Four Slap Fingerprint Segmentation

Nishant Singh, Aditya Nigam, Puneet Gupta and Phalguni Gupta

Department of Computer Science and Engineering,
Indian Institute of Technology Kanpur
Kanpur 208016, INDIA
{singhn,naditya,puneet and pg}@.iitk.ac.in

Abstract. This paper proposes an efficient segmentation algorithm to extract multiple fingerprints from a four slap fingerprint image. The proposed algorithm divides each slap image into non-overlapping square blocks and identifies foreground and background blocks using average pixel intensity in that block. Blocks are joined to find the foreground block's connected components. Finally clusters containing finger-tips are selected using geometric characteristics of a finger. It has been tested on IITK four slap fingerprint database of 13380 four slap fingerprint images collected from 1115 subjects in 2 sessions. The algorithm has achieved segmentation accuracy of 98.8%.

Keywords: Four Slap Fingerprint, Segmentation, Clustering, Connected Components, Fingerprint

1 Introduction

Fingerprint is one of the well accepted biometrics in the current scenario. Unfortunately it has many limitations such as poor accuracy, high failure to error rate, missing fingers etc. In order to make the fingerprint based biometric system more effective and accurate, one can think to make use of multiple fingers, instead of single finger. Fingerprint images of multiple fingers can be acquired by the usual fingerprint scanner. But use of such a scanner for multiple fingers increases not only the time to acquire images but also the chances of forgery. These features can be minimized if one uses four slap fingerprint scanner through which one can get the date of all four fingers simultaneously. A sample image of four fingers obtained through a slap fingerprint device is shown in Fig 1.

A simple way to design a slap fingerprint based biometric system is to extract features like minutiae points from the area above the knuckle line of each finger and to use all these features for matching. So there is a challenge to segment all the fingers from the acquired image because the process of segmentation should meet the following constraints:

1. Segmentation should be accurate. That means, the process should be able to segment fingers in such a way that no finger image contains the image of other finger partially or completely.
2. It should be near real time so that it can be used in recognition purpose.

3. It should be able to classify correctly each finger into one of the four classes, point, middle, ring and little fingers.

There exists quite a good number of papers which have dealt with this type of problem. In [4], a heuristic based algorithm has been presented to segment fingerprints from the image. It is a multi-pass algorithm. In order to minimize the cost of computation, the original image is downscaled and binarized before segmentation. The algorithm performs well if all fingers are in the same direction. In [3], relative orientation and placement of disjoint components of fingerprint image are used to segment fingers. It has used spitting and joining heuristic approach to segment images. It performs well if the degree of orientation is negligible.

The algorithm discussed in [1] is based on the assumption that all fingers are elliptical is shape. It first determines all possible finger components and then uses ellipse-fitting algorithm twice to prune all non-fingers components. It can also detect the hand of the fingerprint. However, it has a limitation that it works well when fingertips are elliptical in shape. In [6], a better algorithm for noise removal and foreground segmentation of a slap fingerprint has been discussed. This algorithm has been modified in [5] to achieve the improvement with respect to speed and accuracy. An application on slap fingerprint image in the field of E-commerce for authentication has been described in [2].

The paper proposes an efficient algorithm which can be used to segment fingerprints from a given slap fingerprint image. Its performance is found to be better than all well known algorithms and it has minimized some of limitations of existing algorithms. The paper is organized as follows. Next section discusses the proposed algorithm. Experimental results have been analyzed in Section 3. Conclusions are given in the last section.

Fig. 1: Sample Four Slap Fingerprint Images

2 Proposed Four Slap Fingerprint Segmentation Algorithm

This section discusses an efficient algorithm which can be used to segment fingers from the four slap fingerprint images. The algorithm consist of 4 major steps viz.

preprocessing, clustering and finger mapping. These identified fingers can be used by any fingerprint based recognition system to authenticate an individual. At preprocessing step, the normalized downsized image of the four slap fingerprint image is obtained. This image is used in the next phase to obtain the connected components and to determine finger clusters. Finally, clusters are classified into a class such as point, middle, ring and little finger.

2.1 Preprocessing

Let I be a four slap fingerprint image of size $m \times n$. The image is binarized by using any well known algorithm and is downscaled to an image of size $m_1 \times n_1$ where $m_1 = \lfloor \frac{m}{b} \rfloor$ and $m_1 = \lfloor \frac{n}{b} \rfloor$. This can be done in the following way. The binarized image is divided into $\lfloor \frac{m}{b} \rfloor \times \lfloor \frac{n}{b} \rfloor$ non-overlapping blocks, each of size $b \times b$. The $(i,j)^{th}$ block of the image represents the $(i,j)^{th}$ element of the downscaled image. If J is the downscaled image, then $J(i,j)$ is defined by

$$J(i,j) = \begin{cases} 1 & \text{if the number of 1s in } (i,j)^{th} \text{ block of I is more than } \frac{b^2}{2} \\ 0 & \text{otherwise} \end{cases}$$

It can be noted that in order to access all neighborhood pixels of I for obtaining J, the original image of I may be appropriately padded by white pixels. Results of preprocessing steps are shown in Fig 2.

(a) (b) (c) (d)

Fig. 2: (a) Original Four Slap Image I of size 1600×1500, (b) Binarized Image, (c) Foreground Image, and (d) Downscaled Image of size 133×125

2.2 Clustering

Given the binarized downsampled image J, this subsection discusses the method to obtain the connected components. This is done by traversing the image in row major order. It can be noted that a pixel in the image can be a member of a connected component if and only if it is a dark pixel. To assign a cluster to such

Algorithm 1 $clustering(I_p, C)$

Require: I_p: Preprocessed 4-slap fingerprint image of size $(\lfloor \frac{m}{bs} \rfloor, \lfloor \frac{n}{bs} \rfloor)$.
$unique(X)$: Returns a vector of non zero non repetitive elements of any 2D or 1D matrix X in ascending order.
$Merge(A, b, c)$: Replaces all elements in a 2D matrix A having value c with b.
C : Zero cluster matrix of size $\lfloor \frac{m}{bs} \rfloor, \lfloor \frac{n}{bs} \rfloor$.

Ensure: f_{head} : Head pointer of the linked list containing spatial location of each cluster's top left and bottom right pixel in a 4-tuple $\langle x^{TL}, y^{TL}, x^{BR}, y^{BR} \rangle$.

$clusterIndex \leftarrow 1$; $f_{head} \leftarrow NULL$; //Initialization
for $y = 2$ **to** $\lfloor m/bs \rfloor$-1 **do**
  for $x = 2$ **to** $\lfloor n/bs \rfloor$-1 **do**
    if $I_p(y, x) = 1$ //foreground pixel **then**
      $U \leftarrow unique(C(y, x-1), C(y-1, x-1), C(y-1, x), C(y-1, y+1))$
      if $size(U) = 0$ //Only background cluster in 4-neighborhood **then**
        $C(y, x) \leftarrow clusterIndex$ //create new cluster
        $clusterIndex \leftarrow clusterIndex + 1$
      end if
      if $size(U) = 1$ //Only one cluster in 4-neighborhood **then**
        $C(y, x) \leftarrow U(1)$//assign $I_p(y, x)$ to that cluster
      end if
      if $size(U) = 2$ //two clusters in 4-neighborhood **then**
        $C(y, x) \leftarrow U(1)$
        $Merge(C, U(1), U(2))$ // Merge both clusters
      end if
    end if
  end for
end for
$clusters \leftarrow unique(C)$//Vector having all assigned cluster indices
for $p = 1$ **to** $size(clusters)$ **do**
  $\langle x_1, y_1, x_2, y_2 \rangle \leftarrow B_{box}$(clusters(p))//Cluster's rect. bounding box
  Insert($\langle x_1, y_1, x_2, y_2 \rangle$)// Inserts at the end of linked list with head f_{head}
end for
return f_{head}

Algorithm 2 $validate\_clusters(C, \langle x_1, y_1, x_2, y_2 \rangle, t_s, t_n)$

Require: C: Cluster matrix, $\langle x_1, y_1, x_2, y_2 \rangle$: coordinates of bounding box of input cluster, t_s: threshold for no. of pixels in a cluster, t_n: number of pixels NOT belonging to background cluster in the line segment L defined as $(\frac{x_1+x_2}{2}$,0) to $(\frac{x_1+x_2}{2}, y_1)$.
Ensure: Return 0 if cluster is invalid else return 1.

1: $N_{nb} \leftarrow NotBackground(L)$//Returns number of pixels NOT belonging to background cluster in the given line segment
2: $N_s \leftarrow (x_2 - x_1) * (y_2 - y_1)$//Size of the bounding box
3: **if** $(N_s > t_s)\&\&(N_{nb} < t_n)$ **then**
4:     **return** 1
5: **else**
6:     **return** 0
7: **end if**

a pixel, clusters of the four already visited nearest neighbor pixels,if any, are checked. There are three possibilities. These possibilities along with the action to be taken to assign a cluster to such a dark pixel are given below.

1. If all 4 neighbor pixels are white, then this pixel is assigned to a new cluster.
2. If some of these 4 neighbors are belonging to one cluster and remaining pixels are white, then this pixel is also assigned to that cluster.
3. If some of these neighbors are lying in one of the two clusters and remaining are white, then these two clusters are merged to one cluster and this pixel is also assigned to the merged cluster.

Finally all these clusters are represented by the bounding boxes. Fig 3 shows an example of clusters of 4 slap fingerprint images represented by bounding boxes obtained by Algorithm 1. It can be observed that some of the bounding boxes may not represent the clusters to valid fingerprints. Clusters are validated using Algorithm 2. Let (x_1, y_1) and (x_2, y_2) be the coordinates of the top left corner and bottom right corner of a rectangular bounding box. Then the cluster in the bounding box is invalid if either

1. the number of pixels in the clusters is less than a threshold, or
2. the number of dark pixels in the top of bounding box is greater than a threshold. This number can be found by obtaining the number of dark pixels in the line segment $(\frac{x_1+x_2}{2}, 0), (\frac{x_1+x_2}{2}, y_1)$.

Using this strategy, Algorithm 2 has been designed to obtain only those bounding boxes that corresponds to valid clusters as shown in Fig 4.

2.3 Finger Mapping

Clusters obtained through the previous step are projected back to the original image and only top 4 clusters are considered for finger mapping. If the length of a cluster is more than a threshold then it is scaled downed to fit within the range. Whole process has been discussed in Algorithm 3. Segmented fingerprints obtained from a four slap fingerprint image are shown in Fig 4.

Algorithm 3 Four Slap Segmentation($I, bs, t_p, T_s, T_n, h_{max}$)

Require: Four Slap image I of dimension $m \times n$, $bs \times bs$: block size, t_p: threshold for average pixel intensity in a block, t_s: threshold for number of blocks in a cluster, t_n: number of cluster blocks allowed over the top of box, h_{max}: maximum height of any finger.

Ensure: Bounding boxes for 4 fingers as $\langle x_f^{TL}, y_f^{TL}, x_f^{BR}, y_f^{BR} \rangle \ \forall f = 1, 2, 3, 4$

```
I_p ← preprocess(I, bs, t_p)    // generate binary image I_p of size (⌊m/bs⌋, ⌊n/bs⌋)
C ← zero matrix of size (⌊m/bs⌋, ⌊n/bs⌋) //Initialization of cluster matrix
f_head ← clustering(I_p, C)    //f_head is the linked list of rect. cluster bounding boxes found in image I_p
f ← 1 //finger count
ptr ← f_head
//Traversing the cluster linked list
while (ptr ≠ NULL) do
  status ← validate_clusters(C, ptr, t_s, t_n)
  if (status)&&(f ≤ 4) then
    ⟨x_f^TL, y_f^TL, x_f^BR, y_f^BR⟩ ← ⟨(ptr ⇒ x_1)·bs, (ptr ⇒ y_1)·bs, (ptr ⇒ x_2)·bs, (ptr ⇒ y_2)·bs⟩// projecting back to original image
    if ((y_f^BR) − (y_f^TL) > h_max then
      y_f^BR ← (y_f^TL) + h_max//Resizing according to the finger dimensions
    end if
    f ← f + 1
    ptr ← ptr ⇒ next
  end if
end while
```

3 Experimental Results

The proposed algorithm has been tested on a database consisting of four slap fingerprint images collected with the help of a four slap scanner at Biometric Lab of Indian Institute of Technology Kanpur(IITK), India. Data has been acquired from the rural areas under non-controlled environment. Since selected people are involved in various kind of hard work. As a result, they carry very poor quality of fingerprints. Data has been collected in two phases with a gap of two months between phases. Each individual has given data of each hand six times. Three samples of each hand are provided in each phase by 1115 people. Total number of four slap images of each hand is 6690. The proposed segmentation algorithm has segmented accurately 13200 four fingers from the four slap fingerprint images of IITK database which gives the accuracy of 98.8%. Images which are failed to be segmented by the proposed algorithm have been critically analyzed. It has been found that the algorithm has the following limitations.

1. Image quality of four slap fingerprint images should be good. Otherwise, the algorithm may fail to obtain the connectivity between foreground blocks. Fig 5a shows an example of such type of bad image. In this experiment, we have found 9 such cases in the first phase and 109 cases in the second phase.

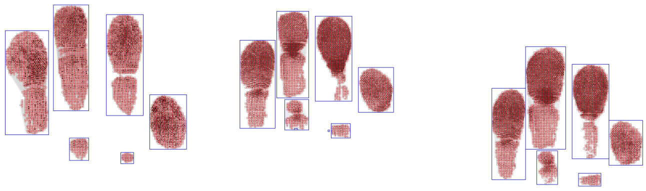

Fig. 3: Boxed Four Slap Fingerprint Images

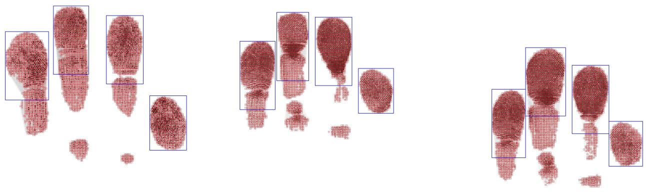

Fig. 4: Segmented Four Slap Fingerprint Image

2. They should not be too much noise in background blocks. An example where the proposed algorithm has failed because of this limitation is shown in Fig 5b. In this experiment we have got 14 images in each phase which contain too much noise in background blocks.
3. Four slap fingerprint images should not be rotated by more than 15°. Fig 5c shows an example where the proposed algorithm has failed to make the proper cluster because of its amount of rotation. In this experiment we have found that the system has failed to cluster images in 7 cases in the first phase and 2 cases of the second phase because they are rotated by more than 15°.
4. It performs poorly when there are some false finger images in a four slap image. False finger images are found when scanner is not properly cleaned before its use. An example of such type of situation is shown in Fig 5d. We have found 5 cases in which there are some false fingerprints.

4 Conclusion

In this paper an efficient four slap fingerprint segmentation algorithm has been presented. Geometric characteristics of fingers have been considered to segment fingerprints. The algorithm is found to perform better than all well known existing four slap fingerprint segmentation algorithms. It has been tested on IITK four slap fingerprint database of 13380 images collected from 1115 subjects in 2 sessions. It has been observed that the proposed algorithm has achieved segmentation accuracy of 98.8%.

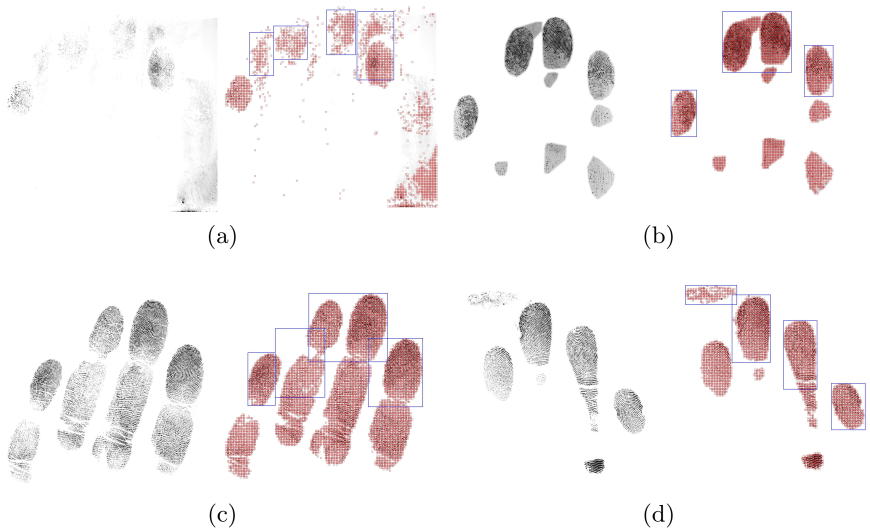

Fig. 5: Example of (a) Bad Quality Images, (b) Background Noise, (c) Rotated Slap Images and (d) False Fingerprints

References

1. Hodl, R., Ram, S., Bischof, H., Birchbauer, J.: Slap fingerprint segmentation. Computer Vision Winter Workshop (2009)
2. Li, Y., Zhang, Y., Lu, J., Liu, C., Fang, S.: Robust rotation estimation of slap fingerprint image for e-commerce authenticate. In IEEE International Conference on Information Theory and Information Security (ICITIS) pp. 66–69, IEEE (2010)
3. Lo, P., Sankar, P.: Slap print segmentation system and method. US Patent 7,072,496 (July 4, 2006)
4. Ulery, B., Hickline, A., Watson, C., Indovina, M., Kwong, K.: Slap fingerprint segmentation evaluation 2004. slapseg04 analysis report (2005)
5. Yong-liang, Z., Yan-miao, L., Hong-tao, W., Ya-ping, H., Gang, X., Fei, G.: Principal axis and crease detection for slap fingerprint segmentation. In 17^{th} IEEE International Conference on Information Processing(ICIP) pp. 3081–3084, IEEE (2010)
6. Zhang, Y.L., Xiao, G., Li, Y.M., Wu, H.T., Huang, Y.P.: Slap fingerprint segmentation for live-scan devices and ten-print cards. International Conference on Pattern Recognition (2010)