSTRACT

Hand-based verification/identification represent a key biometric technology with a wide range of potential applications both in industry and government. Traditionally, hand-based verification and identification systems exploit information from the whole hand for authentication or recognition purposes. To account for hand and finger motion, guidance pegs are used to fix the position and orientation of the hand. In this paper, we propose a component-based approach to hand-based verification and identification which improves both accuracy and robustness as well as ease of use due to avoiding pegs. Our approach accounts for hand and finger motion by decomposing the hand silhouette in different regions corresponding to the back of the palm and the fingers. To improve accuracy and robustness, verification/recognition is performed by fusing information from different parts of the hand. The proposed approach operates on 2D images acquired by placing the hand on a flat lighting table and does not require using guidance pegs or extracting any landmark points on the hand. To decompose the silhouette of the hand in different regions, we have devised a robust methodology based on an iterative morphological filtering scheme. To capture the geometry of the back of the palm and the fingers, we employ region descriptors based on high-order Zernike moments which are computed using an efficient methodology. The proposed approach has been evaluated both for verification and recognition purposes on a database of 101 subjects with 10 images per subject, illustrating high accuracy and robustness. Comparisons with related approaches involving the use of the whole hand or different parts of the hand illustrate the superiority of the proposed approach. Qualitative and quantitative comparisons with state-of-the-art approaches indicate that the proposed approach has comparable or better accuracy.

1. Introduction

Recently, there has been increased interest in developing biometrics-based verification and identification systems which has led to intensive research in fingerprint, face, hand, ear, and iris authentication and recognition. Each biometric has its own strength and weakness depending on the specific application and its requirements. Hand-based biometrics is among the oldest live biometrics-based authentication modalities. The existence of several hand-based authentication commercial systems and patents indicate the effectiveness of this type of biometric. Although hand-based live authentication has a long history and a considerable market share [1], most studies addressing enhancements of this technology are rather recent [2]. Increases in computing power and advances in computer vision and pattern recognition are expected to enable the implementation of more accurate, robust, and easier to use systems. Removal of pegs, to improve convenience, and use of more powerful features to represent the shape of the hand represent promising research directions in this area.

The geometry of the hand contains relatively invariant features of an individual, however, geometric features of the hand (e.g., finger length/width, area/size of the palm) are not as distinctive as fingerprint or iris features. Therefore, hand-based biometric systems have been employed mostly in small-scale person authentication applications. In this study, we demonstrate the application of hand-based biometrics for identification purposes as well. The main difference between verification and identification is that in the case of verification, an unknown subject is compared against a specific subject in the database to verify his/her identity (i.e., "Am I who I claim"). In the case of identification, an unknown subject is compared against all the subjects in the database to establish his/her identity (i.e., "Who am I?"). Therefore, identification can be thought as verifying an unknown subject against all subjects in the database. As a result, identification is more time consuming and prone to errors.

There are several reasons for developing hand-based verification/identification systems. First, the shape of the hand can be easily captured in a relatively user friendly manner by using

* Corresponding author. Fax: +1 775 784 1877.
E-mail addresses: amayeh@cse.unr.edu (G. Amayeh), bebis@cse.unr.edu (G. Bebis), aerol@cse.unr.edu (A. Erol), mircea@cse.unr.edu (M. Nicolescu).

1077-3142/$ - see front matter © 2008 Elsevier Inc. All rights reserved.
doi:10.1016/j.cviu.2008.11.007

conventional CCD cameras. Second, this technology is more acceptable by the public in daily life mainly because it lacks a close connection to forensic applications. Finally, there has been some interest lately in fusing different biometrics to increase system performance [3,4]. The ease of use and acceptability of hand-based biometrics make hand shape a good candidate in these heterogeneous systems.

In this paper, we propose a novel, peg-free approach to hand-based verification and identification which does not require extracting any landmark points on the hand and it is not sensitive to hand and finger motion. The proposed approach operates on 2D hand images acquired by placing the hand on a planar lighting table without any guidance pegs. There are several important ideas behind the proposed approach. First, to deal with the issue of hand and finger motion, we decompose the silhouette of the hand in different regions corresponding to the back of the palm and fingers. This is performed using a robust methodology based on an iterative morphological filtering scheme. To avoid touching fingers and simplify segmentation, subjects are required to stretch their hand prior to placing it on the lighting table. No other restrictions are imposed on the subjects. Second, in contrast to traditional approaches that represent the shape of the explicitly using hand-crafted geometrical measurements, we represent the geometry of each part of the hand implicitly using high-order Zernike moments [5]. Finally, to improve verification/identification accuracy and robustness, we fuse information from different parts of the hand. It is worth mentioning that the use of high-order moments is not practical for many applications due to their noise sensitivity. However, this is not an issue in the context of our application since we use a robust image acquisition process which provides very high quality hand images as shown in Section 3.

Moments have been used before in a wide range of applications in image analysis, and object recognition [6]. In the area of biometrics, preliminary results have been reported using various types of moments (e.g., geometric, Zernike, pseudo-Zernike, and Legendre moments) for palmprint verification [7,8]. Zernike moments are quite attractive for representing the geometry of the hand due to having minimal redundancy (i.e., employ orthogonal basis functions [9]), providing invariance to translation, rotation, and scale, and demonstrating robustness to noise [6]. In most applications, the use of Zernike moments has been limited to low-orders only or small low-resolution images due to high computational requirements and lack of accuracy due to numerical errors. Capturing the shape of the hand in sufficient detail, however, would require computing moments of rather high-orders. Although there have been several efforts to reduce computational complexity by employing quantized polar coordinate systems, such transformations have an effect on accuracy. In this study, we employ an improved algorithm, proposed in one of our earlier studies [10], which can speed up the computation of high-order Zernike moments without sacrificing accuracy. To keep computational complexity low, we avoid redundant computations by detecting common terms and using look-up tables. To preserve accuracy, we avoid any coordinate transformations and employ arbitrary precision arithmetic.

Fusing information from different biometric modalities (i.e., face, fingerprint, hand) has received considerable attention lately, however, fusing information from different parts of the same biometric has been considered to a lesser extent. For example, Ross and Govindarajan [11] have reported a feature-level fusion scheme which combines hand and face features. Kumar and Zhang [12] have investigated feature selection of hand shape and palm print features. Cheung et al. [13] have proposed a two-level fusion strategy for multimodal biometric verification. Jiang and Su [14] have proposed fusing faces and fingerprints to improve verification accuracy. Our approach is mostly related to component-based approaches in object detection and recognition [15,16], face detection/recognition [17], and person detection [18]. The key idea behind them is representing objects in terms of their parts and geometrical relationships. Among them, the most relevant approach to ours is the face recognition approach reported in [17]. In that study, information from different parts of the face was fused at the feature-level using Support Vector Machines (SVMs) [19]. Here, we report results using several different fusion strategies including feature-level, score-level and decision-level. Earlier versions of our work have appeared in [20–22].

The rest of the paper is organized as follows: Section 2 contains a review of hand-based verification and identification systems. An overview of the proposed approach is presented in Section 3. Section 4.2 reviews Zernike moments and presents an efficient algorithm for computing high-order Zernike moments. Section 5 presents our methodology for separating the hand from the arm and decomposing the hand silhouette in different parts corresponding to the back of the palm and the fingers. Representing the geometry of the shape of the palm and the fingers using Zernike moments is discussed in Section 6. Section 7 presents the fusion strategies investigated in this study. Experimental results and comparisons are presented in Section 8. Finally, Section 9 provides our conclusions and directions for future work.

2. Review of hand-based biometrics

The majority of hand-based biometric systems employ geometric measurements and are based on research limited to considerably old patents and commercial products [23]. In these systems, users are asked to place their hand on a flat surface and align it, with the help of some guidance pegs. The alignment operation simplifies feature extraction and allows for high processing speeds. A mirror is usually used to obtain a side view of the hand using a single camera. In most cases, a few hand-crafted geometric features (e.g., length, width and height of the fingers, thickness of the hand, aspect ratio of fingers and palm, etc.) are extracted, making it possible to construct a small template (i.e., 9 bytes in some commercial systems).

Removal of pegs, to improve convenience, and use of more powerful feature extraction techniques to capture the shape of the hand more accurately represent promising research directions in this area. Several studies have reported that peg-based alignment is not very satisfactory and represents in some cases a considerable source of failure [24,25]. Although peg removal provides a solution to reduce user inconvenience, it also raises more challenging research issues due to the increase in intra-class variance. Nevertheless, most recent studies have concentrated on the design of peg-free systems.

Extracting extremities of the hand contour, such as finger valleys and finger tips, is usually the first processing step in these systems. In peg-free systems, fingers are not guaranteed to be at the same position and orientation at different acquisition times; therefore they need to be segmented and identified in the input images. Analysis of the silhouette contour to locate fingertips and palm–finger intersections, which basically corresponds to curvature local maxima, provides an effective solution to the segmentation problem [26,27]. Once the fingers have been segmented, geometric features such as finger length and width can be measured at predefined points along the finger axes [4,27–29].

Using geometric features helps to reduce storage requirements but can not represent hand shape in detail. Moreover, accurate localization of various landmark points on the fingers is not a straightforward task. Some studies have introduced new features capturing the full finger shape. Jain and Duta [30] have used the silhouette contour of the fingers and an iterative closest point (ICP) alignment algorithm to compute a *shape distance* which is used as a measure of similarity. Ma et al. [31] have followed a similar

approach using B-Spline curves. Xioang et. al. [26] introduced a semi-geometric approach by extracting geometric features after aligning the fingers which are represented using ellipses. Kumar et. al [32] used palmprint and hand geometric features where the extremities of the hand contour were used to measure finger length and palm width. Recently, Yoruk et. al. [33] introduced a more accurate and detailed representation of the hand using the Hausdorff distance of the hand contour, and Independent Component Analysis (ICA)[19]. Their approach requires registering the silhouettes of the hand images using the locations of fingertips and valleys. This study is among a few studies where hand-based biometrics have been demonstrated both for verification and recognition purposes.

A marginally different feature extraction approach, which involves reconstructing the 3D surface of the hand, was proposed in [34]. Using a range sensor to reconstruct the dorsal part of the hand, local shape index values of the fingers were used as features in matching. In a related study, Lay [35] projected a parallel grating onto the dorsal part of the hand to extract features that indirectly capture 3D shape information. Use of multiple enrollment templates is an effective method to improve the recognition accuracy of any biometric system. In the hand-based biometrics domain, using multiple enrollment templates is vital part of any system mainly due to the lower distinctiveness of hand shape. User-specific statistical models, such Mixtures of Gaussians [36,37], have shown to improve system accuracy [27,29,36,37].

3. System overview

Fig. 1 shows the main stages of the proposed system. Our image acquisition system consists of a VGA resolution CCD camera and a flat lighting table, which forms the surface for placing the hand. The direction of the camera is perpendicular to the lighting table as shown in Fig. 2(a). The camera has been calibrated to remove lens distortion. In practical settings, both the camera and the lighting table can be placed inside a box to completely eliminate light interferences from the surrounding environment. Also, the whole system can be made much smaller than the one shown in Fig. 2(a) which is very bulky and was built for experimentation reasons only. Nevertheless, the experimental set up in our laboratory, shown in Fig. 2(a), provides high quality images without requiring us to put much effort to control the environment. It should be mentioned that capturing high quality hand images is critical for

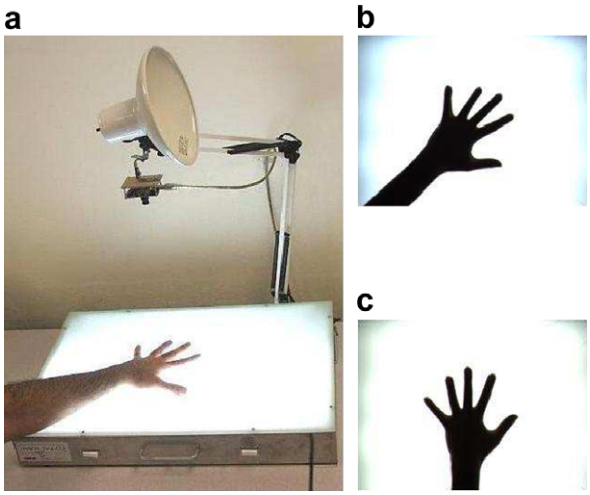

Fig. 2. (a) Image acquisition system, (b and c) images of the same hand acquired by the system.

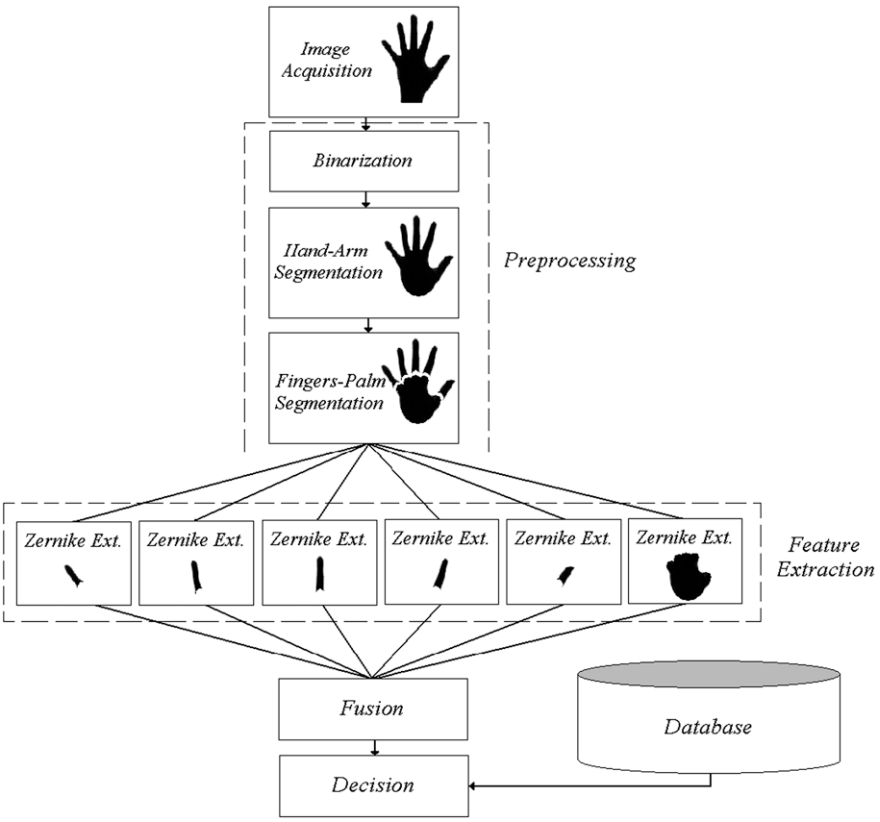

Fig. 1. Main stages of the proposed system.

our application as it allows us to use high-order Zernike moments without worrying about noise sensitivity issues.

When users place their hand on the surface of the lighting table, an almost binary, shadow and noise free, silhouette of the hand is obtained as shown in Fig. 2(b) and (c). During acquisition, subjects are required to stretch their hand and place it inside a rectangular region marked on the surface of the table. This is to avoid touching fingers, ensure the visibility of the whole hand, and avoid perspective distortions. No restrictions were imposed on the orientation of the hand. The image acquired is then binarized and goes through the segmentation module. During segmentation, the arm is separated from the hand and discarded from further processing. Then, the hand is further processed to segment the palm and the fingers. Feature extraction is performed by computing the Zernike moments of each part of the hand separately. The resulting representation is invariant to translation, rotation and scaling. Verification/identification is performed by fusing information from different parts of the hand. We have experimented with different fusion strategies including feature-level, score-level, and decision-level fusion. We employ multiple enrollment templates per subject and compute similarity scores using the minimum distance between a query image and the templates of the subjects.

Next, we present the algorithm adopted in this study for the efficient computation of high-order moments. Then, we describe in detail each stage of the proposed system.

4. Efficient computation of high-order zernike moments

4.1. Motivation and issues

The majority of peg-free systems extract a number of landmark points on the hand (e.g., finger joints) and represent hand shape by explicitly measuring certain geometric features. Alternatively, one can imagine utilizing more general shape descriptors to provide a richer representation of the shape of the hand, replacing the conventional hand-crafted geometric features. In this study, we propose representing the geometry of the hand implicitly using Zernike moments. Zernike moments have the potential to provide a more powerful representation of the shape of the hand, however, capturing important shape details for verification/identification purposes would require computing high-order moments. This bring up the issues of speed and accuracy.

Although there exist some fast algorithms that rely on approximate polar coordinate transformations [38–40], they do not yield satisfactory results in the context of our application due to lack of accuracy. To deal with these issues, we have adopted an efficient algorithm that keeps computational complexity low without sacrificing accuracy. To preserve accuracy, our algorithm avoids any form of coordinate transformation by using arbitrary precision arithmetic. To reduce computational complexity, it avoids recomputing common terms and employs look-up tables.

4.2. Review of zernike moments

Zernike moments are based on a set of complex polynomials that form a complete orthogonal set over the interior of the unit circle [5]. They are defined as the projection of the image on these orthogonal basis functions. Specifically, the basis functions $V_{n,m}(x,y)$ are given by

$$V_{n,m}(x,y) = V_{n,m}(\rho,\theta) = R_{n,m}(\rho)e^{jm\theta} \tag{1}$$

where n is a non-negative integer, m is a non-zero integer subject to the constraints $n-|m|$ is even and $|m|<n$, ρ is the length of the vector from origin to (x,y), θ is the angle between the vector ρ and the x-axis in a counter clockwise direction, and $R_{n,m}(\rho)$ is the Zernike radial polynomial which is defined as follows:

$$R_{n,m}(\rho) = \sum_{k=|m|,n-k=even}^{n} \frac{(-1)^{\frac{n-k}{2}}\frac{n+k}{2}!}{\frac{n-k}{2}!\frac{k+m}{2}!\frac{k-m}{2}!}\rho^k = \sum_{k=|m|,n-k=even}^{n} \beta_{n,m,k}\rho^k \tag{2}$$

Note that $R_{n,m}(\rho) = R_{n,-m}(\rho)$. The basis functions in Eq. (1) are orthogonal, therefore, satisfying the constraint:

$$\frac{n+1}{\pi}\int\int_{x^2+y^2\leqslant 1} V_{n,m}(x,y)V^*_{p,q}(x,y) = \delta_{n,p}\delta_{m,q} \tag{3}$$

where

$$\delta_{a,b} = \begin{cases} 1 & \text{if } a=b \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

The Zernike moment of order n with repetition m for a *digital* image function $f(x,y)$ is given by [41]:

$$Z_{n,m} = \frac{n+1}{\pi}\sum\sum_{x^2+y^2\leqslant 1} f(x,y)V^*_{n,m}(x,y) \tag{5}$$

where $V^*_{n,m}(x,y)$ is the complex conjugate of $V_{n,m}(x,y)$. To compute the Zernike moments of a given image, the center of mass of the object is taken to be the origin. The magnitude of the Zernike moments is rotation invariant by its definition (see Eq. (5)). Taking the center of mass of the object as the origin of the coordinate system makes them translation invariant as well. Additionally, to provide scale invariance, the object is scaled inside the unit circle.

The function $f(x,y)$ can then be reconstructed by the following expression [41]:

$$\tilde{f}(x,y) = \sum_{n=0}^{N}\frac{C_{n,0}}{2}R_{n,0}(\rho) + \sum_{n=1}^{N}\sum_{m>0}(C_{n,m}cosm\theta + S_{n,m}sinm\theta)R_{n,m}(\rho) \tag{6}$$

where N is the maximum order of Zernike moments used, while $C_{n,m}$ and $S_{n,m}$ denote the real and imaginary parts of $Z_{n,m}$, respectively.

4.3. Computation of high-order zernike moments

A method to improve the speed of Zernike moments computation involves using a quantized polar coordinate system. In [38], Mukundan and Ramakrishnan proposed a recursive algorithm for the computation of Zernike and Legendre moments using polar coordinates. In [39], Belkasim et al. introduced a different recursive algorithm using radial and angular expansions of Zernike orthonormal polynomials. For an $M\times M$ image, the angles were quantized to $4M$ and the radii were quantized to M levels. In a more recent study, Gu et al. [40] employed the "square to circle" transformation of Mukundan and Ramakrishnan [38] and more efficient recursive relationships to develop an even faster algorithm, however, its accuracy was still limited to that of [38] due to the quantization step in the coordinate transformation.

A side effect of quantization is that errors are introduced in the computation of high-order Zernike moments (see Section 4.4). In this work, we have adopted a novel algorithm which avoids using any quantization, therefore, the computation of the moments is as accurate as in the traditional approach (i.e., no approximations). To save computation time, the improved algorithm finds the terms that occur repeatedly in various orders and avoids recomputing them. Additional computations can be saved using a look-up table. To ensure high accuracy, it uses arbitrary precision arithmetic.

Specifically, by substituting Eqs. (2) and (1) in Eq. (5) and reorganizing the terms, the Zernike moments can be computed as follows:

$$Z_{n,m} = \frac{n+1}{\pi} \sum_{x^2+y^2 \leqslant 1} \sum \left(\sum_{k=|m|}^{n} \beta_{n,m,k} \rho^k \right) e^{-jm\theta} f(x,y)$$
$$= \frac{n+1}{\pi} \sum_{k=|m|}^{n} \beta_{n,m,k} \left(\sum_{x^2+y^2 \leqslant 1} \sum e^{-jm\theta} \rho^k f(x,y) \right)$$
$$= \frac{n+1}{\pi} \sum_{k=|m|}^{n} \beta_{n,m,k} \chi_{m,k} \quad (7)$$

The terms $\chi_{m,k}$, defined in Eq. (7), are common in the computation of moments having the same repetition as shown in Fig. 3 in the case of repetition $m = 0$. In general, to compute Zernike moments up to order N, we would need to compute $\chi_{m,k}$ for each repetition. However, computing $\chi_{m,k}$ only once would be sufficient for computing Zernike moments of any order and any repetition by simply taking linear combinations as shown in Eq. (7). As an example, Table 1 shows the terms $\chi_{m,k}$ required to be computed up to order 10. Moreover, the coefficients $\beta_{n,m,k}$ (see Eq. (2)) do not depend on the input image or the coordinates; therefore, they can be stored in a small look-up table to save additional computations.

An important issue in the computation of high-order Zernike is the issue of numerical precision. Depending on the image size and maximum order, double precision arithmetic does not provide enough precision and serious numerical errors can be introduced in the computation of the moments. This is demonstrated in Table 2 which shows the differences between Zernike moments up to order 50, computed using double precision and arbitrary precision arithmetic for a 300×300 image. As it can be observed, the error becomes more and more significant with increasing order and decreasing repetition.

Fig. 4 shows the effect of numerical errors on the orthogonality of the basis functions. As it can be observed in Fig. 4(a), obtained using double precision arithmetic, the orthogonality of the basis functions is violated seriously. On the other hand, the orthogonality of the basis functions is preserved using arbitrary precision arithmetic as shown in Fig. 4(b) (i.e., only one delta peak is present).

$$\begin{aligned}
Z_{0,0} &= \beta_{0,0,0}\boxed{\chi_{0,0}} \\
Z_{2,0} &= \beta_{2,0,0}\chi_{0,0} + \beta_{2,0,2}\boxed{\chi_{0,2}} \\
Z_{4,0} &= \beta_{4,0,0}\chi_{0,0} + \beta_{4,0,2}\chi_{0,2} + \beta_{4,0,4}\boxed{\chi_{0,4}} \\
Z_{6,0} &= \beta_{6,0,0}\chi_{0,0} + \beta_{6,0,2}\chi_{0,2} + \beta_{6,0,4}\chi_{0,4} + \beta_{6,0,6}\boxed{\chi_{0,6}} \\
Z_{8,0} &= \beta_{8,0,0}\chi_{0,0} + \beta_{8,0,2}\chi_{0,2} + \beta_{8,0,4}\chi_{0,4} + \beta_{8,0,6}\chi_{0,6} + \beta_{8,0,8}\boxed{\chi_{0,8}} \\
Z_{10,0} &= \beta_{10,0,0}\chi_{0,0} + \beta_{10,0,2}\chi_{0,2} + \beta_{10,0,4}\chi_{0,4} + \beta_{10,0,6}\chi_{0,6} + \beta_{10,0,8}\chi_{0,8} + \beta_{10,0,10}\boxed{\chi_{0,10}}
\end{aligned}$$

Fig. 3. Common terms in the computation of Zernike moments up to order 10 with zero repetition.

Table 1
The terms $\chi_{m,k}$ required to be computed up to order 10.

Repetition m	$\chi_{m,k}$
0	$\chi_{0,0}, \chi_{0,2}, \chi_{0,4}, \chi_{0,6}, \chi_{0,8}, \chi_{0,10}$
1	$\chi_{1,1}, \chi_{1,3}, \chi_{1,5}, \chi_{1,7}, \chi_{1,9}$
2	$\chi_{2,2}, \chi_{2,4}, \chi_{2,6}, \chi_{2,8}, \chi_{2,10}$
3	$\chi_{3,3}, \chi_{3,5}, \chi_{3,7}, \chi_{3,9}$
4	$\chi_{4,4}, \chi_{4,6}, \chi_{4,8}, \chi_{4,10}$
5	$\chi_{5,5}, \chi_{5,7}, \chi_{5,9}$
6	$\chi_{6,6}, \chi_{6,8}, \chi_{6,10}$
7	$\chi_{7,7}, \chi_{7,9}$
8	$\chi_{8,8}, \chi_{8,10}$
9	$\chi_{9,9}$
10	$\chi_{10,10}$

4.4. Comparisons with other methods

We have compared the accuracy of our algorithm with several other algorithms [38–40] using the fidelity of reconstruction as a criterion. The test image that we used in our experiments is shown in Fig. 5(top). This is a 64×64 image and Zernike moments up to order 40 were utilized for reconstruction. Fig. 5(a–c) show the results using Mukundan's method [38], Gu's method [40] and our method, respectively. As it can be observed, the former two algorithms give poor reconstructions mainly because of the square to circle transformation. The effect of the transformation is clearly visible in the reconstructed images.

The reconstruction results using Belkasim's [39] method and Zernike moments up to order 60 is shown in Fig. 6(a) while the reconstruction results using our method is shown in Fig. 6(b). We used arbitrary precision arithmetic in the implementation of Belkasim's method to make the comparison fair. It can be observed that Belkasim's method introduces some distortions at the edges while our method produces smoother edges in general.

To make the differences between the two methods more clear, we have computed reconstruction errors for each method, shown in Table 3, using different orders. The formula use to compute the error is shown below:

$$\varepsilon_r = \frac{\sum_x \sum_y |\tilde{f}(x,y) - f(x,y)|^2}{\sum_x \sum_y f(x,y)^2} \quad (8)$$

where $f(x,y)$ is the original image and $\tilde{f}(x,y)$ is the reconstructed image (up to order N).

In general, it would be reasonable to expected that the reconstruction error decreases as the order of moments increases. Our method exhibits this behavior, however, Belkasim's method behaves quite differently which indicates that the quantization of the polar coordinates has a serious effect on the computation of higher-order moments.

Table 4 shows the number of multiplications and additions required by each method. We have assumed an image of size $M \times M$ pixels and Zernike moments up to order N. In the case of our method, first we need M^2N multiplications to compute $\rho^k f(x,y)$, $k = 0, 1, \ldots, N$. Then, we must compute $\chi_{m,k} = \sum_x \sum_y e^{-jm\theta} \rho^k f(x,y)$. The number of $\chi_{m,k}$ required to compute Zernike moments up to order $N(even)$ is $\frac{N}{2}(\frac{N}{2}+1)$. Since there is no need for any multiplication when $m = 0$ and $\chi_{m,k}$ is a complex number, this step requires $M^2N(\frac{N}{2}+1)$ multiplications and $2(M^2-1)(\frac{N}{2}+1)^2$ additions. For large N and M, the number of mul-

Table 2
Differences between Zernike moments up to order 50, computed using double precision and arbitrary precision arithmetic for a 300×300 image.

Order, repetition	0	2	4	6	8	10	...	40	42	44	46	48	50
42	7.28e−4	6.60e−4	1.91e−4	2.72e−4	1.72e−4	6.54e−6	...	1.17e−17	3.82e− 17				
44	3.50e−3	5.57e−3	1.11e−3	1.18e−3	1.05e−4	1.49e−4	...	1.52e−15	1.30e−17	1.04e−17			
46	3.97e−1	6.48e−3	5.25e−3	2.04e−3	2.57e−3	1.07e−3	...	2.12e−14	1.48e− 15	9.06e−17	2.60e−18		
48	1.86e0	6.91e−2	4.39e−2	2.83e−2	1.66e−2	3.50e−3	...	5.23e−14	5.92e−14	3.11e−16	1.20e−16	3.47e−18	
50	1.38e1	1.81e0	1.06e−1	9.39e−2	6.92e−2	7.12e−2	...	7.52e−12	2.67e−13	1.60e−14	8.60e−16	4.65e−17	2.17e−18

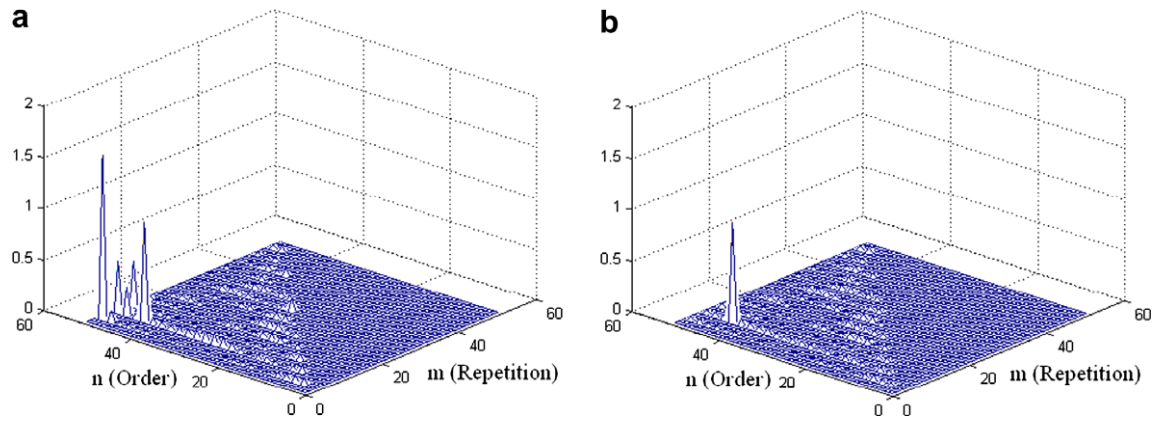

Fig. 4. Dot product between basis function of order $n = 43$ and repetition $m = 7$ with other basis functions up to order 50 using (a) double precision arithmetic and (b) arbitrary precision arithmetic.

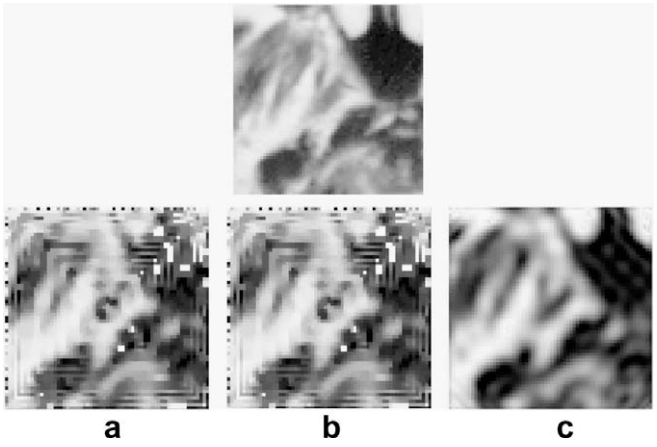

Fig. 5. Original (top) and reconstructed images using moments of order up to 40: (a) Mukundan's method, (b) Gu's method, and (c) our method.

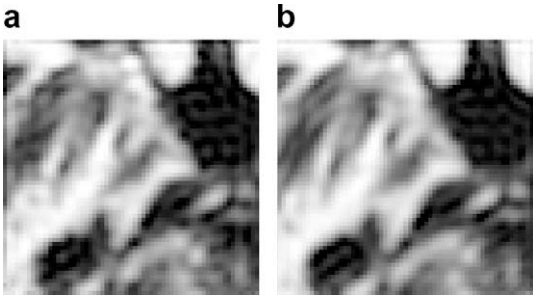

Fig. 6. Reconstructed images using moment of order up to 60: (a) Belkasim's method, and (b) our method.

Table 3
Reconstruction error using our method and Belkasim's method.

Order	Our method	Belkasim's method
35	0.0647	0.0648
40	0.0621	0.0628
45	0.0596	0.063
50	0.0370	0.0557
55	0.0203	0.0645
60	0.0133	0.0665

Table 4
Computational complexity of various methods.

	Number of addition	Number of multiplication
Mukundan's method	$\frac{N^2M^2}{2} + \frac{1}{8}NM^3$	$2N^2 + N^2M^2 + \frac{1}{4}MN^3$
Belkasim's method	$N(M+2)(M-1)$	$\frac{N^2M^2}{2} + 2MN$
Gu's method	$\frac{3}{8}N^2M + 2NM^2 + \frac{1}{12}N^3M + \frac{1}{4}N^2M^2$	$\frac{N^2M}{2} + 2M^2N$
Our method	$2(\frac{N}{2}+1)^2(M^2-1)$	$\frac{N^2M^2}{2} + 2M^2N$

tiplications and additions required to compute $Z_{n,m}$ is negligible according to Eq. (7). Asymptotically, our method has comparable computational complexity with Belaksim's method (i.e., $O(N^2M^2)$ multiplications) although Belaksim's method performs less additions (i.e., $O(NM^2)$ versus $O(N^2M^2)$).

5. Component-based hand representation

This stage includes the binarization of the hand image and its segmentation into different regions corresponding to the arm, hand, palm, and fingers. Our current set up yields very high quality images, which are almost free of shadows and noise. As a result, binarization can be performed using a fixed threshold. To separate the forearm from the hand, first we detect the palm by finding the largest circle that can be prescribed inside the hand–arm silhouette. Then, we take the intersection of the forearm with the circle's boundary. To separate the fingers from the palm, first we filter out the fingers using morphological closing [42]. Then, the palm is subtracted from the hand silhouette. Specific details are provided below.

5.1. Binarization

The hand images can be captured using a grayscale camera; however, we used a color CCD camera as it was already available in our laboratory. To obtain a grayscale image, we used the luminance value $Y_{i,j}$ of each pixel (i,j):

$$Y_{i,j} = 0.299R_{i,j} + 0.587G_{i,j} + 0.114B_{i,j} \tag{9}$$

where $R_{i,j}, G_{i,j}, B_{i,j}$ denote the RGB values of the pixel. Fig. 7(a) and (b) show the original color and grayscale images, respectively. The binary value $B_{i,j}$ of a pixel (i,j) was calculated as follows:

$$B_{i,j} = \begin{cases} 1 & \text{if } Y_{i,j} < T \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

where T is a fixed threshold which was determined experimentally. In all of the experiments reported in this study, $T = 0.5$ was used.

Fig. 7. (a) Color image, (b) grayscale image and (c) binarized image.

Fig. 7(c) shows the output of the binarization process. The resulting silhouettes are very accurate and consistent due to the image acquisition set up. This is critical as high-order Zernike moments are sensitive to small changes in silhouette shape.

5.2. Hand–arm segmentation

The binary silhouette obtained during image acquisition is the union of the hand with the arm. The arm does not have many distinctive features while its silhouette, at different acquisition sessions, is not expected to be the same due to clothing and freedom in hand placement (see Fig. 2(b) and (c)). To segment the arm, we assume that the user is not wearing very loose clothing on the arm. Under this assumption, the palm becomes the thicker region of the silhouette, which enables its detection it by finding the largest circle that can be prescribed inside the silhouette. We use a robust, iterative morphological closing algorithm based on a circular structuring element [42] to find the largest circle. The main steps of our algorithm can be summarized as follows:

(1) Initialize the radius of the circular structuring element D to a large value (e.g., $R = 85$).
(2) Apply morphological closing on the image using D.

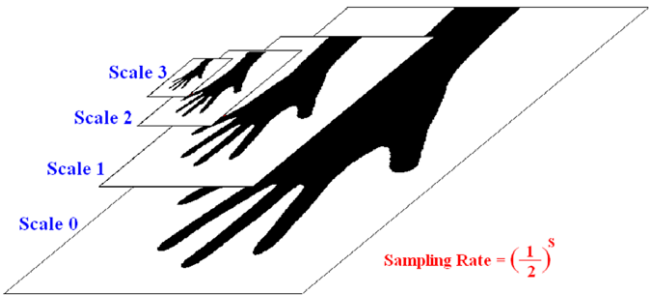

Fig. 8. Illustration of the multi-resolution hierarchy of hand images for reducing the computational cost of the hand–arm segmentation module. The hierarchy is obtained by down-sampling the binarized image three times, each time by a factor of two.

(3) If the output is an empty image, then set $R = R - 1$ and go to Step 2; otherwise, the resulting image corresponds to the largest circle inside the silhouette.

This algorithm has shown to work well in our experiments, however, its main drawback is that it is time consuming due to its iterative nature and the use of morphological operators. This is especially true when the size of the hand is relatively small. For example, it requires 29 morphological closing operations on a 480×640 assuming that the radius of the largest circle inside the hand silhouette is about 57 pixels which is typical for smaller hands. Implementing the algorithm in MATLAB 7.4.0 on a 3.19 GHz 64-bits machine with 2 GB of RAM, it would take more than 6 s to segment the hand and forearm in this case.

One way to speed up processing is by reducing the number of iterations. This can be done by initializing the radius of the structuring element D more conservatively. To address this issue, we have developed a multi-resolution scheme which operates on different resolution images of the hand. First, the largest circle inside the hand silhouette is found approximately but fast using a low-resolution image of the hand. Next, to refine the position and size of the circle found, the same process is repeated at a higher resolution. To reduce the number of iterations, the radius of the structuring element at higher resolutions is initialized using the radius of the circle found at lower resolutions. This process is repeated until the highest resolution hand image (i.e., original hand image) is processed.

To represent a hand image at different resolution levels, we scale it down by simply down-sampling it. Fig. 8 illustrates the hierarchy of hand images obtained by down-sampling the input image three times, each time by a factor of two. The multi-resolution algorithm can be summarized as follows:

(1) Generate a hierarchy of different resolution hand images by down-sampling the input image.
(2) Initialize the radius of the circular structuring element D to a small value (e.g., $R_0 = 11$).
(3) Consider the lowest resolution hand image.
(4) Find the largest circle (R_{max}) inside the hand silhouette.

Fig. 9. (a) Binarized image, (b) largest circle that can be prescribed inside the hand–arm silhouette, and (c) segmented hand silhouette.

(5) Set the radius of the circular structuring element D to $2 \times R_{max} + 2$.
(6) If a higher resolution image is available, go to Step 4; otherwise, stop.

In our experiments, the multi-resolution hierarchy contains four levels, that is, we scale down the original image three times. The algorithm starts by processing the lowest resolution hand image which is eight times smaller than the original one. At this level, the largest circle prescribed inside the hand silhouette can be found very quickly (i.e., typically, within 4–5 iterations). When considering higher resolutions, the number of iterations stays low by initializing the radius of the structuring element conservatively based on the size of the circle found at lower resolutions. Considering the small hand example mentioned earlier, it takes five iterations at the lowest resolution image and only two iterations at the highest resolution (i.e., original) image. Overall, segmenting the hand and forearm reduces processing time from 6 to 0.69 s for this example. The average processing time on 250 sample images was 0.58 s.

Fig. 9(b) shows the output of the algorithm on the sample image shown in Fig. 9(a). Once the largest circle has been found, the arm can be segmented by finding its intersection with the circle and the boundary of the hand–arm region. Fig. 9(c) shows the resulting hand silhouette after discarding the arm region.

5.3. Palm–finger segmentation

To simplify finger segmentation, subjects were instructed to stretch their hand during image acquisition in order to avoid touching fingers; however, finger motion was unavoidable. Several sample images collected from the same subject are shown in Fig. 2. As it can be observed, the relative position of the fingers varies significantly from sample to sample. The processing steps of the finger segmentation module are shown in Fig. 10. First, a morphological closing operator based on a circular disk is applied on the hand image as shown in Fig. 10(a). The radius of the structuring element was experimentally set to 25 pixels, making it thicker than the widest finger in our database. The closing operation filters out the fingers from the silhouette as shown in Fig. 10(c). The remaining part of the silhouette corresponds to the palm, which is subtracted from the hand image to obtain the finger segments as shown in Fig. 10(d). It should be mentioned that an alternative way to segment the fingers from palm is by detecting certain landmark points on the hand, such as fingertips and valleys. This solution, however, would be more prone to errors due to inaccuracies in landmark detection.

To identify each of the fingers quickly, we assume that hand rotations are less than 45°. In our prototype system, larger rotations would correspond to purposeful, unnatural hand placement by the users. In general, it would be possible to deal with larger rotations by using additional information for each finger such as length, width, aspect ratio, and area. To extract each finger and the back of the palm, we use connected component analysis [43].

5.4. Post-processing of finger regions

A closer examination of the results shown in Fig. 10(d) reveal that the segmented fingers have sharp tails at the locations where they meet the palm. The curvature of the hand contour at these locations is smoother for the little, point, and thumb fingers as shown in Fig. 11(left). As a result, there might be significant differences in the length of the tails corresponding to these fingers as shown in Fig. 11(a), where different samples from the same subject are shown. In some cases, especially when the hand is small (i.e., mostly for female hands), there are significant differences in the length of the tails, which introduces significant errors in the computation of the Zernike moments. This can be illustrating by

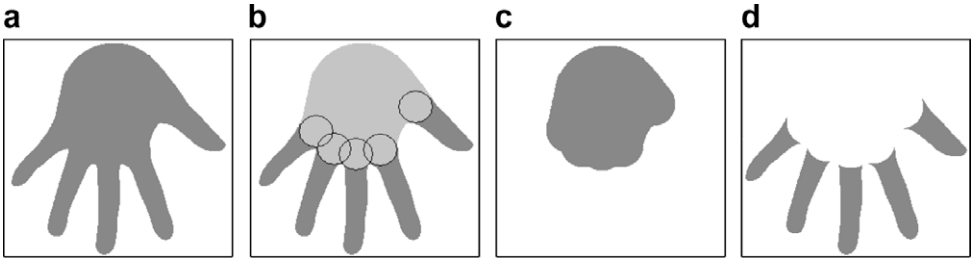

Fig. 10. (a) Hand silhouette, (b) structuring element, (c) the result of morphological closing and (d) the result of subtracting the back of the palm from the hand silhouette.

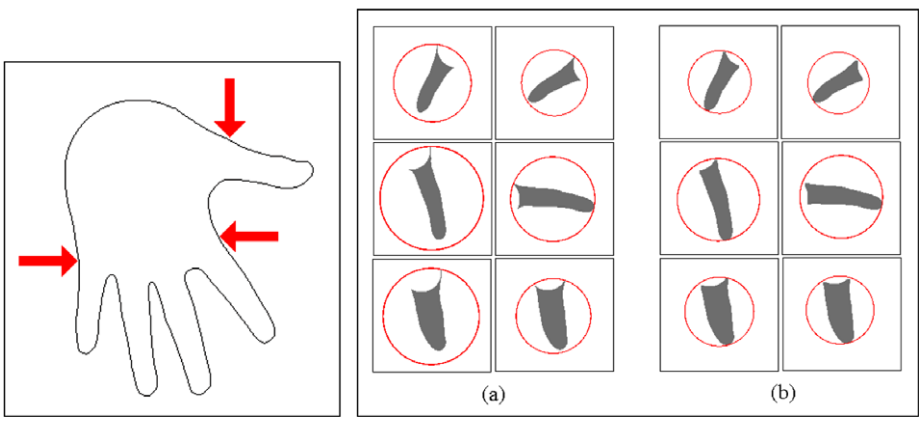

Fig. 11. (Top) The junctions of finger with the palm in the hand counter where their curvature is too smoother than the others. (Right) Pairs of segmented little, point, and thumb fingers. Each pair corresponds to two different samples of the same subject. (a) Before applying the additional step, and (b) after applying the additional step.

Table 5
The effect of the extra morphological closing operator on the normalized distances between the Zernike moments (up to order 20) of the segmented finger pairs before (Fig. 11(a)) and after (Fig. 11(b)) the extra step.

Pair of fingers	d_{before}	d_{after}
Little	0.5904	0.0901
Point	0.7881	0.1135
Thumb	0.7424	0.1253

Table 6
Mean and *Variance* of matching distances for each finger before and after post-processing.

Finger	μ_{before}	μ_{after}	σ_{before}	σ_{after}
Little	0.1724	0.0998	0.1039	0.0873
Ring	0.1085	0.0817	0.1004	0.0952
Middle	0.0823	0.0810	0.0983	0.0995
Point	0.1928	0.0716	0.1287	0.0886
Thumb	0.1843	0.1205	0.1262	0.0843

Table 7
Mean and *Variance* of non-matching distances of each finger before and after post-processing.

Finger	μ_{before}	μ_{after}	σ_{before}	σ_{after}
Little	0.3564	0.2869	0.1162	0.0341
Ring	0.3072	0.2861	0.0531	0.0289
Middle	0.2859	0.2870	0.0290	0.0268
Point	0.3672	0.2715	0.1454	0.0248
Thumb	0.4136	0.3120	0.1216	0.0616

observing the differences in the size of the circles enclosing the fingers in Fig. 11(a) versus those in Fig. 11(b) where the tails have been removed using post-processing.

To keep these errors as low as possible, we post-process each finger by applying an extra morphological closing step as shown in Fig. 11(b). The structuring element was chosen experimentally to be a simple 4 by 4 square with values set to one. Table 5 illustrates the effect of this step by showing the normalized distances between the pairs of corresponding fingers shown in Fig. 11. Obviously, this step improves matching scores considerably.

Tables 6 and 7, present statical results (i.e., mean and variance) to further support the benefits of this step in terms of matching and non-matching distances. For each finger, we have computed all possible matching and non-matching distances in our database, before and after post-processing, using Zernike features up to order 20 (121 features). Since our database contains 101 people with 10 images per person, there were 4545 matching distances and 1,010,000 non-matching distances for each finger. Our results indicate that finger post-processing reduces the overlap between matching and non-matching distances significantly in the case of the little, point, and thumb fingers; however, it does not seem to have an important effect in the case of the ring and middle fingers. This was an expected result since there are more segmentation problems with these fingers due to their greater motion freedom.

6. Feature extraction

In this step, we represent the geometry of the back of the palm and the fingers implicitly using Zernike moments. A critical issue at this stage is choosing the order of Zernike moments appropriately in order to capture sufficient shape information for verification and identification purposes. Our experimental results indicate that capturing important shape details for verification/identification purposes requires using high-order moments.

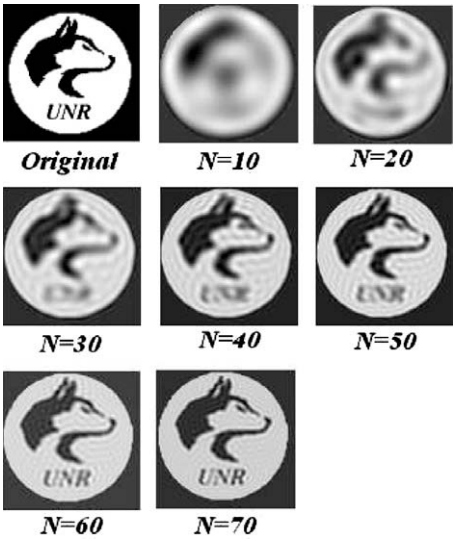

Fig. 12. Original and reconstructed images using different orders of Zernike moments.

In general, using very high-order moments would preserve more and more information. Fig. 12 demonstrates this idea using a 300×300 binary image, shown at the top left corner, which contains information at various levels of detail. The reconstructed images using moments up to order 20 contain only a rough silhouette of the wolf. The reconstructions using moments up to order 50 show the head of wolf clearly, however, the letters in the logo are still blurred. Using orders up to 70 improves the reconstruction of the letters in the logo as well. Using very high-orders is not practical, however, due to information redundancy and computational complexity issues. Moreover, Liao and Pawlak [44] have shown that there is an inherent limitation in the precision of arbitrary high-order Zernike moments due to the circular geometry of the domain.

Here, we used the average reconstruction error (i.e., Eq. (8)) on a large number of palm and finger images to decide the appropriate order for our application. Our objective was to preserve important details while keeping the orders as low as possible. Specifically, by analyzing the reconstruction error, the maximum order chosen for the fingers was 20 while the maximum order chosen for the back of the palm was 30. Fig. 13(a) shows several finger reconstructions using different orders. Fig. 13(b) shows the reconstruction error using different orders in this case. As it can be observed, the error almost saturates for orders higher than 40. This is also visually evident from the finger reconstructions shown in Fig. 13(a). In general, using orders higher than 20 does not offer major improvements. Therefore, to keep computational cost low, the highest order chosen in the case of fingers was 20. Similar experiments and analysis in the case of the back of the palm revealed that the highest order useful for verification/identification purposes was 30. It should be mentioned that the reconstruction criterion used here to select the order of Zernike moments might not yield the most discriminant moments [45]. In the future, we plan to investigate feature selection schemes [46] to select a subset of discriminant Zernike moments for each part of the hand.

Using a similar analysis to represent the geometry of the whole hand, we found that orders as high as 70 were required. Fig. 14(a) shows a hand image while Fig. 14(b) shows several reconstructions using different orders. The reconstruction error is shown to the right of Fig. 14. Clearly, using a component-based representation of the hand offers major computational savings.

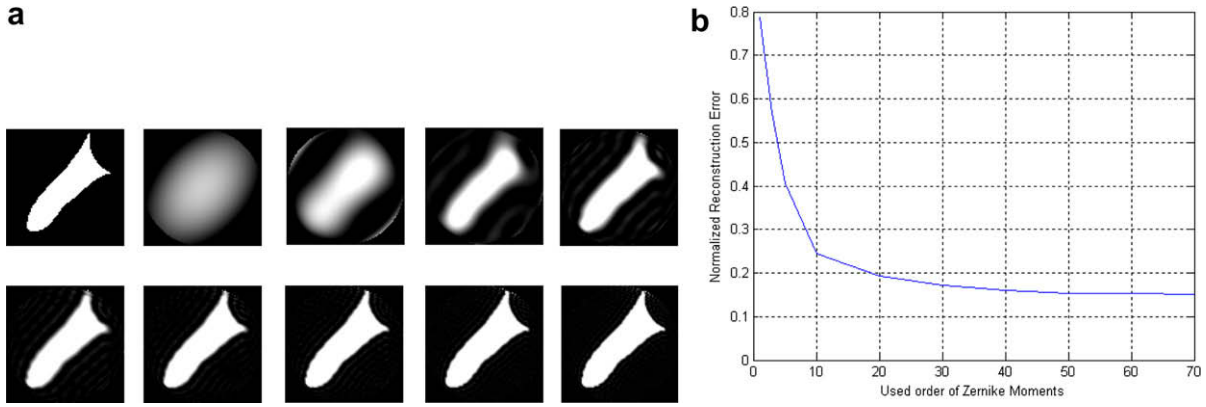

Fig. 13. (a) Original image (top left) and reconstructed images (left to right, top to bottom) up to order 2, 5, 10, 20, 30, 40, 50, 60, and 70, (b) reconstruction error.

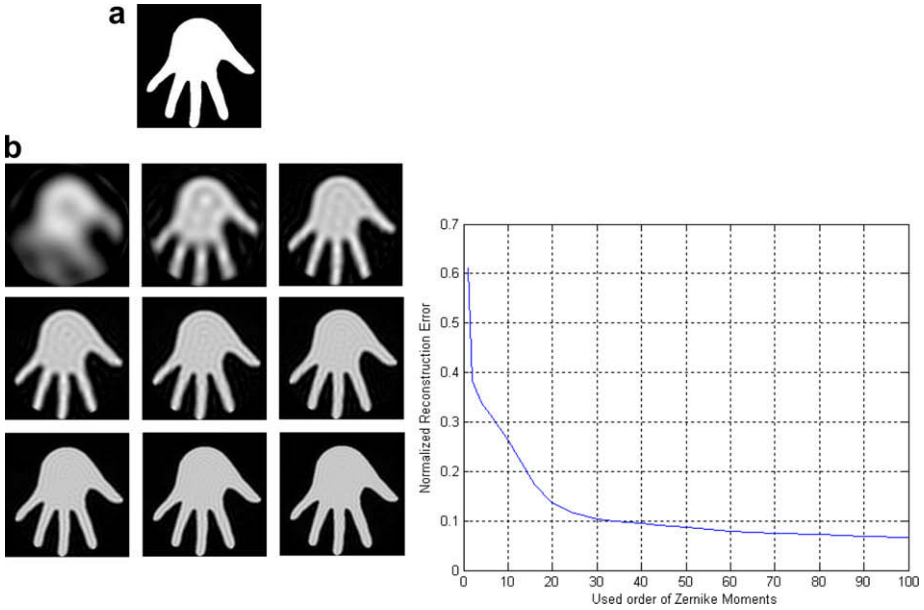

Fig. 14. (a) Original and (b) reconstructed images (left to right, top to bottom) up to order 10, 20, 30, 40, 50, 60, 70, 80, and 90, (right) reconstruction error.

Computing very high-order Zernike moments is quite computationally expensive, especially when precision is a requirement. The algorithm proposed in Section 4.2 was initially implemented in C++ using arbitrary precision arithmetic (i.e., 200 digits) on a 2.66 GHz pentium *IV* with 256 MB memory. In this case, it takes about 6 min to compute Zernike moments up to order 70, while it only takes 35 s to compute Zernike moments up to order 30. We have verified experimentally that moments up to order 30 can be computed quite accurately without resorting to arbitrary precision arithmetic. In our application, using double precision instead of arbitrary precision to compute moments up to order 36 yields an error less than 0.5%. Using double precision in C++ on a 3.19 GHz 64-bits machine with 2 GB of RAM, it takes less than 0.01 s on the average to compute moments up to order 30. The time savings using double precision are significant and can be further improved by computing the Zernike moments of different parts of the hand in parallel. In practice, a hybrid implementation can be employed where the use of arbitrary precision arithmetic is restricted only to orders higher than 36, therefore, reducing computational complexity. It should be mentioned that using feature selection [46] to choose a subset of discriminant Zernike moments, as mentioned earlier, will further decrease time requirements. Hardware implementations could also be considered for real time applications [47,48].

7. Fusion

At this step, we fuse information from different parts of the hand to improve verification/identification accuracy and robustness. In general, fusion can be implemented at different levels. In this paper, we have experimented with three different fusion strategies: *feature-level*, *score-level*, and *decision-level* fusion.

In feature-level fusion, the features extracted from the fingers and the back of the palm can be fused to create a more compact and powerful feature set. Commonly, feature-level fusion is performed using dimensionality reduction or feature selection [45]. In score-level fusion, the matching scores of the fingers and the palm can be fused to obtain an overall score. The sum rule or the weighted-sum rule are common score-level fusion techniques [49]. In decision-level fusion, verification/identification results based on different parts of the hand can be fused to obtain an overall authentication decision. Majority voting, and AND/OR rules are widely used decision-level fusion techniques [49]. We provide more details in the following subsections.

7.1. Feature-level fusion using Principal Component Analysis

Using Principal Components Analysis (PCA) [45] for feature-level fusion is a very common approach. According to this approach, the feature vectors of the back of the palm and the fingers are combined into a single feature vector. Then, PCA is applied to map them into a lower dimensional space. The resulting PCA features are linear combinations of the original finger and palm features.

7.2. Score-level fusion using weighted sum

The weighted-sum rule has been extensively investigated in the literature and it is probably the most straightforward fusion strategy at the score-level. First, we compute matching scores between corresponding parts of the hand (i.e., back of the palm and fingers) in the query and the template. Then, the matching scores are combined into a single score using a weighted-sum rule as shown below:

$$S(Q,T) = \sum_{i=1}^{6} \alpha_i S(Q_i, T_i) \tag{11}$$

where S is the similarity measure (e.g., Euclidean distance) between the query Q and the template T. Q_i and T_i represent the ith part of the hand, $i = 1, 2, \ldots, 6$. In our system, the first five parts correspond to the little, ring, middle, point and thumb fingers while the sixth part corresponds to the back of the palm. The parameters α_i are the weights associated with the ith part of the hand; they need to satisfy the following constraint:

$$\sum_{i=1}^{6} \alpha_i = 1 \tag{12}$$

The key issue with this method is determining a set of appropriate weight values.

7.3. Score-level fusion using Support Vector Machines

A Support Vector Machine (SVM) is a binary classifier that maps input patterns X to output labels $y \in -1, 1$ [45]. In general, an SVM has the following form:

$$f(X) = \sum_{i \in \Omega} \alpha_i y_i K(X, X_i) + b \tag{13}$$

where α_i are the Lagrange multipliers, Ω corresponds to the indices of the support vectors for which $\alpha_i \neq 0$, b is a bias term, X is an input vector, and $K(X, X_i)$ is a kernel function. Classification decisions are based on whether the value $f(X)$ is above or below a threshold. We have employed SVM to implement an alternative score-level fusion strategy. Given a pair of hands to be verified, the input vector X is composed of the matching scores between corresponding parts of the hand. Assigning the input vector to the class "1" implies that both hands come from the same subject while assigning it to the class "−1" implies that they come from different subjects.

7.4. Decision-level fusion using majority voting

Majority voting is among the most straightforward decision-level fusion strategies. In this case, the final decision is based on the output results of several matchers. In the context of our application, first we verify/identify each subject using different parts of the hand (i.e., fingers and palm). Then, if three or more parts of the hand yield a positive verification/identification, then verification/identification is considered successful; otherwise, the subject is rejected.

8. Experimental results and comparisons

In order to evaluate the proposed system, we have collected hand images from 101 people of different age, sex and ethnicity. For each subject, we collected 10 images of their right hand during the same session. To test the performance of our system on time passage, additional hand images were collected in a separate session from 20 of these subjects 9 months later. During each session, subjects were asked to stretch their hand and place it inside a square area drawn on the surface of the lighting table; no other restrictions were imposed on the subjects. To capture different samples within each session, subjects were asked to remove their hand from the lighting table, relax it for a few seconds, and then place it back again. We report results both on hand-based verification and recognition.

8.1. Hand-based verification results

For person verification, one must differentiate a genuine hand from imposter hands as the user provides his/her hand image in support of his/her claimed identity. For this purpose, we calculate the Euclidean distance between the hand of the applicant and each of his/her templates in the database and take the minimum distance D:

$$D = min_i\{||Q - T_i||\}, \quad i = 1, \ldots, k \tag{14}$$

where Q corresponds to the query hand, T_i corresponds to the ith template of a given subject in the database, and k corresponds to the number of templates of that subject. If D is below a threshold, verification is successful; otherwise, the subject is rejected.

In the following subsections, we present the results of several different experiments. First, we investigate the performance of a baseline system which uses the whole hand for verification. Then, we investigate the verification power of different parts of the hand by implementing several systems that perform verification using each part of the hand separately. Finally, we evaluate the proposed system which fuses information from different parts of the hand for verification.

8.1.1. Verification using whole hand

To provide a baseline for comparisons, first we experimented with a simpler system that uses the whole hand for verification. In this case, a global representation of the hand is used for verification. Preliminary results based on this approach have been reported in an earlier work [20], however, this section presents results based on more comprehensive experiments and a larger database.

The first step in this baseline system is to separate the arm from the hand using the methodology presented in Section 5. Then, the geometry of the silhouette of the whole hand is represented using Zernike moments. As mentioned in Section 6, capturing the shape details of the whole hand requires computing Zernike moments up to order 70; this yields feature vectors containing 1296 components.

To test the approach, we used different number of samples (i.e., 3, 4, and 5) for each subject as enrollment templates. To account for regularities in the choice of the templates, we repeated the experiments 30 times, each time choosing the enrollment templates randomly. The remaining samples were used to construct matching and non-matching sets and estimate the False Acceptance Rate (FAR) and False Reject Rate (FRR) of the system. Fig. 15(a) shows the average ROC curves obtained using this procedure. As it can be observed, using more templates improves verification accuracy, however, it this would also increase verification time.

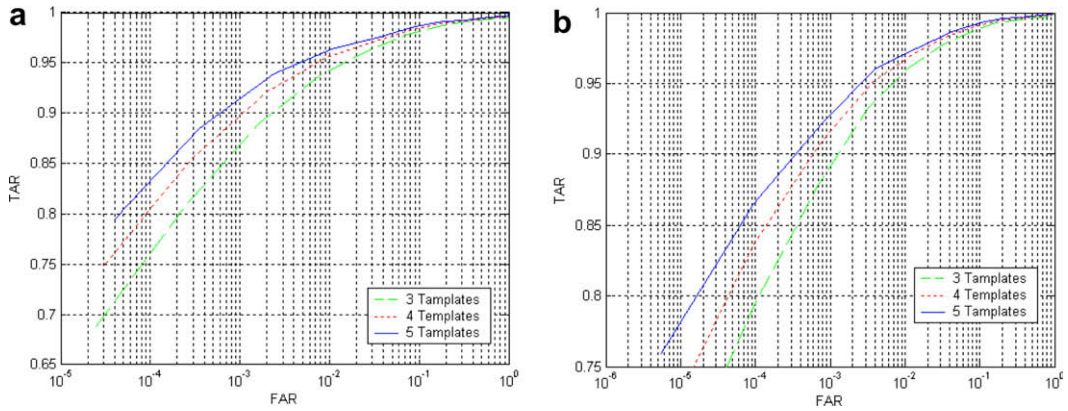

Fig. 15. Average ROC curves using whole hand for verification: (a) raw features, (b) PCA features. Each experiment was repeated 30 times, using 3, 4, and 5 enrollment templates per subject.

Table 8
Verification using whole hand: comparison using raw and PCA features.

Enrollment size	3		4		5	
Features	Raw	PCA	Raw	PCA	Raw	PC
Number of features	1296	182–203	1296	221–242	1296	252–274
$\overline{EER}$ (%)	3.55	2.69	2.95	2.38	2.78	2.21
$\overline{TAR}$ (%) (FAR = 1%)	94.22	95.84	95.62	96.66	96.26	97.06
σ_{TAR} (%) (FAR = 1%)	1.62	1.60	1.61	1.26	1.27	1.16

Since the size of the feature vectors was very high, we have also experimented with PCA to reduce their dimensionality. Using a similar procedure, we repeated the experiments 30 times, choosing 3, 4, and 5 templates randomly each time. In each experiment, the eigenvectors were computed from the covariance matrix of the enrolled templates by preserving 99.9% of the information. Fig. 15(b) shows the average ROC curves obtained using PCA features. Table 8 provides comparative results in terms of the Equal Error Rate (EER), as well as the mean, and standard deviation of the True Acceptance Rate (TAR) when FAR = 0.1%. As it can be observed, PCA improves verification results by increasing TAR while at the same time reducing its standard deviation. Overall, the best verification performance using the whole hand was obtained with PCA features and five enrollment templates per subject.

8.1.2. Verification using different parts of the hand

To investigate the verification power of different parts of the hand, we experimented with several systems, each performing verification using a different part of the hand. In this case, local representations of the hand were used for verification. Each system was tested using five enrollment templates (i.e., using less templates results in lower accuracy) and repeating the experiments 30 times as before by choosing the enrollment templates randomly each time. For each system, we report the average ROC curves obtained. To ensure that the comparison was fair, we used the same training and test data as in the case of the whole hand. To calculate the distance between corresponding parts of the hand (i.e., fingers or back of the palm) in the query and the template hands, we used Eq. (14) as before.

Fig. 16 (blue solid line) shows the average ROC curves obtained for each part of the hand. In addition, we performed experiments using PCA to reduce the dimensionality of the feature vectors. In each case, we preserved 99.9% of the information. Fig. 16 (red dashed line) shows the results obtained in this case. Also, Table 9 shows specific details for each case using raw and PCA features when FAR = 0.1%. As it can be observed, PCA features improve accuracy slightly only in the case of the back of the palm.

To illustrate performance differences between various parts of the hand more clearly, we have plotted all six ROC curves, corresponding to raw features, on the same graph shown in Fig. 17. As it can be observed, the best performance was obtained using the index, middle, and ring fingers. Among them, the index yielded the best results. On the other hand, the thumb yielded the lowest performance among all parts. This can be explained by the fact that the thumb has higher degree of freedom than any other part, making it difficult to fix its position.

8.1.3. Verification by fusing information from different parts of the hand

In this subsection, we report results by fusing information from different parts of the hand for verification. To ensure that our results are comparable to the previous experiments, we used the same evaluation methodology as well as the same training and test sets. Using feature-level fusion, we combined the feature vectors of each part of the hand into a single feature vector yielding 861 features. Using PCA and keeping 99.9% of the information, yields between 72 and 81 features. In the case of score-level fusion using the weighted-sum rule, we experimented with different sets of weight values, using the results from the previous section as a guide. The best results, reported below, were obtained using the following values: $w1 = 0.5/12$ (little finger), $w2 = 2.5/12$ (ring finger), $w3 = 3.0/12$ (middle finger), $w4 = 4.5/12$ (index finger), $w5 = 0.5/12$ (thumb), and $w6 = 1.0/12$ (back of the palm). The

Table 9
Comparison using different parts of the hand for verification: raw versus PCA features.

Finger	Little		Ring		Middle		Index		Thumb		Palm	
Feature	Raw	PCA	Raw	PCA	Raw	PCA	Raw	PCA	Raw	PCA	Raw	PCA
Number of features	121	23–24	121	18	121	16	121	16	121	43–45	256	87–96
$\overline{EER}$ (%)	1.77	1.78	1.62	1.66	1.28	1.44	0.93	0.98	3.62	3.53	2.05	1.90
$\overline{TAR}$ (%) (FAR = 1%)	96.9	96.6	97.3	97.1	98.3	97.8	99.2	99.1	92.2	92.0	96.9	97.5
σ_{TAR} (%) (FAR = 1%)	1.3	1.41	0.79	0.8	0.54	0.72	0.34	0.42	1.7	1.76	1.23	1.16

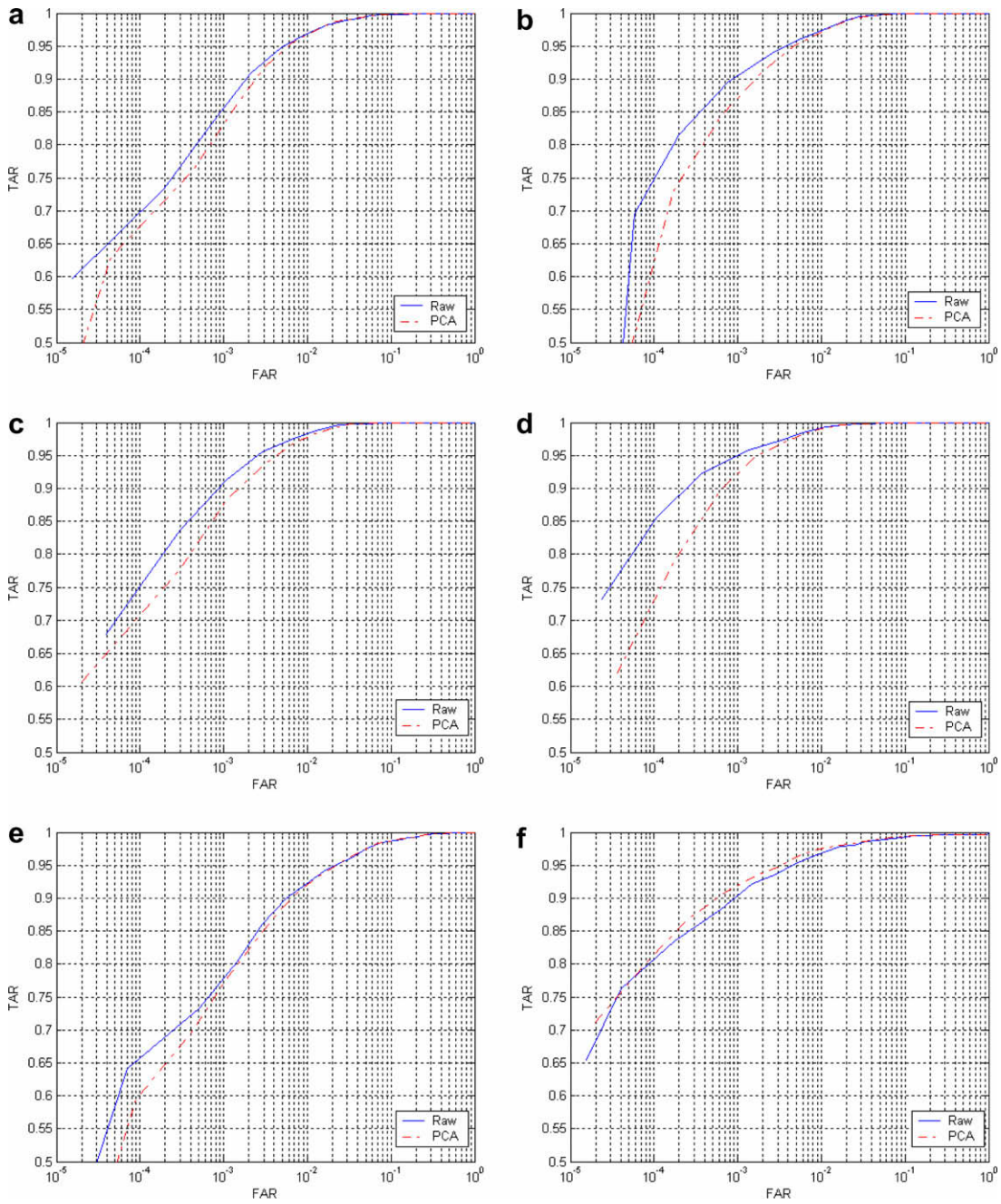

Fig. 16. Average CMC curves using different parts of the hand for identification: (a) little, (b) ring, (c) middle, (d) index, (e) thumb, and (f) back of the palm. Each experiment was performed 30 times using five samples for each subject as enrollment templates.

weights were fixed in all experiments. In the case of score-level fusion using SVMs, we experimented using different parameter values. The best results (i.e., on the average) were obtained using the Gaussian kernel with $\sigma = 0.01$ and $C = 1$ (i.e., cost term). These parameter values were kept fixed in all of our experiments.

Fig. 18 shows the average ROC curves obtained for each fusion strategy using 3, 4, and 5 templates per subject. In general, using more enrollment templates per subject improves verification performance although it would also increase verification time. Among the four fusion strategies considered, decision-based fusion performs best. Between the two different decision-based fusion schemes considered, majority voting performs best. Feature level fusion based on PCA had the lowest performance, however, it should be mentioned that PCA reduces the size of templates more than 10 times.

Fig. 19 compares all fusion strategies on the same graph assuming five enrollment templates. Additional details are shown in Table 10 which compares the fusion strategies, using five enrollment templates, in terms of *EER* and the mean and the standard deviation of TAR when FAR = 0.1%. As it can be observed, all fusion have improved verification performance, for example, TAR is more than 99.4% when FAR is more than 0.1%. Table 11 shows specific details in the case of majority voting.

8.2. Hand-based identification results

For person identification, the user does not provide any identity claim, but the system must find out the user's identity by comparing him/her to a database of enrolled users. Assuming that there are N subjects in the database and that each subject i has k_i tem-

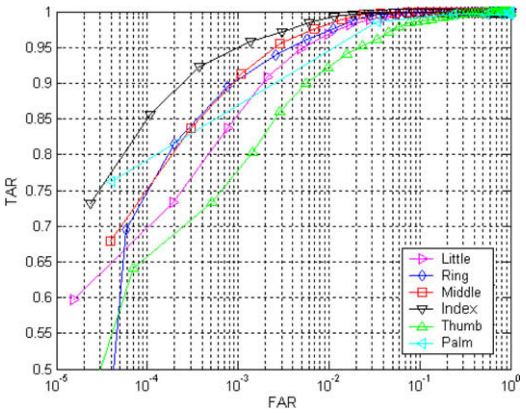

Fig. 17. Verification results using different parts of the hand and raw features.

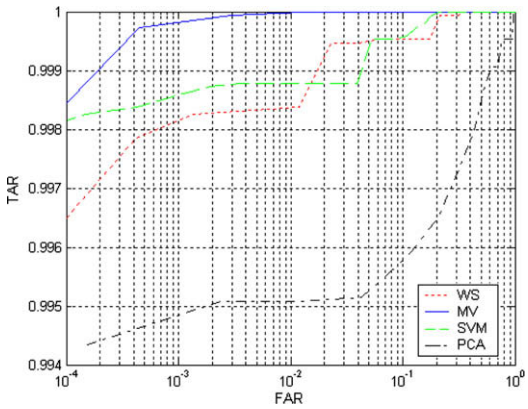

Fig. 19. Comparison of the four different fusion strategies for verification using five enrollment templates per subject.

plates, then the total number of templates stored in the database is $K = k_1 + k_2 + \cdots + k_N$. Given a query hand Q, we compute the Euclidean distance between Q and all the templates $T_j, j = 1, \ldots, K$ in the database. Then, the identity of the user is established by finding the minimum distance, that is, by finding the subject whose template(s) best match the query hand:

$$i^* = arg_i min\{||Q - T_j^i||\}, \quad j = 1, \ldots, k_i, \; i = 1, \ldots, N \tag{15}$$

where T_j^i corresponds to the jth template of the ith subject in the database. In our experiments, we have assumed a "closed-universe" [50], that is, we have assumed that the user is among the subjects stored in the database (i.e., has already enrolled). The "closed-universe" model allows to investigate how good is our recognition algorithm at identifying a query hand by not only asking the question "is the top match correct?" but also the question "is the correct match among the top n matches?". This allows to determine how many templates must be examined in order to get a desired level of performance. In this context, we report identification results by using Cumulative Match Characteristic (CMC) curves which are plots of true match rate versus rank [50]. It should be mentioned that in the case of an "open-universe" [50] (i.e., the user has not enrolled), the minimum distance must also be below a threshold in order to be able to reject imposters.

In the following subsections, we report identification results based on similar experiments as in the case of verification. First, we investigate the performance of a baseline system which uses

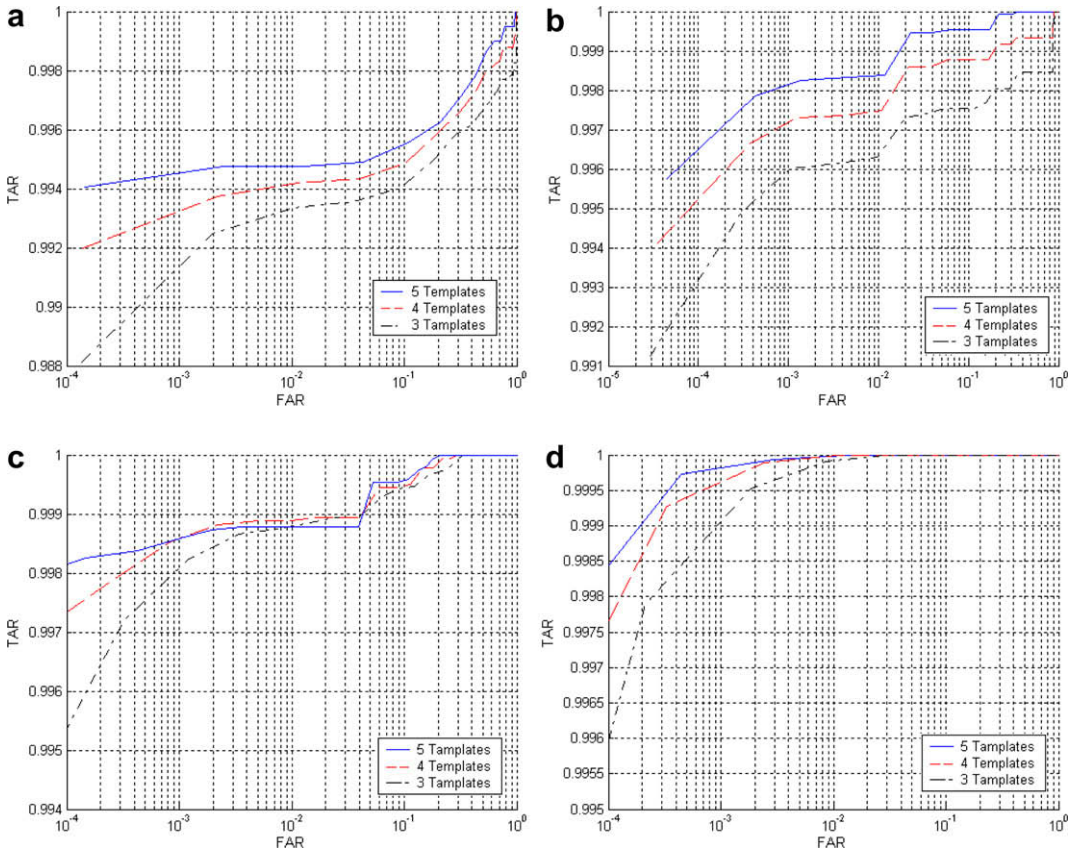

Fig. 18. Average ROC curves for verification: (a) feature-level fusion based on PCA, (b) score-level fusion based on the weighted-sum rule, (c) score-level fusion based on SVMs, and (d) decision-level fusion based on majority voting. In each case, we performed the experiments 30 times using 3, 4, and 5 templates per subject.

Table 10
Detailed comparison of different fusion strategies for verification using five enrollment templates per subject.

Method	PCA	Weighted sum	Majority voting	SVM
$\overline{\text{EER}}$ (%)	0.523	0.052	0.044	0.136
$\overline{\text{TAR}}$ (%) (FAR = 0.1%)	99.47	99.98	99.98	99.86
σ_{TAR} (%) (FAR = 0.1%)	0.231	0.052	0.059	0.12

Table 11
Fusion using majority voting for verification: mean and standard deviation of TAR when FAR = 0.1%.

No. of training vectors	3	4	5
$\overline{\text{TAR}}$ (%)	99.92	99.96	99.98
σ_{TAR} (%)	0.1115	0.0697	0.0594

the whole hand for identification purposes. Then, we investigate the recognition power of different parts of the hand by implementing several systems that perform identification using each part of the hand separately. Finally, we evaluate the proposed system which fuses information from different parts of the hand for identification.

8.2.1. Identification using whole hand

To provide a baseline for comparisons, first we experimented with a simpler system that uses the whole hand for identification. In this case, a global representation of the hand was used for identification. Fig. 20(a) shows the average CMC curves obtained using this procedure while Fig. 20(c) shows the corresponding standard deviations. As it can be observed, using more templates improves recognition accuracy, however, it also increases recognition time. Since the size of the feature vectors was very high, we have also experimented with PCA to reduce their dimensionality. Using a similar procedure like in the case of verification (i.e., see Table 8), we obtained the average CMC curves shown in Fig. 20(b). The corresponding standard deviations are shown in Fig. 20(d). As it can be observed, PCA has almost identical performance to the approach using raw features.

8.2.2. Identification using different parts of the hand

To investigate the identification power of different parts of the hand, we experimented with several systems, each performing recognition using a different part of the hand. Each system was tested using five enrollment templates and repeating the experiments 30 times as before by choosing the enrollment templates randomly each time. For each system, we report the average CMC curve obtained. To ensure that the comparison was fair, we used the same training and test data as in the case of the whole hand. To calculate the distance between corresponding parts of the hand (i.e., fingers or back of the palm) in the query and the template hands, we used Eq. (15) as before. Fig. 16 (blue solid line) shows the average CMC curves obtained for each part of the hand. In addition, we performed experiments using PCA to reduce the dimensionality of the feature vectors. In each case, we preserved 99.9% of the information (i.e., see Table 9, for details). Fig. 16 (red dashed line) shows the results obtained in this case. As it can be observed, PCA has slightly worse recognition accuracy to the approach using raw features, especially for low ranks.

To illustrate performance differences between various parts of the hand more clearly, we have plotted all six CMC curves, corresponding to raw features, on the same graph shown in Fig. 22(a). The corresponding standard deviations are shown in Figs. 20(b) and 22(b). As it can be observed, the best performance was obtained using the index, middle, and ring fingers.

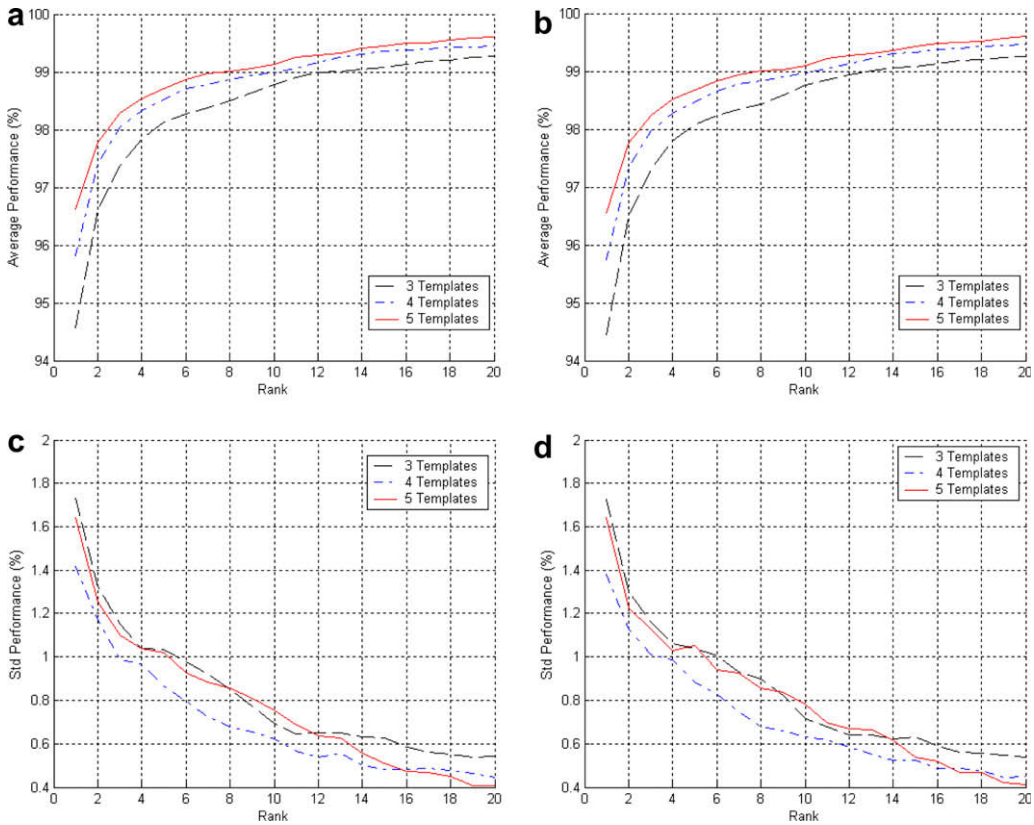

Fig. 20. Average CMC curves using whole hand for identification: (a) raw features, (b) PCA features, (c) standard deviation of raw features, and (d) standard deviation of PCA features. Each experiment was repeated 30 times, using 3, 4, and 5 enrollment templates per subject.

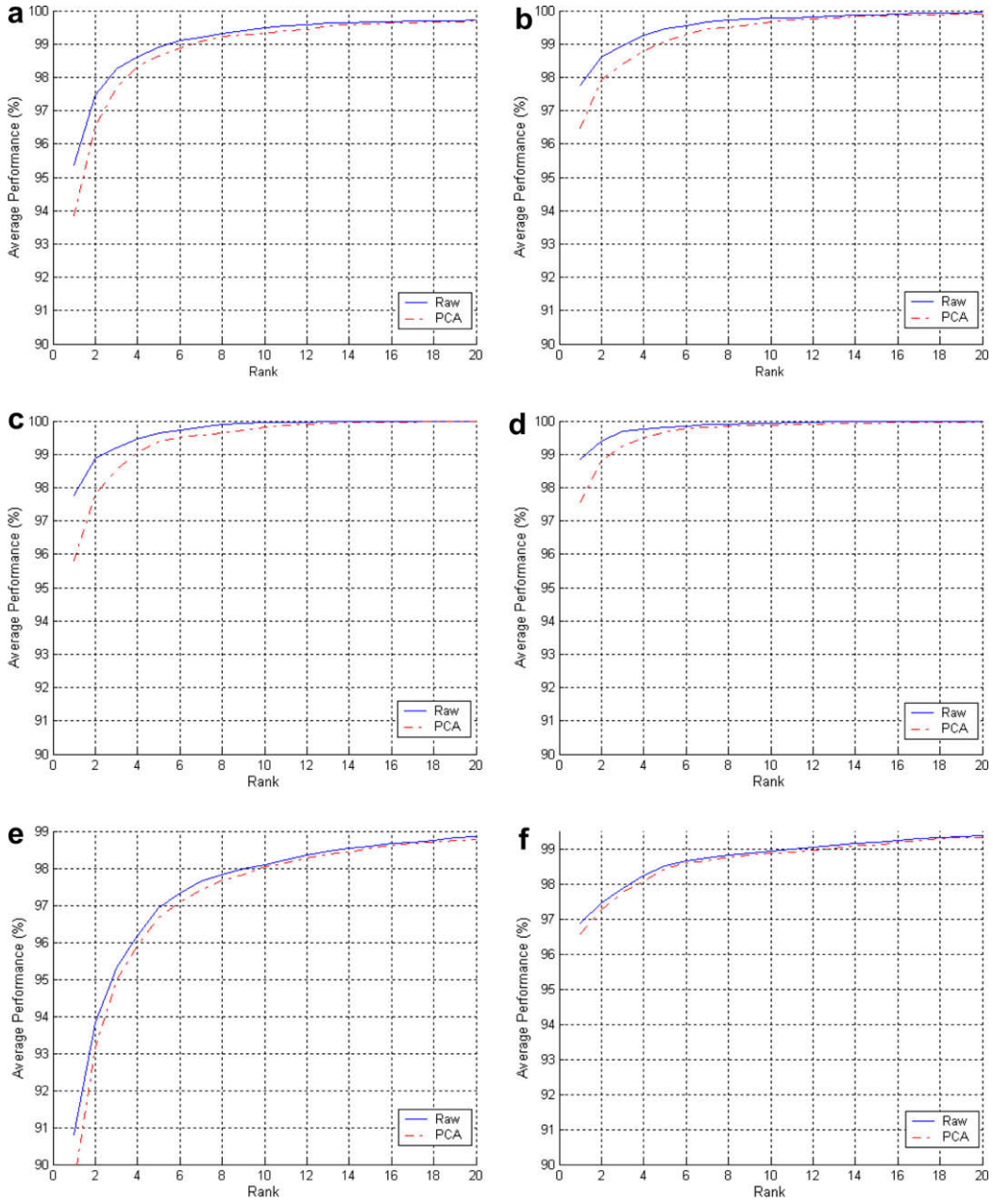

Fig. 21. Average ROC curves using different parts of the hand for verification: (a) little, (b) ring, (c) middle, (d) index, (e) thumb, and (f) back of the palm. Each experiment was performed 30 times using five samples for each subject as enrollment templates.

Among them, the index yielded the best recognition results (i.e., both higher accuracy and lower standard deviation). On the other hand, the thumb yielded the lowest recognition performance among all parts (i.e., both worst accuracy and higher standard deviation). These results are consistent with those obtained for verification.

8.2.3. Identification by fusing information from different parts of the hand

In this subsection, we report results by fusing information from different parts of the hand for identification. In particular, we tested the same fusion strategies except score-level fusion using SVMs since the use of SVMs for identification would require extending SVMs to multiple-class classification. Such an extension would require a large number of training samples (i.e., enrollment templates) per subject to guarantee good performance. As previously, we used the same evaluation methodology as well as the same training and test sets for consistency. Fig. 23 shows the average CMC curves obtained for each fusion strategy using 3, 4, and 5 templates per subject. In general, using more enrollment templates per subject improves identification performance although it would also increase identification time. Among the three fusion strategies considered, score-level fusion had slightly better performance (i.e., higher accuracy and lower standard deviation) for low ranks. Fig. 24(a) compares all three fusion strategies on the same graph assuming five enrollment templates. Fig. 24(b) shows the corresponding standard deviations.

8.3. Comparison between global-based and component-based hand representations

Representing the hand it terms of its components and performing verification or identification using different parts of the hand separately or fusing information from different parts

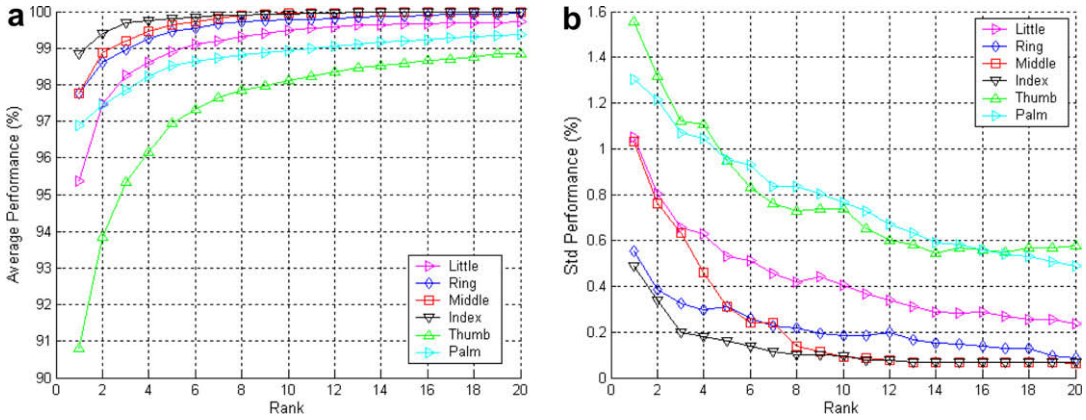

Fig. 22. (a) Identification results using different parts of the hand and raw features; (b) standard deviation.

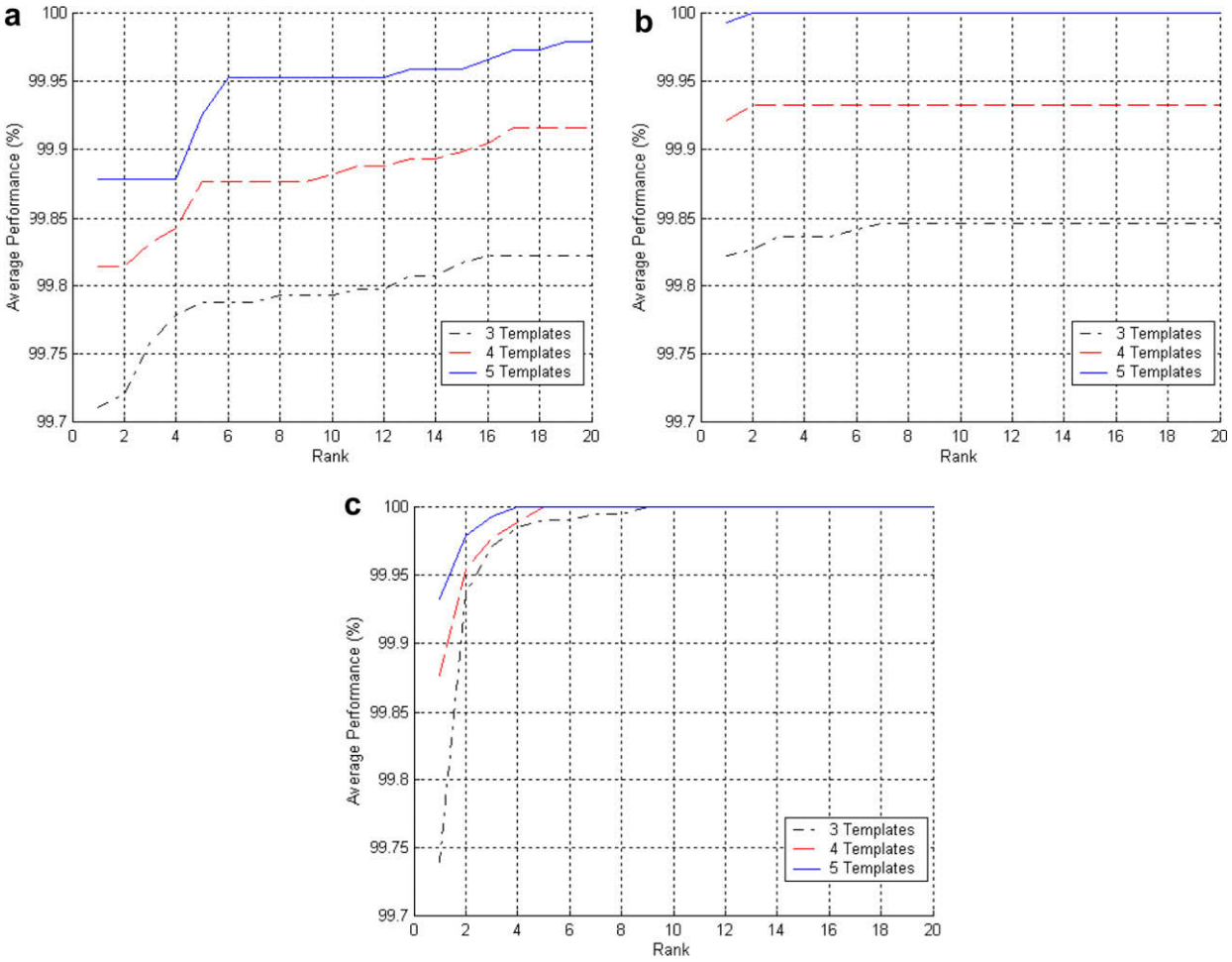

Fig. 23. Average CMC curves for identification using (a) feature-level fusion based on PCA, (b) score-level fusion based on the weighted-sum rule, and (c) decision-level fusion based on majority voting. In each case, we performed the experiments 30 times using 3, 4, and 5 templates per subject.

of the hand offers important advantages both in terms of time and accuracy. In terms of time, a component-based representation of the hand allows for representing shape information using a smaller set of features and lower Zernike moment orders. In terms of accuracy, a component-based representation, using fusion or individual parts of the hand, improves verification and identification performance compared to using a global-based representation.

Fig. 25(a) shows the ROC curves corresponding to the three most representative verification approaches tested here: (i) whole hand, (ii) index finger only, and (iii) fusion based on majority voting. Obviously, fusion improves performance significantly (e.g., when FAR = 1%, TAR increases from 96.06% in the case of whole hand to 100% in the case of fusion). Similarly, Fig. 25(b) shows the CMC curves corresponding to the three most representative identification approaches tested here: (i) whole hand, (ii) index fin-

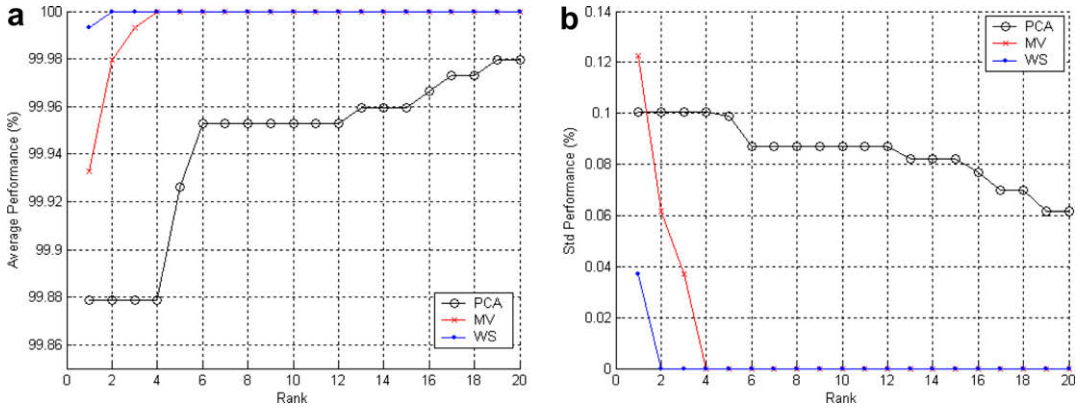

Fig. 24. (a) Comparison of different fusion strategies for identification using five enrollment templates per subject; (b) Standard deviation.

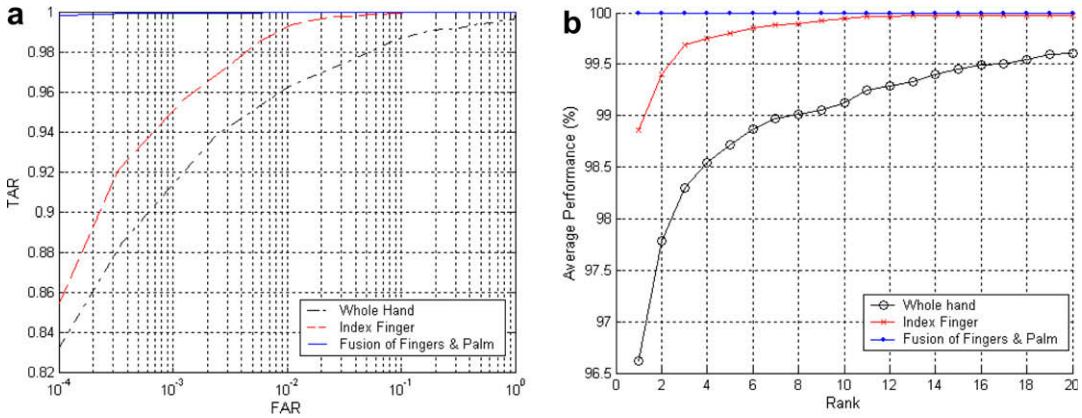

Fig. 25. Comparison of the three most representative approaches for (a) verification and (b) identification, using five enrollment templates per subject: (i) whole hand, (ii) index finger, and (iii) majority voting (for verification) and weighted sum (for identification).

ger only, and (iii) fusion based on weighted sum. Obviously, fusion improves identification performance significantly (e.g., when rank = 1, recognition accuracy increases from 96.75% in the case of whole hand to almost 100% in the case of fusion).

It should be mentioned that the main reason that the baseline approach (i.e., whole hand) did not perform very well is because it cannot tolerate well finger motion. As shown in Fig. 2(b) and (c), finger motion is unavoidable in different sample images of the same subject. Although Zernike moments can tolerate some degree of finger motion(e.g., 6° rotation about the axis being perpendicular to the joint of the finger with the palm), they are sensitive to larger finger motions. Moreover, they cannot tolerate well situations where the hand is bent at the wrist. Fig. 26, illustrates that finger motion affects the Zernike moments of all orders. Segmenting the hand in different parts alleviates these problems.

8.4. Comparisons with other approaches

In this section, we report both qualitative and quantitative results between our method and methods reported in the literature. Table 12 shows a qualitative comparison of the performance of our system and methods reported in the literature. Since there is no

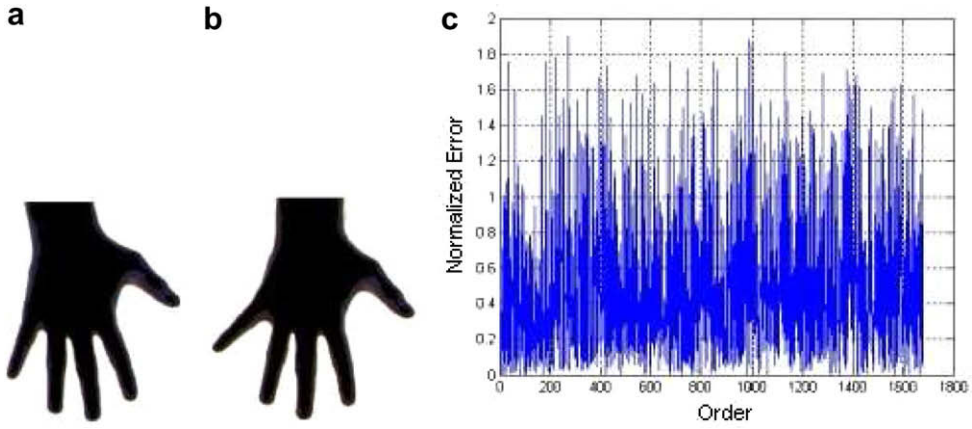

Fig. 26. (a and b) Images of the same hand containing finger motion, (c) normalized Zernike moment differences.

Table 12
Qualitative comparison with existing methods.

System(s)	# of people	# of sample per person	Pegs	# of template(s)	Feature(s)	Similarity	Verification performance	Identification performance
Jain [2]	50[a]	10	Yes	2	Geometric features[k] (16 features)	Mahalanobis	FAR = 0.01 FRR[b] ≈ 0.17	–
Wong [27]	22[c]	12–15	No	9	Thirteen geometric features and three fingertip regions[k]	GMM	FAR = 0.022 FRR = 0.1111	–
Jain [30]	53[d]	2–15	Yes	1[e]	Contour of five fingers	Shape Distance	FAR = 0.01 FRR[b] ≈ 0.06	–
Reillo [36,37]	20	10	Yes	5	Geometric features[k] deviation and angles between the inter-finger points (25 features)	Euclidean Hamming GMM	EER[h] = 0.049 Error rate[f] ⩽ 0.1	97.0×10^{-2}
Ma [31]	20	6	No	1	4 B-Spline curves, length of thumb and width of palm	Shape distance	Error rate = 0.05	–
Kumar [4]	100	10	No	5	Geometric features[k] hand area (16 features) and	Correlation coefficient	FAR = 0.01 FRR[i] ≈ 0.32	–
Bulatov [29]	70[g]	10	No	5	Thirty geometric features[k]	Nearest box	FAR = 0.01 FRR = 0.03	96.5×10^{-2}
Ribaric [28]	130	5	No	1	Twenty geometric features[k]	Euclidean	FAR = 0.153 FRR[j] = 0.13	–
Xiong [26]	108	5	No	1	Width of four fingers at 45 different location for each figure	Shape distance	EER = 2.41×10^{-2}	–
Yoruk [51]	100[l]	3	No	2	Independent Component Analysis (ICA) (200 features)	Cosine of the angle between vectors	EER = 1.15×10^{-2}	98.81×10^{-2}
Oden [55]	35	Not clear	No	Total 20	Combination of implicit polynomials and geometric features (total 16 features)	Mahalanobis	FAR = 0.01 FRR = 0.01	95.0×10^{-2}
Our method	100	10	No	5	Zernike moments (861 features for fingers and palm)	Euclidean	FAR = 0.01 $\overline{FRR} = 0.0$ $\overline{EER} = 4.38 \times 10^{-4}$ Error rate[f] = 7.42×10^{-4}	99.98×10^{-2}

[a] Out of 500 images, only 360 images were used and 140 images were discarded.
[b] Estimated from ROC curve in [2].
[c] A total of 288 images were used.
[d] A total of 353 images were used.
[e] Not all possible non-matching pairs were used.
[f] The minimum error rate, which is the sum of FAR and FRR.
[g] A total of 714 images were used.
[h] This is the best EER using five training vectors and GMM for verification [37].
[i] Hand geometry was used to improve the performance of palmprint-based verification. We have estimated FRR using only hand geometry information from the ROC curve in [4] when FAR = 0.01.
[j] A multi-modal biometric system was designed in [28] using fingerprint, palmprint, and hand geometry. The FAR and FRR reported here relates to hand geometry only, see [28].
[k] Geometric features such as length and width of the fingers, width of palm, thickness of hand and middle finger, etc.
[l] The database includes 458 people with three samples per person. EER has been reported for different populations (i.e., 20,35,50,100, and 458).

standard acquisition method and no benchmark databases, quantitative comparisons of different systems should be considered only indicative and not conclusive. To make the comparison more fair, for each study considered, we report several other factors including the number of subjects, the number of images per person, the number of enrollment templates, the use/no-use of pegs, the type of features, and the distance measure. The results reported for our system in Table 12 correspond to using five enrollment templates. Our database size is comparable to most of the systems reported in the table while our error rates are better than or equal even to the ones reported on much smaller databases.

As it can be observed from Table 12, the majority of existing systems employ hand geometric features for verification or identification. It has been illustrated in the literature that these features work well and can be computed efficiency. To better assess the performance of our method, we have performed quantitative comparisons, using the same database, to investigate whether Zernike descriptors offer any potential advantages over geometric features in terms of robustness and accuracy. The geometric features used in our experiments is a subset of the features introduced by Sanchez-Reillo et al. [36,37]. Fig. 27(a) shows a sample image taken by their image acquisition system. They used 31 features (see Fig. 27(b)): width of four fingers and palm in different locations (18 features), height of middle and little fingers and palm (3 features), distances between the three inter-finger points (3 features) and angles between the inter-finger points and horizontal line (3 features), distances between a middle point of the finger and the middle point of the straight line between the inter-finger point

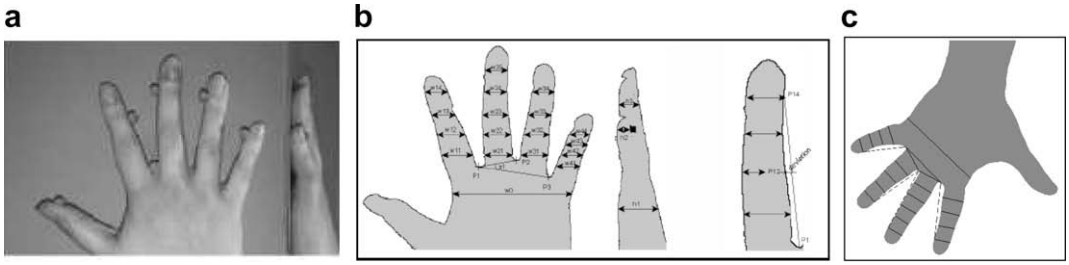

Fig. 27. (a) A sample hand image taken using the image acquisition system in [36,37], (b) location of landmarks and features measured using the approach of [36,37], (c) the main features measured based the on approach of [36,37] on a sample hand image from our database.

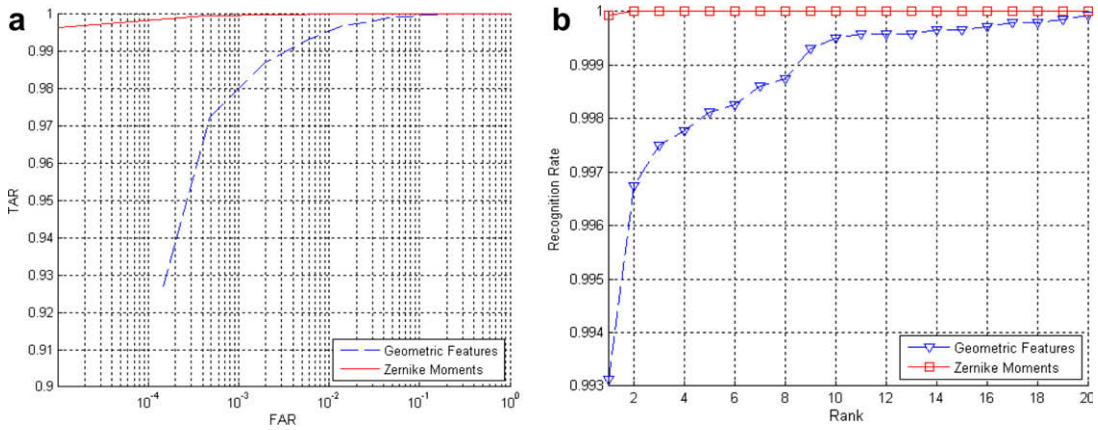

Fig. 28. Comparison between geometric features and Zernike descriptors: (a) authentication results, (b) identification results.

and the last height where the finger width is measured (4 features). However, their image acquisition system uses a mirror to capture a side view of the hand in addition to a top view of the hand as shown in Fig. 27(a). Since our system captures a top view of the hand only, we cannot extract the height of the little and middle fingers as well as the palm (3 features). Therefore, we have used only 28 features in our experimental comparisons. Fig. 27(c) shows the main distances measured on the binarized hand images from our database.

Systems employing pegs to fix the position of the hand, such as [36,37], use predestined axes to facilitate feature extraction. In the case of peg-free systems, several landmarks on the hand, such as fingertips and valleys, must be extracted in order to define the same or similar axes [26,31,27,51]. Here, we compute the curvature of the hand boundary to extract the fingertip and valley locations by detecting curvature minima and maxima. The same methodology has been employed in several other peg-free systems including [31,26,27]. We provide more details about the landmarks extraction algorithm in the next section.

Fig. 28(a) and (b) shows the performance of hand-based authentication (i.e., using majority voting) and identification (i.e., using weighted sum) between our method based on Zernike descriptors and the method of [36,37] based on geometric features. In these experiments, five templates per person were employed for enrollment while the rest of them were used for testing. We report the average performance as before by repeating each experiment 30 times. As it can be observed, system performance using Zernike descriptors is superior to using geometric features both in the case of verification and identification. Therefore, Zernike descriptors seem to be more powerful compared to geometric features. In terms of time, as we discussed in Section 6, it takes less than 0.01 s on the average to compute moments up to order 30 on a 3.19 GHz 64-bits machine with 2GB of RAM, assuming double precision. Table 13 compares the time requirements of the proposed method and the method based on geometric features. Although the preprocessing step of the proposed method is more time consuming, it is most robust than detecting landmark points on the hand as illustrated in the next section.

Table 13
Processing time for traditional and proposed methods. All numbers are in milliseconds.

Method	Preprocessing		Feature extraction
Traditional	Constructing axes/30[a]		Geometric/20[a]
Proposed	Hand–arm segmentation/580[a]	Palm–finger segmentation/140[a]	Zernike/10[b]

[a] ms in MATLAB 7.4 on a 3.19 GHz, 64-bit machine with 2 GB RAM.
[b] ms in Visual Studio 2005 on a 3.19 GHz, 64-bit machine with 2 GB RAM.

8.5. Comparison between morphological-based finger segmentation versus landmarks-based finger segmentation

It could be claimed that segmenting the fingers and the palm could be accomplished more efficiently using landmark points on the hand than the algorithm described in Section 5 based on morphological operators. The purpose of the experiment reported in this section is to investigate this claim by comparing finger segmentation using morphological operators, i.e., versus landmark points. There are two main objectives behind our comparisons: (i) to investigate the computational efficiency of each method and (ii) to investigate the effect of segmentation errors on verification and identification performance.

Fig. 29 illustrates how landmark points on the hand could be used for finger segmentation. In particular, Fig. 29(a) shows the location of the fingertips (blue dots)[1] and valleys (red dots) while

[1] For interpretation of the references to color in the text, the reader is referred to the web version of this paper.

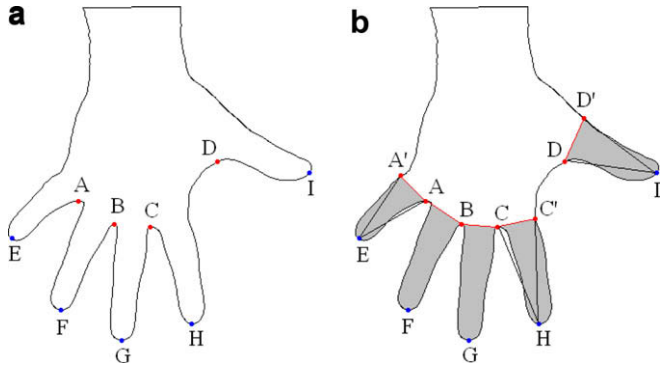

Fig. 29. (a) A sample hand contour and its landmarks. Points $E-I$ and $A-D$ show the location of the fingertips and valleys; (b) finger segmentation using the landmarks in (a) and some auxiliary points.

Fig. 29(b) illustrates how to segment the fingers from the hand using these landmarks. It should be noted that, to segment the thumb, little finger, and index finger, some auxiliary points, namely A', C', and D', need to be considered where ID' is equal to ID, HC' is equal to HC, and EA' is equal to EA [26,31,27,51].

Accurate extraction of landmark points on the hand is a crucial step for peg-free systems [26,31,27,51]. A common approach to detect and extract the fingertips and valleys involves using curvature information on the boundary of the hand [31,26,27]. The main idea is detecting curvature minima (i.e., fingertips) and maxima (i.e., valleys); we have adopted this methodology here. The curvature k of a planar curve, at a point on the curve, is defined as the instantaneous rate of change of the slope of the tangent at that point with respect to arc length, and it can be expressed as follows:

$$k(t) = \frac{(\dot{x}(t)\ddot{y}(t) - \dot{y}(t)\ddot{x}(t))^2}{(\dot{x}(t)^2 + \dot{y}(t)^2)^{\frac{3}{2}}} \tag{16}$$

where $\dot{x}(t)$, $\ddot{x}(t)$, $\dot{y}(t)$, and $\ddot{y}(t)$ are the first and second derivatives of $x(t)$ and $y(t)$, respectively, and $(x(t), y(t))$ is the parametric representation of the curve. To account for noise, $x(t)$ and $y(t)$ are typically smoothed using a Gaussian function $g(t,\sigma)$ [52,53]. The smoothed curve curvature $k(t,\sigma)$ can be expressed as follows:

$$k(t,\sigma) = \frac{(\dot{X}(t,\sigma)\ddot{Y}(t,\sigma) - \dot{Y}(t,\sigma)\ddot{X}(t,\sigma))^2}{(\dot{X}(t,\sigma)^2 + \dot{Y}(t,\sigma)^2)^{\frac{3}{2}}} \tag{17}$$

where $\dot{X}(t,\sigma)$ and $\ddot{X}(t,\sigma)$ are defined at the convolution of $x(t)$ withe the first and second derivatives of $g(t,\sigma)$ correspondingly. $\dot{Y}(t,\sigma)$ and $\ddot{Y}(t,\sigma)$ can be defined similarly. Before computing the curvature, the hand boundary is re-sampled at 1024, equal-distant, points [52,53]. Fig. 30(b) shows the curvature of the hand contour shown in Fig. 30(a). Choosing the value of σ is critical to ensure both good detection and localization. In general, smaller σ values lead to better localization, however, noise could give rise to false positives. On the other hand, larger σ values reduce false positives but good localization is difficult. To address this issue, multi-resolution schemes have been proposed (i.e., curvature scale-space [52]), however, time requirements are higher. Since our system produces images of high quality, we have found that a σ value equal to 20 yields good detection and localization results.

It can be noted by observing Fig. 30(b) that the curvature of the finger valley between the index finger and the thumb is much smaller compared to the other valleys. As a result, detecting and localizing point D is more difficult than the other three valleys. Moreover, it is easy to confuse its location with other points on the hand boundary, having similar curvature values such as point X. To deal with this issue, we use a reference point (i.e., R) as shown in Fig. 30(a). A disadvantage of using the landmarks shown in Fig. 30(a) for partitioning the hand is that the palm cannot be separated from the hand silhouette. To address this issues, additional landmark points would be needed, for example, on the opposite sides of the wrist. However, detecting and localizing these points reliably would be difficult since the curvature in the wrist region is quite as Fig. 30(b) illustrates.

The goal of our first experiment in this section is to investigate how errors in partitioning the hand could affect verification and identification performance. Therefore, we have performed experiments to compare landmarks-based segmentation versus morphological-based segmentation. Since the palm cannot be segmented using landmark points, our comparisons used information from the fingers only. First, we computed Zernike descriptors for each finger up to order 20, segmented using landmarks or morphological operators. Then, we performed verification experiments using each finger separately in the spirit of the experiments reported in Section 8.1.2.

Fig. 31 shows the verification results obtained for each finger. Each experiment was performed 30 times, each time using five samples per person for enrollment. Our results indicate that morphological-based segmentation has better performance than landmarks-based segmentation in the case of the little, ring, middle, and index fingers. In the case of the thumb, morphological-based segmentation performs better for FAR rates smaller than 0.04. Obviously, segmenting the thumb from the hand is more challenging than the rest of the fingers due to its higher flexibility. These results suggest that combining morphological-based with

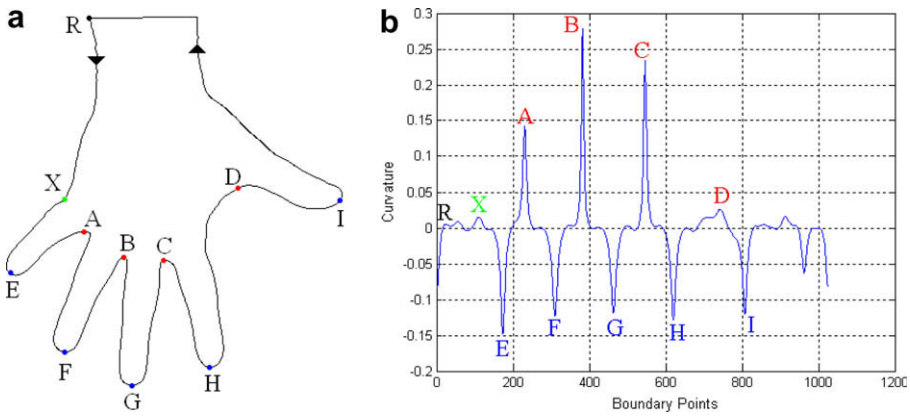

Fig. 30. (a) A sample hand contour, (b) the curvature of the hand contour. R is a reference point which is used for identifying the landmarks. The hand boundary has been re-sampled at 1024, equal-distant, points.

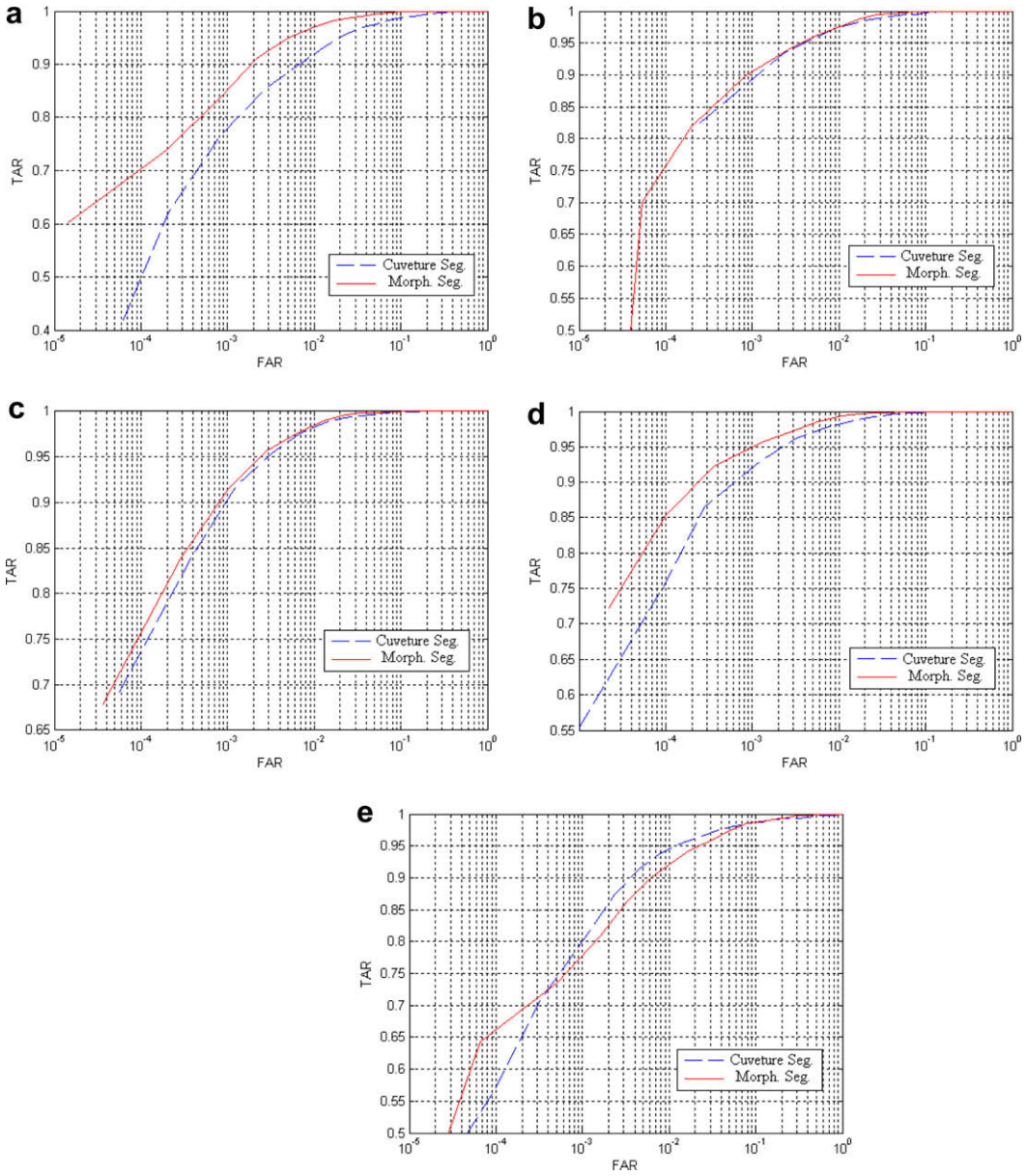

Fig. 31. Average ROC curves using morphological-based and landmarks-based segmentation for verification using each finger separately: (a) little, (b) ring, (c) middle, (d) index, and (e) thumb. Each experiment was repeated 30 times using five enrollment templates per subject and the average is reported.

landmarks-based segmentation for separating the thumb from the hand might yield better results than either approach alone although it would be more time consuming. Next, we performed both verification and identification experiments by fusing information from the fingers using both majority voting and weighted sum. Fig. 32 shows the results obtained in this case for each segmentation method. As it can be observed, landmarks-based segmentation has similar performance to morphological-based segmentation in the case of verification using majority voting. However, morphological-based segmentation performs better than landmarks-based segmentation in all other cases.

It is worth noting by comparing Fig. 19 to Fig. 32(a) and (b) that using information from the palm in addition to information from the fingers for verification (i.e., Fig. 19) does not lead to significantly better results than using information from the fingers only (i.e., Fig. 32(a) and (b)). Similar conclusions can be made in the case of identification by comparing Fig. 24 with Fig. 32(c) and (d). On the other hand, segmenting the palm from the hand is much more expensive than segmenting the fingers from the hand. This is because segmenting the fingers from the hand requires applying a single morphological closing operation using a fixed radius disk structure element. However, segmenting the palm from the hand requires applying the iterative process described in Section 5. If the palm is disregarded, then there are no significant differences in terms of time between landmarks-based segmentation and morphological-based segmentation. In particular, implementing both methods in MATLAB 7.4.0 on a 3.19 GHz 64-bits machine with 2 GB of RAM, landmarks-based segmentation takes 0.09 s on average while morphological-based segmentation takes 0.14 s on average.

8.6. System performance over time

In this section, we report several results to illustrate the performance of the proposed method over large lapses of time. In this context, we recorded 10 new samples from 20 of the 101 subjects

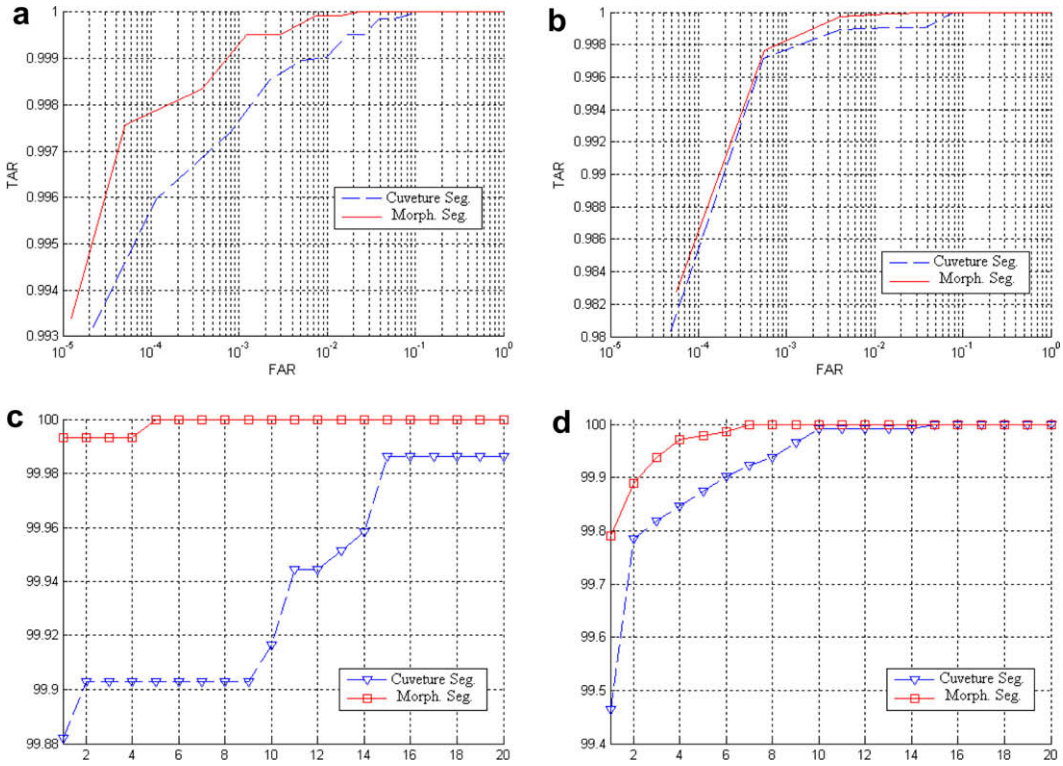

Fig. 32. Comparison of morphological-based and landmarks-based segmentation using different fusion strategies: (a) verification using weighted sum, (b) verification using majority voting, (c) identification using weighted sum, and (d) identification using majority voting. Each experiment was repeated 30 times using five enrollment templates per subject and the average is reported.

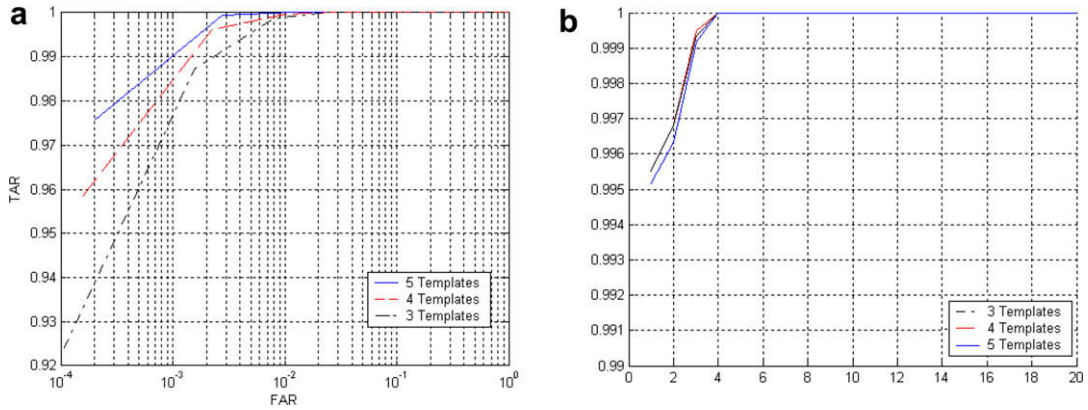

Fig. 33. Effect of time lapse: (a) average ROC curves for verification based on decision-level fusion (majority voting) using 3, 4, and 5 templates per subject; (b) average CMC curves for identification based on score-level fusion (weighted sum) using 3, 4, and 5 templates per subject. The experiment was performed 30 times using the same enrollment templates as in the previous experiments. For testing, we used 200 images from 20 of the 101 subjects, taken 9 months later.

after a period of 9 months (i.e., 200 images). These samples were used to test the performance of our system when there is a substantial passage time between the acquisition of the template and test images. In a similar manner as before, we repeated each experiment 30 times using 3, 4, and 5 samples from our initial data collection as enrolment templates. To keep results consistent, we used exactly the same enrolment templates in each experiment as in our previous experiments. Fig. 33(a) and (b) show the average ROC and CMC curves respectively obtained in this case. As it can be observed by comparing Fig. 33(a) with Fig. 18(d), and Fig. 33(b) with Fig. 23(b) there is a small deterioration in system performance over time, however, this is quite reasonable and acceptable.

To further test the performance of our method on time lapse, we performed more experiments using a publicly available hand database provided by University of Notre Dame [54]. This database was created by collecting data on three different sessions. In the first session, two images from 132 subjects were collected. In the second session, which was conducted a week later, three images were collected from the same 132 subjects. The third session, which was conducted 15 weeks later from the second session, three images were collected from 177 subjects of which 86 had participated in the first two data collections [54]. The database contains both range and color images, each being 640×480 in size.

In our experiments, we used the color images of the same 86 subjects who participated in all three sessions. To extract the hand silhouette, we used the same algorithm described in [54]. However, since lighting was not uniform in all images, some areas of the palm have low contrast. As a result, the hand silhouette was defected

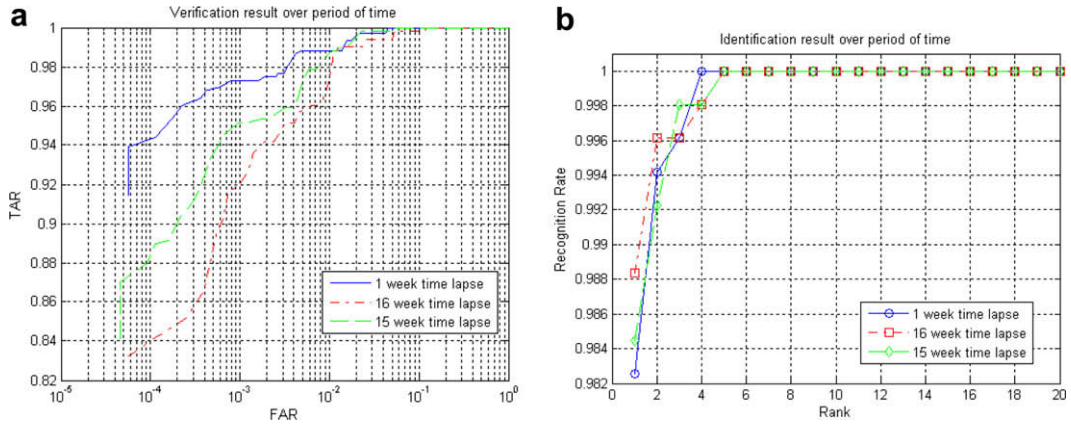

Fig. 34. Effect of time lapse: (a) average ROC curves for verification using score-level fusion (weighted sum) of fingers; (b) average CMC curves for identification using score-level fusion (weighted sum) of fingers. 86 subjects participated in three separate sessions in which 2, 3 and 3 images were taken from each subject (total 688 images).

many times and we were not able to segment the palm satisfactorily. Therefore, we decided to use only information from the fingers in our experiments. Similarly to other experiments, we computed Zernike descriptors for each finger up to order 20. Verification was performed using the majority voting rule while identification was performed using the weighted-sum rule. In both cases, we used the samples from one of the three session as enrollment templates and the samples from the other two sessions for testing.

For consistency reasons, we adopted the same setup as in [54] to form the gallery (i.e., enrollment) and probe (i.e., test) image sets. That is, gallery images were chosen to be images collected prior to those chosen as probe images [54]. Following this rule, only images collected during the second week could serve both as probe and gallery images. For each time lapse, we performed two experiments by switching the enrollment samples with the test samples. Since the number of samples was not equal in all sessions (e.g., two samples per person in the first session and three samples per person in the second and third sessions), we report average performance for each time lapse.

Fig. 34 shows the results obtained for each time lapse. Fig. 34(a) shows that the performance of our method is close to 98% when FAR is equal to 0.01; these results do not change much for different time lapses. Fig. 34(b) illustrates that recognition is robust over time. There is a slight inconsistency for ranks 1 and 2 (i.e., the recognition rate based on 16 weeks time lapse is higher than 1 and 15 weeks time lapse), however, this is probably due to the unequal size of the data sets.

Table 14 reports the Equal Error Rate (EER) and TAR for each method when FAR is equal to 5% assuming a 16 week time lapse as reported in [54]. As it can be observed, our approach shows better performance. Table 15 reports the identification results for each method assuming 1 week and 16 weeks time lapse as reported in [54]. Again, our methods shows better performance. Moreover, our method seems to be more robustness over time since the identification rate does not change significantly.

Table 14
Time lapse verification performance comparison between our method and Woodard's method [54] using the Notre Dame University database.

Time lapse	16 week	
Method	Woodard [54] (%)	Proposed (%)
EER	5.5	1.72
TAR (FAR = 5%)	94	99.3

Table 15
Time lapse identification performance comparison between our method and Woodard's method [54] using the Notre Dame University database.

Time lapse	1 week		16 week	
Method	Woodard [54]	Proposed	Woodard [54]	Proposed
Recognition Rate	91%	97.7%	94%	98.4%

9. Conclusion

We have presented a new approach to hand-based verification and identification using a component-based representation of the hand and fusion. The proposed method has several advantages including that it is peg-free, it does require the extraction of any landmark points on the hand, it is independent of the position and orientation of the hand, and tolerates finger motion very well. The only restriction imposed by our system is that users must stretch their hand during image acquisition to avoid touching figures.

Our system represents the geometry of the fingers and the back of the palm using translation, rotation, and scale invariant Zernike moments. To improve the computational efficiency and accuracy of high-order Zernike moments, we have adopted an improved algorithm that avoids redundant computations and uses arbitrary precision arithmetic and look-up tables. Using a database of 1010 images from 101 subjects and five enrollment templates per subject, we obtained a TAR = 99.98% when FAR = 0.1% and EER = 0.044 for verification, and 99.98% accuracy for identification. Comparisons with alternative approaches using the whole hand or individual parts of the hand, illustrate the superiority of the proposed approach both in terms of speed and accuracy. Also, qualitative comparisons with systems reported in the literature indicate that our system performs comparable or better.

Implementing the proposed system on a 3.19 GHz 64-bits machine with 2 GB of RAM, the computation of Zernike moments up to order 20/30 for the fingers and the palm using double precision architecture in Visual C++ Studio 2005 is less than 0.01 s. The total preprocessing time for hand–arm segmentation and finger–palm segmentation using MATLAB 7.4.0 is less than 0.73 s on average. Time can be further improved without sacrificing accuracy significantly by disregarding the palm as discussed in Section 8.5. Therefore the proposed system can be employed for on-line applications.

For future work, first we plan to perform larger scale verification and identification experiments by increasing the size of our database. This would allow us to obtain more accurate error estimates. Moreover, we plan to perform additional tests to evaluate the robustness of our method when there is substantial passage time between the template and test images. Second, we plan to investigate the idea of combining multiple templates into a single,

"super-template", in order to build more accurate models for each subject and reduce storage requirements. Third, we plan to investigate feature selection schemes in order to reduce the dimensionality of the feature vectors without sacrificing discrimination power. This would also reduce time requirements since we would need to compute a small number of Zernike moments only. Finally, we plan to perform additional comparisons with other methods in the literature using the same database.

Acknowledgements

This research was supported in part by NSF under EPSCoR RING-TRUE III Grant No. 0447416 and in part by NASA under EPSCoR Grant No. NCC5-583.

References

[1] Biometrics market and industry report 2004–2008. Available from: <http://www.biometricgroup.com/reports/public/market_report.html>.
[2] A.K. Jain, A. Ross, S. Pankanti, A prototype hand geometry-based verification system, in: Proceedings of the 2nd International Conference on Audio- and Video-based Personal Authentication (AVBPA), Washington, USA, March 1999, pp. 166–171.
[3] A. Ross, A.K. Jain, Information fusion in biometrics, Pattern Recognition Letters 24 (13) (2003) 2115–2125.
[4] A. Kumar, D.C.M. Wong, H.C. Shen, A.K. Jain, Personal verification using palmprint and hand geometry biometric, in: Time-Varying Image Processing and Moving Object Recognition, Guildford, UK, June 2003, pp. 668–678.
[5] F. Zernike, Beugungstheorie des schneidenverfahrens und seiner verbesserten form, derphasenkontrastmethode, Physica 1 (1934) 689–704.
[6] C. Teh, R. Chin, On image analysis by the methods of moments, IEEE Transactions on Image Analysis and Machine Intelligence 10 (4) (1988) 496–513.
[7] J. Noh, K. Rhee, Palmprint identification algorithm using hu invariant moments and otsu binarization, in: ACIS International Conference on Computer and Information Science, 2005.
[8] Y.H. Pang, A. Teoh, D. Ngo, H.F. San, Palmprint verification with moments, Journal of WSCG 12 (2) (2004) 325–332.
[9] M.R. Teague, Image analysis via the general theory of moments, Journal of the Optical Society of America 70 (1980) 920–930.
[10] G. Amayeh, A. Erol, G. Bebis, M. Nicolescu, Accurate and efficient computation of high order zernike moments, in: First International Symposium on Visual Computing, (ISVC 2005), Lake Tahoe, NV, December 5–7, 2005, pp. 462–469.
[11] A. Ross, R. Govindarajan, Feature level fusion using hand and face biometrics, in: SPIE Conference on Biometrics Technology for Human Identification, vol. 5779, No. 2, March 2005, pp. 196–204.
[12] A. Kumar, D. Zhang, Personal recognition using hand shape and texture, IEEE Transactions on Image Processing 15 (2006) 2454–2461.
[13] M. Cheung, M. Mak, S. Kung, A two level fusion approach to multimodal biometrics verification, in: IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'05), vol. 5, 18–23 March 2005, pp. 485–488.
[14] C. Jiang, G. Su, Information fusion in face and fingerprint identity verification system, in: The Third International Conference on Machine Learning and Cybernetics, 26–29 August 2004, pp. 3529–3535.
[15] S. Agarwal, A. Awan, D. Roth, Learning to detect objects in images via a sparse, part-based representation, IEEE Transactions on Pattern Analysis and Machine Intelligence 26 (11) (2004) 1475–1490.
[16] H. Schneiderman, T. Kanade, Object detection using the statistics of parts, International Journal of Computer Vision 56 (3) (2004) 151–177.
[17] B. Heisele, Visual object recognition with supervised learning, IEEE Intelligent Systems (2003) 38–42.
[18] A. Mohan, C. Papageorgiou, T. Poggio, Example-based object detection in images by components, IEEE Transactions on Pattern Analysis and Machine Intelligence 23 (4) (2001) 349–361.
[19] R. Duda, P. Hart, D. Stork, Pattern Classification. second ed., 2000, 640p.
[20] G. Amayeh and G. Bebis, A. Erol, M. Nicolescu, Peg-free hand shape verification using high order zernike moments, in: IEEE Workshop on Multi-modal Biometrics, (in conjunction with CVPR 2006), New York City, NY, June 17–18, 2006.
[21] G. Amayeh, G. Bebis, A. Erol, M. Nicolescu, A new approach to hand-based authentication, in: SPIE Defense and Security Symposium: Biometric Technology for Human Identification IV, Orlando, FL, April 9–10, 2007.
[22] G. Amayeh, G. Bebis, A. Erol, M. Nicolescu, A component-based approach to hand verification, in: IEEE Workshop on Biometrics,(in conjunction with CVPR 2007), Minneapolis, MN, June 18, 2007.
[23] A.K. Jain, R. Bolle, S. Pankanti, Biometrics: Personal Identification in Networked Society, Kluwer Academic Publishers, 1999.
[24] R. Sanchez-Reillo, Hand geometry pattern recognition through gaussian mixture modelling, in: Proceedings of the 15th International Conference on Pattern Recognition (ICPR'00), vol. 2, 2000, pp. 937–940.
[25] A.K. Jain, N. Duta, Deformable matching of hand shapes for verification, in: Proc. IEEE International Conference on Image processing, Kobe, Japan, October 1999, pp. 857–861.
[26] W. Xiong, C. Xu, S.H. Ong, Peg-free human hand shape analysis and recognition, in: Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'05), vol. 2, March 18–23 2005, pp. 77–80.
[27] L. Wong, P. Shi, Peg-free hand geometry recognition using hierarchical geometry and shape matching, in: IAPR Workshop on Machine Vision Applications, Nara, Japan, 2002, pp. 281–284.
[28] S. Ribaric, D. Ribaric, N. Pavesic, Multimodal biometric user-identification system for network-based applications, in: IEE Proceedings on Vision Image and Signal Processing, vol. 150, No. 6, December 2003, pp. 409–416.
[29] Y. Bulatov, S. Jambawalikar, P. Kumar, S. Sethia, Hand recognition using geometric classifiers, in: ICBA'04, Hong Kong, China, July 2004, pp. 753–759.
[30] A.K. Jain, N. Duta, Deformable matching of hand shapes for verification, in Proceedings of the IEEE International Conference on Image Processing, Kobe, Japan, October 1999, pp. 857–861.
[31] Y.L. Ma, F. Pollick, W.T. Hewitt, Using b-spline curves for hand recognition, in: Proceedings of the 17th International Conference on Pattern Recognition (ICPR'04), vol. 3, August 2004, pp. 274–277.
[32] A. Kumar, D.C.M. Wong, H.C. Shen, A.K. Jain, Personal verification using palmprint and hand geometry biometric, in: Time-Varying Image Processing and Moving Object Recognition, Guildford, UK, June 2003, pp. 668–678.
[33] E. Yoruk, E. Konukoglu, B. Sankur, J. Darbon, Shape-based hand recognition, IEEE Transactions on Image Processing 15 (7) (2006) 1803–1815.
[34] D.L. Woodard and P.J. Flynn, Personal identification utilizing finger surface features, CVPR, San Diego, CA, USA, 2005.
[35] Y.L. Lay, Hand shape recognition, Optics and Laser Technology 32 (2000) 1–5.
[36] R. Sanchez-Reillo, C. Sanchez-Avila, A. Gonzalez-Marcos, Biometric identification through hand geometry measurements, IEEE Transactions on Pattern Analysis and Machine Intelligence 22 (10) (2000) 1168–1171.
[37] R. Sanchez-Reillo, Hand geometry pattern recognition through gaussian mixture modelling, in: Proceedings of the 15th International Conference on Pattern Recognition (ICPR'00), vol. 2, 2000, pp. 937–940.
[38] R. Mukundan, K.R. Ramakrishnan, Fast computation of legendre and zernike moments, Pattern Recognition 28 (9) (1995) 1433–1442.
[39] S.O. Belkasim, M. Ahmadi, M. Shridhar, Efficient algorithm for fast computation of zernike moments, in: IEEE 39th Midwest Symposium on Circuits and Systems, vol. 3, August 1996, pp. 1401–1404.
[40] J. Gu, H.Z. Shua, C. Toumoulinb, L.M. Luoa, A novel algorithm for fast computation of Zernike moments, Pattern Recognition 35 (12) (2002) 2905–2911.
[41] A. Khotanzad, Y.H. Hong, Invariant image recognition by zernike moments, IEEE Transactions on Pattern Analysis and Machine Intelligence 12 (1990) 489–498.
[42] R.C. Gonzalez, R.E. Woods, Digital Image Processing, Prentice-Hall, 2002.
[43] R.M. Haralock, L.G. Shapiro, Computer and Robot Vision, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1991.
[44] S.X. Liao, M. Pawlak, On the accuracy of zernike moments for image analysis, IEEE Transactions on Pattern Analysis and Machine Intelligence 20 (12) (1998) 1358–1364.
[45] R.O. Duda, P.E. Hart, D.G. Stork, Pattern Classification, John-Wiley, 2001.
[46] Z. Sun, G. Bebis, R. Miller, Object detection using feature subset selection, Pattern Recognition 37 (2004) 2165–2176.
[47] L. Kotoulas, I. Andreadis, Real-time computation of zernike moments, IEEE Transactions on Circuits and Systems for Video Technology 15 (6) (2005) 801–809.
[48] O. Aubreton, L.F.C. Lew Yan Voon, B. Lamalle, G. Cathebras, F. Moniot, A cmos retina for zernike moments estimation, in: SPIE Conference on Electronic Imaging, vol. 5677, 2005, pp. 119–128.
[49] J. Kittler, M. Hatef, R. Duin, J. Matas, On combining classifiers, IEEE Transactions on Patterns Analysis and Machine Intelligence 20 (1998) 226–239.
[50] J. Phillips, H. Moon, S. Rizvi, P. Rauss, The feret evaluation methodology for face-recognition algorithms, IEEE Transactions on Patterns Analysis and Machine Intelligence 22 (10) (2000) 1090–1104.
[51] E. Yoruk, E. Konukoglu, B. Sankur, J. Darbon, Shape-based hand recognition, IEEE Transactions on Image Processing 15 (7) (2006) 1803–1815.
[52] F. Mokhtarian, A. Mackworth, A theory of multiscale, curvature-based shape representation for planar curves, IEEE Transactions on Pattern Analysis and Machine Intelligence 14 (8) (1992) 789–805.
[53] G. Bebis, G. Papadourakis, S. Orphanoudakis, Curvature scale space driven object recognition with an indexing scheme based on artificial neural networks, Pattern Recognition 32 (7) (1999) 1175–1201.
[54] D.L. Woodard, P.J. Flynn, Finger surface as a biometric identifier, Computer Vision and Image Understanding 100 (2005) 357–384.
[55] Cenker Oden, Aytul Ercil, Burak Buke, Combining implicit polynomials and geometric features for hand recognition, Pattern Recognition Letters 24 (13) (2003) 2145–2152.