Applied Soft Computing 29 (2015) 411–423

Contents lists available at ScienceDirect

Applied Soft Computing

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

A robust singular point detection algorithm

Puneet Gupta*, Phalguni Gupta

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

ARTICLE INFO

Article history:
Received 31 October 2013
Received in revised form 13 January 2015
Accepted 20 January 2015
Available online 29 January 2015

Keywords:
Biometrics
Fingerprint classification
Singular points
Core
Delta

ABSTRACT

This paper proposes an efficient algorithm to extract the singular points which can be used to classify the given fingerprint. It makes use of a novel algorithm which is a hybrid of orientation field, directional filtering and Poincare Index based algorithms to detect singular points, even when the fingerprint is of low quality or singular point is occluded. Locations of detected singular points are not much accurate and thus they are further refined. Also, some delta points which lie near to the border, may be missed out at the time of detection. Efforts are made to retrieve these missed points. The proposed algorithm also determines the direction of a singular point along with its type (either core or delta). It uses these detected singular points to classify accurately arch, tented arch, left loop, right loop, double loop and whorl type fingerprint patterns. It can handle efficiently the cases of missing delta points during fingerprint classification. The proposed algorithm has been tested on three publicly available databases. It reveals that the proposed algorithm exhibits better singular points detection and fingerprint classification performance in comparison to other well known algorithms.

1. Introduction

A fingerprint (FP) contains patterns of ridges and valleys that flow in a coherent manner. It is one of the most extensively studied and well accepted biometric traits because it satisfies various necessary characteristics of a suitable biometric trait like uniqueness, permanence, difficult to forge, etc. Besides ridge-valley flow, a FP also contains (i) minutiae which are randomly distributed local features; (ii) global features called as 'singular points' (SPs); and (iii) specific type of FP pattern [1]. Any SP can be categorized as either core or delta point which is defined as a concentrate region where the ridge curvature converges to a local maximum or local minimum respectively [2]. This paper proposes an efficient algorithm focused to extract accurately SPs and to detect the FP pattern.

Accurate matching, correct classification, efficient indexing and enhancement are of paramount importance in a FP based biometric system [3]. Most of the times, these are difficult to achieve due to bad quality FPs [4]. Thus, accurate detection of SPs can play a crucial role in solving many problems like (i) indexing large FP database [5], (ii) smoothing OF by reconstruction [6], (iii) synthesizing FPs [7], (iv) alignment of FP which assists in global FP matching [8] and (v) classification of FPs [9]. Thus, reliable extraction of SP is an important problem to be considered.

Extracted SPs are used for FP classification [9–11]. But such SP based classification systems may give erroneous classification results when (i) a SP is left out during acquisition; (ii) a SP is not accurately localized; or (iii) direction of a SP is incorrect. SP can be missed during acquisition when a FP is partially captured due to fixed area of FP acquisition sensor. These are mostly the case with delta points. Thus, unlike existing algorithms, it is better if FP classification can successfully work even in the absence of delta points. Missing of genuine SPs and generation of spurious SPs are the common phenomenon in low quality FP images. Such SPs are inevitably generated [12] due to (i) inappropriate interaction between environment and user which causes partial FP acquisition, (ii) user condition like cuts and bruises on fingertip, (iii) sensor's condition like small area and presence of latent prints, and (iv) occupation or age which can smoothen the ridge-valley structure. Even in good quality FP images, it is observed that SPs can be missed if two core points lie close to each other like in the case of whorl pattern or SPs can be spuriously detected like in the arch type pattern.

Advantages of the proposed algorithm are manifold. The proposed algorithm can accurately detect the SPs even if the FP has low quality or SP is occluded. It reconstructs such areas using a model based algorithm. It detects low false generated minutiae and more genuine minutiae in a time efficient manner. Further, it gives highly accurate location and direction of a SP along with a SP type (core or delta). Moreover, it has proposed a hybrid SP extraction algorithm which recaptures the missed delta points present near

* Corresponding author. Tel.: +91 9559754489; fax: +91 5122597579.
E-mail addresses: puneet@cse.iitk.ac.in (P. Gupta), pg@cse.iitk.ac.in (P. Gupta).

http://dx.doi.org/10.1016/j.asoc.2015.01.027
1568-4946/

FP borders. The proposed FP classification algorithm makes use of these extracted SPs for better performance. It reliably detects the arch pattern and even marks a reference point in such pattern. It can correctly distinguish between left/right loop type FPs even in the absence of delta points, which is usually the case when FP is captured in an unconstrained environment. Also, it can accurately distinguish between double loop and whorl pattern which is a challenging task in FP classification. The experiment results conducted on three publicly available databases, reveal that the proposed system exhibit better SP detection and FP classification performance than other well known systems.

The paper is organized as follows. Next section discusses some of the algorithms which are used to design the proposed algorithm. The proposed algorithm has been presented in Section 3, which can be used to extract SPs and to classify FPs. Experimental results are analyzed in Section 4. Conclusions are given in the last section.

2. Preliminaries

This section discusses various algorithms for SP detection and FP classification that are used to design the proposed algorithm.

2.1. SP detection

SP extraction algorithms based on OF can be categorized as (i) Poincare Index (PIindex) based, (ii) directional partitioning based, (iii) template based, (iv) orientation curvature based algorithms, and (v) OF modeling based algorithms.

2.1.1. Poincare Index (PIindex) based algorithm

An algorithm based on PIindex evaluation has the following advantages: (i) it is robust against image rotation, (ii) it gives highly accurate SPs location and (iii) it gives fixed values at each SP location, using which arch type pattern can easily be identified. PIindex value is calculated at each pixel along a closed path (i.e. a block). Let ϑ be such a pixel, surrounded by a block having n_ϑ boundary pixels. Let $(o_1, o_2, \ldots, o_{n_\vartheta})$ be OF of these boundary pixels in a clockwise fashion. Then PIindex value at the central pixel ϑ, $PI(\vartheta)$, is given by

$$PI(\vartheta) = \sum_{i=1}^{n_\vartheta - 1} [f(o_{i+1} - o_i)] + f(o_1 - o_{n_\vartheta}) \quad (1)$$

where f is

$$f(x) = \begin{cases} x, & \text{if } |x| \le \pi/2 \\ \pi - x, & \text{if } x > \pi/2 \\ \pi + x, & \text{if } x < -\pi/2 \end{cases} \quad (2)$$

$PI(\vartheta)$ is equal to $+\pi$ or $-\pi$ at the core or delta point location respectively. Directional image can be used instead of OF for calculating PIindex [13,14].

If PIndex is computed using small block-size, then SPs can be reliably located. But many spurious SPs can also be generated due to noise or bad quality. On the contrary, if larger block-size is used then, true SPs may be missed but it removes spurious SPs. Hence, there is a trade-off between true-miss and false-accept. Other limitations of such algorithms are that (i) these can generate false SP or miss genuine SPs for low quality FP images, (ii) direction of SPs is not estimated and (iii) these are time consuming [15].

Two important properties [16] of PIindex are (i) if a FP image is captured completely then, it should have the same number of core and delta points, and (ii) PIindex value is independent with the integral paths till these paths are homotopic i.e., does not contain any new SP. These properties can be used to remove some limitations of PIindex based algorithms [17,16,18].

2.1.2. Directional partitioning based algorithm

Such algorithms clusters similar orientation, direction [19] or gradient [15]. Such clusters are separated by several boundaries (or *transition lines* [15]). Intersections of these transition lines give the points where the ridge curvatures attain local maxima/minima, i.e. SPs [2]. Estimates of orientation, direction or gradient may be spurious due to the bad quality, hole or minutiae, etc. thus, smoothing operations can be performed which can shift the location of SPs. Another important point to note is that if the number of clusters is increased, then all intersections may not be SPs, but at SP, there must be an intersection. Though these algorithms are computationally efficient, but they may fail to extract SPs when SPs lie close to each other like whorl pattern. Also, this type of algorithms cannot distinguish arch pattern.

2.1.3. Template based algorithms

In template based algorithms [20–22], SP type filters (or templates) are convolved with the FP image to extract singularities. An advantage of such algorithms is that position and spatial orientation of a SP can simultaneously be extracted and this orientation estimate is highly accurate. But such algorithms do not have any defined value for SP. These algorithms use either a threshold or a global maxima to extract SPs. Disadvantage of a threshold based algorithm is that such a threshold which can detect multiple genuine SPs and can avoid spurious SPs, cannot be effectively determined. On the other hand, global maxima based SP detection can detect one core and delta point location. Also, in the absence of clearly defined value of SP, the arch type pattern cannot be determined.

2.1.4. Orientation curvature based algorithms

As areas near SPs have large orientation change, this leads to high curvature [23,24]. Curvature can be measured by various ways [25,26]. Disadvantages of such algorithms are that (i) good number of spurious SPs are generated, (ii) SP can be easily missed due to low quality, (iii) SP type and direction cannot be determined and (iv) it fails to separate arch type pattern as there is no fixed value for SP.

2.1.5. OF modeling based algorithms

If OF used to detect SP is erroneous due to low quality, then it results in a spurious SPs generation or missing of genuine SPs. Therefore, it is smoothed or modeled by using global structure before the detection of SP to remove local noise. Also, missing of genuine SPs is reduced as OFs are accurately interpolated in low quality areas using global constraints. OF modeling algorithms can be divided into two categories based on whether prior knowledge of SPs is required or not. First category which requires prior knowledge of SPs for modeling OF [27–29] and is not of much use in SP detection. Another category which does not require any prior knowledge of SPs for orientation modeling [6,16,30–33] is time-efficient and highly robust against spurious SPs. But OF modeled by it deviates from actual OF, especially near SPs. Thus, the extracted SPs may have poor localization.

2.2. FP classification

FP classification algorithms can be classified as:

1. *Syntactic based algorithms* cluster similar elements [34] and assign a symbol to each cluster. These symbols build grammar for each FP class. For the given FP image, the grammar is obtained and a parsing strategy is used to find the most appropriate FP class. But some FP classes require complex symbol grammar.
2. *Rule-based algorithms* rely on the heuristics and are mainly used with other features (like ridge line shape [35]) for better FP

classification. Such algorithms give improper results if given FP is of bad quality or SPs are missed during FP acquisition.

3. *Structural algorithms* build a hierarchical organization from low-level to higher-level features to classify FP. OF of the given FP image is clustered and these clusters are represented in the form of a graph (i.e., hierarchical organization) [36]. For classification, inexact graph matching technique is used for matching this graph with each class prototype graph. For accurate results, these algorithms require accurate clustering, which in turn need accurate OF.
4. *Hybrid algorithms* can also be used to classify the FPs. Probabilistic neural network can be used with ridge tracing algorithm [9]. Similarly, rule-based algorithm can be used with ridge tracing and support vector machine [37].

It is seen that FP classification is mainly dependent on OF estimation in one way or the other. As accurate OF estimation is not possible in low quality FP images, therefore FPs can be wrongly classified.

3. Proposed algorithm

This section proposes an efficient algorithm to extract SPs from a given FP image. It smoothen OF by OF modeling and a candidate set of SPs are extracted from it by applying modified Directional partitioning algorithm. This candidate set of SPs is validated by PIindex evaluation. This whole algorithm has the benefits of model based, directional partitioning based and PIindex based algorithms. That is, using the proposed algorithm, only genuine SPs are reliably extracted. Subsequently, a post-processing operation is performed to localize detected SPs more accurately and detect the missing delta points, if any. These detected SPs are used to classify the FPs. The flow diagram of the proposed algorithm is shown in Fig. 1.

3.1. Preprocessing

An image enhancement technique [38] is applied on the given FP, I. It extracts FP foreground blocks and enhances ridge-valley pattern using anisotropy measures and STFT analysis respectively. Multi-scale Gaussian filter based algorithm [17] is used to extract the OF θ from the enhanced image. Gradient vectors, coherence data and sinusoidal components are smoothed in it by using a Gaussian filter to remove local errors. Quality of I is estimated at each pixel by using symmetric filters [39]. It is required in subsequent steps for the extraction of missing delta points. Besides Q, symmetric filters also give core and delta filter responses, i.e., $\hat{S}_1$ and $\hat{S}_{-1}$ respectively.

3.2. OF modeling

It is observed that OF θ is error prone near bad quality areas and if it is used for SP detection, then it generates a large number of spurious SPs. This can be visualized by using Fig. 2(1a)–(4a) which show θ for the FPs corresponding to Fig. 2(1)–(4) respectively. Thus, OF is first smoothed by using the OF modeling algorithm [6]. Modeled OF are constructed from the projection of actual OF onto the Fourier basis [6]. The number of Fourier basis are decided by the order N used to create the Fourier basis. Thus, N plays a crucial role in determining the effectiveness of OF modeling. If N is small then modeled OF is very different from the actual OF, especially near the areas consisting of large orientations changes (like areas near SPs). But for large value of N, abrupt changes arise due to bad quality, are also modeled. Experimentally, it is observed that order $N = 6$ is sufficient to fit the θ with low residual error rate [6]. Let $\overline{\theta}$ represent the modeled OF obtained from θ. To visualize modeled OF $\overline{\theta}$, consider Fig. 2(1b)–(4b) which show $\overline{\theta}$ for the FP images corresponding to Fig. 2(1a)–(4a) respectively. Also, θ for the FP images are shown in Fig. 2(1a)–(4a). Therefore, it is clearly indicated from Fig. 2 that θ contains local noises which are removed by using modeled OF $\overline{\theta}$.

3.3. Genuine SPs extraction

This section proposes an efficient algorithm for extracting the SPs by using modeled OF $\overline{\theta}$. It consists of two major tasks. Initially, a candidate set of SPs is obtained by using proposed modified directional partitioning algorithm. Its SPs are validated by using a PIindex algorithm which has high localization ability and can remove spurious SPs obtained in arch type pattern.

In the modified directional partitioning algorithm, transition lines are first detected for extracting a candidate set of SP. Clustering of modeled OF $\overline{\theta}$ is used for detection. It does not require any preprocessing or multiple resolution analysis due to smooth mature of $\overline{\theta}$. Thus, accurate SPs are located in less time computation. Let Z be the number of bins required for clustering. Then directional image D_I which represents a unique number for each cluster is given by

$$D_I = \left\lceil \frac{Z \cdot \overline{\theta}}{\pi} \right\rceil \tag{3}$$

Experimentally, it is found that $Z = 8$ is sufficient for the accurate results. Transistion line is detected from D_I by using Algorithm 1. In Algorithm 1, binary image FL_{bin} is formed by setting those pixels to 1 which have a change of cluster along X or Y direction. Therefore, FL_{bin} contains 1 at the transition line location; otherwise it is 0. As intersections of transition lines are regarded as the SPs, these are determined by applying Harris-corner detector [40] on FL_{bin}. All the intersections may not be SPs as can be observed in the arch type pattern thus, these are referred as a candidate set of SPs.

Algorithm 1. *TransistionLine*$(D_I, length, breadth)$

Require: Size of Directional image D_I is given by *length* and *breadth*
Ensure: Binary image FL_{bin} contain 1 at transition line otherwise 0.

```
FL_bin = zeros(length, breadth)
for i = 1 to length-1 do
  for j = 1 to breadth-1 do
    if D_I(i, j) ≠ D_I(i+1, j) OR D_I(i, j) ≠ D_I(i, j+1) then
      FL_bin = 1
    end if
  end for
end for
return
```

Algorithm 2. *ValidationCandidateSet*$(\overline{\theta}, X, Y)$

Require: X and Y store x and y coordinates of candidate SPs while $\overline{\theta}$ is modeled OF
Ensure: Arrays C and D store locations of core and delta points respectively

```
a = b = 1
C = D = [ ]    //Empty arrays.
/*let |X| is number of elements in X*/
for i = 1 to |X| do
  Consider (X[i], Y[i]) and its 8-connected neighbors
  Calculate there PIindex value by using θ̄ at a block-size of 3 × 3.
  Pixel locations whose PIindex value is equal to π are appended in C
  While pixel locations whose PIindex value is equal to −π are appended in D
end for
return
```

Extracted candidate set of SPs is validated to extract genuine SPs by using Algorithm 2, which gives the location of genuine core or delta points in C and D arrays respectively. It is observed that a candidate SP can deviate from actual SP location by 1 pixel and sometimes it is not a genuine SP, like in arch pattern. To accurately localize and validate a candidate SP, PIindex values are evaluated at a candidate SP pixel and its 8-connected neighboring pixels. If PIindex value at a pixel is π or $-\pi$ then that pixel is marked as genuine core or delta point respectively. Use of PIindex guarantees that spurious SPs generated due to arch pattern are removed. PIindex

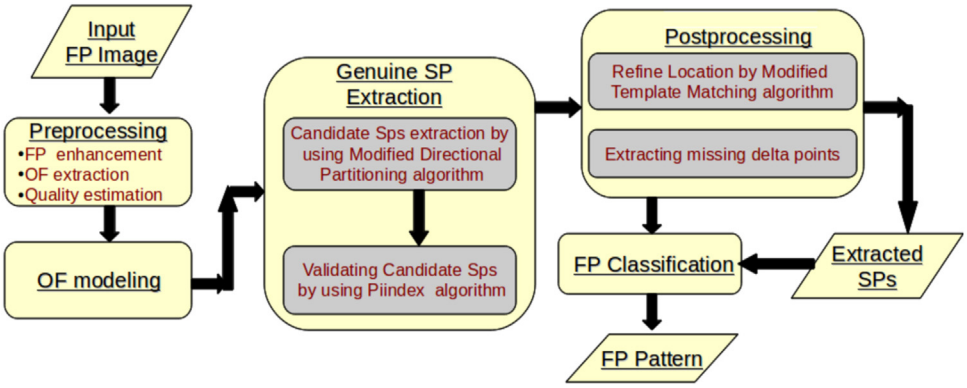

Fig. 1. Flow diagram of the proposed algorithm.

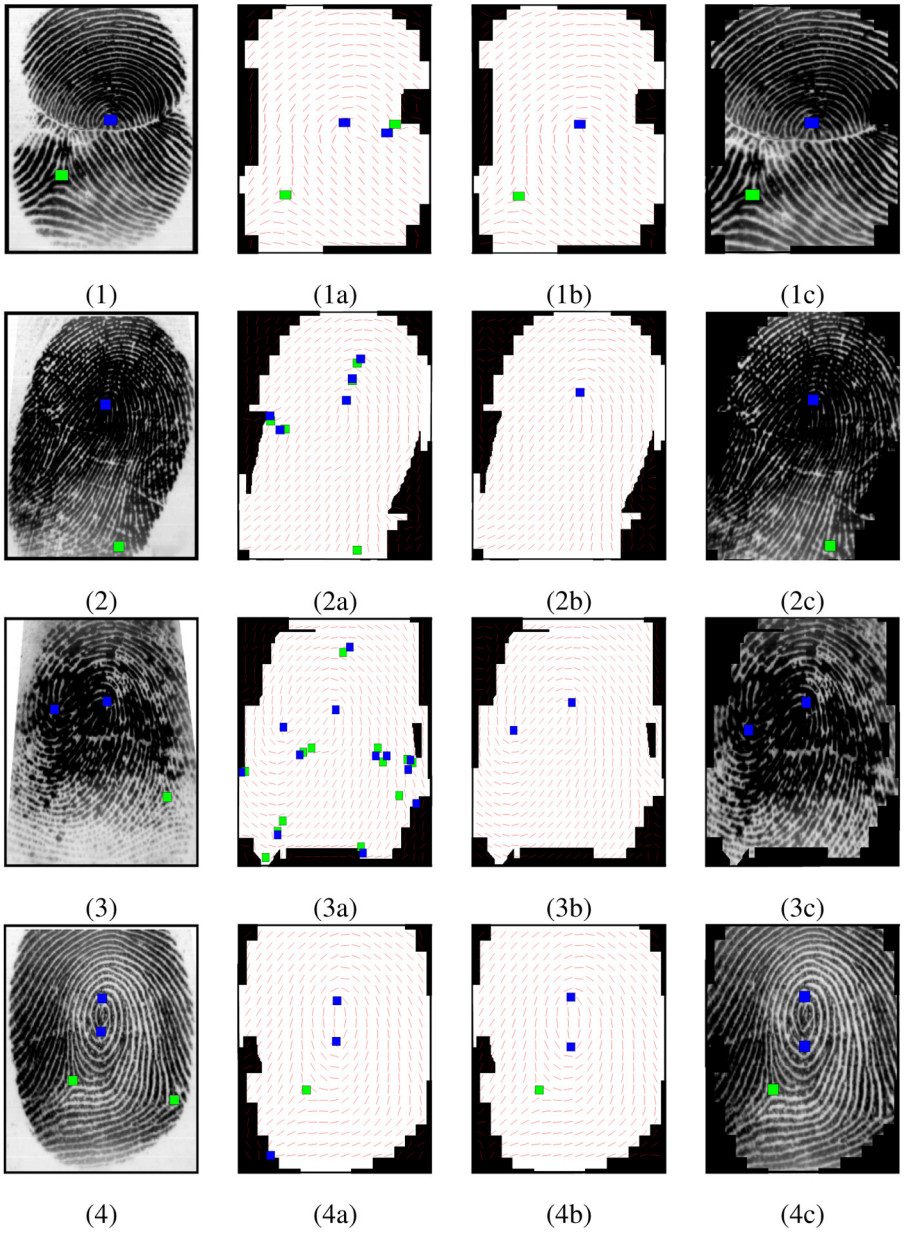

Fig. 2. Various stages of SP detection. Each row shows an example showing following from left to right: (i) actual FP with manually marked SPs, (ii) actual OF with detected SPs, (iii) modeled OF with detected SPs, and (iv) detected SPs after post-processing. (For interpretation of the references to color in the text, the reader is referred to the web version of the article.)

values are evaluated by using $\overline{\theta}$ in a minimum possible block-size, i.e. 3×3. If PIndex is calculated in small block-size, then SPs can be reliably located but it can generate many spurious SPs in case of erroneous OF. Since it is applied on modeled OF, problem of spurious SPs generation is avoided. Besides accurate localization, use of minimum possible block-size also assures that maximum one SP can be detected corresponding to a candidate SP due to the smooth nature of $\overline{\theta}$. If larger block-size is used, then SPs are detected at multiple locations corresponding to a candidate SP and these are not accurately localized, thus, it is avoided.

Algorithm 3. *SingularPointDetection*($\overline{\theta}$, z)

Require: Z is number of bins for clustering modeled OF, $\overline{\theta}$.
Ensure: Arrays C and D contain core and delta points respectively
$D_I = \left\lceil \frac{Z \cdot \overline{\theta}}{\pi} \right\rceil$
(*length*, *breadth*) = *sizeof*(D_I)
FL_{bin} = *TransistionLine*(D_I, *length*, *breadth*) //Refer Algorithm 1
Find candidate SPs by applying Harris Corner Detector on FL_{bin}
x and y coordinates of candidate SPs are stored in X and Y respectively.
[C, D] = *ValidationCandidateSet*(X, Y) //Refer Algorithm 2
return

The proposed genuine SPs extraction algorithm is explained in Algorithm 3 where genuine core or delta points are stored in arrays C and D respectively. It extracts SPs even if these SPs lie close to each other. As an example, refer Fig. 2(4b) which shows detected core and delta points in blue and green squares respectively, for the FP shown in Fig. 2(4). This is due to the use of the robust extraction of pixel-level OF using multi-scale Gaussian filter based algorithm. Further, it is observed that the use of modified directional partitioning before PIndex method helps to reduce the search space substantially, thereby reducing the computational cost. Examples of such SP detections are shown in Fig. 2(1b)–(4b).

3.4. Post-processing

SP detection accuracy is further improved by using various post-processing operations. If a SP is extracted by using modeled OF ($\overline{\theta}$), then its location may be shifted from its actual position. Thus, a SP localization algorithm has been proposed. Also, delta points present near FP borders can be missed which are recaptured by using a hybrid SP extraction algorithm.

3.4.1. Refine location by modified template matching

Modeled OF $\overline{\theta}$ is error prone near SPs; thus locations of SPs are refined by using actual OF θ. The proposed algorithm can detect SPs even if SPs lie in bad quality block. If θ is used to refine such SPs lying in bad quality blocks, then it results in spurious localization of SPs. To avoid such cases, if a SP has quality value less than some predefined threshold $Q_{threshold}$ (experimentally chosen to be 0.1), then location of that SP is not considered for refinement. Otherwise, there is a need to refine its location.

Algorithm 4. *FindBlockSizeCore*($\overline{\theta}$, $\overline{f}$, $\overline{x}$, $\overline{y}$)

Require: Modeled OF $\overline{\theta}$ and maximum block-size $\overline{f}$
Ensure: k is the adaptive block-size.
$k = 2 \times \overline{f}$
repeat
  Extract a block of size $k \times k$ centered at $(\overline{x}, \overline{y})$
  Calculates PIndex value of the extracted block by using $\overline{\theta}$.
  $k = \left\lceil \frac{k}{2} \right\rceil$
until (*PIvalue* == π)
$k = 2 \times k$
return

The proposed algorithm uses Hill Climbing algorithm for refining SP location. It is based on the assumption that the deviation of SP localization in modeled OF from that of actual OF is not large. Let us consider that there is a core-type SP whose location needs to be refined and is labeled as C with the initial location given as $(\overline{x}, \overline{y})$. For refining it, a block $\hat{B}$ of size $k \times k$ and center $(\overline{x}, \overline{y})$ are extracted from S_1. In the proposed algorithm, value of k is adaptively chosen because:

1. It is observed that in whorl type pattern, core points lie closed to each other. Hence, if a high value of k for a block is selected, it may result in loss of actual core point when two core points lie in a block.
2. Similarly, if value of k is low, then the algorithm may take a large number of iterations to converge and may give spurious results due to less number of elements in a block for finding global maxima in core filter response $\hat{S}_1$.

To adaptively estimate the value of k, initially it is set to twice the average width of a ridge-valley pair [41], that is $2 \times \overline{f}$. This k is iteratively updated satisfying two constraints which are: (i) no two SPs lie inside a block and (ii) value of k should not be very small. To update k, PIndex is calculated at $(\overline{x}, \overline{y})$ by using a block-size of $k \times k$. If PIndex value is equal to π then it indicates the presence of only one core point. It results from two observations: (i) difference between the total number of core and delta points is 1 and (ii) minimum Euclidean distance between a core and a delta point is always greater than k, hence a core and a delta point cannot coexist in any considered blocks. If evaluated PIndex value is not equal to π, then value of k is updated by setting $k = k/2$ and the whole procedure of PIndex evaluation is repeated till PIndex value is equal to π. Algorithm 4 is used to find the adaptive block-size, k.

Algorithm 5. *RefineLocationCore*($\overline{\theta}$, S_1, C, Q, $Q_{threshold}$, $\overline{f}$)

Require: Modeled OF $\overline{\theta}$, core-filter response S_1, core-point locations C, quality map Q, threshold $Q_{threshold}$ and maximum block-size $\overline{f}$
Ensure: An array ($\overline{C}$) where refined locations of core points are stored
for i = 1 to *sizeof*(C) **do**
  Store the locations of $C[i]$ in $(\hat{x}, \hat{y})$
  if $Q(\overline{x}, \overline{y}) > Q_{threshold}$ **then**
    repeat
      $\overline{x} = \hat{x}$ and $\overline{y} = \hat{y}$
      k = *FindBlockSizeCore*($\overline{\theta}, \overline{f}, \overline{x}, \overline{y}$) //Refer Algorithm 4
      Extract the block from S_1 which has size $k \times k$ and center $(\overline{x}, \overline{y})$
      Detect global maxima of the block and store its location in $(\hat{x}, \hat{y})$
    until ($\overline{x} == \hat{x}$ *AND* $\overline{y} == \hat{y}$)
    $\overline{C}[i] = (\hat{x}, \hat{y})$
  else
    $\overline{C}[i] = C[i]$ //For bad quality, locations are not modified.
  end if
end for
return

This k is used to extract block $\hat{B}$ (of size $k \times k$ and center $(\overline{x}, \overline{y})$) from $\hat{S}_1$ which is used to refine $(\overline{x}, \overline{y})$. Let global maxima of $\hat{B}$ be $(\hat{x}, \hat{y})$ which gives the most probable location of core point (by template based algorithm [39]) inside $\hat{B}$. Hence, if $(\overline{x}, \overline{y})$ is different from $(\hat{x}, \hat{y})$ then it is refined by setting it to $(\hat{x}, \hat{y})$. This procedure is repeated till global maxima location and initial location coincide. Algorithm 5 is used to refine the locations of core points in C and to store these refined locations in $\overline{C}$. Similarly location of delta points is modified.

The proposed algorithm for refining SPs locations can accurately detect the locations of SPs even if SPs lie in bad quality regions. Also, by choosing the adaptive block-size, accurate locations of SPs can be found even if two SPs lie close to each other.

Algorithm 6. *FindMissingDelta* (S_{-1}, γ, θ)

Require: Delta-filter response S_{-1}, actual OF θ and number of maxima to be used γ
Ensure: Array *DeltaList* which contains the locations of all missed delta points

```
DeltaList = []
Find the locations of top γ global maxima in S_-1
Store these locations in ((x_1, y_1), (x_2, y_2) ... (x_γ, y_γ))
for i=1 to γ do
    Extract the block which has size 7 × 7 and center at (x_i, y_i).
    Calculate Plindex value of this block by using θ and store it in PIvalue
    if PIvalue == −π then
        append(DeltaList,L[i])
    end if
end for
return
```

3.4.2. Extracting missing delta points

Use of modeled OF in SP detection may result in missing of some genuine SPs (especially delta points) which are closed to the borders. Reasons for this are: (i) FP enhancement and segmentation, (ii) location of a SP may be shifted a lot in modeled OF such that it lies outside the foreground region, or (iii) the behavior of a SP is localized in a small area, hence such a SP is not accurately modeled. Such cases can be seen from Fig. 2(2b) and (4b) where delta points (in green color) are missed, when modeled OF is used for SP detection. Hence, missed delta points which lie in the foreground are recaptured. Initially, cases of missing delta points are determined. It can be noted that if a FP image is captured completely, then it should have the same number of core and delta points [16,18]. Therefore, if a FP has an unequal number of core and delta points, then it implies that either full FP is not captured, and/or some SPs are not detected. Hence, if an image has a number of core points greater than the number of delta points, then that image is further analyzed. For analysis, top γ global maxima near the borders of delta filter response S_{-1} are determined where γ is the difference between the total number of core and delta points. To validate these top γ maxima, Plindex value on these locations are calculated by using θ at a block-size of 7×7. If Plindex value at any of these maxima is $-\pi$ then that point is marked as genuine delta. Algorithm 6 is used to recapture the missing delta points lying in the foreground and to store these in an array *DeltaList*. To perceive this, consider Fig. 2(2c) where missed delta point (on the left in Fig. 2(2b)) is retrieved.

3.5. FP classification

This section proposes an algorithm to classify a given FP which uses the number, location and direction of core points. Direction of each core point plays a crucial role in FP classification and is estimated by using a template matching algorithm. Let location of a core point be $(\bar{x}_c, \bar{y}_c)$. Then the direction of core filter response S_1 at the pixel $(\bar{x}_c, \bar{y}_c)$ is referred as the direction of this core point. Separate cases are considered for FP classification, based on the number of extracted core points.

3.5.1. Case 1

If a FP image contains no core point, then it has arch pattern. In such cases, there is no core point which can be used as a reference point. Determining the reference point in arch pattern is very important [42,43], therefore the global maxima for core filter response, S_1 is used as a reference point.

3.5.2. Case 2

If a FP image contains one core point, then it can be classified among left loop, right loop or tented arch pattern. Assume that (x_{c_1}, y_{c_1}) and θ_{c_1} are the location and the direction of the extracted core point respectively. And there is a line l_1 which passes through (x_{c_1}, y_{c_1}) and has direction given by θ_{c_1}. There are two possibilities which are (1) there is one delta point or (2) there is no delta point. These cases are separately analyzed as:

1. *Possibility 1*: Let (x_{d_1}, y_{d_1}) be the location of delta point. It is observed that tented arch patterns have vertical trend and they do not have any ridge re-curving between core and delta points [1]. Therefore, a distance measure, *SD* [44], is used for finding tented arch patterns. It gives minimum possible distance between delta point and line l_1 which is oriented at θ_{c_1} and passes through (x_{c_1}, y_{c_1}). It is given by

$$SD = \frac{|y_{d_1} - mx_{d_1} - c|}{\sqrt{1 + m^2}} \tag{4}$$

where $m = \tan\theta_{c_1}$ and $c = y_{c_1} - mx_{c_1}$. If *SD* is less than a threshold Th_{TA}, then FP image has tented arch pattern. Since there should not be any re-curving ridge in tented arch pattern [1], so the threshold Th_{TA} is set to a width of ridge-valley pattern, i.e., $\bar{f}$. If FP does not belong to the tented arch pattern, then it is checked for left or right loop. Assume that there is a line l_2 which joins (x_{c_1}, y_{c_1}) and (x_{d_1}, y_{d_1}). Direction of this line l_2 is given by

$$\Theta = \tan^{-1}\left(\frac{y_{c_1} - y_{d_1}}{x_{c_1} - x_{d_1}}\right) \tag{5}$$

and angle between lines l_1 and l_2 is given by:

$$\Delta = (\theta_{c_1} - \Theta) \bmod 2\pi \tag{6}$$

If $\Delta < \pi$, then FP image consists of left loop; otherwise it is right loop. To comprehend these cases, refer Fig. 3 which shows l_1 (in blue line), *SD* (in green line), Δ (in yellow arc), core points (in red square) and delta points (in green square).
2. *Possibility 2*: Delta points are frequently missed during FP capture. If FP classification is based on the existence of SP, then such cases are either left out [45] or generate spurious results. Both these cases can degrade FP classification accuracy, therefore an algorithm to effectively handle such cases is proposed in this possibility.

Points in the transition lines have high orientation changes. One of these transition lines has the highest orientation change which is equal to or more than π. This happens due to change in OF from $-\frac{\pi}{2}$ to $\frac{\pi}{2}$. Let such a transition line exhibiting the orientation change of equal to or more than π be referred as *Prominent line*. Assume $\theta_{c_1} = \frac{\pi}{2}$ and there is a delta point. Then it is observed that the line l_1 (oriented at θ_{c_1} and passes through (x_{c_1}, y_{c_1})) divides the image into two sides, viz. left and right side of l_1. It is observed that nearly all the points on the prominent line and delta point lie on the same side of l_1. Hence, all points on the prominent line that lie in foreground region are used to detect the loop type, when delta point is absent in given FP. For prominent line detection, θ_{c_1} should be $\frac{\pi}{2}$. Therefore, $\bar{\theta}$ is modified using

$$\theta_{modified}(x, y) = \bar{\theta}(x, y) - \theta_{c_1} + \frac{\pi}{2} \tag{7}$$

at each pixel (x, y) of $\bar{\theta}$. A foreground pixel (x, y) is referred as a point on prominent line if

$$|\theta_{modified}(x, y) - \theta_{modified}(x + 1, y)| > \pi \tag{8}$$

or,

$$|\theta_{modified}(x, y) - \theta_{modified}(x, y + 1)| > \pi \tag{9}$$

Let $\{(\bar{x}_1, \bar{y}_1), (\bar{x}_2, \bar{y}_2), \ldots, (\bar{x}_{\bar{k}}, \bar{y}_{\bar{k}})\}$ represent $\bar{k}$ prominent pixels. If $\bar{k}$ is less than a threshold $\bar{k}_{th}$ then pattern present in FP image is marked as *cannot determine*. Threshold $\bar{k}_{th}$ represents the minimum number of points that should be present on the prominent

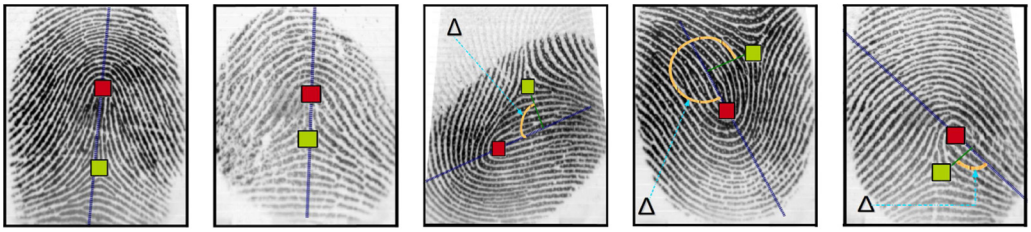

Fig. 3. Examples of FP classification for Case 2 and Possibility 1. FP pattern used from left to right: (a) tented arch, (b) tented arch, (c) left loop, (d) right loop and (e) left loop. (For interpretation of the references to color in the text, the reader is referred to the web version of the article.)

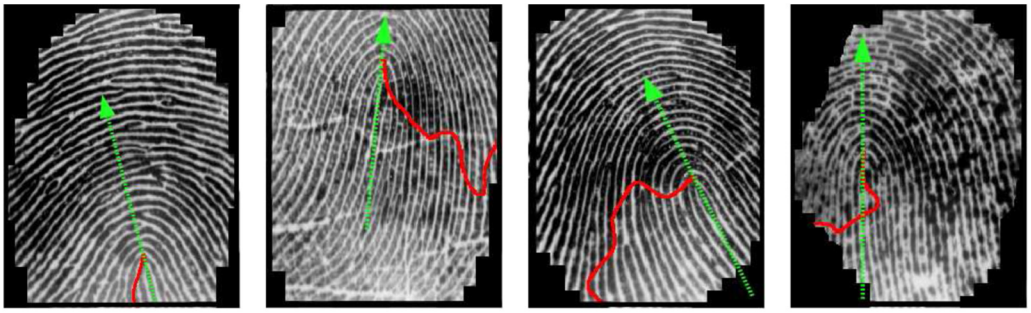

Fig. 4. Examples of FP classification for Case 2 and Possibility 2. FP pattern used from left to right: (a) *cannot determine*, (b) left loop, (c) right loop and (d) *cannot determine*.

line for effective FP classification. Therefore, it should be large and is set to six times of $\bar{f}$ (average width of a ridge-valley pair). Such a case can be seen in Fig. 4(a) depicting a *cannot determine* pattern due to less points on prominent line. While if $\bar{k}$ is more than the threshold $\bar{k}_{th}$ then FP is further analyzed for classification. For analysis, each point $(\bar{x}_b, \bar{y}_b)$ on the prominent line is considered and a line $\bar{l}_b$ is formed by joining (x_{c_1}, y_{c_1}) to $(\bar{x}_b, \bar{y}_b)$. Direction of $\bar{l}_b$ is

$$\overline{\Theta}_b = \tan^{-1}\left(\frac{y_{c_1} - y_b}{x_{c_1} - x_b}\right) \tag{10}$$

Therefore, angle between lines l_1 and l_b is given by

$$\Delta_b = (\theta_{c_1} - \overline{\Theta}_b) \bmod 2\pi \tag{11}$$

These angles are evaluated for each point on the prominent line and are represented as $(\Delta_1, \Delta_2, \ldots, \Delta_{\bar{k}})$. Evaluated angles are averaged as

$$\overline{\Delta} = \frac{(\Delta_1 + \Delta_2 + \cdots + \Delta_{\bar{k}})}{\bar{k}} \tag{12}$$

By using $\overline{\Delta}$, the pattern present in the given FP is estimated by

$$pattern = \begin{cases} Rightloop, & \text{if } \overline{\Delta} - \pi < -TH_{\overline{\Delta}} \\ Leftloop, & \text{if } \overline{\Delta} - \pi > TH_{\overline{\Delta}} \\ cannot\ determine, & \text{otherwise} \end{cases} \tag{13}$$

where threshold $TH_{\overline{\Delta}}$ separates left loop, right loop and *cannot determine* patterns, and is set to $\bar{f}$ (average width of a ridge-valley pair). Examples of the cases arise due to *cannot determine* patterns are shown in Fig. 4 where prominent lines and core-point directions are shown in red and green color on segmented FP image.

3.5.3. Case 3

If a FP image contains two core points, then either it has whorl or double loop (or twined loop) pattern. Our whorl class consists of central pocket loops and lateral pocket loops. Let (x_{c_1}, y_{c_1}) and (x_{c_2}, y_{c_2}) represent the locations of two core points, which are oriented at an angle of θ_{c_1} and θ_{c_2} respectively. Assume that there is a line $\hat{l}_1$ which passes through (x_{c_1}, y_{c_1}) and is oriented at an angle of χ, where χ is given by

$$\chi = \frac{\theta_{c_1} + (\theta_{c_2} + \pi) \bmod 2\pi}{2} \tag{14}$$

Then distance measure *SD* is used to determine the existence of the pattern in the given image. In this case, *SD* (Eq. 4) is used which refers to the minimum possible distance between (x_{c_2}, y_{c_2}) and line $\hat{l}_1$ in this case. If *SD* is less than threshold Th_{dl}, then input image has whorl pattern; otherwise it has a double loop pattern. In double loop, it is observed that at least three ridges pass between the two core points. Therefore, threshold Th_{dl} is set to three times of an average width of a ridge-valley pair ($\bar{f}$). An example of this case is given in Fig. 5 which show $\hat{l}_1$ (in blue line), *SD* (in green line), core points (in red) and delta points (in green).

4. Experimental results

The performance of the proposed algorithm has been analyzed on publicly available FP databases: Biostar's database [46] and FVC2004 (DB1 and DB2) [47]. Biostar's database consists of 500 samples, each of size 355 × 390 pixels [48]. These images are collected in an uncontrolled environment with no restriction on finger placement. This may cause various problems such as: large rotation and displacement, wet/dry FPs, scars and low-quality images. This database also provides the manual annotation of SPs using Henrys definition [1]. Total 537 core points and 235 delta points are manually annotated in the FPs of this database. There are some cases where no SP is annotated, even though they exist. For example, in Fig. 6(1) there is no annotated SP. Likewise, 25

Table 1
Usefulness of OF modeling.

SP type	*TP*	Precision	Recall	*F*-measure	Accuracy[a]
θ					
Core	539	0.84	0.93	0.79	93%[a]
Delta	196	0.75	0.93	0.69	
$\overline{\theta}$					
Core	568	0.99	0.98	0.97	100%[a]
Delta	194	0.96	0.91	0.89	

[a] FP classification are evaluated on a subset of size 200 FP images.

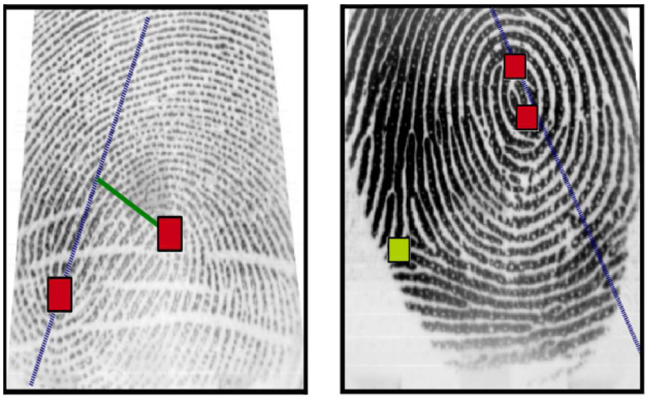

Fig. 5. Examples of FP classification for Case 3. These have following patterns: (a) double loop and (b) whorl. (For interpretation of the references to color in the text, the reader is referred to the web version of the article.)

annotated delta points are found to be missing after FP enhancement and segmentation. This happens because this type of delta points lies in the bad quality region or near border of FP area. Hence, these delta points are marked in the background region after FP segmentation. Thus, 577 core points and 211 delta points are considered for evaluating the performance of the proposed SP detection algorithm. Other databases used for evaluation of the proposed algorithm by using FP classification are FVC2004 DB1 and DB2 which contain 880 images of 110 fingers (eight impressions per finger) captured from optical sensors namely, CrossMatch and Digital Persona respectively. The size of each image in FVC2004 DB1 is 640×480 while that of each image in FVC2004 DB2 is 328×364.

For correct classification, ground truth is manually assigned to the FP images in each database. It requires tuning of various parameters like $\overline{k}_{th}$, $TH_{\overline{\Delta}}$, Th_{dl} and TH_{arch} based on a database. In this paper, average width of a ridge-valley pair, $\overline{f}$ is used to define all these parameters. It is calculated by applying [41] on 10 randomly selected FP images. The integer which is nearest to the average and power of 2 is set as $\overline{f}$. In this case, $\overline{f}$ are found to be 16 for Biostar, FVC2004 DB1 and FVC2004 DB2 database. In addition, the proposed algorithm uses Precision, Recall and *F*-measure for performance evaluation of SP detection [32] which are given by:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{15}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{16}$$

$$F\text{-measure} = \frac{2 \times \text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}} \tag{17}$$

where *TP*, *FN* and *FP* are the number of true positive, false negative and false positive cases respectively [49,50]. High values of Precision, Recall and *F*-measure indicate a good SP detection algorithm. Likewise, Accuracy is used for performance evaluation of FP classification which is given by:

$$\text{Accuracy} = \frac{\text{Number of correctly detected cases}}{\text{Total number of cases}} \times 100 \tag{18}$$

4.1. Experiments I

Aim of this set of experiments is to show the usefulness of OF modeling. For this, experiments are performed on Biostar's database. Test results are shown in Table 1. In the proposed algorithm, OF modeling is used to extract SPs and estimation of the direction of core points. Hence, it influences the FP classification accuracy. Thus, in Table 1, performance of core and delta point detection are analyzed along with FP classification. In order to use FP classification accuracy as a metric to evaluate the performance of the estimation of core point direction by using OF modeling, only 200 FP images which give same SPs irrespective of the use of OF modeling are used during FP classification. Also, $\overline{\theta}$ is set to θ (which avoid the use of OF modeling) to analyze the usefulness of OF modeling.

It is evident that when $\overline{\theta}$ is used, both the number of missed genuine SPs and that of falsely detected SPs are found to be less. This is due to the fact that $\overline{\theta}$ can reconstruct the OF in bad quality regions by using global constraints. Another crucial observation is that use of $\overline{\theta}$ gives better FP classification accuracy. But there exist same SPs for θ and $\overline{\theta}$. Hence, change in FP classification in Table 1 is mainly due to direction of core points. This points out that use of $\overline{\theta}$ gives a better estimate of core point direction.

4.2. Experiments II

This set of experiments (i) evaluates the performance of the proposed genuine SPs extraction algorithm which is a hybrid of model based, directional partitioning based and Piindex based algorithms and (ii) show the usefulness of the post-processing algorithms. Tests are conducted on Biostar's and FVC 2004 DB1 databases. Experimental results are shown in Table 2.

The proposed algorithm has been compared with PIindex algorithms having block sizes of 3×3 and 16×16, which are denoted by $PIindex_3$ and $PIindex_{16}$ respectively. Similarly, [32] which uses the Legendre basis for OF modeling followed by $PIindex_3$ is also used for SPs extraction. For effective evaluation, all these algorithms are evaluated on a common platform. Thus, these are applied on the OF, θ, of the enhanced FPs, which is extracted by the proposed algorithm. Also, let $Proposed_A$ and $Proposed_B$ be the results of the algorithm without and with post-processing respectively. Results shown in Table 2 indicates the effectiveness of post-processing step of the proposed algorithm. For comparison number of correctly detected, falsely detected and missed genuine SPs (both core and delta points) are considered along with the error in localization of the detected SPs. Localization of a SP is a crucial parameter in FP indexing [5] and in global matching [8]. There is always a difference between the SPs extracted from OF and manually detected. Since PIindex algorithm is regarded as the best known method to find

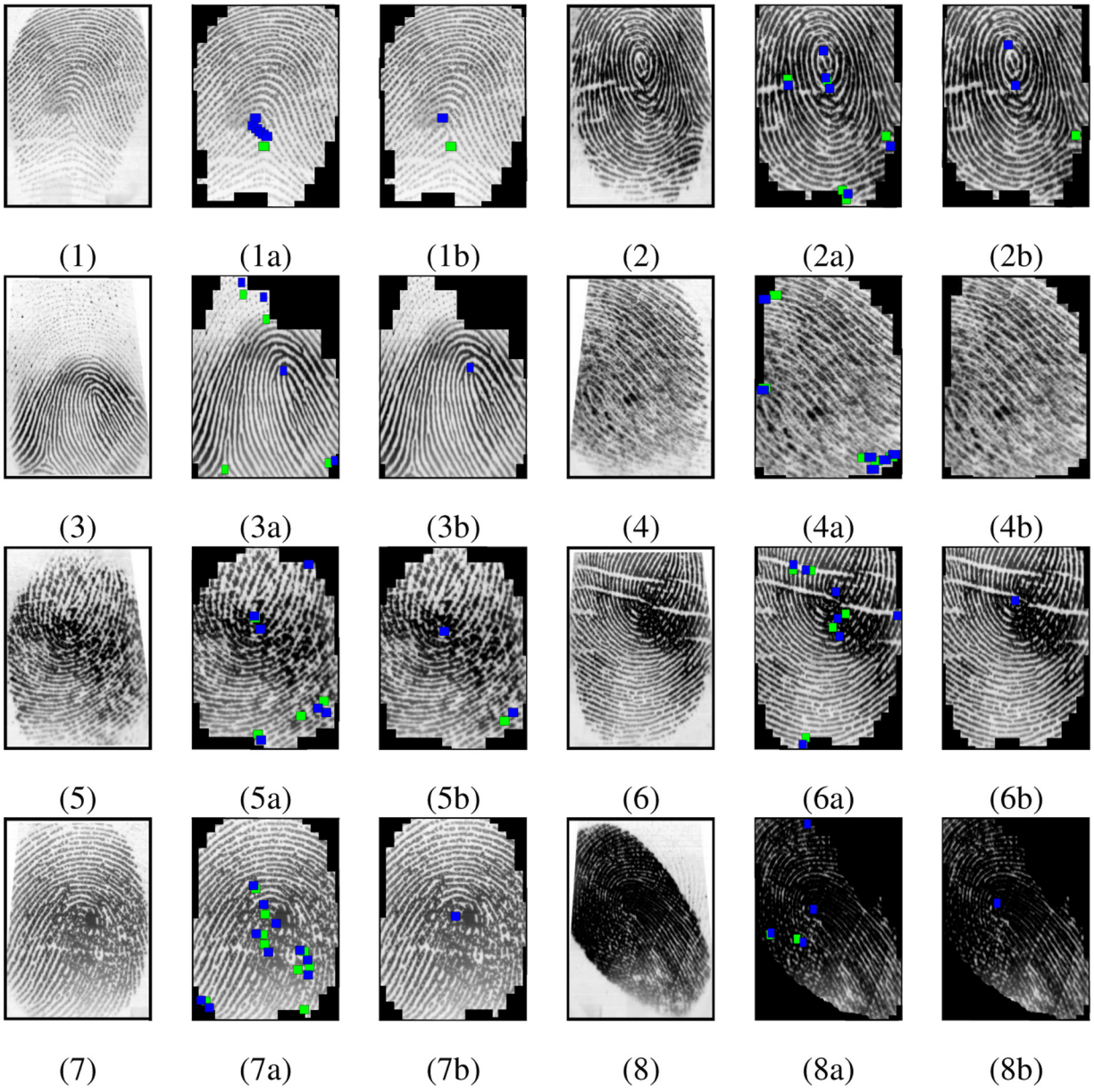

Fig. 6. Comparision of proposed SP detection with $PIndex_3$. Each row shows two examples and each example is a triplet having FP image and its results by using $PIndex_3$ and proposed SP detection.

the locations of SPs, if it is evaluated in a small neighborhood, thus results of $PIndex_3$ is used as a ground truth. But $PIndex_3$ cannot always extract the SPs correctly, 230 images having a correct SPs detection by $PIndex_3$ are used to calculate the localization error in terms of mean absolute difference (MAD). Localization errors for PIndex based algorithms are not calculated as $PIndex_3$ is used as ground truth and $PIndex_{16}$ uses a block-wise SP detection. Total SP detection accuracy by the proposed algorithm is found to be 96.7%. It can be inferred from Table 2 that:

1. Though $PIndex_3$ has better delta point detection accuracy than the proposed algorithm but it has high false delta point detections. It is evident that the proposed algorithm has a better detection accuracy along with less false detection and missed genuine SPs as compared to other algorithms.
2. Localization errors in [32] and $Proposed_A$ are large because the modeled OF reconstructs the actual OF near occluded SPs, but with some deviation from the actual OF. Also, localization errors in [32] are more as compared to $Proposed_A$ though both use the modeling of OF. But the deviation is much more in the case of [32] as compared to [6] near SPs.
3. SPs can be detected with a small localization error by the proposed algorithm. In case of core points, localization error between the proposed algorithm and $PIndex_3$ is given by 1 and 2 pixels along the x and y direction respectively. If post-processing refinement of location of the modified template matching is not applied in the proposed algorithm, then it has high localization errors. Therefore, refinement step is important for accurate localizations of SPs in the proposed algorithm.
4. If the extraction of missing delta points is not applied in the proposed algorithm, then there are 35 missed genuine cases, instead of 17 in the detection of delta points. Hence, detection of missing delta points plays a useful role in the proposed algorithm. An example is shown in Fig. 6(2b) which shows the recapturing of delta point.

Table 2
Comparative results of core and delta point detection on Biostar's database.

Algorithm	*TP*	Precision	Recall	*F*-measure	Accuracy	Error (x, y)[a]
Delta						
Plindex$_3$	197	0.75	0.93	0.71	93.36%	–
Plindex$_{16}$	182	0.84	0.86	0.72	86.26%	–
[19]	158	0.80	0.75	0.60	74.88%	(13,17)[a]
[20]	192	0.72	0.91	0.66	90.99%	(5.6)[a]
[32]	162	0.90	0.77	0.69	76.77%	(20,23)[a]
Proposed$_A$	176	0.96	0.83	0.80	83.41%	(12,14)[a]
Proposed$_B$	*194*	*0.97*	*0.92*	*0.89*	*91.94%*	*(6,8)*[a]
Core						
Plindex$_3$	542	0.85	0.93	0.80	93.93%	–
Plindex$_{16}$	504	0.92	0.87	0.81	87.35%	–
[19]	554	0.94	0.96	0.90	96.01%	(2,2)[a]
[20]	566	0.86	0.98	0.85	98.09%	(3,4)[a]
[32]	564	0.99	0.98	0.97	97.74%	(12,13)[a]
Proposed$_A$	568	0.99	0.98	0.98	98.44%	(8,10)[a]
Proposed$_B$	*568*	*0.99*	*0.98*	*0.98*	*98.44%*	*(1,2)*[a]

–: not required.
Italic is used to represent the algorithms, which are designed by the authors and are not existing in the literature.
[a] This is with respect to *Plindex*$_3$ and on a subset (200 images) of database.

Table 3
Comparative results of core and delta point detection on FVC 2004 DB1.

Algorithm	*TP*	Precision	Recall	*F*-measure
Delta				
[6]	374	0.78	0.69	0.54
[33]	379	0.96	0.70	0.68
Proposed	*473*	*0.94*	*0.87*	*0.82*
Core				
[6]	910	0.89	0.93	0.83
[33]	918	0.97	0.94	0.92
Proposed	*961*	*0.99*	*0.98*	*0.97*

Italic is used to represent the algorithms, which are designed by the authors and are not existing in the literature.

Table 4
Computational time for SP detection (in ms).

Algorithm	OF modeling	SP detection	Post-processing	Total time
[6]	2312	512	–	2824
[16]	2267	3391	–	5658
[32]	2108	1743	–	3851
Proposed	*2312*	*89*	*198*	*2599*

–: not used.
Italic is used to represent the algorithms, which are designed by the authors and are not existing in the literature.

Thus, it is evident from Table 2 that the proposed algorithm performs better than other existing algorithms. Reasons behind this are that the proposed algorithm gives: (i) accurate SP detection even if there is low quality is some area of FP; (ii) better than other OF modeling algorithms because it retrieve back the missed SPs; and (iii) precise locations of SPs. Similarly, the results of SP detection on FVC 2004 DB1 are presented in Table 3. OF modeling algorithms are compared with the proposed algorithm. It indicates that the proposed algorithm has less number of false detections and missed genuine which is mainly due to the use of post-processing algorithms.

Table 4 presents the computational time comparison between the proposed SP detection algorithm and various state of the art algorithms. All these algorithms are implemented on a desktop computer having Intel Pentium 8 processor, 2.8 GHz with 4 GB RAM and MATLAB 7.2. Algorithms in Table 4 take nearly same time for the OF modeling. But it can be seen that the time taken for SP detection from the modeled OF is least when the proposed algorithm is used. Reason behind is that the proposed algorithm exploits the smooth nature of modeled OF and prune most of the search space by using directional partitioning. In contrast, other algorithms apply Plindex based algorithm at each location which is time expensive. Algorithm [6] uses a less time consuming Plindex based algorithm but it still takes a large amount of time when compared with the proposed algorithm. Also, the time taken for post-processing is 198 ms which is negligible to the total time taken by the other algorithms. Some algorithms (like [20]) do not require OF modeling and thus, take less time for SP detection. But they have poor SP detection performance as shown in Table 3 and hence, avoided. Also, the proposed algorithm can classify a fingerprint in 59ms if modeled OF and SPs are given.

Sometime genuine SPs can be missed or spurious SPs can be detected by the proposed algorithm. This may occur when: (i) SPs may lie close to the border (for example, Fig. 6(3b) and (4b)); and/or (ii) bad quality in large area of the FP (for example, Fig. 6(5b)). Also OF modeling is a time consuming step in the proposed algorithm. But it is unavoidable as it reduces the number of missed genuine SPs and falsely detected SPs, as discussed in Section 4.1. A better OF modeling algorithm which can faithfully reconstruct OF near the borders or large bad quality regions can give better results. In addition, computation time for OF modeling can be somewhat reduced by using: (i) OF estimates at block-level; (ii) parallelization; and (iii) implement it in C.

4.3. Experiments III

Another way to show the effectiveness of the proposed algorithm is to classify the FPs by using the detected SPs. This set of experiments mainly focuses on this aspect. In order to test the classification results on Biostar's database, the database is divided into two parts namely, *GenuineDB* and *MissedDB*. If a FP image is completely captured, then it has the same number of core and delta points [16]. Such images are kept in *GenuineDB*. Often in some FP images, SPs (mostly deltas) are missed during the enrollment phase due to inappropriate capture or bad quality of FP. These images are kept in *MissedDB*. There are 179 and 321 FP images in *GenuineDB* and *MissedDB* database respectively. The classes used for classifying the images in *GenuineDB* are: arch, tented arch (T), left loop (LL), right loop (RL), double loop and whorl. On the other hand, classifications for the images in *MissedDB* are: left loop (LL), right loop (RL), double loop, whorl and *cannot determined* (C). There is a difference between classification classes used because classification of a tented arch pattern requires one core and one delta point

Table 5
Comparative results of FP classification on full database.

Algorithm	Accuracy on		
	Biostar's	FVC2004 DB1	FVC2004 DB2
$PIindex_3$	82.62%	87.56%	84.81%
$PIindex_{16}$	80.58%	84.39%	82.91%
[19]	88.32%	89.83%	86.43%
[20]	82.44%	83.83%	81.58%
[32]	96.90%	97.40%	95.22%
[36]	–	93.84%	94.89%
[35]	–	83.18%[a]	88.52%[a]
$Proposed_A$	97.10%	97.84%	95.96%
$Proposed_B$	*97.80%*	*98.12%*	*96.38%*

–: not reported.
Italic is used to represent the algorithms, which are designed by the authors and are not existing in the literature.
[a] Considering only four classes viz., A, W, LL and RL.

Table 6
Comparative results of FP classification on 400 FP images.

Algorithm	Accuracy on	
	FVC2004 DB1	FVC2004 DB2
[51]	91.92%	92.55%
[52]	95.17%	96.38%
Proposed	*98.76%*	*97.03%*

Italic is used to represent the algorithms, which are designed by the authors and are not existing in the literature.

while classification of a *cannot determined* pattern requires one core and no delta point. FP classification accuracies for *GenuineDB* and *MissedDB* databases are found out to be 99.44% and 96.88% respectively. Database *MissedDB* has low accuracy because some of its C patterns are classified as LL or RL. Accuracy of FP classification on full Biostar's database is 97.80%.

Other databases used for evaluation of the proposed algorithm by using FP classification are FVC2004 DB1 and DB2. All FPs are enhanced, segmented and binarized which are then classified manually. In FVC2004 DB1, there are 40 images consisting of *cannot determined* (C) type pattern. These C type cases exist when there is one core point and no delta point such that either point on the prominent line is less or pattern formed is ambiguous in terms of RL or LL, and mostly resembles T. There are 2 and 3 C type cases which are wrongly classified as LL and RL respectively in FVC2004 DB1. Along with this, there are 10 cases where the actual core point is missed out due to the bad quality or when a core point lies close to the border. Overall in FVC2004 DB1, there are 15 wrongly classified cases and its classification accuracy is 98.29%. Similarly, classification accuracy of FVC2004 DB2 database comes out to be 96.70% with 29 wrongly classified FP images.

FP classification accuracy of the proposed algorithm along with other known algorithms is shown in Table 5. To evaluate it, SPs extracted from different algorithms are used. From Table 5, it can be inferred that the SPs extracted by the proposed algorithm ($Proposed_B$) gives the best results. Also, it can be observed from Tables 2 and 5 that the effect of delta point on FP classification is very less. This is because of the fact that the proposed FP classification can correctly classify the pattern present in a FP image even in the absence of delta points. Another crucial observation inferred from Table 5 is that the proposed classification algorithm based on SP detection performs better than directional partitioning algorithm [36] and rule based algorithms [35]. The reasons behind the low performance of [35] are: (i) extracted SPs are not accurately localized; (ii) large number of genuine SPs are missed while the spurious ones are generated; and (iii) classification rules are not effective in the case of poor quality FP images. In contrast to the proposed system, only four classes, viz., A, W, LL and RL are considered in [35]. The class-wise accuracy of A, W, LL and RL on FVC2004 DB1 are 99.11%, 96.76%, 70.49% and 79.17% respectively, while that on FVC2004 DB1 are 93.06%, 97.5%, 77.16% and 87.84% respectively. Hence, it is apparent that the proposed system performs much better than system [35]. A support vector machine (SVM)-based classifier has been also used to evaluate the proposed algorithm. SVMs are efficient supervised classifiers that have been applied to several pattern recognition problems in signal and image processing [53,54]. Table 6 shows the performance evaluation of the proposed algorithm with SVM based classification systems [51,52]. Its SVM based systems make use of 400 FP images for testing while the remaining images are used for training. For consistency, the proposed algorithm has also used 400 images for evaluation. It can be observed from Table 6 that the proposed algorithm has better performance than the existing SVM based classification systems [51,52].

The performance of each FP class has been shown in Table 7. It illustrates that the proposed FP classification algorithm has correctly distinguish between TL, RL and LL cases which is usually the most problematic cases. In addition, patterns belonging to A class has been correctly classified due to PIindex based algorithm. As an epitome, the proposed algorithm is beneficial for FP classification

Table 7
Performance of the proposed FP classification.

FP pattern	Performance metric	Biostar's		FVC2004 DB1	FVC2004 DB2
		Genuine	Missed		
A	Precision	1	–	1	0.93
	Recall	1	–	1	1
TA	Precision	0.97	–	1	0.91
	Recall	1	–	1	0.78
LL	Precision	1	0.96	0.99	0.95
	Recall	0.98	1	1	0.99
RL	Precision	1	0.97	0.99	0.96
	Recall	1	1	1	0.99
DL	Precision	1	1	1	1
	Recall	1	0.8	1	1
W	Precision	1	0.95	1	1
	Recall	1	1	1	0.96
C	Precision	–	1	0.88	0.92
	Recall	–	0.5	1	0.71

–: not required.

because: (i) parameter tuning for classification parameters is automatically achieved in it; (ii) it works accurately even if delta points are missed which is usually the case; (iii) it gives an accurate FP classification for several classes; (iv) even for low quality FP it gives a correct classification due to the use of OF modeling; and (v) it can automatically reject the problematic cases of poor quality FP or partial FP capture by placing them in 'C' class. But it has shown some spurious classification because: (i) only one core is captured during FP acquisition, which consists of two cores (for example, Fig. 6(6b) and (5b)), (ii) missed core point (for example, Fig. 6(4b)), and (iii) false SPs detection due to which wrong case analysis is applied during classification (for example, Fig. 6(3b) and (5b)). Further, like any other FP classification algorithm, the proposed FP classification algorithm may fail for accidental pattern or altered FP pattern.

4.4. Experiments IV

This set of experiments is performed to analyze the robustness of the proposed algorithm against rotation and noise. To test the algorithm against rotation, 50 images (containing each FP pattern) of the database are rotated at 36 different angles starting from 0 to 360. It is observed that rotated images can still be classified correctly and their SPs locations are accurately determined. Further, the proposed algorithm can detect SPs even if the FP consists of bad quality (either dry or wet) in small areas or SPs are present in occluded areas. It can be visualized from Fig. 6(3b), (6b), (7b) and (8b). For a more reliable evaluation, 100 images from FVC2004 DB2 database are selected which have low quality values. To evaluate the quality, NFIQ [55] is used. It has been observed that the proposed algorithm reliably extracted the SPs which can be manually detected. Again to evaluate the noise invariance in a more difficult situation, 60 images are randomly taken from FVC2004 DB2 database and their SPs are occluded with a block of size 33×33. It is observed that SPs can still be detected with a mean absolute difference (MAD) in localization, given by 4 pixels along the x and y directions. The reason behind this is that the modeled OF can reconstruct OF near the occluded SPs, but with some deviation from actual OF.

5. Conclusions

This paper has proposed SP detection and FP classification algorithms. Large number of false SPs can be generated due to spurious OF which are obtained near bad quality areas of fingerprints. Such SPs have been eliminated by using OF modeling. SPs have been accurately extracted from modeled OF in a time efficient manner by using a novel hybrid algorithm which consists of a modified directional partitioning and a Plindex algorithm. Locations of detected SPs have been further refined. Sometimes delta points lie closed to the borders and are missed. These delta points have been retrieved for better SP detection accuracy. Extracted SPs have been used for FP classifications. But delta points may be missed out during FP capture which can restrict the use of SPs in FP classification. Hence, a new rule based algorithm has been proposed which can classify the FP even in the absence of delta points. It has classified the FPs into following classes: arch, tented arch, left loop, right loop, double loop and whorl. Arch type FPs along with correctly localized reference point are accurately determined by it. It has correctly distinguished the left loop FPs from the right loop type FPs and double loop FPs from whorl type FPs which are challenging task in FP classification. It does not need any parameter tuning. The proposed algorithm has been tested on three publicly available databases, which are Biostar's; FVC2004 DB1 and DB2 databases. Experimental results have revealed that it performs better than the various existing algorithms in terms of SP extraction and FP classification.

Acknowledgments

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

References

[1] E.R. Henry, Classification and Uses of Finger Prints, George Rutledge and Sons, 1900.
[2] V. Srinivasan, N. Murthy, Detection of singular points in fingerprint images, Pattern Recognit. 25 (2) (1992) 139–153.
[3] P. Gupta, P. Gupta, A dynamic slap fingerprint based verification system, in: International Conference on Intelligent Computing, 2014, pp. 812–818.
[4] P. Gupta, P. Gupta, An efficient slap fingerprint segmentation and hand classification algorithm, Neurocomputing 142 (2014) 464–477, http://dx.doi.org/10.1016/j.neucom.2014.03.049.
[5] T. Liu, G. Zhu, C. Zhang, P. Hao, Fingerprint indexing based on singular point correlation, in: IEEE International Conference on Image Processing, vol. 3, IEEE, 2005, pp. 293–296.
[6] Y. Wang, J. Hu, D. Phillips, A fingerprint orientation model based on 2D Fourier expansion (FOMFE) and its application to singular-point detection and fingerprint indexing, IEEE Trans. Pattern Anal. Mach. Intell. 29 (4) (2007) 573–585.
[7] R. Cappelli, A. Lumini, D. Maio, D. Maltoni, Fingerprint image reconstruction from standard templates, IEEE Trans. Pattern Anal. Mach. Intell. 29 (9) (2007) 1489–1503.
[8] S. Chikkerur, N. Ratha, Impact of singular point detection on fingerprint matching performance, in: Fourth IEEE Workshop on Automatic Identification Advanced Technologies, IEEE, 2005, pp. 207–212.
[9] G. Candela, P. Grother, C. Watson, R. Wilkinson, C. Wilson, PCASYS – A Pattern-Level Classification Automation System for Fingerprints, NIST Technical Report NISTIR 5647, 1995.
[10] Q. Zhang, H. Yan, Fingerprint classification based on extraction and analysis of singularities and pseudo ridges, Pattern Recognit. 37 (11) (2004) 2233–2243.
[11] K. Karu, A. Jain, Fingerprint classification, Pattern Recognit. 29 (3) (1996) 389–404.
[12] F. Alonso-Fernandez, J. Fierrez, J. Ortega-Garcia, J. Gonzalez-Rodriguez, H. Fronthaler, K. Kollreider, J. Bigun, A comparative study of fingerprint image-quality estimation methods, IEEE Trans. Inf. Forensics Secur. 2 (4) (2007) 734–743.
[13] M. Kawagoe, A. Tojo, Fingerprint pattern classification, Pattern Recognit. 17 (3) (1984) 295–303.
[14] Z. Han, C. Liu, Fingerprint classification based on statistical features and singular point information, in: Advances in Biometric Person Authentication, Springer, 2005, pp. 119–126.
[15] P. Ramo, M. Tico, V. Onnia, J. Saarinen, Optimized singular point detection algorithm for fingerprint images, in: International Conference on Image Processing, vol. 3, IEEE, 2001, pp. 242–245.
[16] J. Zhou, F. Chen, J. Gu, A novel algorithm for detecting singular points from fingerprint images, IEEE Trans. Pattern Anal. Mach. Intell. 31 (7) (2009) 1239–1250.
[17] C. Jin, H. Kim, Pixel-level singular point detection from multi-scale Gaussian filtered orientation field, Pattern Recognit. 43 (11) (2010) 3879–3890.
[18] J. Zhou, J. Gu, D. Zhang, Singular points analysis in fingerprints based on topological structure and orientation field, in: Advances in Biometrics, Springer, 2007, pp. 261–270.
[19] C. Huang, L. Liu, D. Hung, Fingerprint analysis and singular point detection, Pattern Recognit. Lett. 28 (15) (2007) 1937–1945.
[20] K. Nilsson, J. Bigun, Localization of corresponding points in fingerprints by complex filtering, Pattern Recognit. Lett. 24 (13) (2003) 2135–2144.
[21] X.J. Hang Yin, S. Sun, A novel algorithm for fingerprint singular points detection based on vector orthogonal theory, in: IEEE International Conference on Cloud Computing and Intelligence Systems, IEEE, 2011, pp. 486–489.
[22] A.K. Jain, S. Prabhakar, L. Hong, S. Pankanti, Filterbank-based fingerprint matching, IEEE Trans. Image Process. 9 (5) (2000) 846–859.
[23] J. Qi, M. Xie, A robust algorithm for fingerprint singular point detection and image reference direction determination based on the analysis of curvature map, in: IEEE Conference on Cybernetics and Intelligent Systems, IEEE, 2008, pp. 1051–1054.
[24] H. Chen, L. Pang, J. Liang, E. Liu, J. Tian, Fingerprint singular point detection based on multiple-scale orientation entropy, IEEE Signal Process. Lett. 18 (11) (2011) 679–682.
[25] X. Jiang, M. Liu, A.C. Kot, Reference point detection for fingerprint recognition, in: Proceedings of the 17th International Conference on Pattern Recognition, vol. 1, IEEE, 2004, pp. 540–543.
[26] X. Wang, J. Li, Y. Niu, Definition and extraction of stable points from fingerprint images, Pattern Recognit. 40 (6) (2007) 1804–1815.
[27] B. Sherlock, D. Monro, A model for interpreting fingerprint topology, Pattern Recognit. 26 (7) (1993) 1047–1055.
[28] N. Wu, J. Zhou, Model based algorithm for singular point detection from fingerprint images, in: International Conference on Image Processing, vol. 2, IEEE, 2004, pp. 885–888.

[29] L. Fan, S. Wang, H. Wang, T. Guo, Singular points detection based on zero-pole model in fingerprint images, IEEE Trans. Pattern Anal. Mach. Intell. 30 (6) (2008) 929–940.
[30] S. Huckemann, T. Hotz, A. Munk, Global models for the orientation field of fingerprints: an approach based on quadratic differentials, IEEE Trans. Pattern Anal. Mach. Intell. 30 (9) (2008) 1507–1519.
[31] S. Dass, Markov random field models for directional field and singularity extraction in fingerprint images, IEEE Trans. Image Process. 13 (10) (2004) 1358–1367.
[32] S. Ram, H. Bischof, J. Birchbauer, Modelling fingerprint ridge orientation using Legendre polynomials, Pattern Recognit. 43 (1) (2010) 342–357.
[33] Z. Hou, H.-K. Lam, W.-Y. Yau, Y. Wang, A variational formulation for fingerprint orientation modeling, Pattern Recognit. 45 (5) (2012) 1915–1926.
[34] B. Moayer, K.S. Fu, A syntactic approach to fingerprint pattern recognition, Pattern Recognit. 7 (1) (1975) 1–23.
[35] J.-M. Guo, Y.-F. Liu, J.-Y. Chang, J.-D. Lee, Fingerprint classification based on decision tree from singular points and orientation field, Expert Syst. Appl. 41 (2) (2014) 752–764.
[36] R. Cappelli, A. Lumini, D. Maio, D. Maltoni, Fingerprint classification by directional image partitioning, IEEE Trans. Pattern Anal. Mach. Intell. 21 (5) (1999) 402–421.
[37] K. Cao, L. Pang, J. Liang, J. Tian, Fingerprint classification by a hierarchical classifier, Pattern Recognit. 46 (12) (2013) 3186–3197.
[38] S. Chikkerur, A. Cartwright, V. Govindaraju, Fingerprint enhancement using stft analysis, Pattern Recognit. 40 (1) (2007) 198–211.
[39] H. Fronthaler, K. Kollreider, J. Bigun, J. Fierrez, F. Alonso-Fernandez, J. Ortega-Garcia, J. Gonzalez-Rodriguez, Fingerprint image-quality estimation and its application to multialgorithm verification, IEEE Trans. Inf. Forensics Secur. 3 (2) (2008) 331–338.
[40] C. Harris, M. Stephens, A combined corner and edge detector, in: Fourth Alvey Vision Conference, Manchester, UK, 1988, pp. 147–151.
[41] Z.M. Kovacs-Vajna, R. Rovatti, M. Frazzoni, Fingerprint ridge distance computation methodologies, Pattern Recognit. 33 (1) (2000) 69–80.
[42] C.-H. Park, J.-J. Lee, M.J. Smith, K.-H. Park, Singular point detection by shape analysis of directional fields in fingerprints, Pattern Recognit. 39 (5) (2006) 839–855.
[43] T.H. Le, H.T. Van, Fingerprint reference point detection for image retrieval based on symmetry and variation, Pattern Recognit. 45 (9) (2012) 3360–3372.
[44] P. Gupta, P. Gupta, Slap fingerprint segmentation, in: 2012 IEEE Fifth International Conference on Biometrics: Theory, Applications and Systems, IEEE, 2012, pp. 189–194.
[45] A. Tariq, M.U. Akram, S.A. Khan, An automated system for fingerprint classification using singular points for biometric security, in: International Conference for Internet Technology and Secured Transactions, IEEE, 2011, pp. 170–175.
[46] F. Magalhaes, H.P. Oliveira, A. Campilho, SPD 2010 – Fingerprint Singular Points Detection Competition Database, 2010 http://paginas.fe.up.pt/spd2010/
[47] FVC 2004, 2004 http://bias.csr.unibo.it/fvc2004/databases.asp
[48] H. Oliveira, F. Magalh aes, Two unconstrained biometric databases, in: Proceedings of the 9th International Conference on Image Analysis and Recognition, Volume Part II, Springer-Verlag, 2012, pp. 11–19.
[49] F. Yaghouby, A. Ayatollahi, R. Bahramali, M. Yaghouby, A.H. Alavi, Towards automatic detection of atrial fibrillation: a hybrid computational approach, Comput. Biol. Med. 40 (11) (2010) 919–930.
[50] F. Yaghouby, A. Ayatollahi, R. Bahramali, M. Yaghouby, Robust genetic programming-based detection of atrial fibrillation using RR intervals, Expert Syst. 29 (2) (2012) 183–199.
[51] Y. Yao, G.L. Marcialis, M. Pontil, P. Frasconi, F. Roli, Combining flat and structured representations for fingerprint classification with recursive neural networks and support vector machines, Pattern Recognit. 36 (2) (2003) 397–406.
[52] L. Ji, Z. Yi, SVM-based fingerprint classification using orientation field, in: Third International Conference on Natural Computation, IEEE, 2007, pp. 724–727.
[53] S. Chowdhury, J.K. Sing, D.K. Basu, M. Nasipuri, Face recognition by generalized two-dimensional FLD method and multi-class support vector machines, Appl. Soft Comput. 11 (7) (2011) 4282–4292.
[54] F. Yaghouby, A. Ayatollahi, An arrhythmia classification method based on selected features of heart rate variability signal and support vector machine-based classifier, in: World Congress on Medical Physics and Biomedical Engineering, Springer, 2010, pp. 1928–1931.
[55] E. Tabassi, C. Wilson, C. Watson, NIST Fingerprint Image Quality, NIST Res. Rep. NISTIR7151, 2004.

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

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