Slap Fingerprint Segmentation using Symmetric Filters based Quality

Puneet Gupta, Phalguni Gupta
Department of Computer Science and Engineering
Indian Institute of Technology Kanpur
Kanpur-208016, India
Email: {puneet,pg}@cse.iitk.ac.in

***Abstract*—In this paper, a slap fingerprint segmentation algorithm is proposed which can accurately segment the fingerprints from a given slap-image and classify them as index, middle, ring or little finger of left/right hand. To improve its performance, quality of components is used to eliminate several non-fingerprint components. It is defined by using symmetric filters which measure the uniformity in ridge-valley structure irrespective of linear or curved pattern. Some of the remaining non-fingerprint components are removed by using geometrical locations. For performance evaluation, a challenging database is used which contains 500 slap-images. Experimental results reveal the proposed algorithm can correctly segment the fingerprints from the slap-images with an accuracy of 99.40%, which is found to be better than other existing algorithms.**

***Keywords*—Biometrics, Slap fingerprint, Symmetric filters, Hand detection**

I. INTRODUCTION

Data security is a major issue in the modern era due to the availability of large digital data and cheap hardware. A personal authentication system which can restrict the access for the impostors, but allow the access to genuine users is of paramount importance in this regard. Systems based on keys or passwords can be easily stolen, lost or forgotten, thus these are not much useful. Use of biometrics in authentication system is increasing day-by-day because it contains all the properties required for an authentication system like: permanence, acceptability, uniqueness and universality [1]. The fingerprint is one of the most applicable biometric trait which is also considered as legal proof of evidence in courts of law or in forensics. It is the oriented texture pattern of ridges and valleys on the surface of a fingertip which maintains a coherent flow. There are several factors which can restrict the utilization of fingerprints in personal authentication. Some of these inevitable factors are: (i) improper interaction between user and sensor like undue pressure on some area of sensor; (ii) presence of dirt or latent print on sensor; (iii) environment conditions which causes wet/dry prints due to humidity or temperature; and (iv) smoothening of ridge-valley structure due to occupation or age. Therefore, use of multiple fingerprints is preferable to achieve better performance [2]. Multiple fingerprints can be acquired by acquiring single fingerprints either separately or simultaneously. Separately acquiring single fingerprints of different fingers is a time consuming process which require large operator intervention, thus it is usually avoided [3]. Therefore, a slap fingerprint device is used to acquire multiple fingerprints in this paper. It can simultaneously acquire all the fingerprints present in a hand. Besides increasing the authentication accuracy, use of slap-image can also reduce the problem of spoofing as it is difficult to forge all fingerprints. Due to these factors, usage of slap-image in personal authentication in proliferating.

In a slap-image based personal authentication system, fingerprints present in a slap-image are first segmented and labeled as index, middle, ring or little fingerprint of left/ right hand. Subsequently, single fingerprint feature extraction, single fingerprint matching and consolidation of matching scores are performed for authentication. Process of extracting and labeling the single fingerprint images from a given slap-image is termed as slap fingerprint segmentation. An accurate slap fingerprint segmentation algorithm is crucial for a slap fingerprint based authentication system. This has motivated us to design a slap fingerprint segmentation algorithm which can satisfy the following constraints: (i) it should be accurate, i.e., all single fingerprints are correctly extracted; (ii) it can correctly classify each of the extracted fingerprint into one of the four fingers viz., index, middle, ring or little finger; and (iii) it can be accomplished in near real time.

Several slap fingerprint segmentation algorithms are present in the literature which require geometrical constraints of hand geometry and domain knowledge of sensors for accurate single fingerprint extraction. To the best of our knowledge, fingerprint quality is not used in any of the existing algorithm. There are various advantages of using the quality parameters during slap fingerprint segmentation. Some of these are: (i) most of the spurious single fingerprint extraction from a slap-image can be avoided because these are eliminated by using quality; (ii) slap fingerprint segmentation for the bad quality slap-images can be avoided which can give spurious segmentation and slap-image matching results and thus, degrade the performance of slap fingerprint based authentication system; and (iii) bad quality slap-images can be re-enrolled during enrollment. Thus, a slap fingerprint segmentation based on quality parameters is proposed in this paper. It is observed that non-fingerprint components present in slap-images contain non-uniform flow. Therefore, a quality measure based on symmetric filters is defined which measures the uniformity in ridge-valley flow irrespective of linear or curved pattern which are present near singularities. The major contributions of this paper are: (i) it defines a quality measure which can effectively measure the uniformity; (ii) it designs an algorithm to remove the non-fingerprint components by using quality; and (iii) it

978-1-4799-7458-0/15/$31.00 ©2015 IEEE

presents a geometrical locations based algorithm to eliminate some non-fingerprint components. Both the algorithms for removing the non-fingerprint components are systematically arranged in the slap-image segmentation framework to achieve better performance than existing algorithms.

The paper is organized as follows. The framework of a slap fingerprint segmentation algorithm along-with various existing segmentation algorithms are discussed in the next section. Section III presents the proposed algorithm which can segment the single fingerprints for a given slap-image. Experimental results are analyzed in Section IV. Conclusions are given in the last section.

II. Preliminaries

A. Framework for Slap Fingerprint Segmentation

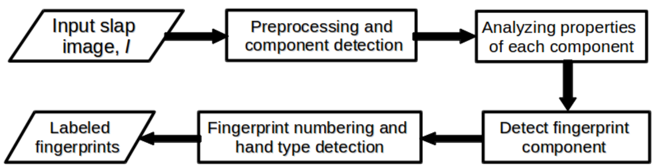
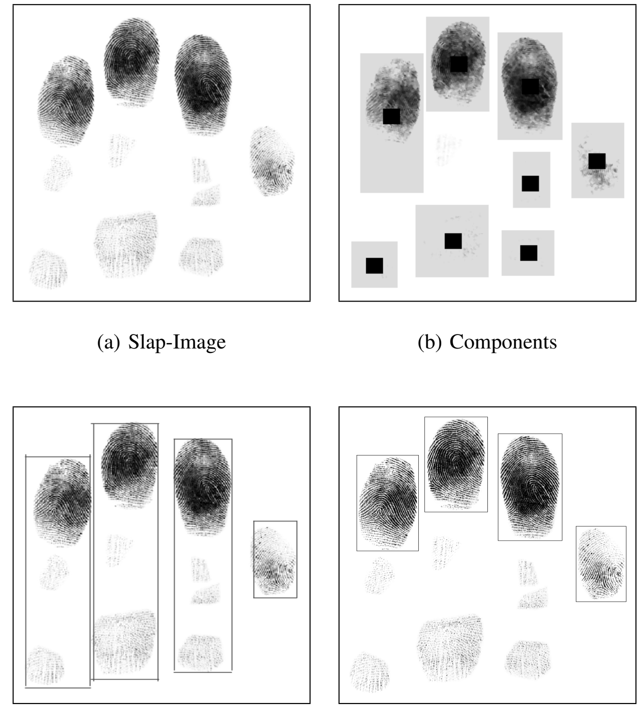


Fig. 1. Framework for Slap-image Segmentation

All the slap-image segmentation algorithms follow some steps which are explained in this section. A slap-image is first preprocessed which consists of: (i) down-sampling; (ii) removing the noise; and (iii) binarization which separates the foreground pixels (pixels belonging to a hand) and background pixels. It is observed that the foreground pixels possess higher intensity variations as compared to the background pixels, thus binarization can be carried out by using anisotropic measures. The foreground pixels are clustered by applying neighborhood connectivity algorithms to form the components. Amongst these components, some components derived from fingertips, which are referred as fingerprint components. Several geometrical properties can be used to describe a component like center, minor-axis, major-axis and orientation. These properties are analyzed to cluster, each of the components into four classes which represent different fingers. An effective clustering should ensure that the components derived from a finger are clustered together. It requires various rules which are defined by using the domain knowledge of acquisition device, hand geometry and hand placements. In a finger, fingerprint always lie at the top, thus topmost component from each class is marked as fingerprint component. Extracted fingerprint components are eventually labeled as index, middle, ring or little fingerprint of left/ right hand by using the domain knowledge of hand geometry. For visualization, consider Figure 1 which shows this framework.

B. Literature Survey

A slap-image segmentation algorithm, NFSEG clusters the detected components into four finger classes by using the assumption that the fingers in a hand can be separated by using straight parallel lines which have equal spacings between them [4]. It fails in the case of open fingers. Component detection is a crucial stage for correct slap-image segmentation. Mean-shift algorithm [5] or various anisotropic measures [6], [7] can be used for component detection. The mean-shift algorithm is usually avoided because it is highly time consuming. Clustering of component can be improved by using better estimates of geometrical properties along-with better constraints on hand geometry [8], [9], [10]. All these algorithms perform poorly if non-elliptical shaped components or dull prints are present in the slap-image. Algorithm in [7] is designed to handle these issues to a large extend. It uses local entropy along-with 8-neighborhood connectivity for component detection. Components are subsequently clustered into four classes by using the geometrical property of the components and the domain knowledge of hand geometry. Fingerprint components (the topmost component from each finger) are numbered from left to right placement. Numbering and the relationship between alternate fingerprint components are used to determine hand type in [7]. As an example of fingerprint component detection and labeling is shown in Figure 2. This algorithm performs highly accurate segmentation because it can accurately detect the components even if partial dull-prints are present (by using entropy) or if the components have non-elliptical shape (by avoiding ellipse fitting). This algorithm is modified in [11] by using better fingerprint labeling constraints.



(a) Slap-Image (b) Components

(c) Fingers (d) Fingertips

Fig. 2. Extracted Fingertips

III. Proposed Algorithm

An algorithm which can extract the fingerprints from a given slap-image is proposed in this section. It contains three steps. In the first step, components present in the slap-image are detected. If more than four components are detected, then some of the components are non-fingerprint components which need to be removed. This is carried out in the second step where each component is classified into fingerprint and non-fingerprint components by using its quality. The quality of each component is evaluated by using symmetric filters. Non-fingerprint components are removed for further processing. Some non-fingerprints are still present which are wrongly classified as fingerprint components because they contain uniform linear flow. These are subsequently removed by using the geometrical locations of the components. Finally, fingerprint components are detected and labeled as index, middle, ring and little finger of left/ right hand.

A. Preprocessing and Component Detection

Algorithm used to extract the components from a given slap-image is explained in this section. Given slap-image cannot be segmented in near real time because it has a large size of 1600×1500 pixels. Thus, it is down-sampled to to one-sixteenth of its original size by using Bicubic interpolation. Down-sampled image may contain salt and pepper noise which are removed by applying median filter. An anisotropic measure, local entropy is applied on the enhanced image to separate the foreground pixels from the background pixels [7]. Detected foreground pixels are clustered to form the components by applying 8-neighborhood connectivity algorithm [12]. It is observed that small sized components can be generated due to noise. Therefore, if the area of a component is less than a threshold T, then it is removed. T is determined by using the contextual knowledge of given slap fingerprint image. Assume that the remaining components are represented by $C_1, C_2, ..., C_n$.

B. Removing Components using Quality

If more than four components are detected, then some components are non-fingerprint components. The aim of this section is to detect such non-fingerprint components by using the quality of component and remove these components. To fulfill this, the quality of a component is evaluated such that it has high value if a component belongs to fingerprint otherwise low value. Hence, following properties of a fingerprint are used to define the quality measure: (i) a fingerprint is an oriented pattern of ridges and valleys; (ii) flow of ridges and valleys is uniform in fingerprint; and (iii) a fingerprint contains singularities like core and delta points which have high curvature. Several ways are present in the literature to evaluate such quality parameters, which can be based on the principle of (i) measuring using the uniformity; (ii) analyzing the pixels; or (iii) using the template based approaches. A template based algorithm is proposed in this paper to define the quality of a component which assigns large weights to the area containing uniform ridge-valley flow irrespective of the linear or curved (near singularities) pattern. Uniformity based measures like orientation consistence level, power spectrum based approach, Gabor filter based approach, etc [13] measure the linear flow and thus, they give low values at the areas near singularities. Since these quality measures cannot be accurately determined near singularities, these are avoided. Similarly, pixel based measures like mean, variance, local contrast, local clarity, FFT measures, etc. are based on measuring the clarity between the ridge-valley structure. These are dependent on fingerprint enhancement and thus, usually avoided.

Consider a component C_i whose quality needs to be estimated. It may consist of background and noisy pixels, thus the fingerprint enhancement algorithm proposed in [14] is used to enhance the ridge-valley structure and to detect the foreground mask, M_i. Multi-scale Gaussian filter based algorithm proposed in [15] is applied to the enhanced image to estimate the orientation field of C_i. It evaluates the orientation field at each pixel and eventually smoothen the orientation field to remove local errors due to bad quality. Let the obtained orientation field corresponding to C_i is represented by θ_i. The orientation field of a component is represented as an orientation tensor, z_i which is the complex field of squared orientation field. That is,

$$z_i = \cos(2\theta_i) + i\sin(2\theta_i) \tag{1}$$

It contains the information of ridge-valley flow, which can be decomposed into a set of symmetric descriptors that are useful for quality estimation. Filter corresponding to n^{th} symmetry is given by

$$h_n = \begin{cases} (x+iy)^n \cdot g & if\ n \geq 0 \\ (x-iy)^{|n|} \cdot g & otherwise \end{cases} \tag{2}$$

where g follows a 2D Gaussian distribution. Orientation tensor is projected onto various symmetric filters for decomposition. Let S_n represents such a projection for symmetry of order n, then it is given by

$$S_n = \frac{\langle z_i, h_n\rangle}{\langle |z_i|, h_0\rangle} \tag{3}$$

Such projections are highly useful because these are translation, scale and rotation invariant when considered in a local neighborhood. According to the requirement, one can apply various symmetric filters on a given tensor to extract the symmetries. In our case, symmetries applicable in the field of fingerprint are required. Uniform linear flow in a fingerprint can be represented by linear symmetry which is modeled by symmetric filter of order 0, i.e. h_0. Likewise, areas near minutiae and singular point (i.e. core and delta points) are identical to parabolic symmetry which is modeled by symmetric filters of order 1 or -1, i.e. h_1 and h_{-1} [16]. Thus, z_i is decomposed into linear and parabolic symmetries (S_0, S_1 and S_{-1}) by using h_0, h_1 and h_{-1}. These symmetric filters and symmetries are calculated by using Equation (2) and Equation (3) respectively, for n = 0, 1 and -1. Further, it has been observed that in a block, only one type of symmetry can be present. If several symmetries are present in a block then it indicates that the block has bad quality due to noise or blur. In such cases, it is better to reduce the value of all the symmetries. Thus, S_n (for n = 0,1,-1) is modified by using an inhibition scheme such that the modified symmetry has low values at the location of low quality areas. Modified symmetry for order n, $\hat{S}_n$ is given by

$$\hat{S}_n = S_n \prod_{k\in N\backslash n} (1 - |S_k|) \tag{4}$$

where N represents all the symmetries while k represents the symmetries except n. Let W_i represent the quality of component C_i at each pixel. It is obtained by

$$W_i(x,y) = \max\left(\hat{S}_0(x,y), \hat{S}_1(x,y), \hat{S}_{-1}(x,y)\right) \tag{5}$$

where (x, y) represents the pixel location while max operator gives the maximum value. The global quality of C_i, Q_i is calculated by averaging the quality (W_i) of foreground pixels (given by M_i). That is,

$$Q_i = \frac{\sum_x \sum_y W_i(x,y) \times M_i(x,y)}{\sum_x \sum_y M_i(x,y)} \tag{6}$$

where (x, y) indicate the pixel locations.

By using this symmetric based quality estimation, quality of each component is evaluated. All the components whose quality parameters are less than a preassigned threshold, are marked as non-fingerprint component and removed for further evaluation. Assume that the r number of components are left after removing the non-fingerprint components and these are represented by $(\bar{C}_1, \bar{C}_2, ..., \bar{C}_r)$. It is possible that r is less than 4 which implies that all fingerprints are not captured during acquisition or some of the acquired fingerprints have bad quality. Such cases can result in false segmentation or spurious matching results due to poor quality in the later stages, thus any further processing of such cases is avoided.

C. Removing Components using Geometrical Locations

If the number of remaining components, r is equal to 4, then pruning based on location is not required. But if r, is greater than 4, then some non-fingerprint components are still present in the remaining components. In this section, some of the remaining non-fingerprint components are removed by using the geometrical locations.

For pruning, location of a component is used which is given by its center. Let $(x_{\bar{C}_i}, y_{\bar{C}_i})$ represent the center for $\bar{C}_i$ component. Then, it is given by:

$$x_{\bar{C}_i} = \frac{\sum_{i=1}^{m} x_i}{m} \qquad y_{\bar{C}_i} = \frac{\sum_{i=1}^{m} y_i}{m} \tag{7}$$

where m is the total number of foreground pixels in $\bar{C}_i$ and (x_i, y_i) represent the pixel coordinate of foreground pixel. The centers of all the components constitute a set whose convex hull, C_h is obtained. By using the hand geometry, it can be observed that boundary of C_h contains the centers of all fingerprint components along-with some non-fingerprint components. But all the components whose centers lie inside the C_h are non-fingerprint components which are detected and removed. The remaining components are stored as $(\hat{C}_1, \hat{C}_2, ..., \hat{C}_t)$ where t is the number of the remaining components. Remember in case of r = 4, pruning based on location is not used and $(\bar{C}_1, \bar{C}_2, \bar{C}_3, \bar{C}_4)$ are stored as $(\hat{C}_1, \hat{C}_2, \hat{C}_3, \hat{C}_4)$.

D. Fingerprint Components Detection and Labeling

If t is equal to 4, then all the components are marked as fingerprint components and these are used for fingerprint labeling. But if its t is greater than 4, then it contains some non-fingerprint components which should be removed. Therefore, remaining components $(\hat{C}_1, \hat{C}_2, ..., \hat{C}_t)$ are first clustered into four classes representing four fingers by using the component clustering proposed in [11]. Topmost component from each finger is marked as fingerprint component.

Numbering is provided to the detected fingerprint components based on their left to right placement in a hand.

Algorithm 1 QualityBasedComponentPruning (T, th, n)

Require: Array T stores the components; th is the threshold; and n is number of components.
Ensure: Array Y stores the components after pruning based on quality.

Y = []
if $n > 4$ **then**
  for $i = 1$ to n **do**
    $C = T(i)$
    (C^E, M) = Enhancement(C)
    /* Component C is enhanced by [14] which gives enhanced image C^E and foreground mask M */
    θ = Orientation(C^E)
    //Orientation is estimated by using [15]
    $z = \cos(2\theta) + i \sin(2\theta)$
    //Use it to form orientation tensor at each pixel
    Symmetric filters h_0, h_1 and h_{-1} are formed by using Equation (2), for n = 0,1 and -1.
    Symmetries S_0, S_1 and S_{-1} are the projections z onto h_0, h_1 and h_{-1} calculated by using Equation (3).
    for each pixel (x, y) in C_i **do**
      $\hat{S}_0(x,y) = S_0(x,y)(1 - |S_1(x,y)|)(1 - |S_{-1}(x,y)|)$
      $\hat{S}_1(x,y) = S_1(x,y)(1 - |S_{-1}(x,y)|)(1 - |S_0(x,y)|)$
      $\hat{S}_{-1}(x,y) = S_{-1}(x,y)(1 - |S_1(x,y)|)(1 - |S_0(x,y)|)$
      //Applying inhibition scheme for each symmetry
      $W(x,y) = \max\left(\hat{S}_0(x,y), \hat{S}_1(x,y), \hat{S}_{-1}(x,y)\right)$
      //Quality estimation at each pixel
    end for
    $Q = \frac{\sum_x \sum_y W(x,y) \times M(x,y)}{\sum_x \sum_y M(x,y)}$
    if $Q > th$ **then**
      Append C in Y.
    end if
  end for
  Store the size of Y in r
  if $r < 4$ **then**
    Y = [] // Avoid in subsequent stages.
  end if
end if
if $n = 4$ **then**
  $Y = T$
end if
return (Y)

Algorithm 2 LocationBasedComponentPruning (T)

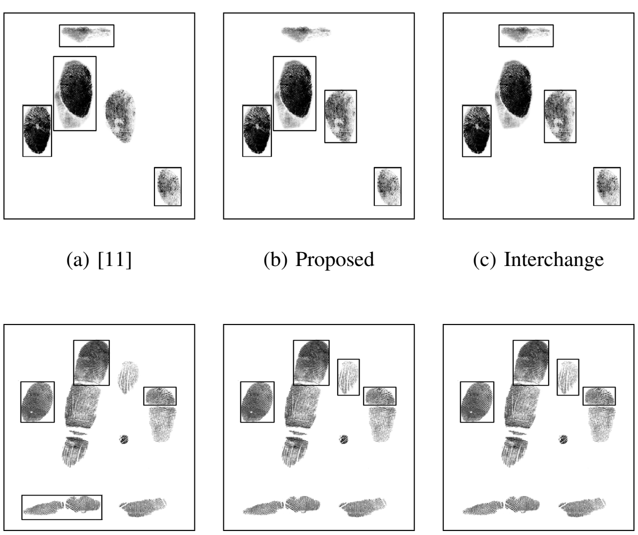
Require: Array T stores the components that require pruning.
Ensure: Array Y stores the components after pruning based on location.

Size of T is stored in r.
if $r > 4$ **then**
  Find center of each component by using Equation (7).
  Find the convex hull, C_h by using these centers.
  Find the components whose centers lie on the boundary of C_h and store these in an array Y.
else
  $Y = T$ // no pruning is required
end if
return (Y)

It requires various hand geometry constraints along-with the Euclidean distance between the geometric center of fingerprint components [7]. Let x represent the difference in length between little finger and ring finger while y represent the difference in length between index finger and middle finger. It is intuitive from the right hand geometry that $x > y$. This intuition along-with fingerprint numbering is used to detect the hand type present in the slap-image. The difference between the lengths of two alternate fingers is calculated by using a measure $Relativelength$ defined in [11]. By using the fingerprint numbering and hand type, each fingerprint component is classified as index, middle, ring or little finger of the left or right hand.

IV. Experimental Results

In the absence of any publicly available database for slap-images, we have created our own database for performance evaluation of the proposed algorithm. This is created by acquiring 540 slap-images from 45 subjects. Subjects belong to the rural areas and from different age groups. Slap-images are acquired under non-controlled environment. In addition, subjects are involved in significant manual work due to which their fingerprints are of average or low quality. Due to these factors, our database consists of challenging slap-images which cover all possible variations and thus, it can be applied in real world scenarios. Size of acquired slap-image is 1600×1500 pixels. Data is collected in two separate sessions such that the average time gap between the two sessions is approximately three months. For each subject, slap-images of both hands are acquired. Total 3 slap-images per subject hand and per session are acquired. Hence, twelve slap-images are acquired from each user. Out of 540 slap-images, there are 40 slap images which are not used for evaluation because they do not contain all four fingerprints. Rest of 500 slap-images are used for evaluation.



(a) [11] (b) Proposed (c) Interchange

(d) [11] (e) Proposed (f) Interchange

Fig. 3. Some Segmentation Results

Table I shows the performance of the proposed algorithm in comparison to the other well known algorithms. All the algorithms are implemented in Matlab and tested on the same machine. The results of these algorithms are manually examined. In Table I, accuracy is used as a performance metrics which is given by:

$$accuracy = \frac{\#\ correctly\ labeled\ fingerprint\ component}{\#\ fingerprint\ components} \tag{8}$$

Since there are 500 slap-images, 2000 fingerprint components are used during the evaluation. Table I shows that the proposed algorithm is able to correctly detect the most number of fingerprint component as compared to other algorithms. The main reasons for this are:

1) Most of the non-fingerprint components which belong to hand are eliminated by using symmetric filters based quality estimation
2) In addition, sometimes erroneous components are formed due to sweat, other hand components and noises. Such components are eliminated by using quality.
3) In some cases, good quality non-fingerprint components can be misinterpreted as fingerprint components. Some of these can be removed by using the constraints based on geometrical locations.

Some of the segmentation results are shown in Figure 3 for better understanding of these reasons. Due to this, maximum number of correctly labeled fingerprint components are detected by the proposed algorithm as compared to other algorithms in Table I.

TABLE I. Performance of the Proposed Algorithm

	# detected fingerprint components	# correctly labeled fingerprint component	Accuracy (in %)
[5]	1856	1823	91.15
[17]	1694	1582	79.10
[7]	1963	1947	97.35
[11]	1963	1952	97.60
Proposed	1988	1988	99.40

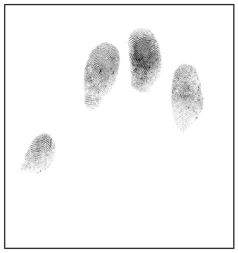
But there are cases of wrong fingerprint component detections and incorrect labeling due to:

1) *Improper hand geometry*: Sometimes hand geometry is changed due to disease in such a way that fingerprint components cannot be correctly extracted.
2) *Bad Quality*: A bad quality fingerprint component can be missed.
3) *Inappropriate hand placement*: Sometimes a hand is placed such that orientation differences between adjacent fingers are large, due to which $RelativeLenght$ does not provide an accurate hand estimation.
4) *Merging of fingerprints*: In some cases, different fingerprints in a slap-image can be merged to form a single component due to sweat. In such cases, accurate fingerprint component detection is not possible.

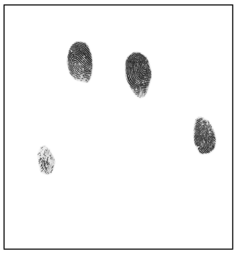
For visualization, consider Figure 4 which shows an example of these cases.

It is to be noted that a proper sequencing of removing a non-fingerprint components is essential for better performance.

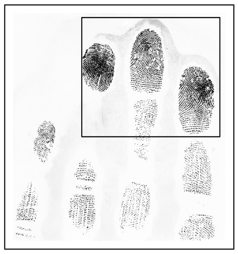
(a) Hand Geometry



(b) Bad Quality



(c) Hand Placement



(d) Merged fingertip

Fig. 4. Reasons for wrong results

To understand this, an algorithm referred as Interchange is used. In Interchange algorithm, non-fingerprint components are first removed by using geometrical locations and then quality is used to remove non-fingerprint components. Only 1969 correctly labeled fingerprint components are detected by using Interchange algorithm, instead of 1988 as in case of the proposed algorithm. Thus, it can be inferred that an appropriate ordering of the algorithms is crucial for successful slap fingerprint segmentation. Reason behind the failure of Interchange algorithm is that, sometimes fingerprint components are detected as non-fingerprint components because they lie inside the convex hull. Such cases arise when spurious components are formed due to sweat or dirt. An example is shown by Figures 3(b) and 3(c).

V. Conclusions

A slap fingerprint segmentation algorithm has been proposed in this paper. It can accurately segment the fingerprints from a given slap-image and can correctly classify these fingerprints into one of following fingerprint classes: index, middle, ring and little fingers of the left or right hand. Fingerprint and non-fingerprint components present in a slap-image has been extracted from the slap-image by using entropy. Some non-fingerprint components have been detected and removed by using quality of components. Such a quality is estimated by measuring the uniformity in ridge-valley flow irrespective of linear or curved pattern. Uniformity for linear and curved pattern are measured by using various symmetric filters which are eventually consolidated to obtain the quality. Subsequently, geometrical locations of the remaining components have been analyzed to remove some non-fingerprint components. The remaining components are analyzed to detect fingerprint components and labeling by using [11].

The performance of the proposed algorithm has been evaluated on 500 slap-images. Experimental results have shown that the proposed algorithm has correctly segment the fingerprints from the slap-images with an accuracy of 99.40%. The proposed algorithm has shown better performance than other existing algorithms because it eliminates several non-fingerprint components (formed due to noise and sweat) which can result in spurious fingerprint component detection and labeling.

References

[1] P. J. Flynn, A. K. Jain, and A. A. Ross, *Handbook of biometrics*. Springer, 2008.

[2] 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 Dept. of Commerce, National Institute of Standards and Technology, 2004.

[3] S. Prabhakar and A. K. Jain, "Decision-level fusion in fingerprint verification," *Pattern Recognition*, vol. 35, no. 4, pp. 861–874, 2002.

[4] C. Watson, M. Garris, E. Tabassi, C. Wilson, R. McCabe, S. Janet, and K. Ko, *User's guide to Non-Export Controlled Distribution of NIST Biometric Image Software*. NIST Report, 2004.

[5] R. Hödl, S. Ram, H. Bischof, and J. Birchbauer, "Slap fingerprint segmentation," in *Computer Vision Winter Workshop*, 2009.

[6] Y. Zhang, G. Xiao, Y. Li, H. Wu, and Y. Huang, "Slap fingerprint segmentation for live-scan devices and ten-print cards," in *20th International Conference on Pattern Recognition (ICPR)*, 2010, pp. 23–26.

[7] P. Gupta and P. Gupta, "Slap fingerprint segmentation," in *5th IEEE International Conference on Biometrics: Theory, Applications and Systems (BTAS)*, 2012, pp. 189–194.

[8] Z. Yong-liang, L. Yan-miao, W. Hong-tao, H. Ya-ping, X. Gang, and G. Fei, "Principal axis and crease detection for slap fingerprint segmentation," in *17th IEEE International Conference on Image Processing (ICIP)*, 2010, pp. 3081–3084.

[9] Y. Li, Y. Zhang, J. Lu, C. Liu, and S. Fang, "Robust rotation estimation of slap fingerprint image for e-commerce authentication," in *2010 IEEE International Conference on Information Theory and Information Security (ICITIS)*, 2010, pp. 66–69.

[10] N. Singh, A. Nigam, P. Gupta, and P. Gupta, "Four slap fingerprint segmentation," in *Intelligent Computing Theories and Applications*, 2012, pp. 664–671.

[11] P. Gupta and P. Gupta, "An efficient slap fingerprint segmentation and hand classification algorithm," *Neurocomputing*, vol. 142, pp. 464–477, 2014.

[12] P. Gupta and P. Gupta, "A dynamic slap fingerprint based verification system," in *International Conference on Intelligent Computing*, 2014, pp. 812–818.

[13] F. Alonso-Fernandez, J. Fierrez, J. Ortega-Garcia, J. Gonzalez-Rodriguez, H. Fronthaler, K. Kollreider, and J. Bigun, "A comparative study of fingerprint image-quality estimation methods," *IEEE Transactions on Information Forensics and Security*, vol. 2, no. 4, pp. 734–743, 2007.

[14] S. Chikkerur, A. Cartwright, and V. Govindaraju, "Fingerprint enhancement using stft analysis," *Pattern Recognition*, vol. 40, no. 1, pp. 198–211, 2007.

[15] C. Jin and H. Kim, "Pixel-level singular point detection from multi-scale gaussian filtered orientation field," *Pattern Recognition*, vol. 43, no. 11, pp. 3879–3890, 2010.

[16] H. Fronthaler, K. Kollreider, J. Bigun, J. Fierrez, F. Alonso-Fernandez, J. Ortega-Garcia, and J. Gonzalez-Rodriguez, "Fingerprint image-quality estimation and its application to multialgorithm verification," *IEEE Transactions on Information Forensics and Security*, vol. 3, no. 2, pp. 331–338, 2008.

[17] N. Singh, K. Tiwari, A. Nigam, and P. Gupta, "Fusion of 4-slap fingerprint images with their qualities for human recognition," in *World Congress on Information and Communication Technologies (WICT)*, 2012, pp. 925–930.