Neurocomputing 188 (2016) 178–189

Contents lists available at ScienceDirect

Neurocomputing

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

An accurate slap fingerprint based verification system ☆

Puneet Gupta *, Phalguni Gupta

Department of Computer Science and Engineering, Indian Institute of Technology Kanpur, Kanpur 208 016, India

ARTICLE INFO

Article history:
Received 4 October 2014
Received in revised form
13 January 2015
Accepted 29 January 2015
Available online 19 December 2015

Keywords:
Biometric template update
Slap-image verification
Co-update

ABSTRACT

A slap fingerprint based verification system generally is highly secure. But it cannot handle the problems of temporal deformation and pose variation. In this paper, a template update algorithm has been proposed to ensure better performance of the system. During verification, each single fingerprint is matched and matching scores of all fingerprints are fused using an adaptive score level fusion. The performance of the proposed system has been evaluated on a challenging database containing 1800 slap-images acquired from 150 subjects. Experimental results show that the accuracy is increased by more than 3.5%.

1. Introduction

In the modern era, the advent of large digital data and improved information technology require data protection or security, which allows the access to genuine users while restricting the impostor. Traditional methods like keys or passwords are not much used in this regard, because these can be easily stolen, lost or spoofed. On the other hand, a biometric based recognition system is proliferating because it satisfies the properties like universality, uniqueness, permanence and acceptability [1]. The fingerprint is one such biometric trait which is extensively studied for recognition purpose [2]. There exist various feature extraction and matching algorithms for fingerprints which perform accurately under different conditions. Thus, multiple classifiers representing complimentary information can be fused to enhance the performance of a fingerprint based recognition system [3]. In spite of this, effective usage of fingerprints in recognition is restricted due to (i) environmental conditions which can generate wet/dry fingerprints, (ii) creases, cuts and bruises on the fingertip, (iii) occupation or age, which sometime smoothen the ridge-valley structure; and (v) applying undue pressure during fingerprint acquisition which can introduce elastic deformation in the acquired fingerprint image. These issues can be handled if multiple fingerprints are used during recognition [4]. If multiple fingerprints are acquired one by one then it is highly user unfriendly and time consuming. Therefore, multiple fingerprints are acquired simultaneously by using a slap-image scanner [5]. A slap fingerprint device acquires all fingerprints of a hand simultaneously. Some example of the acquired slap image are shown in Fig. 1.

A slap fingerprint based verification system works as follows. Initially, each single fingerprint present in a slap-image is extracted. This process is referred as *slap-image segmentation*. Its example is shown in Fig. 2. Each single fingerprint is used for fingerprint matching. Matching scores of all the fingerprints are fused to obtain the slap-image matching score. Since multiple fingerprints are used in a slap-image, it has less intra-class variation and high inter-class variation [6]. But it is observed that sometime partial fingerprints or bad quality fingerprints are acquired. These unavoidable factors can lead to improper slap fingerprint segmentation [7] and large intra-class variation that can degrade the effectiveness of slap fingerprint matching [8]. Both these factors deteriorate the verification accuracy [9].

It is advisable to use accurate slap fingerprint segmentation and slap fingerprint matching algorithm to achieve better results [10]. Further, a user can be enrolled multiple times to handle the problems of partial fingerprints or bad quality fingerprints using data fusion or template improvement algorithms. A system based on data fusion requires multiple fingerprint images during enrollment to obtain the accurate features [11]. It is user unfriendly and cost expensive during acquisition, thus avoided. Further, it cannot handle the temporal variations which can be introduced in the fingerprint [12]. On the other hand, a template improvement can be useful to handle the intra-class variations. It fuses the matched samples with stored templates for accurate feature extraction.

In this paper, an accurate slap-image based verification system has been designed. Its main contributions are as follows:

1. It uses template update method to reduce the intra-class variation in the slap-image. The proposed template update uses various constraints on fingerprint quality to ensure that beneficial result can be obtained. It has improved the performance because it removes spurious minutiae and retrieves the missed one.

☆Preliminary version of this paper [Paper Id. 413] has been accepted in International Conference on Intelligent Computing (ICIC'14), 2014 and sub-selected for Neurocomputing.

* Corresponding author. Tel.: +91 512 2597579; fax: +91 512 2597647.
E-mail address: puneet@cse.iitk.ac.in (P. Gupta).

http://dx.doi.org/10.1016/j.neucom.2015.01.111
0925-2312/

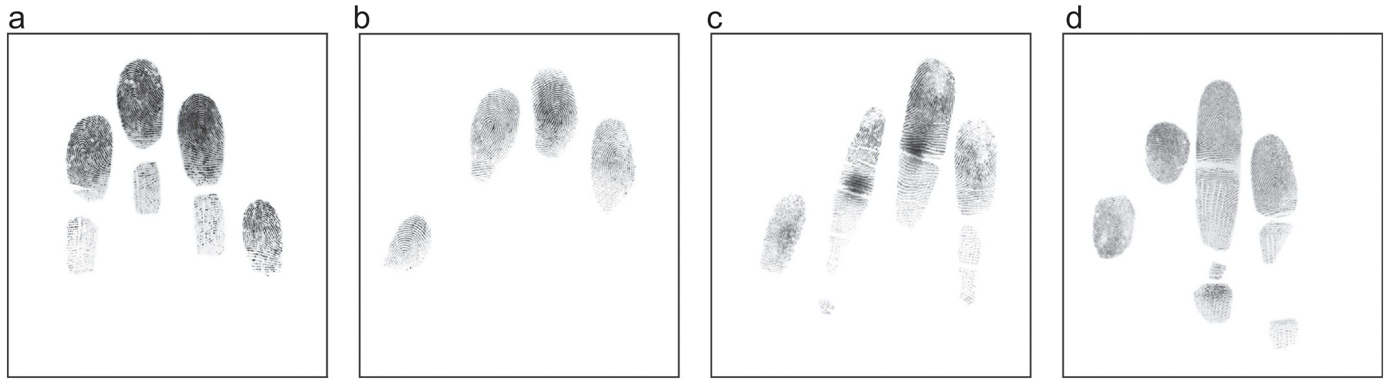

Fig. 1. Examples of slap-images.

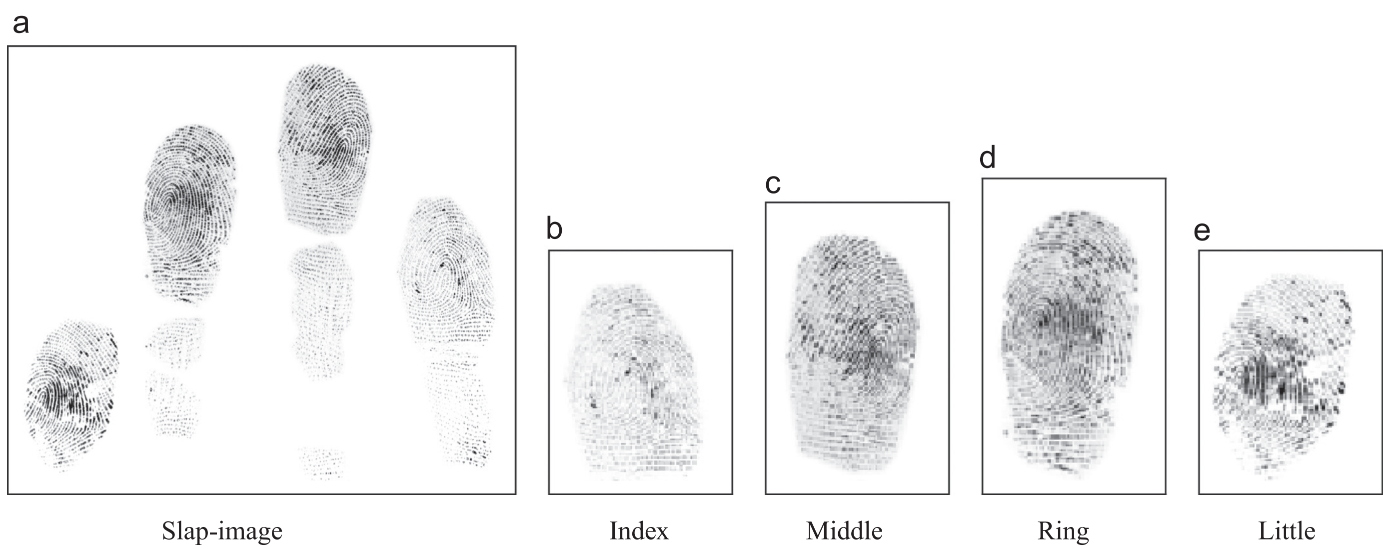

Fig. 2. Slap-image and its segmented fingerprints.

2. It proposes a line sweep algorithm to extract accurately the single fingerprints from the given slap image. This improves the slap fingerprint segmentation that eventually enhances the verification accuracy.
3. An adaptive score level fusion is used in this paper to fuse the matching scores of segmented single fingerprints. Adaptive weights are assigned based on the uniformity in ridge-valley structure. Symmetric filters are used to measure the uniformity in high curvature areas. Experimental results reveal that the proposed fusion is better than other existing fusion strategies.

This paper is organized as follows. Applicability of the template update, preliminary for slap fingerprint segmentation and slap fingerprint matching algorithms are discussed in the next section. In Section 3, the proposed template update technique has been presented. The design of slap-image based verification system along with the usage of the template update is presented in Section 4. Experimental results are analyzed in Section 5. Conclusions are given in the last section.

2. Literature survey

2.1. Template update

This section is divided into three subsections, viz., (i) requirement of template update; (ii) template improvement; and (iii) importance of classification in template update.

2.1.1. Requirement of template update

In the enrollment phase, a biometric sample is acquired, enhanced and processed to extract features. Extracted features are stored in the database corresponding to the user identity and are referred as enrolled templates. During the verification phase, a user claims an identity which is confirmed by matching the templates of acquired biometric sample with that of a claimed identity. Performance of a verification system is dependent on several factors and some of these are:

1. *Number of enrolled samples*: Templates stored in a database should capture all possible deformations that can be introduced in the biometric sample, for effective matching. It requires large data enrollment which is infeasible.
2. *Temporal variations*: Even though features in a biometric sample are assumed to be stable against time, there are some conditions introduced by time factor which can alter the features in a biometric sample [13]. As an instance, cuts or scars can be formed on fingerprints due to injury which can generate spurious minutiae.
3. *Sensor condition*: Sometimes due to improper interaction between a user and the acquisition sensor, partial fingerprints or highly deformed fingerprints can be acquired. This degrades the matching performance.
4. *Environmental condition*: The role of environmental condition in determining the verification performance is indispensable [14]. Sometime due to humidity and temperature, a normal fingerprint can be captured as a wet or dry fingerprint. Such a deterioration in

fingerprint quality can result in the generation of false minutiae and missing of genuine minutiae.

These issues are solved if multiple templates of a user can be acquired at different instances of time. This can be accomplished by using template improvement [15].

2.1.2. Template improvement

During template improvement, stored templates are modified by using the matched samples such that (i) intraclass similarities can be increased; and (ii) interclass similarities can be decreased. If the matching score between an acquired sample and a stored template is above a predetermined threshold, then the sample can be used to improve the stored template. Hence, threshold plays a significant role in deciding the performance of template updates. If it is high, then only few matched samples are found to be suitable for template update. All possible deformations cannot be captured. On the other hand, if the threshold value is low, then non-matching samples can also update the stored templates. Such a wrong update can severely degrade the system performance. In order to update the stored templates, multiple templates or single template can be used. In case of using multiple templates, one option is to store the templates containing high similarities. During verification, all templates of a user are used for matching. Thus, in case of using multiple templates, large number of templates should be used for effective results. But it is a time consuming stage. Another possible option by which multiple templates can be used is to store only those templates which can capture the maximum amount of deformation. In such cases, template size is restricted by a fixed number. But detection of suitable templates is a non-trivial task which requires more time and space. On the other hand, storing a single template is beneficial in terms of time and space. But its success primarily requires the correct determination of the biometric sample that can be used for template update which can modify a stored template by three ways, which are:

1. *Periodic template update*: It requires user re-enrollment after a fixed duration of time. Thus, it involves large operator intervention to ensure that templates are updated by using genuine user samples. Further, periodical re-enrollment makes it user inconvenient. But such a supervised template update is highly effective to abolish temporal changes.
2. *Template selection*: It replaces a stored template with the matching sample where matched samples are detected by using the following criteria: (i) it has maximum dis-similarity with other classes; (ii) it has a maximum similarity within a class; and (iii) the performance of the system should not decrease after replacement. Two such systems are proposed in [16]. One system detects the template having maximum intra-class variations and use this template for template selection. It is termed as DEND which focuses on capturing on variability. Another system in [16] termed as MDIST tries to maximize the similarity in users' samples.
3. *Template update*: Template of a matched sample is used to upgrade the stored template rather than replacement. It is categorized as either self updating or co-updating [17]. Difference in classification process is responsible for such a categorization. In self-update [18], only one biometric trait is used for classification while in co-update [19] multiple modalities are used. Thus, one can expect that co-update performs better than self-update because of better classification. Templates should be upgraded after classifying the matched sample such that the performance of the system improves against time. Like in [20], minutiae features present in multiple fingerprint templates of a user are consolidated such that false minutiae are eliminated while genuine minutiae from all the templates are preserved. It is further refined in [21] where local fingerprint quality is also used along with minutiae representation.

2.1.3. Importance of classification in template update

Despite of various advantages of template update, its utilization is restricted because these are fragile to impostor attacks. It has been observed that by using the domain knowledge, one can breach the verification system employing template update [22]. Hence, one can observe that accurate classification technique is of prime importance to restrict the impostor entry.

2.2. Slap fingerprint segmentation

2.2.1. Framework

In this subsection, the basic steps required in a slap-image segmentation are discussed. Each slap-image is preprocessed which involves (i) noise reduction; (ii) sometimes down-sampling; and (iii) separation of foreground and background pixels. Foreground pixels are the pixels belonging to a hand. Since these possess high intensity variation in comparison to background pixels, various anisotropic measures can be used for separating foreground and background pixels. Neighborhood constraints are applied to the detected foreground pixels to detect the *components*. A component is represented by its geometrical properties like center and orientation. Components are clustered into four classes such that components belonging to a common finger share same class while components belonging to different fingers lie in different classes. Such a clustering requires knowledge of hand geometry, constraints on hand placements, sensor conditions and geometrical properties of components [23]. Since fingerprint lies at the top in a finger, the topmost component from each class is referred as fingerprint component. A slap-image segmentation also requires to label the fingerprint components as index, middle, ring or little finger of left/right hand. Processes required in a slap-image segmentation framework are shown in Fig. 3.

2.2.2. Background

Components can be detected using anisotropic measures followed by neighborhood connectivity algorithms [23,24] or mean-shift algorithm [25]. The mean-shift algorithm is avoided because it involves more computational time. Component detection algorithms often fail if components contain non-elliptical shape or dull prints. Algorithm [24] handles these issues using local entropy. Detecting the fingerprint component from the extracted component is considered to be the most important stage in slap fingerprint segmentation [26]. The extracted components can be clustered into four finger classes using the intuition that fingers can be separated by straight parallel lines [27]. The intuition fails if slap images contain open fingers. Clustering can be improved if one uses better hand geometry constraints and better estimates of geometrical properties as mentioned in [28–30]. Further, fingerprint components can be labeled and hand type (either left or right) can be determined if one uses domain knowledge of hand [24].

2.3. Slap fingerprint matching

In this section, matching scores of single fingerprint matching are fused to obtain the matching score of slap-image matching. In [31], fingerprints are fused by using serial fusion. Initially, a selected fingerprint is matched to prune out several non-matching candidates. Then, another single fingerprint is considered to prune

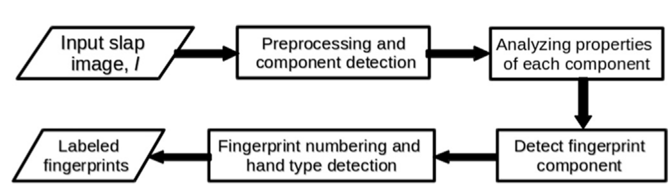

Fig. 3. Framework for slap-image segmentation.

the non matching one of the remaining candidates. It is iteratively repeated to eliminate the non-matched candidates in each iteration till no fingerprint is left or a suitable decision (either match or mismatch) is taken with high confidence. Since ordering used to match the different modalities play a crucial role in the success of serial fusion, serial fusion in [31] has performed poorly in various cases. Further, it requires that all extracted fingerprints should have good quality which is not always possible. In [32], feature level fusion is used to fuse the matching scores of fingerprints. In [29], fixed weighted score level fusion is used for fusing the matching scores of all four fingerprints. These weights are given by minimizing the total error rate. Similarly, in [33], weighted score level fusion is used. But its weights are adaptively chosen based on the fingerprint quality.

3. Proposed template update

In this section, the system which updates the enrolled fingerprint templates using matched sample, is presented. It assumes the presence of all four fingerprints in the slap-image. Thus, four fingerprint data are stored corresponding to a user during enrollment. A fingerprint data consists of minutiae and fingerprint image present in it. In this paper, all minutiae are used for matching and are referred as a template. Fingerprint images are useful for fingerprint registration which helps in template update.

3.1. Database creation during enrollment

In order to create the database during enrollment, slap-image is first segmented into four distinct fingerprints. Extracted single fingerprints are labeled into index, middle, ring or little finger of left/ right hand. Minutiae from labeled fingerprints are detected by using [27]. Corresponding to each fingerprint, a template storing the extracted minutiae is formed. A user identity is stored in the database by storing the fingerprint images and templates of all four fingerprints.

Algorithm 1. *TemplateUpdation*$(ST_i, F_i, T_i, \overline{T}_i)$.

Require: Stored image ST_i and stored template T_i which needs to be updated by using image F_i and template $\overline{T}_i$
Ensure: ST_i and T_i store the updated image and template respectively.

```
Let ST_i^q and F_i^q denote the fingerprint quality corresponding to ST_i and F_i respectively.
/* Applying [34] for fingerprint registration */
(A, T_F) = FingerprintRegistration(ST_i, F_i)
/* A is overlapping area and T_F is registered image of F_i with respect to ST_i */
/* th_1 and th_2 are predetermined thresholds */
if ((ST_i^q > th_1) AND (F_i^q > th_1) AND (A > th_2)) then
    ST_i = BinaryOR(ST_i, T_F)    // Concatenating images
    Extract minutiae from ST_i and store in T_i
else if ST_i^q < F_i^q then
    T_i = T̄_i    // Replacing templates
    ST_i = F_i // Replacing images
end if
return T_i and ST_i
```

3.2. Template update

Assume that stored fingerprint images and templates corresponding to the query slap-image, Q, are represented by F_i and $\overline{T}_i$ (for $i=1, 2, 3, 4$) respectively. Also, assume that stored fingerprint images and templates of user B are used to update the template which are represented by ST_i and T_i (for $i=1, 2, 3, 4$) respectively. The proposed template update requires that genuine minutiae present in both the templates are present in the updated template while spurious minutiae are removed. It is observed that minutiae present in both the templates are slightly deformed due to skin elasticity. Such displaced minutiae should be determined and used only once to create the updated template. This requires an accurate fingerprint registration, which is robust against skin elastic deformations.

Fingerprint registration can be accomplished by using only the minutiae features present in a fingerprint [35]. But it is not much accurate due to the local nature of minutiae features. More sophisticated fingerprint registration requires fingerprint skeletons for registration. Use of the skeleton is more advantageous than the use of minutiae during fingerprint registration because:

1. Since minutiae points are sparsely distributed in restricted fingerprint areas and skeleton is distributed over the entire fingerprint, use of skeleton offers a more accurate estimate of fingerprint registration over entire fingerprint.
2. As skeleton is continuous and contains a large number of feature points as compared to minutiae, registration can be established by using a large number of features. In addition, skeleton is described by a large number of ridge points and thus it contains more information than minutiae. These factors make the skeleton correspondence more accurate than the minutiae correspondence.
3. In addition, minutiae cannot be accurately detected in case of bad quality fingerprints, but skeleton can be effectively extracted.

Therefore, skeleton based image registration [36] is used where skeleton is obtained from fingerprint image [37] and is divided into several ridge curves. These ridge curves are used for matching and obtaining ridge curve correspondences. Fingerprint registration is achieved by using these ridge curve correspondences. Let transformation parameter T_E registers F_i with respect to ST_i and T_F is the transformed image of F_i.

Since registered images can handle the elasticity effect of the skin, minutiae information present in the registered images can be easily consolidated. But in some cases, it is observed that the system performance can be degraded even if the genuine matched sample is used [38] for updating the stored templates. Therefore, performance improvement is first assured before template update by satisfying two conditions, which are:

1. *Sufficient uniform structure*: If a matching sample has less foreground area or poor quality (dry or wet fingerprints) then such a matching sample should not be used to update the stored template. Reasons behind this are that (i) such samples can easily generate false minutiae points and spurious ridge-valley flow; and (ii) it has less features required for accurate fingerprint registration. Thus, if either ST_i or F_i has low uniform ridge valley structure, then template update is avoided. In this paper, the uniformity is given by fingerprint quality determined by algorithm NFIQ [39]. Assume that the quality of ST_i and F_i determined after applying the NFIQ [39] are given by ST_i^q and F_i^q respectively. Therefore, if ST_i^q or F_i^q is less than a predetermined threshold, then template update is avoided.
2. *Sufficient overlapping area*: It is observed that sometime fingerprint registration is not effective. It occurs when the ridge curve correspondence cannot be established due to low overlapping area. In the absence of accurate fingerprint registration, minutiae consolidation would be inappropriate. Let A represent the overlapping area between ST_i and T_F which is obtained after fingerprint registration. Thus, if A is less than a predetermined threshold, then the template update is avoided.

If both the above conditions are fulfilled, then one should update T_i by using $\overline{T}_i$. Initially, ST_i is updated by applying pixel-wise binary OR operation on ST_i and T_F. All minutiae present in ST_i are extracted and are stored as T_i in the database. All genuine minutiae are present and counted only once in T_i because fingerprint deformations are taken care by fingerprint registration. Further, spurious minutiae are eliminated by fingerprint registration. Sometime biometric sample acquired during enrollment has poor quality which can generate spurious template. If such spurious templates are stored in the database then it can lead to poor performance. Thus, such templates are replaced by using good quality matched samples. Therefore, if the quality of F_i (F_i^q) is greater than the quality of ST_i (ST_i^q) then T_i is replaced by $\overline{T}_i$. Also, ST_i is replaced by F_i in such a case. The proposed template update is explained in Algorithm 1.

4. Slap fingerprint verification system

In this section, a verification system based on slap fingerprint is presented. All individual fingerprints present in a slap-image are accurately extracted. Each fingerprint is matched with the stored templates to generate the matching score. Eventually, matching scores of all fingerprints are fused for slap-image matching. Flow-graph of the proposed verification system is shown in Fig. 4. Major steps of the proposed slap fingerprint based verification system are (i) slap fingerprint segmentation; (ii) single fingerprint matching; and (iii) quality based fusion of matching scores (Fig. 5).

4.1. Slap fingerprint segmentation

4.1.1. Component detection

In this subsection, components present in a slap-image along with their geometrical properties are determined. Foreground pixels (i.e., pixels belonging to a hand) in a slap-image contains large variation as compared to the background pixels. Therefore, an anisotropic measure, local entropy, is used to separate the foreground and background pixels [24]. This allows accurate detection of foreground pixels even if a slap-image consists of dull prints. Foreground pixels are united by using neighborhood connectivity algorithm [10] to accurately extract the slap-image components even if these components are non-elliptical in shape. Several detected components are formed due to noise in the slap-image which are removed by using the observation that such components have small area. Extracted components are represented by using the geometrical properties, viz., center and orientation, which are estimated by using an ellipse fitting algorithm [25]. That is, center (x_{C_i}, y_{C_i}) for C_i component is given by:

$$x_{C_i} = \frac{\sum_{i=1}^{m} x_i}{m} \quad y_{C_i} = \frac{\sum_{i=1}^{m} y_i}{m} \tag{1}$$

where m is the number of foreground pixels in C_i while x_i and y_i are the x and y coordinates of foreground pixels. Similarly to obtain orientation θ_i for C_i component, variance-covariance matrix, A_{C_i}, is calculated by using (x_{C_i}, y_{C_i}) [40]. Let eigenvalues for A_{C_i} are represented by λ_a and λ_b.

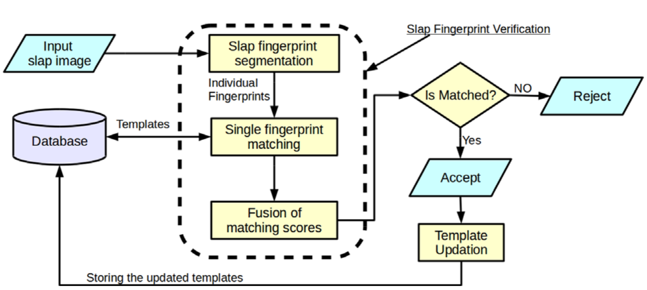

Fig. 4. Flow-graph of the proposed verification system.

Without loss of generality, consider that λ_a is greater than λ_b and eigenvector corresponding to it is given by e. Then θ_i is given by

$$\theta_i = \tan^{-1}\left(\frac{e_2}{e_1}\right) \tag{2}$$

Steps required for detecting the components are shown in Algorithm 2.

Algorithm 2. ComponentDetection (I).

Require: Slap-image I
Ensure: X and Y store the x and y coordinates of centers while Θ store the orientations for all the components present in I.
1: Apply local entropy on I to obtain I_e.
2: Apply global thresholding on I_e to separate foreground and background pixels.
3: Apply neighborhood connectivity to detect the components.
4: Remove small sized components by using constraints on area. Let p components are detected.
5: Find center and orientation of each component by using ellipse fitting algorithm.
Let (x_{C_i}, y_{C_i}) is the center and θ_i is the orientation of C_i component.
6: $X = \left(x_{C_1}, x_{C_2}, \ldots, x_{C_p}\right)$
7: $Y = \left(y_{C_1}, y_{C_2}, \ldots, y_{C_p}\right)$
8: $\Theta = (\theta_1, \theta_2, \ldots, \theta_p)$
9: **return** (X, Y, Θ)

4.1.2. Fingerprint component detection

The crucial stage in slap-image segmentation is the detection of fingerprint components from the detected components. In this subsection, a line sweep based algorithm is proposed for such a detection. Initially, global orientation of hand, g_h, is evaluated by using:

$$g_h = \frac{\sum_{i=1}^{p} \theta_i}{p} \tag{3}$$

where p is the total number of components detected in the slap-image. Consider a line perpendicular to global orientation of hand plane which is given by H_L. Line H_L is swept across the slap-image plane such that it starts from the top of the slap-image and stops when the topmost four components are detected. These components are marked as fingerprint component. Let C^1, C^2, C^3 and C^4 represent the fingerprint components. Steps involved to detect the fingerprint components are given in Algorithm 3.

Algorithm 3. FingerprintComponentDetection (X, Y, Θ, I).

Require: X and Y store the x and y coordinates of centers while Θ store the orientations for all the components detected in slap-image, I.
Ensure: C^1, C^2, C^3, C^4 contains the fingerprint component.
1: Find global orientation of hand, g_h, by using $g_h = \frac{\sum_{i=1}^{p} \theta_i}{p}$
2: Consider a line H_L in the direction of $(g_h + 90)^\circ$ and passes through (0,0).
3: Apply line sweep algorithm on I by using H_L till topmost four components are detected and stored as (C^1, C^2, C^3, C^4).
4: **return**: (C^1, C^2, C^3, C^4)

4.1.3. Fingerprint labeling

In this subsection, detected fingerprint components are labeled as index, middle, ring or little finger of the left or right hand. Initially, fingerprint components are numbered to indicate left to right placement in a hand. Fingerprint numbering is established by using the

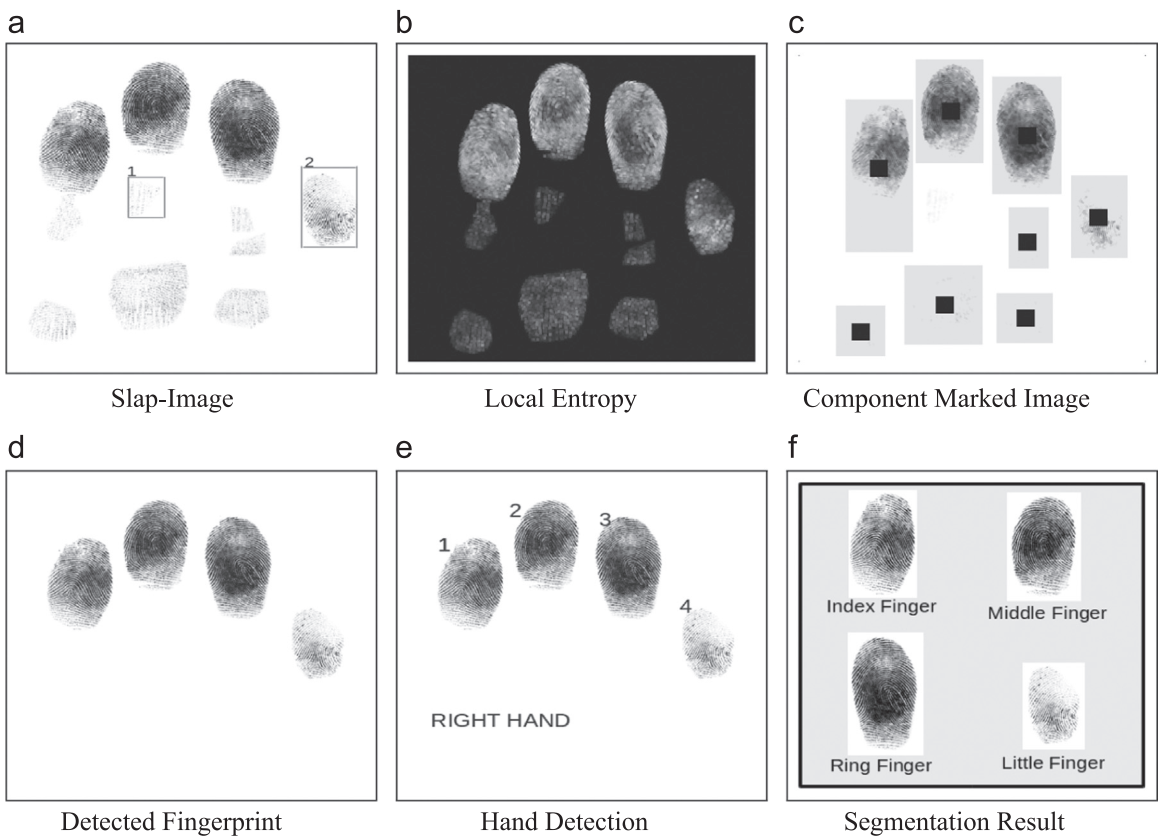

Fig. 5. Various stages of the proposed slap fingerprint segmentation system.

Euclidean distance between centers of fingerprint components along with the hand geometry constraints explained in [24]. These numbering is useful for hand detection. It is apparent that in a hand geometry of right hand, the difference between lengths of the middle finger and the index finger is less than that of the ring finger and the little finger. Mathematically, consider x is the difference between the length of the middle finger and the index finger while y is the difference between the length of the ring finger and the little finger. Then, if $x<y$ then the slap-image belongs to right hand. For better understanding, consider Fig. 6 where a right hand image along with x and y are shown. Such a difference in lengths is calculated by using a measure *Relative length* proposed in [10]. Steps required for hand detection by using *Relative length* measure are given in Algorithm 4. Eventually, fingerprint components in the slap-image are labeled by using fingerprint numbering and hand type detection.

Algorithm 4. *HandTypeDetection*$\left(C^1, C^2, C^3, C^4\right)$.

Require: Four fingertip components, C^1, C^2, C^3 and C^4
Ensure: *Hand* stores the hand-type present in the *I*, i.e., *LeftHand* or *RightHand*.

$x = Relative\ length\left(C^1, C^2\right)$
$y = Relative\ length\left(C^3, C^4\right)$
if $x \le y$ **then**
  $Hand = RightHand$
else
  $Hand = LeftHand$
end if
return *Hand*

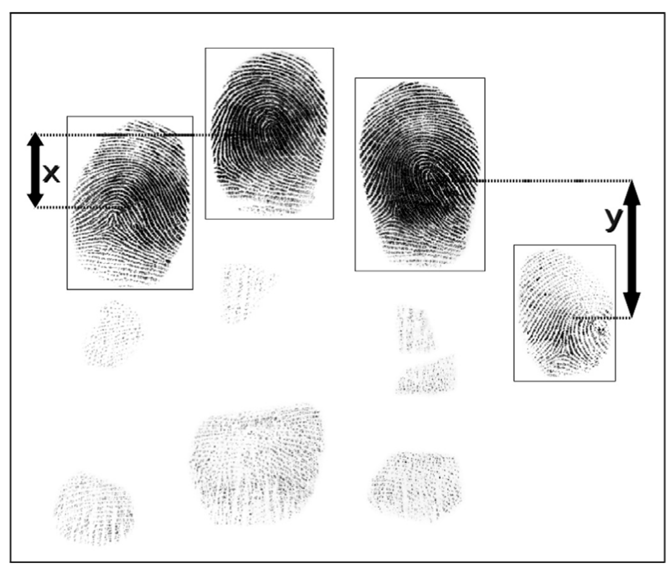

Fig. 6. An example of hand logic.

4.2. Single fingerprint matching

In order to verify the user's identity, each query fingerprint template is matched with a suitable fingerprint template of the claimed identity. For single fingerprint matching, Bozorth's matcher [27] is used because it is highly accurate and robust against geometric deformation. Let S_1, S_2, S_3 and S_4 be the matching scores of four fingerprints of the query.

4.3. Slap fingerprint matching

Let us assume that the identity claimed by the user is represented by $\overline{U}$. To verify this claim, all single fingerprints are matched and their matching scores are fused to obtain slap-image matching score. If fused score is more than a predefined threshold, then the claim by the user is correct and the user belongs to $\overline{U}$. In this paper, adaptive score level fusion is used to fuse matching scores of four fingerprints. The proposed fusion strategy relies on the intuition that if a single fingerprint does not contain uniform ridge-valley flow then, it should be given low weight during fusion. Thus, a uniformity measure is defined corresponding to each single fingerprint based on its uniform ridge-valley structure, which controls its contribution in the score level fusion.

4.3.1. Measuring uniformity in single fingerprint

In this section, uniformity measure of a fingerprint is determined. The proposed algorithm assigns higher weights in fingerprint areas containing uniform ridge-valley flow irrespective of the curvature, as compared to the dry or wet fingerprints areas. There are various possible weight assignment [41]. Some of these measures the linearity of orientation field like orientation certainty level [42]. Since the uniform flow in a singular point close-by fingerprint areas contain large curvature, it is avoided because it assigns low values in such areas [43]. Likewise, weight assignment using clarity between the ridge-valley structure like mean, variance and local contrast are avoided. In this paper, weights are assigned using symmetric filters because uniformity in ridge-valley flow of singular points close-by areas can be measured by parabolic symmetric filters [44]. Fundamentally, a symmetric filter work as follows. Symmetric filter corresponding to a particular shape is created using the domain knowledge. It is convolved with the edge information of the input image to obtain the filter response that represents the similarity between the edges and a particular pattern. Typically, following uniform ridge-valley pattern are present in a fingerprint: linear pattern, pattern near core points and delta points. Symmetric filters of order -1, 0 and 1 can be used to model such patterns [35]. A symmetric filter h_j corresponding to order j (for $j=-1, 0, 1$) is given by:

$$h_j(x,y)=\begin{cases}(x+iy)^j\cdot e^{-\frac{x^2+y^2}{2\sigma^2}} & \text{if } j\ge 0\\ (x-iy)^{|j|}\cdot e^{-\frac{x^2+y^2}{2\sigma^2}} & \text{otherwise}\end{cases} \tag{4}$$

where (x,y) denote a pixel location and σ is a predefined constant. In case of fingerprints, edge information or orientation tensor, z, is given by

$$z=\cos(2\theta)+i\sin(2\theta) \tag{5}$$

where θ denote the orientation field of the fingerprint. In this paper, orientation field at each pixel is estimated by applying the algorithm [45]. Let S_j stores the similarity between a uniform pattern represented by h_j and z. It is obtained by convolving h_j with z [46]. That is,

$$S_j=\frac{\langle z,h_j\rangle}{\langle |z|,h_0\rangle} \tag{6}$$

In this equation, normalization is also performed along-with convolution to restrict S_j between 0 and 1. To model the uniform patterns in a fingerprint, filter responses S_0, S_1 and S_{-1} are evaluated using Eq. (6) corresponding to h_0, h_1 or h_{-1} respectively. A pattern in a fingerprint area should resemble to only one of the symmetric filter. The reason for its resemblance to several symmetric filters is that it contains non-uniform ridge-valley flow generated due to dry or wet fingerprint areas. Thus, filter responses in such areas should be significantly reduced to indicate dry or wet fingerprint areas. Hence, filter responses are modified using

$$\hat{S}_j=S_j\prod_{k\in\{-1,0,1\}\setminus j}(1-|S_k|) \tag{7}$$

where $\setminus$ represent the set difference operation while $\hat{S}_j$ denotes the modified filter response corresponding to S_j. All the modified filter responses $\hat{S}_{-1}$, $\hat{S}_0$ and $\hat{S}_1$ are fused using

$$Q_1=\hat{S}_{-1}+\hat{S}_0+\hat{S}_1 \tag{8}$$

Mean of Q_1 gives the uniformity measure for the fingerprint. Let Q represent the mean. Algorithm 5 shows the steps required for the proposed weight estimation. Its example is shown in Fig. 7. Fig. 7 (h) indicates that by using the proposed weight estimation algorithm, low weights are assigned in non-uniform ridge-valley flow areas while high weights are assigned in uniform ridge-valley flow areas.

Algorithm 5. MeasuringUniformity (I).

Require: Input fingerprint image, I.
Ensure: Q contains the uniformity measure for I.
Symmetric filters h_0, h_1 and h_{-1} are formed by using Eq. (4), for $n=0$, 1 and -1.
Apply algorithm [45] on I to obtain the orientation field, θ.
Find orientation tensor, z using Eq. (5).
Apply Eq. (6) to obtain the filter responses, S_0, S_1 and S_{-1} corresponding to h_0, h_1 and h_{-1} respectively.
for each pixel (x,y) in I **do**
  $\hat{S}_0(x,y)=S_0(x,y)\cdot(1-|S_1(x,y)|)\cdot(1-|S_{-1}(x,y)|)\cdot F_I(x,y)$
  $\hat{S}_1(x,y)=S_1(x,y)\cdot(1-|S_{-1}(x,y)|)\cdot(1-|S_0(x,y)|)\cdot F_I(x,y)$
  $\hat{S}_{-1}(x,y)=S_{-1}(x,y)\cdot(1-|S_1(x,y)|)\cdot(1-|S_0(x,y)|)\cdot F_I(x,y)$
  //Fusion of all the modified filter responses
  $Q_1(x,y)=\hat{S}_0(x,y)+\hat{S}_1(x,y)+\hat{S}_{-1}(x,y)$
end for
Divide Q_1 into non overlapping blocks of block-size, B.
for each block k in Q_1 **do**
  Find mean and replicate it at each pixel of the block k.
end for
return (Q)

4.3.2. Adaptive score level fusion of matching scores

Assume that Q_i^q and Q_i^s denote the uniformity measure for the ith single fingerprint obtained from query and stored template images respectively. Then, slap-image matching score, M is given by:

$$M=\frac{\sum_{i=1}^{4}\left(\min(Q_i^q,Q_i^s)\times S_i\right)}{\sum_{i=1}^{4}\min(Q_i^q,Q_i^s)} \tag{9}$$

If M is greater than a predetermined threshold, then the identity claim by the user is correct, i.e., I is acquired from $\overline{U}$. In such a case, templates of I are used to update the stored templates of $\overline{U}$. Steps involved for fusing the matching scores are explained in Algorithm 6.

Algorithm 6. *Fusion_of_Matching_Scores*($C,\overline{C},S,th$).

Require: C and $\overline{C}$ stores the query and stored single fingerprints templates; S stores the matching scores; and th is a predetermined threshold.
Ensure: D contains whether C and $\overline{C}$ are *Matched* or *MisMatched*.
Uniformity measure of single fingerprints in C are evaluated using Algorithm 5 and are stored as q_i for i_{th} fingerprint.
Similarly, uniformity measure of single fingerprints in $\overline{C}$ are evaluated and are stored as $\overline{q}_i$ for i_{th} fingerprint.
Evaluate $w_i=\max(q_i,\overline{q}_i)$ for $i=1$ to 4 // max is maximum operation
$M=\frac{\sum_{i=1}^{4}(w_i\times S_i)}{\sum_{i=1}^{4}w_i}$
if $M\le th$ **then**

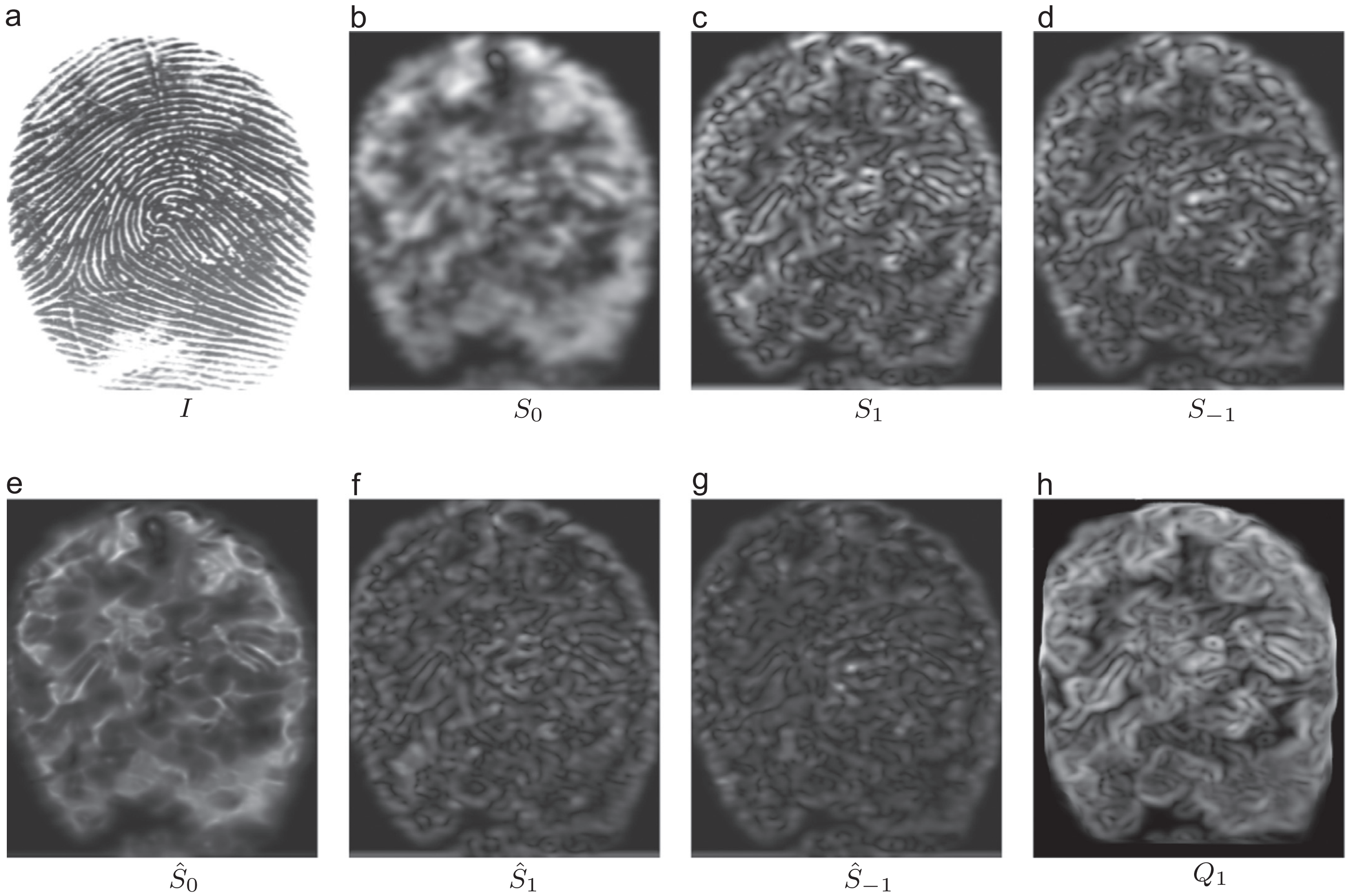

Fig. 7. Example of symmetric filter based fingerprint quality.

    D=*MisMatched*
else
    D=*Matched*
end if
return *D*

5. Experimental results

To the best of our knowledge, there does not exist any publicly available database for slap-images. We have created our own database at Indian Institute of Technology Kanpur, India to evaluate the performance of the proposed system. It consists of 1800 slap-images which are captured from 150 subjects. Data acquisition has been carried out in rural areas under non-controlled environment. Further, subjects are from different age groups who are involved in significant manual work. All these factors result in the creation of a challenging database which contains poor quality fingerprints and can be applied to real world scenarios. Each slap-image has a size of 1600×1500 pixels. Data acquisition has been carried out in two different sessions with a minimum time gap of two months to introduce temporal variation. In each session, 3 slap-images per hand have been acquired from each subject. Thus, total six slap-images per hand are acquired. Since different hands means different class in the proposed system, 300 separate classes are formed to evaluate the performance. A computer desktop of Intel Pentium 8 processor, 2.8 GHz and 4GB RAM is used to conduct the experiments. In addition, MATLAB framework is used for implementation. The performance of the system is measured by

$$accuracy = (100 - EER)\% \tag{10}$$

where *EER* is the equal error rate, that is, the false accept rate which is same as false reject rate.

5.1. Parameter tuning

Parameter used in the proposed algorithm are set in such a way that we achieved the best performance:

1. *Parameters for template update*: Threshold th_1 which is used in template update to satisfy the conditions of sufficient uniform structure is considered as 50% of the uniform structure present in good quality fingerprints. We have selected randomly 20 good quality fingerprints. Mean of their quality [39] is found to be 0.7 and hence, th_1 is set to 0.35. Another threshold th_2 used in template update for satisfying the conditions of sufficient overlapping area is set as 50% of the total overlapping area present in the selected 20 fingerprints.
2. *Parameters for measuring uniformity*: Further, to measure the uniformity in ridge-valley pattern, we have used orientation field and symmetric filters. Parameters used to estimate orientation field are set following the methods presented in [45,47]. Also, parameters given in [48] has been used to design symmetric filters for singular point detection. That is, σ is set to 0.8 for the size of symmetric filter of 42×28.

Table 1
Comparative results in terms of accuracies (in %).

Fingerprint used	Database part used				
	P_2	P_3	P_4	P_5	P_6
Template update					
Without					
Index	84.37	84.13	83.62	83.70	83.05
Middle	87.19	87.06	86.03	86.28	86.12
Ring	86.82	86.57	85.76	85.09	85.39
Little	80.26	80.23	78.71	78.52	78.57
Combined[a]	89.79	89.48	88.26	88.73	88.51
With					
Index	84.37	85.57	85.43	87.23	88.32
Middle	87.19	88.42	88.16	89.58	90.71
Ring	86.82	87.63	87.41	88.28	88.95
Little	80.26	82.12	81.95	84.37	85.94
Combined[a]	89.79	91.78	91.23	92.91	93.74

[a] In this, all fingerprints are fused by using the proposed quality based fusion.

5.2. Performance of template update

Experimental results which demonstrate the effectiveness of the proposed template update are shown in Table 1. These experiments are conducted by dividing the database into six equal parts $P_i, i = 1, 2, \ldots, 6$ such that each part contains one slap-image from each class. The part P_1 is used as a gallery set while other parts constitute probe set during evaluation. Probe set is matched in a sequential manner such that stored templates can be updated after matching. Matching starts from P_2 and ends at P_6. Also, results for matching of single fingerprints and combined multiple fingerprint matching are separately shown in Table 1, for better understanding of the proposed template update. In addition, results are divided into two categories: one uses template update while another does not. It can be inferred from Table 1 that:

1. Performance of single fingerprints and combined fingerprint are almost stable if the template update is not used. Also, accuracy of the system when used for little finger is the least. It is because fingerprint from little finger has the least area (thus less features) as compared to other fingerprint. In addition, the performance of the combined fingerprint is better than single fingerprint.
2. By using the proposed template update, performance of single fingerprint matchings and combined fingerprint matching are substantially improved. Its apparent reason is that the proposed template update can effectively handle the problems of partial fingerprint, fingerprint deformation and template aging.
3. In the results, slight reduction in accuracy can be observed in P_4. It occurs due to high temporal changes. Since P_1, P_2 and P_3 are captured in the first session with a small time lag, small deviation can be observed. In contrast, P_4 which is captured in second session has a high time lag compared to P_1, P_2 or P_3. Thus, various possible deformation like cuts or scars can be found in P_4 which are absent in P_1, P_2 and P_3. One can observe that it is better to use template update.

Comparative study of the proposed system with other systems has been presented using Table 2 and receiver operating characteristics (ROC) curve shown in Fig. 8. Experimental results in Table 2 are focused to examine the usefulness of (i) the proposed co-update system over self update system; and (ii) the proposed template update over template selection system. Template selection systems used for testing are DEND and MDIST [16]. Four different systems are compared with the proposed system, viz., (i) System I uses DEND along with self update method; (ii) System II uses MDIST along with self update method; (iii) System III uses DEND along with the proposed co-update method; and (iv) System IV uses MDIST along with the proposed co-update method.

Table 2
Comparative results in terms of accuracies (in %).

System used	Database part used[a]				
	P_2	P_3	P_4	P_5	P_6
Template update					
Index					
System I	84.37	84.43	83.77	84.03	84.36
System II	84.37	84.52	83.71	84.17	84.49
System III	84.37	84.78	83.86	85.01	85.43
System IV	84.37	84.84	83.95	85.68	85.74
Proposed	84.37	85.57	85.43	87.23	88.32
Middle					
System I	87.19	87.53	86.37	86.64	86.73
System II	87.19	87.67	86.25	86.78	86.79
System III	87.19	87.89	86.41	87.51	87.62
System IV	87.19	87.98	86.58	87.47	87.75
Proposed	87.19	88.42	88.16	89.58	90.71
Ring					
System I	86.82	86.89	85.73	85.91	86.17
System II	86.82	86.94	85.56	85.79	86.21
System III	86.82	87.05	86.21	86.64	86.98
System IV	86.82	87.17	86.25	86.73	87.03
Proposed	86.82	87.63	87.41	88.28	88.95
Little					
System I	80.26	80.72	79.95	80.23	80.32
System II	80.26	80.83	79.83	80.18	80.41
System III	80.26	81.07	80.34	81.81	82.93
System IV	80.26	81.19	80.39	81.74	82.98
Proposed	80.26	82.12	81.95	84.37	85.94
Combined[b]					
System I	89.79	89.89	86.93	87.83	88.09
System II	89.79	89.92	86.88	87.91	88.27
System III	89.79	90.47	88.72	89.22	89.96
System IV	89.79	90.51	88.56	89.38	90.08
Proposed	89.79	91.78	91.23	92.91	93.74

[a] It is sequentially executed.
[b] In this, all fingerprints are fused by using the proposed quality based fusion.

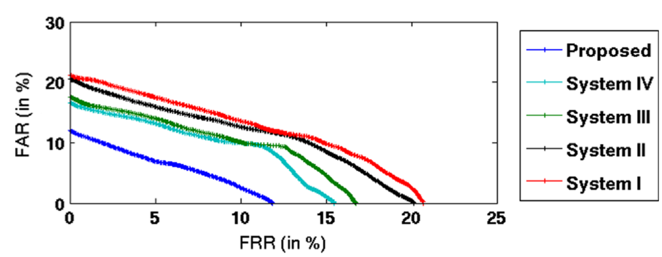

Fig. 8. ROC curve for various template update systems.

In this, self update system considers only a single fingerprint for classification. Some findings from Table 2 are:

1. Both System III and System IV which are based on co-update perform better than System I and System II which are based on self update.
2. The proposed template update system shows better accuracy than other systems which are based on template selection.

Thus, it can be inferred from Table 2 that the proposed co-update based system can significantly improve the verification accuracy of fingerprint based system. Since template is updated in off-line mode, time comparison is not useful.

5.3. Performance of the proposed fingerprint component detection

It has been observed that out of 1800 slap-images, the proposed fingerprint component detection based on line sweep algorithm, has failed to detect single fingerprints from 32 slap-images. On the other hand, algorithms [24,10] have failed to detect single fingerprints from 40 slap-images. Algorithms [24,10] shows more failure cases than the proposed one because it requires the local orientation of each component which can be sometimes spurious due to bad quality or improper acquisition. In addition, average time taken by the proposed fingerprint component detection is 0.03 s while algorithms [24,10] require 0.07 s. Thus, the proposed line sweep based fingerprint component detection performs better than the existing well known system in terms of performance and time computations. Reasons behind the generation of failure cases by the proposed fingerprint component detection are:

1. *Noisy background*: Sometime dust, sweat or latent fingerprints available on the acquisition device can merge with two different components.
2. *Other hand components*: It may occur that thumb or other part of the palm may touch the acquisition device during data acquisition.
3. *Smaller size of fingertip component*: In some slap-images, small part of fingerprint is acquired due to inappropriate hand placement, non-uniform pressure or worn out ridge-valley structure.
4. *Improper orientation*: Wrong orientation of hand sometime leads to wrong fingerprint component detection.

Some examples of wrongly detected fingertip components are shown in Fig. 9.

5.4. Performance of the proposed score level fusion

The performance of a slap fingerprint based system is highly dependent on the fusion of single fingerprint matching scores [49]. The proposed fusion is compared with some well known algorithms and results are presented by Table 3 and ROC curve shown in Fig. 10. In Table 3, all the single fingerprints of a slap-image are extracted using the proposed slap fingerprint segmentation. It can be inferred from Table 3 that:

1. Serial level fusion has the least performance because it requires good quality of fingerprints and a suitable ordering for fingerprint matchings which are infeasible in the case of slap fingerprint. Similarly, feature level fusion has poor performance, where all fingerprints in a slap-image should have good quality.
2. Score level fusion proposed in [29] has less accuracy than other score level fusion because of use of fixed weights for score level fusion.
3. Even though adaptive weights are used for score level fusion in [33] and in the proposed system, the proposed system has shown better performance than [33]. This is due to the fact that the proposed system uses better weight assignment as compared to the weights used in [33]. In [33], uniformity in large curvature areas, i.e., singular points close-by areas are ignored.

Thus, it can be inferred that the proposed slap fingerprint matching exhibit superior performance than the existing slap fingerprint matching systems. It is to be noted that adaptive weight based score level fusion takes larger time than a fixed weight based score level fusion because of the estimation of adaptive weights. The proposed fusion takes an addition time of 0.08 s that is negligible as compared to the total verification time, which is 1.56 s.

5.5. Time computation

Average time taken by the proposed slap fingerprint segmentation algorithm is 0.74 s, out of which, component detection, fingerprint detection and fingerprint labeling take 0.67 s, 0.02 s and 0.05 s respectively. Also, the average time taken by the proposed slap fingerprint matching is 0.82 s, which includes single fingerprint matching, measuring uniformity and fusion of matching scores. Thus, a user identity can be verified in 1.56 s. In addition, average time taken by the template update is 9.53 s, out of which, fingerprint registration is the most time consuming and inevitable step. Since template update is

Table 3
Results for various fusion algorithms.

Systems	Fusion type	Accuracy
[31]	Serial fusion	71.73
[32]	Feature level fusion	78.51
[29]	Score level fusion	86.17
[33]	Score based fusion	87.24
Proposed	Score based fusion	**91.89**

To maintain uniformity, slap fingerprints are extracted by the proposed slap fingerprint segmentation.

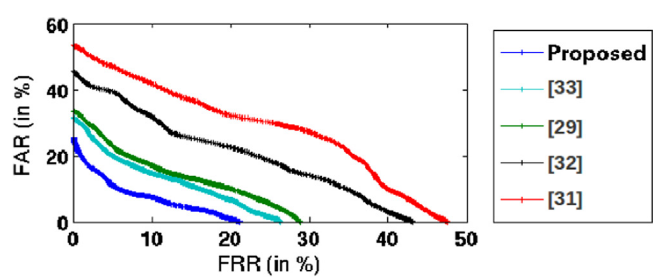

Fig. 10. ROC curve for slap fingerprint matching.

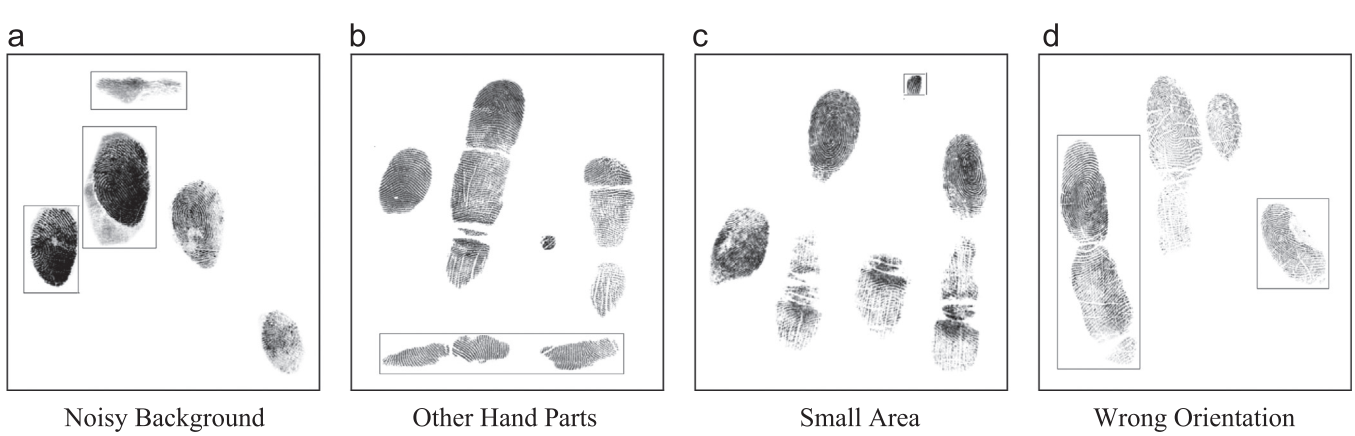

Fig. 9. Wrongly detected fingertip components.

carried out in off-line mode, it does not impact the verification time. Thus, it can be inferred that the proposed system is well suited for verification.

6. Conclusions

A slap fingerprint based verification system has been proposed in this paper. Such systems are highly secure, but they cannot handle the problem of temporal deformation and different pose variation. This paper has presented a template update algorithm for slap-image based verification system. It has improved the accuracy of the system because it reduces intra-class variation, increases inter-class variation and minimizes the temporal deformation. Also, the slap fingerprint segmentation system presented in this paper can extract the single fingerprints from a slap-image. Eventually, matching scores of single fingerprints have been fused using adaptive score level fusion. The performance of the proposed system has been evaluated on a challenging database containing 1800 slap-images acquired from 150 subjects. Experimental results have shown that significant improvement in verification accuracy can be observed if the template update is used.

Acknowledgment

Authors are thankful to the anonymous reviewers for their valuable suggestions to improve the quality of the paper. This work is partially supported by the Department of Information Technology (DIT), Government of India.

References

[1] P.J. Flynn, A.K. Jain, A.A. Ross, Handbook of Biometrics, Springer, 2008.
[2] D. Maltoni, D. Maio, A.K. Jain, S. Prabhakar, Handbook of Fingerprint Recognition, Springer, 2009.
[3] S. Prabhakar, A.K. Jain, Decision-level fusion in fingerprint verification, Pattern Recognit. 35 (4) (2002) 861–874.
[4] C. Wilson, R. Hicklin, H. Korves, B. Ulery, M. Zoepfl, M. Bone, P. Grother, R. Micheals, S. Otto, C. Watson, et al., Fingerprint vendor technology evaluation 2003: summary of results and analysis report, US Department of Commerce, National Institute of Standards and Technology, 2004.
[5] AADHAAR AUTOMATED, Uid enrolment proof-of-concept report, ⟨http://uidai.gov.in/images/FrontPageUpdates/uid_enrolment_poc_report.pdf⟩.
[6] B. Ulery, A. Hicklin, C. Watson, M. Indovina, K. Kwong, Slap fingerprint segmentation evaluation, SlapSeg04 Analysis Report, 2004.
[7] P. Gupta, P. Gupta, A dynamic slap fingerprint based verification system, in: Intelligent Computing Theory, Springer, 2014, pp. 812–818.
[8] R. Cappelli, M. Ferrara, D. Maltoni, Minutia cylinder-code: a new representation and matching technique for fingerprint recognition, IEEE Trans. Pattern Anal. Mach. Intell. 32 (12) (2010) 2128–2141.
[9] A.A. Ross, K. Nandakumar, A.K. Jain, Handbook of Multibiometrics, vol. 6, Springer, 2006.
[10] P. Gupta, P. Gupta, An efficient slap fingerprint segmentation and hand classification algorithm, Neurocomputing 142 (0) (2014) 464–477.
[11] M. Faundez-Zanuy, Data fusion in biometrics, IEEE Aerosp. Electron. Syst. Mag. 20 (1) (2005) 34–38.
[12] S.E. Baker, K.W. Bowyer, P.J. Flynn, P.J. Phillips, Template aging in iris biometrics, in: Handbook of Iris Recognition, Springer, 2013, pp. 205–218.
[13] P. Gupta, P. Gupta, A vein biometric based authentication system, in: Information Systems Security, Springer, 2014, pp. 425–436.
[14] P. Gupta, P. Gupta, An accurate finger vein based verification system, Digit. Signal Process.
[15] G. Marcialis, A. Rattani, F. Roli, Biometric template update: an experimental investigation on the relationship between update errors and performance degradation in face verification, in: Structural, Syntactic, and Statistical Pattern Recognition, Springer Berlin Heidelberg, 2008.
[16] U. Uludag, A. Ross, A. Jain, Biometric template selection and update: a case study in fingerprints, Pattern Recognit. 37 (7) (2004) 1533–1542.
[17] F. Roli, L. Didaci, G. L. Marcialis, Adaptive biometric systems that can improve with use, in: Advances in Biometrics, Springer, 2008, pp. 447–471.
[18] A. Rattani, G.L. Marcialis, F. Roli, Self adaptive systems: an experimental analysis of the performance over time, in: IEEE Workshop on Computational Intelligence in Biometrics and Identity Management, 2011, pp. 36–43.
[19] L. Didaci, G.L. Marcialis, F. Roli, Analysis of unsupervised template update in biometric recognition systems, Pattern Recognit. Lett. 37 (2014) 151–160.
[20] X. Jiang, W. Ser, Online fingerprint template improvement, IEEE Trans. Pattern Anal. Mach. Intell. 24 (8) (2002) 1121–1126.
[21] C. Ryu, H. Kim, A.K. Jain, Template adaptation based fingerprint verification, in: IEEE International Conference on Pattern Recognition, 2006, pp. 582–585.
[22] Z. Wang, A. Serwadda, K.S. Balagani, V.V. Phoha, Transforming animals in a cyber-behavioral biometric menagerie with frog-boiling attacks, in: IEEE International Conference on Biometrics: Theory, Applications and Systems, 2012, pp. 289–296.
[23] Y. Zhang, G. Xiao, Y. Li, H. Wu, Y. Huang, Slap fingerprint segmentation for live-scan devices and ten-print cards, in: IEEE International Conference on Pattern Recognition, 2010, pp. 23–26.
[24] P. Gupta, P. Gupta, Slap fingerprint segmentation, in: IEEE International Conference on Biometrics: Theory, Applications and Systems, 2012, pp. 189–194.
[25] R. Hödl, S. Ram, H. Bischof, J. Birchbauer, Slap fingerprint segmentation, in: Computer Vision Winter Workshop, 2009.
[26] P. Lo, P. Sankar, Slap print segmentation system and method, US Patent 7,072,496, 2006.
[27] C.I. Watson, M.D. Garris, E. Tabassi, C.L. Wilson, R.M. McCabe, S. Janet, User's guide to NIST fingerprint image software 2 (NFIS2), National Institute of Standards and Technology, 2004 .
[28] Z. Yong-liang, L. Yan-miao, W. Hong-tao, H. Ya-ping, X. Gang, G. Fei, Principal axis and crease detection for slap fingerprint segmentation, in: IEEE International Conference on Image Processing, 2010, pp. 3081–3084.
[29] Y. Li, Y. Zhang, J. Lu, C. Liu, S. Fang, Robust rotation estimation of slap fingerprint image for e-commerce authentication, in: IEEE International Conference on Information Theory and Information Security, 2010, pp. 66–69.
[30] N. Singh, A. Nigam, P. Gupta, P. Gupta, Four slap fingerprint segmentation, in: Intelligent Computing Theories and Applications, 2012, pp. 664–671.
[31] P.Z. Lo, B. Bavarian, Adaptive fingerprint matching method and apparatus, US Patent 7,515,741, 7 April 2009.
[32] A.Z. Bendale, T.E. Boult, id-privacy in large scale biometric systems, in: IEEE International Workshop on Information Forensics and Security, 2010, pp. 1–6.
[33] N. Singh, K. Tiwari, A. Nigam, P. Gupta, Fusion of 4-slap fingerprint images with their qualities for human recognition, in: IEEE World Congress on Information and Communication Technologies, 2012, pp. 925–930.
[34] A.M. Bazen, S.H. Gerez, Fingerprint matching by thin-plate spline modelling of elastic deformations, Pattern Recognit. 36 (8) (2003) 1859–1867.
[35] H. Fronthaler, K. Kollreider, J. Bigun, J. Fierrez, F. Alonso-Fernandez, J. Ortega-Garcia, J. Gonzalez-Rodriguez, Fingerprint image-quality estimation and its application to multialgorithm verification, IEEE Trans. Inf. Forensics Secur. 3 (2) (2008) 331–338.
[36] A. Ross, S.C. Dass, A.K. Jain, Fingerprint warping using ridge curve correspondences, IEEE Trans. Pattern Anal. Mach. Intell. 28 (1) (2006) 19–30.
[37] L. Lam, S.-W. Lee, C.Y. Suen, Thinning methodologies-a comprehensive survey, IEEE Trans. Pattern Anal. Mach. Intell. 14 (9) (1992) 869–885.
[38] A. Rattani, G.L. Marcialis, E. Granger, F. Roli, A dual-staged classification-selection approach for automated update of biometric templates, in: IEEE International Conference on Pattern Recognition, 2012, pp. 2972–2975.
[39] E. Tabassi, C. Wilson, C. Watson, Fingerprint image quality, NISTIR 7151, National Institute of Standards and Technology, August 2004.
[40] M. Turk, A. Pentland, Eigenfaces for recognition, J. Cognit. Neurosci. 3 (1) (1991) 71–86.
[41] F. Alonso-Fernandez, J. Fierrez, J. Ortega-Garcia, J. Gonzalez-Rodriguez, H. Fronthaler, K. Kollreider, J. Bigun, A comparative study of fingerprint image-quality estimation methods, IEEE Trans. Inf. Forensics Secur. 2 (4) (2007) 734–743.
[42] E. Lim, X. Jiang, W. Yau, Fingerprint quality and validity analysis, in: IEEE International Conference on Image Processing, vol. 1, 2002, pp. I–469.
[43] S.J. Xie, S. Yoon, J. Shin, D.S. Park, Effective fingerprint quality estimation for diverse capture sensors, Sensors 10 (9) (2010) 7896–7912.
[44] J. Bigun, T. Bigun, K. Nilsson, Recognition by symmetry derivatives and the generalized structure tensor, IEEE Trans. Pattern Anal. Mach. Intell. 26 (12) (2004) 1590–1605.
[45] L. Hong, Y. Wan, A. Jain, Fingerprint image enhancement: algorithm and performance evaluation, IEEE Trans. Pattern Anal. Mach. Intell. 20 (8) (1998) 777–789.
[46] H. Fronthaler, K. Kollreider, J. Bigun, Local features for enhancement and minutiae extraction in fingerprints, IEEE Trans. Image Process. 17 (3) (2008) 354–363.
[47] R. Thai, Fingerprint image enhancement and minutiae extraction, The University of Western Australia, 2003.
[48] K. Nilsson, J. Bigun, Localization of corresponding points in fingerprints by complex filtering, Pattern Recognit. Lett. 24 (13) (2003) 2135–2144.
[49] J. Kittler, M. Hatef, R.P. Duin, J. Matas, On combining classifiers, IEEE Trans. Pattern Anal. Mach. Intell. 20 (3) (1998) 226–239.

Puneet Gupta is a research scholar in the Department of Computer Science and Engineering, Indian Institute of Technology Kanpur, India. His area of research includes Biometrics, Image Processing and Computer Vision.

Phalguni Gupta is associated with Department of Computer Science and Engineering, Indian Institute of Technology Kanpur, India, where currently he is a Professor. He received his PhD degree in computer science and engineering from Indian Institute of Technology Kharagpur, India, in 1986. Prior to joining IIT Kanpur, he was with the Image Processing and Data Product Group of the Space Applications Centre (ISRO), Ahmedabad, India (1983–1987) and was responsible for correcting image data received from Indian Remote Sensing Satellite. His research interest includes sequential algorithms, parallel algorithms, online algorithms, image processing, biometrics, identity and infrastructure management. Currently, he is responsible for several research projects in the area of image processing, biometrics and parallel processing. He has published over 150 peer-reviewed journals and conference papers. He is also the co-author of six books. He is a member of the Association Computing Machinery (ACM) and recipient of 2007 IBM Faculty Award.