

An efficient slap fingerprint segmentation and hand classification algorithm



Puneet Gupta*, Phalguni Gupta

Computer Science and Engineering Department, Indian Institute of Technology Kanpur, Kanpur 208016, India

ARTICLE INFO

Article history:

Received 26 March 2013

Received in revised form

19 March 2014

Accepted 28 March 2014

Communicated by Jinhui Tang

Available online 20 May 2014

Keywords:

Biometrics

Slap fingerprint

Fingerprint segmentation

Knuckle line

ABSTRACT

This paper proposes an efficient algorithm to segment all fingertips from a slap-image and to identify them into their corresponding indices i.e. index, middle, ring or little finger of left/right hand. Geometrical and spatial properties have been used to identify these fingertips. The proposed algorithm can handle various challenges like the presence of dull prints, large rotational angles of the hand, small variation in the orientation of the fingertips and non-elliptical shape of components. It has been tested on a database of 6732 images of 1122 subjects. Experimental results reveal the segmentation of all fingertips from slap-images with an accuracy of 99.02%.

© 2014 Elsevier B.V. All rights reserved.

1. Introduction

Fingerprint consists of the patterns of ridges and valleys on the surface of a fingertip. It is one of the most extensively studied biometric traits and is considered as a proof of evidence in courts of law or in forensics because it exhibits all necessary characteristics such as uniqueness, permanence and difficult to forge. But the use of fingerprint in automatic personal identification is somehow limited due to many factors such as (i) environment (which causes wet/dry prints), (ii) cuts and bruises on fingertip, (iii) sensor's condition (presence of dirt, latent print, etc.), (iv) occupation or age (which may smoothen the ridge-valley structure) and (v) undue pressure on some parts of a finger (which may introduce shear transformation). However, accuracy of any fingerprint based identification system can be increased if images of multiple fingers, henceforth we refer as *multiple fingerprints*, are used [1]. A slap fingerprint device can capture images of multiple fingers in a hand simultaneously. Fig. 1 shows an example of a slap-image. Though images of multiple fingers can be acquired by capturing fingerprint for each finger but it requires more time and operator intervention as compared to capturing a slap-image instantaneously [2]. The use of a slap-image not only increases the authentication accuracy but can also minimize the problem of spoofing. All these factors have ignited the use of slap fingerprint based personal authentication in the modern era [3].

For slap-image based personal authentication, fingerprint of each finger is segmented from the slap-image and features which are used for authentication are extracted from each of these fingerprints. Extraction of all fingerprint images from slap-image is termed as slap fingerprint segmentation. Fingertip components of a slap-image are shown in Fig. 2. Performance of a slap fingerprint based system depends on its segmentation technique. Hence, there is a need to design an efficient slap fingerprint segmentation algorithm which satisfies the following constraints:

1. *Accurate*: All fingerprints in slap-image are correctly located.
2. *Detecting hand*: It can classify correctly whether the given slap-image is of the right or the left hand.
3. *Finger labeling*: Each segmented fingerprint image should correctly be recognized as one of these four fingers viz., index, middle, ring or little finger.
4. *Real time*: This segmentation is done in near real time.

These constraints are necessary for effective usage of slap fingerprint in a personal authentication system. This paper proposes an algorithm which segments slap fingerprint image accurately, detects hand and fingerprint in near real time. It considers full resolution of images which helps to avoid the problem of merging components lying close to each other. It has detected knuckle line accurately even for dry/wet fingerprints, creases or dull prints. The proposed algorithm works efficiently even for a large angle of rotation, for dull/wet prints in small portion of the image and for non-elliptical shaped fingerprint image.

Initially, a given slap-image is partitioned into multiple segments (which can be a pixel or blocks of pixels) such that each

* Corresponding author.

E-mail addresses: puneet@cse.iitk.ac.in (P. Gupta), pg@cse.iitk.ac.in (P. Gupta).

segment belongs to the foreground or the background object. Foreground objects for slap-image are the areas belonging to some part of the hand and have sufficient intensity variation as compared to background objects. These segments are merged using neighborhood connectivity and are referred to as “components” in the slap-image. For example, fingertip components of Fig. 3(b) are clustered into four classes in such a way that components of a finger belong to a class. In human anatomy, terminal phalanges are the terminal limb bones located at the tip of the fingers which are referred to as fingertips. Hence, corresponding to each class (or finger), a “fingertip component” is extracted which lies at the top of the finger. These fingertip components contain fingerprint area. But sometime other parts of the finger (or intermediate phalanges) are also captured in the slap-image and merged with fingertip component at the distal interphalangeal joints which are also referred to as knuckle line. For example, Fig. 3(d) shows all

fingertip components of a slap-image and it can be observed that the leftmost fingertip component contains other part of finger except fingerprint. Merging of fingertip with other finger components can hamper the performance of the slap-image based system because it can change the spatial and geometrical properties like center of mass, mean and orientation. This can lead to the wrong finger/hand classification and/or generation of spurious features (like minutiae). Therefore, each of these extracted fingertip components is analyzed to remove the areas that do not belong to fingerprint. Finally each fingerprint is labeled as index, middle, ring or little finger of the left or the right hand.

The paper is organized as follows. Next section discusses some well known slap fingerprint segmentation techniques. Section 3 presents some methods which are used to design our algorithm. Section 4 presents the proposed algorithm to segment fingertips for the slap-images. Experimental results are analyzed in Section 5. Conclusions are given in the last section.



Fig. 1. Slap fingerprint image.



Fig. 2. Segmented fingerprints components for Fig. 1. (a) Index, (b) middle, (c) ring, and (d) little.

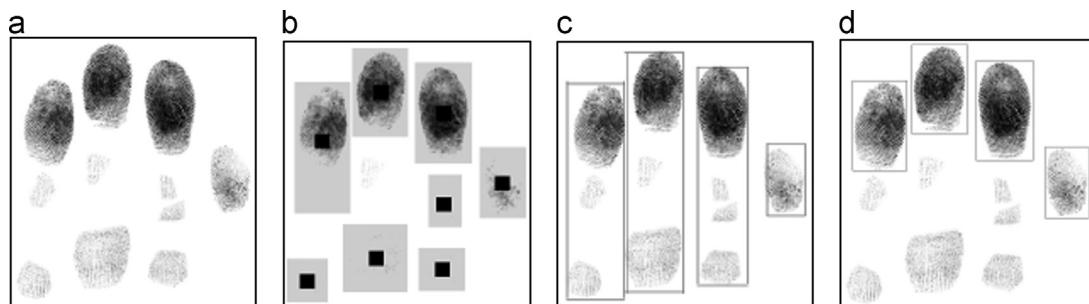


Fig. 3. Extracted fingertips. (a) Slap-image, (b) components, (c) fingers, and (d) fingertips.

2. Literature survey

In [2], a slap fingerprint image segmentation algorithm, NFSEG, has been presented. Each slap-image in NFSEG is binarized and several pairs of parallel lines with equal spacing are drawn on the image at various angles. However, spacings and angles are heuristically determined. Each component belongs to a finger and the pair of parallel lines that can separate these fingers is chosen. The top most component of each class/finger is the fingertip component. This is a multipass algorithm; hence it is computationally expensive. It is designed based on the assumption that all fingers are in one common direction which is the global orientation of hand.

The algorithm presented in [4] uses several hand geometry constraints to obtain the components from the binarized slap-image. In [5], components are extracted using mean-shift algorithm and ellipse-fitting algorithm. Anisotropic measures are used in [6] to extract the components. The hand geometry and the global hand orientation help to put these components into various classes/fingers. It has been improved in [7] by enhancing orientation estimation and detecting knuckle line. Further improvement has been reported in [8] by using some additional constraints. In [9], components are extracted from the binarized slap-image. Since fingerprint lies on the top of hand, four components that lie on the top are marked as fingertip components. But this is not the case with the slap-image having large rotation. This algorithm has been improved in [10] which can handle the problem of rotation. In [10], slap-image has been rotated at various angles (heuristically) and one of these which represents the best possible hand geometry is chosen. This is a multipass algorithm and is also computationally expensive. The algorithm proposed in [11] extracts different components from a slap-image by using local entropy along with 8-neighborhood connectivity. Geometrical

properties are used to cluster these components into four classes. For example, Fig. 3(c) shows the clusters of all components. Fingertip component is extracted by estimating the top most component from each finger. These are labeled from left to right by using hand geometry constraints and it uses local relationship between consecutive labeled fingertip components to determine hand type present in the slap-image. It is found to be more robust than the algorithm proposed in [5] which uses global orientation of hand. Also, it can effectively handle the problems of partial dull-prints and non-elliptical fingertip shaped components. However, it performs poorly when there is a large variation in estimation of local orientation.

Knuckle line detection for a component can improve the performance of the system [6,7]. Assume, there is a fingertip component C with local orientation θ which contains a knuckle line that needs to be determined. It is observed that local ridge direction around the area containing knuckle line is almost perpendicular to the direction of θ . Therefore, lines oriented in the direction perpendicular to θ are analyzed to determine the knuckle line. In [6], orientation and frequency in a block are used to determine the knuckle line. It uses the observations that the block near knuckle line (i) has orientation perpendicular to the direction of θ , and (ii) has low frequency. In [7], measures like coherence, mean and variance are used. It makes the assumption that mean intensity value around knuckle line is low and fingerprint blocks have higher variance due to ridge-valley structures as compared to blocks containing knuckle line. These measures may give spurious results for fingertip components containing dry/wet fingerprints, creases or dull prints.

Each algorithm for segmenting fingerprints from slap-image has some limitations which are mentioned below:

1. In [5,8], shape of fingerprint is assumed to be elliptical which may not be always true. It can be seen in Fig. 2(a).
2. Estimation of global orientation of hand is found to be inaccurate as shown in Fig. 4. Orientation of each component is shown with dotted lines which pass through its center. Global orientations of the hand computed in [5,6] are also shown in the figure with the help of lines L_3 and L_4 respectively. When such a global orientation of hand is used to find the fingertip component, it may generate spurious results.
3. All algorithms are not invariant to large rotation.
4. Different thresholds are used which raise many issues such as
 - (a) Dull prints present in the fingerprint are considered as background due to the thresholds on mean and variance and are left out for fingertip detection. In these cases, binarization can be used [2,5,9,10].
 - (b) Prior knowledge of capturing device and slap-image is helpful to set key parameters for merging strategy [5], ellipse fitting [5] parallel line configuration [2], clustering threshold [6], etc. But it is unknown in advance.
5. All algorithms are computationally expensive because of the use of multiple passes [2], ellipse fitting [5], anisotropic measures [6–8] or local entropy [11]. Hence, images are down-sampled which may lose some critical information present in original slap-image. Different components come closer to each other which may cause merging of separate components. It results in deterioration of the slap-image segmentation accuracy.

3. Mathematical basis

This paper makes use of various well known characteristics to segment slap-image. These are (i) geometrical properties of a component, (ii) local range of an array, (iii) polynomial approximation of 1-D signal, and (iv) radon transform along a direction.

By geometrical properties of a component, we refer to mean and orientation of a component. Local range is used as an anisotropic measure for 1-D and 2-D arrays. Polynomial approximation of 1-D signal is used for smoothing to remove local artifacts. In order to find projections of the component image along a line and to sum these projections, Radon transform is used.

3.1. Geometrical properties of a component

This subsection describes a procedure to estimate some of the geometrical properties of a given component. Let C_1, C_2, \dots, C_n be the components found from a given slap-image. Let (x_i, y_i) be a pixel in C_i and p be the total number of foreground pixels in C_i . Let (x_{C_i}, y_{C_i}) be the center of gravity for component C_i which is given by

$$x_{C_i} = \frac{\sum_{i=1}^p x_i}{p}, \quad y_{C_i} = \frac{\sum_{i=1}^p y_i}{p} \quad (1)$$

Principal component analysis (PCA) [12] can be used to estimate local orientation of C_i . For this, unbiased estimate of variance-covariance matrix, A_{C_i} , is formed using (x_{C_i}, y_{C_i}) . Characteristic polynomial of A_{C_i} is used to obtain eigenvalues λ_a and λ_b . Let e represent the eigenvector corresponding to the maximum eigenvalue. Then estimation of orientation θ_i corresponding to C_i is given by

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

where e_1 and e_2 represent the projection of e along x - and y -axis respectively. Also, square root of both eigenvalues, $\sqrt{\lambda_a}$ and $\sqrt{\lambda_b}$, for A_{C_i} are lengths of axis of the ellipse. Greater among these lengths gives the semi-major axis while other one gives semi-minor axis of the ellipse.

3.2. Local range of an array

For finding local range [13] of an array F at a point ξ , an interval which lies in the neighborhood of point ξ is chosen. For example, in 1-D array, if γ is the neighborhood in one side then the interval is defined as $(F[\xi-\gamma], F[\xi+\gamma])$. Thus, total $(2 \times \gamma + 1)$ elements are used to find local range at a point. The difference between the maximum

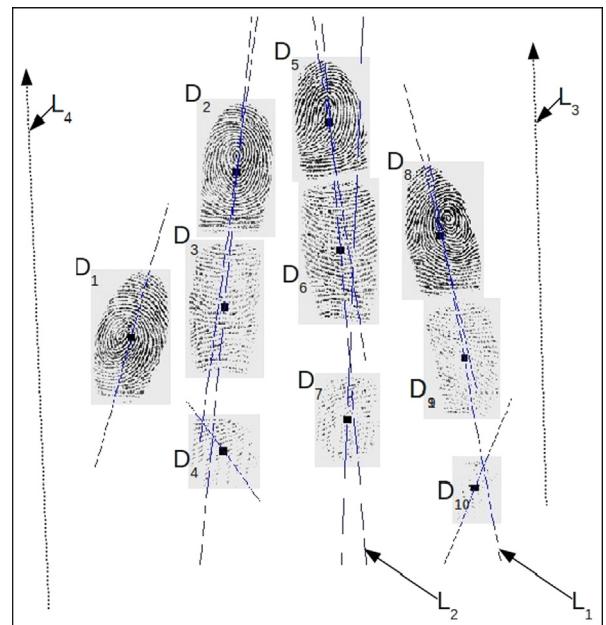


Fig. 4. Finger components with local and global orientations.

and the minimum value of F that lie inside this interval is referred to as local range at a point ξ . Likewise, for a 2-D signal, a block of certain size is used to define an interval. And, local range is given by difference between the maximum and the minimum value of an array lying in that interval. Low local range implies that elements in the array are similar while high local range indicates distinct elements.

3.3. Polynomial approximation of 1-D signal

Let $\psi(x)$ represent the cost function that needs to be smoothed where x refers to the index. Then $\psi(x)$ can be decomposed into polynomial of order n using

$$\psi(x) \approx \sum_{i=0}^n a_i \times x^n \quad (3)$$

where a_i 's are the polynomial coefficients for $\psi(x)$. To find the value of each a_i , least square method can be used. Eq. (3) can be written in matrix form as

$$\psi(x) \approx X^T A$$

where A and X are given by

$$A = [a_0 \ a_1 \ a_2 \dots a_n]^T$$

$$X = [x^0 \ x^1 \ x^2 \dots x^n]^T$$

Let \hat{A} be the estimated A obtained through least square method. Then,

$$\hat{A} = \arg \min_A \sum_x (\psi(x) - X^T A)^2$$

\hat{A} is used to model $\psi(x)$ using

$$\hat{\psi}(x) = \sum_{i=0}^n \hat{a}_i \times x^n$$

where \hat{a}_i 's are given by

$$\hat{A} = [\hat{a}_0 \ \hat{a}_1 \ \hat{a}_2 \dots \ \hat{a}_n]^T$$

3.4. Radon transform along a direction

Radon transform [14] gives the integral projection of the image intensity along a radial line oriented at a specific angle. The transform, F , in R^2 along a straight line t which is oriented at angle θ_{comp} is given by

$$F(s) = \int_{-\infty}^{\infty} \bar{E}_{comp}(x(t), y(t)) dt$$

where s represents search line indices perpendicular to t ; $(x(t), y(t))$ are the parameterized version of the actual pixels in \bar{I}_{comp} and are given by

$$x(t) = (t \sin \theta_{comp} + s \cos \theta_{comp})$$

$$y(t) = (-t \cos \theta_{comp} + s \sin \theta_{comp})$$

4. Proposed algorithm

The algorithm consists of three major steps which are (i) detection of fingertip components; (ii) detection of knuckle line and fingerprints; and (iii) fingerprint labeling.

4.1. Detection of fingertip component

It has been observed that components from the hand (i.e. foreground) contain more local variation and information as compared to the background which is fairly isotropic in nature. Using this local information, each pixel is classified as foreground or background. A

slap-image, I_{actual} , is down-sampled to obtain an image, I_{down} . Each pixel of I_{down} has been multiplied by the normalized standard deviation and normalized local range calculated in a 3×3 neighborhood to get a blob image, B^d . This image contains high and low intensities for foreground and background pixels respectively. Let $I_{binarized}^d$ be the binarized image of B^d . It consists of several connected components which are labeled based on 8-connected neighbors. A few components may be formed due to some noise in the background. These components can be identified using the fact that a component can be eliminated if its area is less than certain threshold T . One such component is shown inside the rectangular box labeled as 1 in Fig. 5(a) and has been eliminated in Fig. 5(f).

Let $C_1^d, C_2^d, \dots, C_n^d$ be the components found from the image after eliminating all components having area smaller than T . Due to downsampling slap-image, components may be merged. To remove such cases, each extracted component is further analyzed at original resolution (i.e., for image I_{actual}). It is observed that merging of separate components due to down-sampling can occur only at the boundary pixels. The image B^d with boundary pixels in I_{down} is projected back to the size of actual image to obtain $B_{original}$ which is used to extract components from I_{actual} . If a component area is found to be less than predefined threshold T then that component is removed. Let C_1, C_2, \dots, C_n be the components extracted from I_{actual} after eliminating all components having area smaller than T .

For aggregation of these detected components into four classes (representing four fingers), geometrical properties of these components are analyzed using the method proposed in [11]. Let (x_{C_i}, y_{C_i}) be the centroid and θ_i be the local orientation of the component C_i . Centers of all components of a finger lie close to a line having slope as that of local orientation of the finger. Consider Fig. 4 where the centers of components (shown with black square) D_8, D_9 and D_{10} lie close to the line L_1 passing through the center of D_9 and have slopes as that of the local orientation of D_9 . But local orientation of finger cannot be effectively determined. Therefore, to find out whether two components, A and B , belong to a common finger or not, the shortest distance from center of one component to the line formed by the center and orientation of another component is used. This gives two distances and if the minimum of these two distances is within some acceptable range then these components are assumed to belong to a common finger. For example, in Fig. 4, D_6 and D_7 should be merged as the shortest distance of the center of D_7 from the line L_2 formed by using center and orientation of D_6 is negligible. Using shortest distance, a measure *Association* is defined. *Association(A, B)* measures how likely two components, A and B , are associated with a finger. It can be determined by Algorithm 1.

Algorithm 1. Association(A, B)

Require: Component A with center (a_1, a_2) and orientation θ_A and, component B with center (b_1, b_2) and orientation θ_B .

Ensure: Return *Association(A, B)*, that is how likely A and B belong to a common finger.

- 1: $m_A = \tan(\theta_A)$
- 2: $c_A = a_2 - m \times a_1$
- 3: $SD(A, B) = \frac{|b_2 - m \times b_1 - c|}{\sqrt{1+m^2}}$
- 4: $m_B = \tan(\theta_B)$
- 5: $c_B = b_2 - m \times b_1$
- 6: $SD(B, A) = \frac{|a_2 - m \times a_1 - c|}{\sqrt{1+m^2}}$
- 7: $Association(A, B) = \min(SD(A, B), SD(B, A))$
- 8: **return**

To illustrate this, consider Fig. 6 where centers and orientations of components C_1, C_2 and C_3 are shown by marks + and dotted lines respectively. Line L_1 passes through the center of C_1

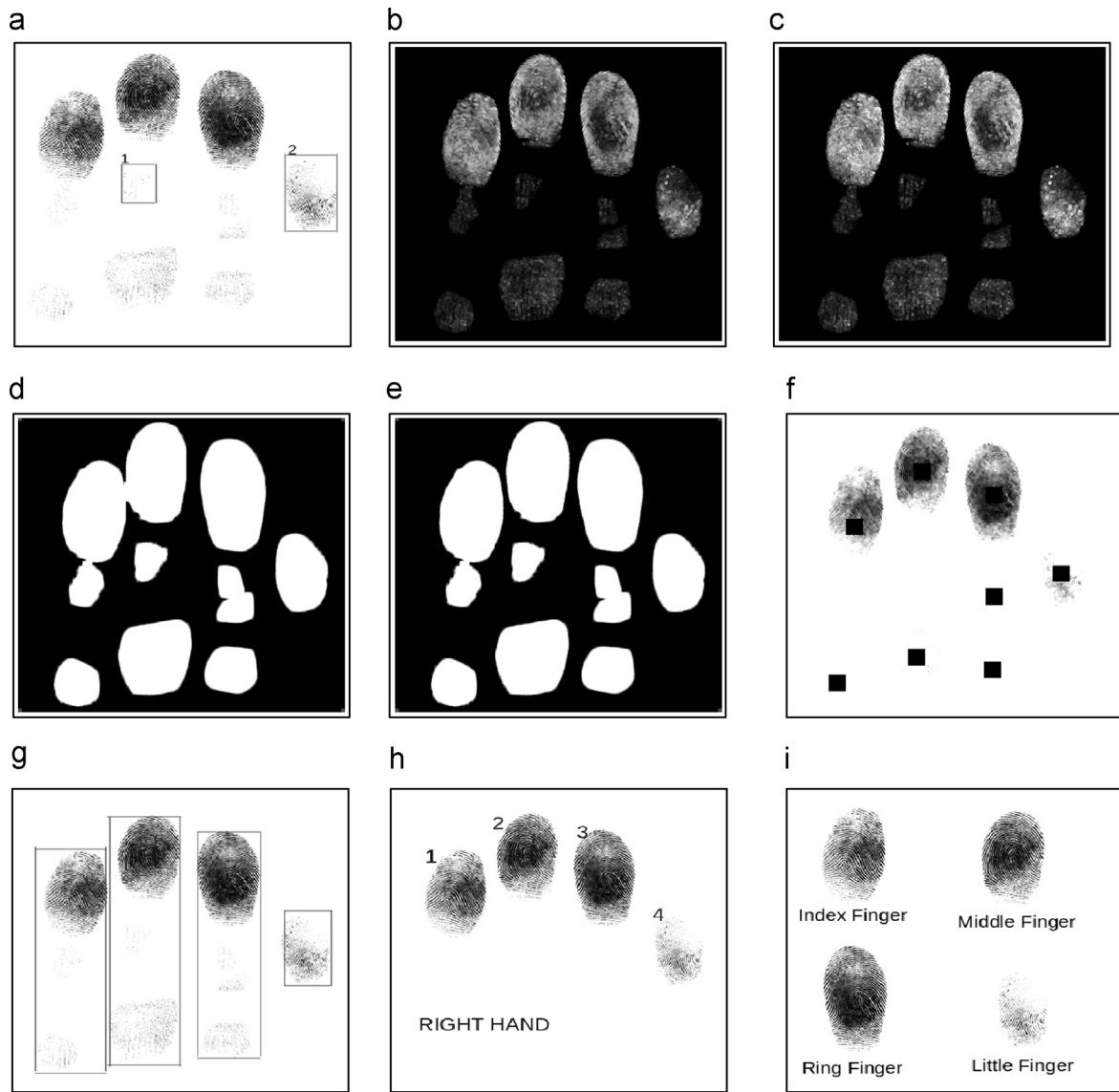


Fig. 5. Finger images at various stages of the proposed algorithm. (a) Slap-image I_{actual} , (b) local standard deviation, (c) local range image, (d) mask image $I_{binized}^d$, (e) mask image $I_{binized}^d$, (f) component marked image, (g) finger marked image, (h) hand detection, and (i) segmentation result.

and is oriented in its direction. Perpendicular distance of line L_1 from the center of C_2 is the shortest distance between C_1 and C_2 . Fig. 6 shows that Association of (C_1, C_2) is less than that of (C_1, C_3) . So, C_1 and C_2 are likely to be the members of the same finger.

Let $Dist$ be a matrix which contains the association measure between each pair of components. The i th row of $Dist$ provides the association measure of the component C_i with every other component. Let $Dist$ matrix contain the minimum value corresponding to components C_i and C_j . Then, these components belong to a common finger and hence, same component number, *FingerNumber*, is assigned to C_i and C_j . This is iteratively applied $n-4$ times where n is the total number of components. For this, Union-Find algorithm [15] can be used. Fig. 5(g) shows the segmented image of the slap-image of Fig. 5(a) where components of a finger are merged and four fingers are formed which are shown in bounding boxes. Instead of using global orientation of hand and local orientation of finger, local orientation of each component is used to merge the components which are part of a finger. The method of aggregating all components to obtain four classes has been given in Algorithm 2.

Algorithm 2. ComponentMerging (C_1, C_2, \dots, C_n).

Require: n components with their centers and orientations.
Ensure: Each component is given a number such that components belong to same finger have same number.

- 1: $Dist = \infty$ /* $Dist$ is a $n \times n$ matrix where each entry is initially set to ∞ */
- 2: **for** $i=1$ to n **do**
- 3: **for** $j=i+1$ to n **do**
- 4: $Dist(i,j) = \text{Association}(A, B)$
- 5: **end for**
- 6: $\text{FingerNumber}(i) = i$
- 7: **end for**
- 8: **for** $k=1$ to $n-4$ **do**
- 9: Obtain (A, B) such that $Dist(A, B)$ is minimum
- 10: Call Union-Find algorithm to merge A and B
- 11: Set $\text{FingerNumber}(A) = \text{FingerNumber}(B)$
- 12: Set $Dist(A, B) = Dist(B, A) = \infty$
- 13: **end for**
- 14: **return**

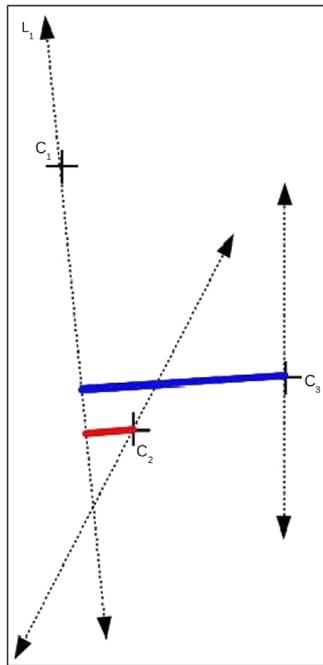


Fig. 6. Shortest distance.

Fingerprint in a finger is placed at the top of the finger. To find such a component, local orientation of that finger is estimated. As all components belonging to a finger are known, local orientation of that finger is approximated by using the orientations of these components. This is repeated for each of the four fingers to extract all four fingertip components. Fig. 5(i) shows the fingerprints of a slap-image of Fig. 5(a).

4.2. Fingerprint extraction using knuckle line

Since extracted fingertip components may contain other parts of the finger which do not belong to fingerprint. For example, the fingertip component shown in Fig. 7(a) consists of areas not belonging to a fingerprint. Location of the knuckle line for each extracted fingertip component is used to define the desired fingertip component.

Each component is enhanced and its edges are summed up across various search lines to extract the line shaped pattern. The local variation among these summed up edge pixels is used to define a cost function that gives the location of knuckle line. But this detection can be affected by creases or bad quality of fingertip components. Thus, this cost function is smoothed based on the domain knowledge and global behavior of the fingertip component.

Initially, each segmented fingertip component is analyzed to figure out whether it contains any unwanted part of fingerprint. If the ratio of length of semi-major axis and semi-minor axis is greater than a threshold, $T_{ellipse}$, then it refers to the fact that the fingertip component image contains areas other than fingertip. If value of $T_{ellipse}$ is large then spurious areas in fingertip component can be considered as fingerprint. If the value of $T_{ellipse}$ is small, then knuckle line detection module is used. As a result, it increases the time required for slap-image segmentation. Experimentally, it is found that an appropriate value of $T_{ellipse}$ is 2.5. Let I_{comp} be such a component. In order to detect knuckle line of I_{comp} , an image enhancement technique [16] is applied on a block size of $b \times b$ to enhance the ridge-valley pattern of I_{comp} . For enhancement, b is chosen such that a block must contain at least two pairs of ridges

and valleys. The enhanced image is binarized to reduce the computational cost. Let this binarized image corresponding to I_{comp} be \bar{I}_{comp} . We have the following observations:

1. If the fingerprint is merged with any other part of the finger in a fingertip component, then its area has low anisotropy due to halo effects. By using the algorithm proposed in [16] for fingerprint enhancement, such halo effects are removed because it uses coherence for measuring the anisotropy and fingerprint segmentation. Hence, merged components are separated and knuckle line is effectively detected.
2. The enhancement technique also removes the areas which do not have uniform pattern (as it should have in a fingerprint image). For example, area inside green colored bounding box in Fig. 8(a) is removed.
3. The enhanced component image has sufficiently good ridge-valley structure; hence estimated orientation of a component is much more accurate.
4. Since this algorithm is based on energy map for fingerprint analysis and enhancement, it helps to remove halo and sweat.
5. A component image may consist of creases. For example, the image in Fig. 8(a) consists of creases. If the enhancement algorithm is applied on such an image then at places of near creases, the enhanced image is either removed/smoothed or continued to act like singularities. Such cases can be visualized in Fig. 8(a) (in red and blue bounding boxes).

The image \bar{I}_{comp} may contain several components and one of these components contains fingerprint. From these components, fingertip component is chosen using the intuition that a fingertip is always located on the top of the fingertip and has sufficiently large area. Fingertip component is extracted from \bar{I}_{comp} and let this be referred to as E_{comp} . Geometrical properties like mean, orientation (i.e. θ_{comp}), bounding box, length of semi-major axis and semi-minor axis of E_{comp} are estimated. If the ratio of semi-major axis and semi-minor axis of E_{comp} is less than a threshold, $T_{ellipse}$, then E_{comp} does not consist of knuckle line; thus mean along with bounding box corresponding to E_{comp} is returned. For example, E_{comp} of the component image in Fig. 8(a) is shown in Fig. 8(b). Otherwise, E_{comp} contains areas other than fingerprint. Knuckle line for such E_{comp} is determined to remove the unwanted parts from the fingertip component.

One can use all search lines, which may be large in number, to detect knuckle line. But it makes the proposed algorithm computationally expensive. In order to reduce the number of search lines, one can consider only those search lines which lie within a specified region. Let this be referred to as search space. Thus, this search space can be defined by the distance between $\alpha \times W_{comp}$ and $(\alpha + \beta) \times W_{comp}$ from the top of E_{comp} where W_{comp} represents the length of semi-minor axis of E_{comp} . Values of α and β are experimentally chosen as 0.4 and 2.6 respectively. Let \hat{E}_{comp} denote the search space of E_{comp} . Edges are detected from \hat{E}_{comp} by applying canny edge detector [17] and are referred to as edge map, \bar{E}_{comp} .

For a component image, ridge-valley pattern around the knuckle joint has a linear flow in the direction that is almost perpendicular to the local orientation of the component. This linear flow is measured by finding the line shaped pattern in \bar{E}_{comp} . For that, all edge pixels in \bar{E}_{comp} are projected onto various search lines such that direction of each search line is perpendicular to the direction of θ_{comp} and all these projections are summed for various search lines. In other words, integral transform of \bar{E}_{comp} over several straight lines (or search lines) oriented perpendicular to the direction of θ_{comp} is computed. For this, the Radon transform is used to obtain a 1-D vector, F , that contains the measurement of linear flow for various search lines. Therefore, search line

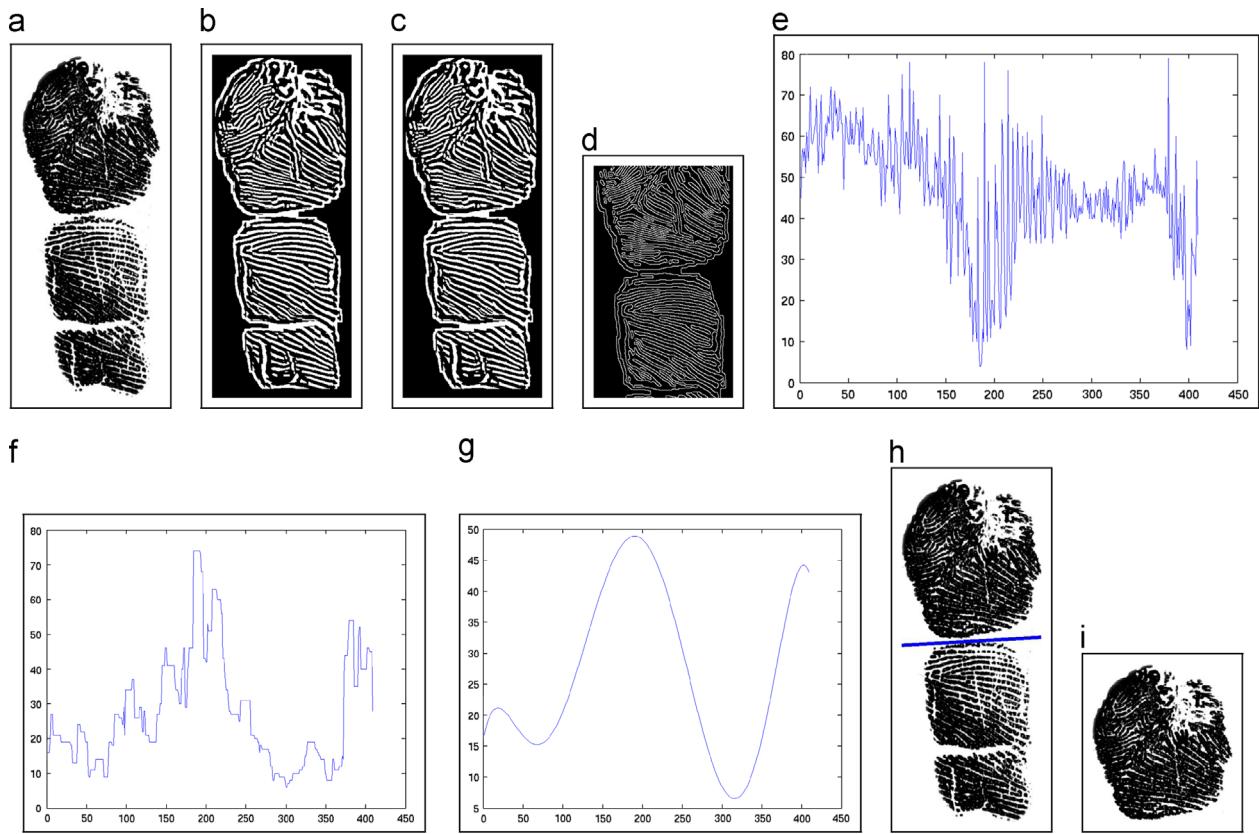


Fig. 7. Various stages of knuckle line extraction and fingerprint extraction algorithms. (a) I_{comp} , (b) \bar{I}_{comp} , (c) E_{comp} , (d) \bar{E}_{comp} , (e) radon transform (F), (f) cost function (ψ), (g) smooth cost function ($\hat{\psi}$), (h) Z , and (i) fingertip.

corresponding to the knuckle line should have a high value and a peak like pattern. But, it is observed that F has also various peaks at various locations where there are no knuckle lines. This is due to the bad quality of captured fingertip component and error in estimating θ_{comp} . Also, it may happen that global maxima do not lie at knuckle line location. This can be seen from Fig. 7(e). Following observations on the nature of F are considered to design a robust knuckle line detection algorithm:

1. At the location of knuckle line, the corresponding search line has a higher F value than its immediate neighbors.
2. Since fingerprint consists of alternate ridge-valley structure flowing uniformly and with high curvature, it has many edges and their sum along the search lines varies uniformly. So the value of F is high and varies uniformly at the place of fingerprints. Similar behavior of F is observed for intermediate phalanges.

From these observations, it can be inferred that to detect knuckle line carefully, local behavior of F should be used. This helps to define a cost function (ψ) which is estimated by local range on F . Low local range value implies that F varies uniformly locally, i.e. in a neighborhood, indicating fingerprint or intermediate phalanges area. High local range value indicates that F has different values with high fluctuations, indicating the presence of knuckle line. But the cost function ψ has also high values at the places consisting of bad quality (like wet or dry fingertips in some part) or creases. Thus, if ψ is used to detect knuckle line then it may also give spurious results. Following observations are useful to understand the nature of ψ across various search lines:

1. If dry/wet fingerprint is captured in some part of the fingertip then after enhancement stage, it contains blurred or no ridge-valley

structure. In case of blur, there is no effect on F . But for no ridge-valley structure, F uniformly varies at places of no ridge-valley structure. But at the boundary of no ridge-valley structure, F has abrupt change; hence, ψ has high value at these boundaries. In the neighborhood of such a high valued ψ , ψ varies uniformly with low values.

2. Similar behavior is observed near creases.
3. Cost function ψ has a spike at the location of knuckle line and also in the neighborhood of knuckle line, there are various spikes and high value of ψ .

Thus, if creases, dry or wet fingerprints are present in some part of the component then ψ contains high value at some places. To minimize the high value of ψ due to spikes caused by creases, dry or wet fingerprints, ψ is smoothed using 1-D polynomial approximation for order \bar{n} and it is referred to as $\hat{\psi}$. This helps in analyzing the global behavior of the cost function, ψ . Global maxima for $\hat{\psi}$ give the index of knuckle line because search line corresponding to the knuckle line has maximum value of $\hat{\psi}$. Thus Z which represents the index of search line corresponding to the knuckle line is given by

$$Z = \arg \max_i (\hat{\psi}[i])$$

where i gives an index of a search line. Area below this knuckle line Z is removed. This image is referred to as modified fingertip component image which only consists of fingerprints. This is stored in E_{comp} along with its geometrical properties.

It can be noted that for polynomial approximation, suitable selection of order \bar{n} plays an important role. When the order \bar{n} of polynomial basis is small then modeling is not quite effective. Again, if the value of \bar{n} is large, then abrupt changes are also modeled, which is called overfitting [18]. This is explained by Runge's phenomenon [19]



Fig. 8. Preprocessing of bad quality and crease containing component image. (a) Component image, and (b) extracted fingerprint. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

which says that increasing the order of polynomial fitting does not necessarily give better fitting. To choose a suitable value for \bar{n} , a test is conducted on 100 fingertip components consisting of knuckle line. In this test, knuckle lines are detected in each image by using 1-D polynomial approximation for various values of order \bar{n} . If the location of a knuckle line can deviate from the location of the actual knuckle line by twice of the ridge width (i.e., $2 \times b$) then it is marked as spurious detection; otherwise it is considered as genuine detection. This is applied on all 100 images and spurious detection corresponding to each order is calculated. Its result is shown in Fig. 9. Best value of \bar{n} is chosen such that spurious detections are minimized. Experimentally, \bar{n} is set to 6.

4.3. Hand/finger classification

All fingertip components consist of fingerprints and are to be indexed into one of the four fingers such as index, middle, ring or little finger of left or right hand. This is carried out in three stages. Initially, these are numbered indicating left to right placement in a hand and then hand is detected using *Relative length*. Lastly, using numbering of these components and hand type, each fingerprint is labeled.

Euclidean distance between centers of the fingerprint along with the constraints of hand geometry proposed in [11] is used for fingerprint numbering. It is observed that if the difference between lengths of the middle finger and index finger is less than that of the ring finger and little finger, then it can be inferred that slap-image consists of right hand. This is illustrated in Fig. 10 which contains right hand image. Let x and y be the difference

between the length of the middle finger and the index finger and that of the ring finger and the little finger, respectively.

Then, the slap-image is of right hand if

$$x < y \quad (4)$$

To determine x and y , a new distance measure, *Relative length*, is proposed which makes use of the relative orientation between adjacent fingers to compute the length of the original fingers.

Consider that C^A and C^B are the adjacent fingertip components and C^A is the left of C^B . Relative orientation $O^{relative}$ between these two components is given by

$$O^{relative} = \frac{W_{C^A} \times O^{C^A} + W_{C^B} \times O^{C^B}}{W_{C^A} + W_{C^B}}$$

where W_{C^A} and W_{C^B} are the weights and O^{C^A} and O^{C^B} are orientation of C^A and C^B respectively. A line $L 1$ which is perpendicular to the direction of relative orientation has a slope m where

$$m = \frac{-1}{\tan^{-1}(O^{relative})}$$

The shortest distance D^A between the center of C^A to the line $L 1$ is given by

$$D^A = \frac{|y^A - m \times x^A - c|}{\sqrt{1+m^2}}$$

where constant c is assumed to be zero. If the shortest distance between the center of C^B to the line $L 1$ is D^B , then *Relative length* is given by

$$\text{Relative length}^{A,B} = |D^A - D^B|$$

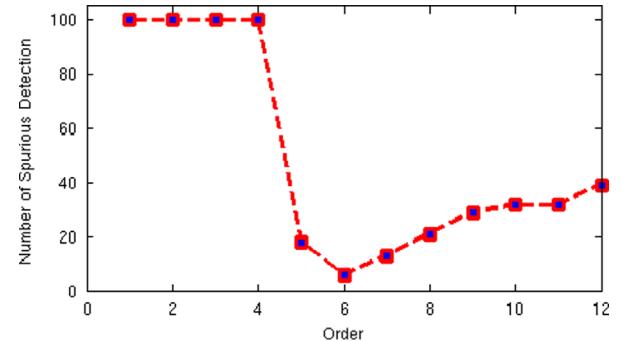


Fig. 9. Effect of order \bar{n} .

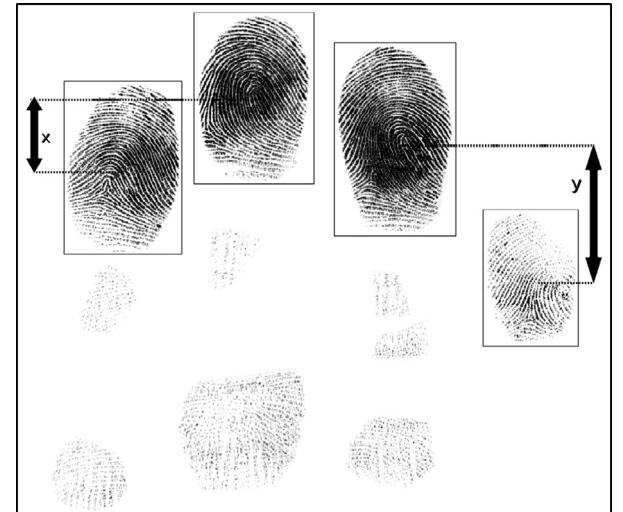


Fig. 10. Hand logic.

where $|.|$ represents the absolute value. To reduce the effect of spurious local orientation of a finger, a weight W_i is assigned which is based on the area of the fingertip component C^i . The rationale behind this is that if a fingertip component is large then estimation of orientation is more accurate. Steps used to compute a distance measure, *Relative length*, are given in [Algorithm 3](#). As an illustration, consider components A, B and the line L_1 perpendicular to direction of their relative orientation as shown in [Fig. 11](#). The shortest distances of each component from L_1 , D^A and D^B are used to get *Relative length* between A and B.

Algorithm 3. *Relative length*(C^A, C^B).

Require: Components C^A and C^B where C^A has centers (x^A, y^A) with orientation O^{C^A} and weight W^{C^A} and C^B has centers (x^B, y^B) with orientation O^{C^B} and weight W^{C^B}

Ensure: Relative length between C^A and C^B .

```

1:  $O^{relative} = \frac{W_{C^A} \times O^{C^A} + W_{C^B} \times O^{C^B}}{W_{C^A} + W_{C^B}}$ 
2:  $m = \frac{-1}{\tan^{-1}(O^{relative})}$ 
3:  $D^A = \frac{|y^A - m \times x^A|}{\sqrt{1+m^2}}$ 
4:  $D^B = \frac{|y^B - m \times x^B|}{\sqrt{1+m^2}}$ 
5:  $length^{A,B} = |D^A - D^B|$ 
6: return

```

In order to find the hand in the given slap-image, Eq. (4) is used where x and y can be calculated using [Algorithm 3](#). Steps involved in hand detection are given in [Algorithm 4](#). Labels of fingertip components from left to right and the hand present in given slap-image help to classify each fingerprint component into its classes as index, middle, ring, or little finger. An example is shown in [Fig. 5](#).

Algorithm 4. *HandTypeDetection*(C^1, C^2, C^3, C^4).

Require: four fingertip components, C^1, C^2, C^3 and C^4 in the image I

Ensure: *LeftHand* or *RightHand* in the I .

```

1:  $x = \text{Relative length}(C^1, C^2)$ 
2:  $y = \text{Relative length}(C^3, C^4)$ 
3: if  $x \leq y$  then
4:    $\text{Hand} = \text{RightHand}$ 
5: else
6:    $\text{Hand} = \text{LeftHand}$ 
7: end if
8: return

```

5. Experimental results

Since there does not exist any publicly available database, so a database has been created at Indian Institute of Technology

Table 1
Comparison with other papers.

Parameters	Algorithm used							Proposed
	[2]	[5]	[6]	[7]	[9]	[10]	[11]	
Shape assumption	X	✓	✓	✓	X	X	X	X
Downsampling	✓	✓	✓	✓	✓	✓	✓	X
Dry fingers	X	X	✓	✓	X	X	✓	✓
Rotational invariance ^a	NA	X	X	X	X	X	✓	✓
Prior information ^b	✓	✓	✓	✓	✓	✓	X	X
Hand detection	X	✓	X	X	X	✓	✓	✓
Knuckle line detection	X	X	✓	✓	X	X	X	✓

^a It excludes hand geometry constraints.

^b It refers to the rotational invariance from -60° to $+60^\circ$.

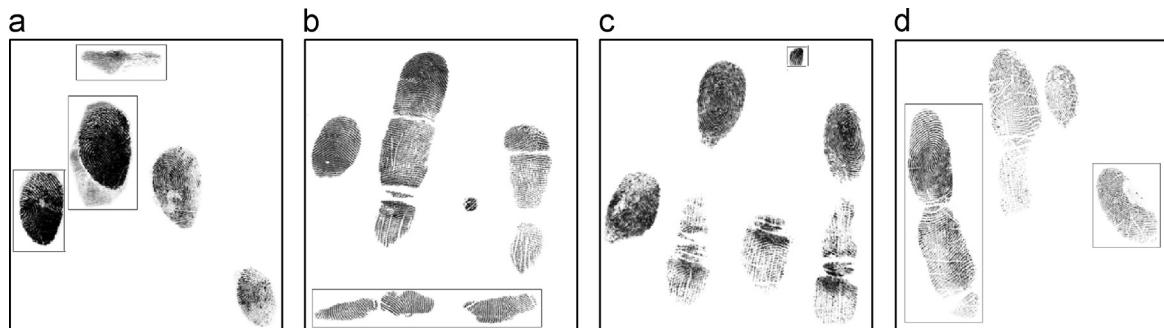


Fig. 12. Wrongly detected fingertip components. (a) Noisy background, (b) other hand parts, (c) small area, and (d) wrong orientation.

Kanpur, India to test the proposed algorithm. It consists of 3366 of slap-images of each hand acquired from 1122 subjects. These subjects are from the rural area, involve significant manual work and are of different age groups. Each grayscale image is of 1600×1500 pixels or nearly 2.3 megabytes in size. These 6732 slap-images are used to find the fingertip components.

Comparison of the proposed algorithm with other well known algorithms is given in [Table 1](#). It can be found that

1. Only the proposed algorithm works on full resolution image and thus can effectively achieve slap fingerprint segmentation even if fingerprints lie close to each other.
2. A dry finger component cannot be accurately determined by an algorithm based on fixed threshold on intensity measures.
3. Even though high rotational invariance can be achieved in [2], it needs multiple passes engulfing rotation information for component extraction. It increases the time complexity.
4. The proposed algorithm is the only algorithm which does not consider three factors responsible for poor segmentation, such as shape, prior information and down-sampling. As a result, it performs better.

It can be noted that algorithm proposed in [8] is not used to compare because it is an extension of the algorithm presented in [6].

To evaluate the performance of the proposed algorithm, it has been tested at each phase, i.e., component detection, knuckle line detection and hand detection. All these experiments are conducted on a personal computer having Intel Pentium 8 processor 2.8 GHz.

5.1. Performance of fingertip component detection

All 6732 slap-images are analyzed for extracting fingertip components. If the total number of components in a slap-image is less than four, then that slap-image is not considered for further

evaluation and is referred to as *failure to enroll*. In the database, there are 175 such images. Experimental results are found to be correct. However, there are improper results in 296 cases which are due to one of the following reasons.

1. *Noisy background*: This occurs generally due to dust, sweat or latent fingerprints marks on the slap-image scanner. This can merge two components from different fingers or a new component is introduced from these noisy areas. An example of this is shown in [Fig. 12\(a\)](#).
2. *Other hand components*: Sometime during enrollment, thumb or other part of the palm may touch the sensor. It can generate a new erroneous component in the slap-image. An example which depicts such cases is shown in [Fig. 12\(b\)](#). This may result in merging of inappropriate components or behave as a valid fingertip component.
3. *Smaller size of fingertip component*: Due to inappropriate hand placement, pressure applied at the time of enrollment or worn out ridge-valley structure, small part of fingertip is captured, as shown in [Fig. 12\(c\)](#). Hence, such components may be left out due to the small in size.
4. *Improper orientation*: Improper placement of hand and bad quality of fingerprint due to worn out ridge-valley pattern or diseases like osteoarthritis, ulnar claw, swan neck deformity, clinodactyly, and dupuytrens contracture often provide improper hand geometry which can form an inappropriate component. An example of such inappropriate component orientation is shown in [Fig. 12\(d\)](#).
5. *Haloring*: It merges different components of separate fingers. Thus, these finger components are not the true representation of the finger.

To test the proposed fingertip component detection algorithm, 1000 slap images not having the above-mentioned problems are chosen from the database. Results on these slap images are evaluated to test the effectiveness of the proposed component detection algorithm and are shown in [Table 2](#).

Table 2
Results for component detection.

Parameters	[2]	[9,10]	[11]	Proposed
Total cases	1000	1000	1000	1000
Downsampling ratio	1:(8 × 8)	1:(12 × 12)	1:(4 × 4)	1:1
Merging separate fingers	18	52	7	0
Wrong component detection	137	158	0	0
Time taken (s)	0.0858	0.1123	0.7284	0.1427

Table 3
Results for knuckle line detection.

Parameters	[6]	[7]	Proposed
Total cases	453	453	453
Spurious detection	109	87	30
Total time taken (s)	0.96	0.65	0.12

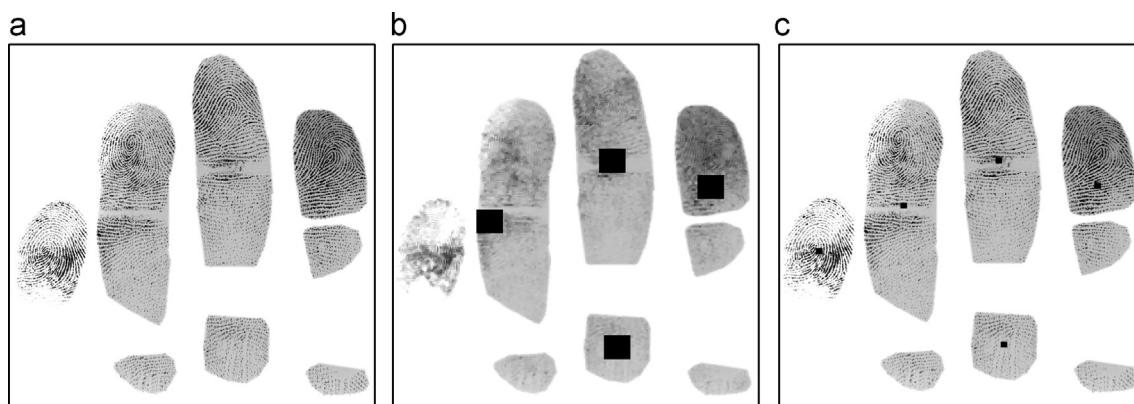


Fig. 13. Detected components by [11] and proposed algorithm. (a) Original image, (b) components [11], and (c) components (proposed).

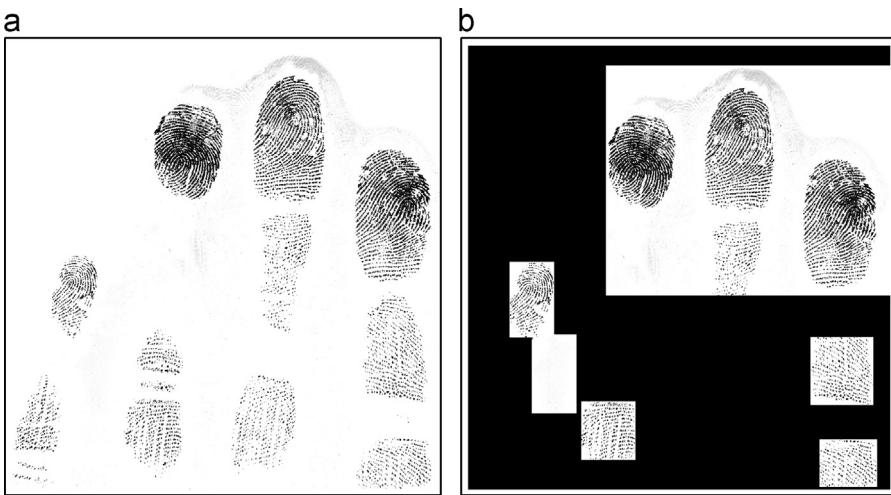


Fig. 14. Merged fingerprint due to Halo effect. (a) Haloing slap-image, and (b) merged fingertip.



Fig. 15. Shifted knuckle line due to bad quality. (a) Component image, and (b) knuckle line.

It can be inferred that due to the increase in downsampling ratio, more number of some nearby components get merged. Such type of components have the following adverse effects:

1. Total number of components is less than four and hence these components are not considered. These cases are referred to as failure-to-enroll.
2. When fingertip components are merged, erroneous fingertip components are present in the end. Also, when components other than fingerprint components are merged, they may result in merging of fingertip components.

This can be visualized in Fig. 13. Such cases are easily handled by the proposed algorithm. Another important inference derived from Table 2 is that the proposed algorithm has not detected any

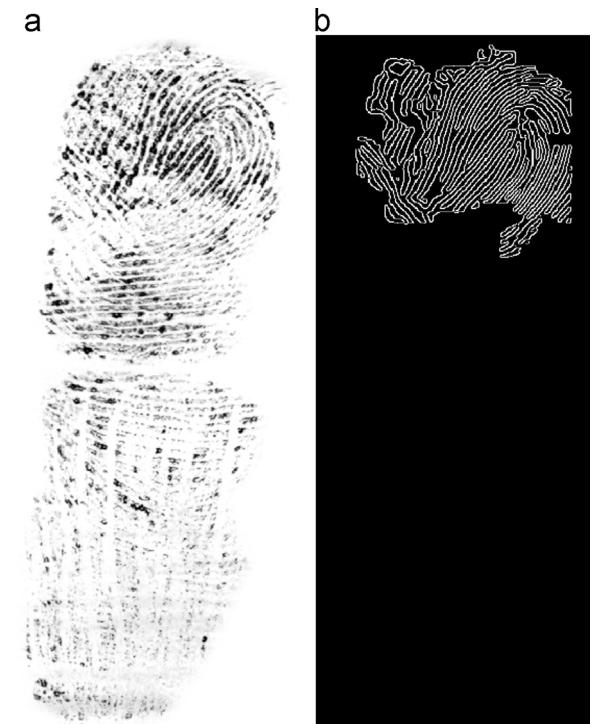


Fig. 16. Partial fingerprint due to bad quality. (a) Component image, and (b) Edgemap.

wrong component. Dry finger in a component is the primary reason of wrong component detection. It results in inaccurate component detection as sometime areas belong to dry areas are marked as background during segmentation. This is often the case when a fixed threshold is applied for binarizing the slap-image which has dry finger (low pixel intensities) in small area while some other areas have high intensities. Fingertip components are detected among these extracted components.

5.2. Results for knuckle line detection

Total 1000 such fingertip components are used to evaluate knuckle line detection algorithm. Among these, there are 547 fingertip components which do not require knuckle line detection. And the remaining 453 fingertip component images are further

evaluated for knuckle line detection using cost function. Out of these components, there are only 30 components where wrong knuckle line is detected.

When all fingerprint segmentations are manually verified, there are some cases of wrong detection of knuckle line. Some of the reasons for wrong detections are

- Merging of fingerprints:** Sometime there may be separate fingerprints in the slap-image which are merged and belong to a common component. This merging happens due to sweat, halo or capture of some part of hand except fingers. An example is shown in Fig. 14(a).
- When the captured fingertip contains huge area of dry or wet prints then there is large shift of knuckle line location. An example is shown in Fig. 15.
- In some cases, the captured fingerprint has poor quality and lack of coherence. When enhancement algorithm is applied on such cases, it may result in removing the fingertip area. For example, in Fig. 16, most of the fingerprint area is removed due to bad quality. This gives only valuable area of fingertip which should be used for feature extraction and matching. But it has limitation like wrong prediction of mean. Thus, it leads to determine wrong hand geometry which can result in wrong hand detection.



Fig. 17. Partial fingerprint.

Table 4
Results for hand/finger labeling.

Parameters	[5]	[10]	[11]	Proposed
Total cases	6328	6328	6328	6328
Spurious detection	532	1240	89	62
Correct detection accuracy (%)	91.59	80.40	98.59	99.02
Total time taken (s)	0.04	2.3	0.06	0.05

- Less area of captured fingerprint:** There are cases when only small part of fingertip is captured which are merged with intermediate phalanges. In such cases, ratio of length of semi-major axis and semi-minor axis is less than $T_{ellipse}$. Hence these cases are not considered for knuckle line detection. An example is shown in Fig. 17.

Knuckle line detection algorithm has been used in [6,7]. These algorithms are compared with the proposed algorithm in Table 3. For knuckle line detection, orientation and frequency in a block are used in [6] while measures like coherence, mean and variance are used in [7]. Algorithm in [7] uses a threshold for knuckle line detection which is set to 1. All the above-mentioned reasons are responsible for spurious knuckle line detection in [6,7]. Besides this, both these algorithms also detect spurious knuckle line when the fingertip components contain dry/wet fingerprints, creases or dull prints in some part. It can be inferred from Table 3 that the proposed algorithm performs better than [6,7].

5.3. Results for hand/finger classification

The modified fingertip components are labeled from left to right placements in a hand and based on this, hand present in the given slap-image is predicted. There exist hand detection algorithms [5,10,11]. The proposed hand detection algorithm is compared with these algorithms in Table 4. For comparison, the fingerprint components extracted by the proposed algorithm after knuckle line detection are used. Only hand detection algorithms can be tested without affecting the slap segmentation accuracy. A spurious detection in Table 4 refers to a case of wrong hand detection. It can be inferred from Table 4 that the proposed hand/fingerprint labeling algorithm gives better results than any other existing hand/fingerprint labeling algorithm. The reason for the large number of spurious hand detection in [5,10] is mainly due to the use of spurious global orientation of hand. And the reasons for the spurious hand detection in the proposed algorithm are given below.

- Partial fingerprint:** Sometime hand is placed in such a way that full fingerprint is not captured. This may be due to unnecessary pressure applied on some part in hand. This results in wrong estimation of orientation and hence hand detection is not proper.
- Change in hand geometry:** In some cases, hand geometry changes due to disease.
- Inappropriate hand placement:** When a user places his/her hand such that there is huge orientation difference in adjacent fingers, *RelativeLength* does not give accurate results.
- Bad quality:** Due to bad quality of the fingerprint components, orientation estimation and center detection are not found correctly.

Some of the incorrect results are shown in Fig. 18.

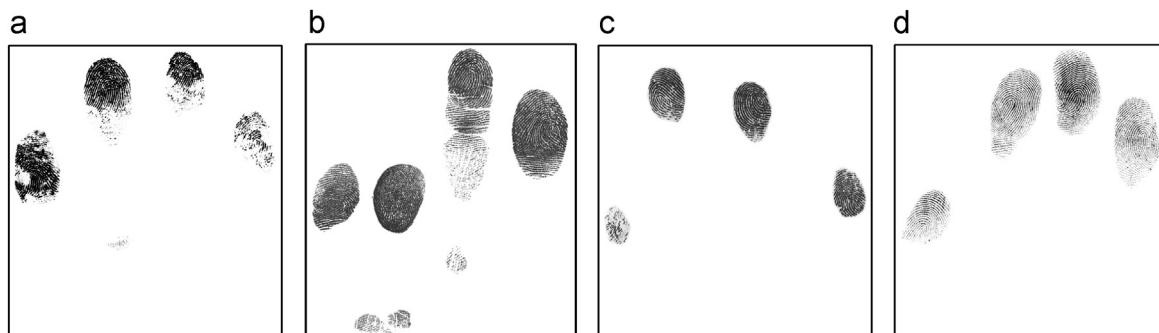


Fig. 18. Wrong hand detection. (a) Partial fingerprint, (b) hand geometry, (c) hand placement, and (d) bad quality.

Table 5
Comparative studies.

Technique used	[11]	Proposed
Fail to enroll cases	404	284
Total considered cases	6330	6450
Total wrong segmentation cases	222	120
Segmentation accuracy (%)	96.52	98.11
Total wrong hand detection cases	130	62
Knuckle line detection	X	✓
Hand detection accuracy (%) (Removing wrong segmentation cases)	97.87 (using 6110 cases)	99.02 (using 6328 cases)
Time taken per slap image (s)	0.9158	0.3094

5.4. Comparison

The proposed algorithm cannot be fully compared to other algorithms because of the (i) non-availability of various parameters used in the well known algorithms and (ii) several algorithms do not use hand detection or knuckle line as shown in Table 1. But by the results presented in Section 5, one can be easily conclude that the proposed algorithm performs better than available algorithms. Results of the proposed algorithm are evaluated on full database and are presented in Table 5. It shows that the proposed algorithm produces better results than [11]. Primary reason for the failure of [11] at some places is that there can be huge variations in the actual and the estimated orientation of fingerprints due to bad quality or partial capture of fingerprint components. The proposed algorithm tries to mitigate these orientation effects by using weights according to the area captured. This can be seen with the better hand detection accuracy of the proposed algorithm as compared to [11]. To remove irrelevant areas from a fingertip component, knuckle line detection algorithm has been used. If it does not use knuckle line detection algorithm then there are 95 wrong hand detection cases and if it is used, then the number of such cases has been reduced to 62. Further, the proposed algorithm is nearly three times faster than [11].

5.5. Rotational invariance

For testing rotational invariance, 100 images have been taken from the database. It has given accurate results for fingerprint component extraction and hand detection. These images are rotated from -60° to $+60^\circ$ with a step-size of 5° . Results are evaluated and are found to be accurate. But as the angle of rotation is increased beyond $+60^\circ$, results start to deteriorate. Local orientation of a finger must lie between -90° and $+90^\circ$; otherwise the lowermost component in finger is marked as the topmost component. Hence, spurious fingertip component is marked and results suffer.

6. Conclusions

This paper has proposed an efficient algorithm to segment a slap-image into fingerprints and to classify them correctly into one of the index, middle, ring and little fingers of left or right hand. It has accurately detected the fingertip components from a given slap-image using anisotropic measures and multi-resolution analysis. Since anisotropic nature of each pixel is independent of any shape constraint and is significant in these dull print regions, it can handle carefully the non-elliptical shape of fingertip and the presence of dull prints in some part. Also, problems like merging of separate nearby components are avoided by considering the slap-image at original resolution. Local orientation of two components is used to find whether these components belong to a finger

or not. This makes the clustering of components into classes/finger more effective. An efficient knuckle line detection algorithm has been used to get rid of the areas that do not consist of fingerprint from a fingertip component. It uses image enhancement to remove the effects of halo/sweat, dry finger or dull prints, if they are present in some part of the fingertip component. Also, to remove the effects of creases and bad quality, an polynomial approximation is applied on a cost function to remove local artifacts and to look for both local and global behavior of the cost function. In addition to this, with the use of edges as features and radon transform, evaluation of the cost function is fast and accurate. Geometrical and spatial constraints of hand geometry are used to classify each fingerprint. Such a classification is robust because it uses the local orientation of two fingertip components along with their areas. Cases where spurious local orientation of one fingertip component can result in wrong hand detection are eliminated. It can handle the problems of large rotational angles, dull-prints, non-elliptical shape, small orientation changes in adjacent fingers and components of different fingers lying closer to each other. It has been tested on 6732 images obtained from 1122 subjects. It has accurately segmented each slap-image into four distinct fingerprints. It is worth to explore the possibility of using the available feature selection techniques like [20] to improve its accuracy.

Acknowledgments

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] 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.
- [2] C. Watson, M. Garris, E. Tabassi, C. Wilson, R. McCabe, S. Janet, K. Ko, User's guide to Non-Export Controlled Distribution of NIST Biometric Image Software, NIST Report, 2004.
- [3] A. Hicklin, C. Reedy, Implications of the IDENT/IAFIS Image Quality Study for Visa Fingerprint Processing, Technical Report, Mitertek Systems, 2002.
- [4] P. Lo, P. Sankar, Slap Print Segmentation System and Method, US Patent 7,072,496, 2006.
- [5] R. Hodl, S. Ram, H. Bischof, J. Birchbauer, Slap fingerprint segmentation, in: Computer Vision Winter Workshop, 2009.
- [6] Y. Zhang, G. Xiao, Y. Li, H. Wu, Y. Huang, Slap fingerprint segmentation for live-scan devices and ten-print cards, in: The 20th International Conference on Pattern Recognition (ICPR), 2010, pp. 23–26.
- [7] 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: The 17th IEEE International Conference on Image Processing (ICIP), 2010, pp. 3081–3084.
- [8] Y. Li, Y. Zhang, J. Lu, C. Liu, 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.
- [9] N. Singh, A. Nigam, P. Gupta, P. Gupta, Four slap fingerprint segmentation, in: Intelligent Computing Theories and Applications, 2012, pp. 664–671.
- [10] N. Singh, K. Tiwari, A. Nigam, 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.
- [11] P. Gupta, P. Gupta, Slap fingerprint segmentation, in: The fifth IEEE International Conference on Biometrics: Theory, Applications and Systems (BTAS), 2012, pp. 189–194.
- [12] M. Turk, A. Pentland, Eigenfaces for recognition, *J. Cogn. Neurosci.* 3 (1) (1991) 71–86.
- [13] S. Zambanini, M. Kampel, Segmentation of ancient coins based on local entropy and gray value range, in: The 13th Computer Vision Winter Workshop, 2008, pp. 9–16.
- [14] L. Murphy, Linear feature detection and enhancement in noisy images via the radon transform, *Pattern Recognit. Lett.* 4 (4) (1986) 279–284.
- [15] A. Aho, J. Hopcroft, J. Ullman, *Data Structures and Algorithms*, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1983.
- [16] S. Chikkerur, A. Cartwright, V. Govindaraju, Fingerprint enhancement using STFT analysis, *Pattern Recognit.* 40 (1) (2007) 198–211.
- [17] R. Jain, R. Kasturi, B. Schunck, *Machine Vision*, vol. 5, McGraw-Hill, New York, 1995.
- [18] D.M. Hawkins, The problem of overfitting, *J. Chem. Inf. Comput. Sci.* 44 (1) (2004) 1–12.
- [19] G. Dahlquist, A. Bjork, *Equidistant Interpolation and the Runge Phenomenon. Numerical Methods*, Prentice Hall, Englewood Cliffs, New Jersey, 1974.
- [20] Z. Li, Y. Yang, J. Liu, X. Zhou, H. Lu, Unsupervised feature selection using nonnegative spectral analysis, in: The 26th AAAI Conference on Artificial Intelligence (AAAI), 2012, pp. 1026–1032.



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 Ph.D. 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.