

## Review

# Long-bone fracture detection in digital X-ray images based on digital-geometric techniques<sup>☆</sup>



Oishila Bandyopadhyay<sup>a,\*</sup>, Arindam Biswas<sup>a</sup>, Bhargab B. Bhattacharya<sup>b</sup>

<sup>a</sup> Department of Information Technology, Indian Institute of Engineering Science and Technology, Shibpur, India

<sup>b</sup> Center for Soft Computing Research, Indian Statistical Institute, Kolkata, India

## ARTICLE INFO

## Article history:

Received 17 March 2015

Received in revised form

3 September 2015

Accepted 8 September 2015

## Keywords:

Medical imaging

Bone X-ray

Adaptive thresholding

Chain code

Segmentation

Digital straight line segment (DSS)

## ABSTRACT

Automated fracture detection is an essential part of a computer-aided tele-medicine system. In this paper, we have proposed a unified technique for the detection and evaluation of orthopaedic fractures in long-bone digital X-ray image. We have also developed a software tool that can be conveniently used by paramedics or specialist doctors. The proposed tool first segments the bone region of an input digital X-ray image from its surrounding flesh region and then generates the bone-contour using an adaptive thresholding approach. Next, it performs unsupervised correction of bone-contour discontinuities that might have been generated because of segmentation errors, and finally detects the presence of fracture in the bone. Moreover, the method can also localize the line-of-break for easy visualization of the fracture, identify its orientation, and assess the extent of damage in the bone. Several concepts from digital geometry such as relaxed straightness and concavity index are utilized to correct contour imperfections, and to detect fracture locations and type. Experiments on a database of several long-bone digital X-ray images show satisfactory results.

© 2015 Elsevier Ireland Ltd. All rights reserved.

## Contents

1. Introduction .....	3
2. Related work .....	3
3. Design considerations .....	4
4. Description of the system .....	6
4.1. Preprocessing of bone X-ray images .....	6
4.1.1. Bone region segmentation from an X-ray image .....	6
4.1.2. Contour generation using adaptive thresholding .....	7
4.2. Contour correction using relaxed digital straight line segments .....	8

<sup>☆</sup> A preliminary version of this paper appeared in the Proceedings, IWCIA, LNCS, vol. 8466, pp. 212–223, 2014 [1].

\* Corresponding author. Tel.: +91 9433887145.

E-mail addresses: [oishila@gmail.com](mailto:oishila@gmail.com) (O. Bandyopadhyay), [barindam@gmail.com](mailto:barindam@gmail.com) (A. Biswas), [bhargab.bhatta@gmail.com](mailto:bhargab.bhatta@gmail.com) (B.B. Bhattacharya).

<http://dx.doi.org/10.1016/j.cmpb.2015.09.013>

0169-2607/© 2015 Elsevier Ireland Ltd. All rights reserved.

4.3.	Fracture detection .....	9
4.3.1.	Fracture detection using concavity index.....	9
4.3.2.	Probable fracture point identification using RDSS.....	9
4.3.3.	Detection of fracture type .....	11
5.	Case studies and experimental experience .....	11
6.	Description of the software tool .....	13
7.	Conclusion .....	13
	Acknowledgements .....	13
	References .....	13

## 1. Introduction

Automated X-ray analysis is an integral part of telemedicine that is widely used in the rural areas where there is an acute shortage of medical practitioners. An analysis of an orthopaedic X-ray image can help the paramedical staff in remote areas to identify the severity of the condition and take proper action in a timely fashion. X-ray analysis can be done in two different phases: (i) segmentation of bone region from the surrounding flesh component followed by the extraction of bone-contour, and (ii) identification of the presence of fracture in the bone X-ray, if any.

Long-bone fracture is a very common ailment among senior citizens. They often suffer from bone injury because of osteoporosis, stress, or accidental fall. Automated fracture detection can assist doctors and paramedical staff to analyze the severity of the situation and to choose a proper course of action on urgent basis. Moreover, several computer-based training (CBT) programs in medical science focus on specialized fields like orthopaedics [22]. Also, simulators based on virtual reality [8,9,29] are now becoming increasingly popular among the medical residents and interns for the purpose of self-tutoring. A tool for automated fracture detection will be very useful in orthopaedic diagnosis, treatment, and management of patients as well.

The nature of bone fracture in different types of bone in human body widely differs from each other. Thus, designing a generalized algorithm for fracture identification in all types of bone is impracticable. In this paper, we have focused on the identification of fractures in a long-bone and proposed a new tool based on digital geometry [1,4]. The segmentation of a bone region of a digital X-ray image from its flesh part and generation of exact bone-contour play a crucial role in fracture detection. The proposed tool integrates an entropy-based segmentation method [3] with an adaptive thresholding-based contour tracing [2] to generate the bone-contour of an X-ray image. The algorithm traverses the bone-contour and rectifies the false discontinuities using certain new geometric techniques. This corrected bone-contour is then used for the detection of fracture locations using the concept of concavity index of a digital curve, and relaxed digital straight line segment (RDSS). When run on a large number of test cases, our tool is observed to have identified various types of fractures, successfully, in long-bone X-ray images.

## 2. Related work

The class of long-bones in human anatomy includes bones like humerus, radius, ulna, femur, tibia, and fibula. Several

researchers have proposed different algorithms for bone-contour generation and fracture detection in long-bones. The extraction of bone-contour plays an important role in X-ray, computerized tomography (CT), and in magnetic resonance (MR) based image analysis. An approach based on 2D-region growing can be applied for the segmentation and labeling of fractured or wrongly-joined bones from a CT image [26]. Other researchers have used 3D Log-Gabor filters to perform phase-symmetry-based surface extraction for segmenting the bone region from a three-dimensional (3D) ultrasound image [14]. Lin et al. have proposed automated teeth segmentation from dental radiographs using Otsu's thresholding, connected component analysis, snake boundary tracking, and morphological operations [18]. In a review paper, Ma et al. [20] have critically surveyed the comparative performance of several medical image segmentation methodologies, and shown their application for the extraction of organ and tissue regions that are present in a female pelvic cavity.

Because of the diversity of human anatomy and the inherent lack of sharp contrast in a bone X-ray image, computer-aided automated fracture detection turns out to be a challenging task. Various computational models have been proposed by earlier researchers for the automation of fracture detection. Tian et al. [28] proposed a method for detecting femur fracture based on the observed angle between the axis of the femur-neck and that of the shaft. Yap et al. [30] have also suggested a method for detecting femur-fracture that relies on the analysis of disruption found in the trabecular-texture patterns present in the femoral neck. Other methods based on deformable models such as active contour, were used for the detection of bone-boundary discontinuities observed in fractured-bone X-ray images [24]. The parametric cylinder model, Monte Carlo simulation, and active contour model are also used by researchers to develop computer-assisted diaphyseal long bone fracture detection techniques [31].

Among others, mathematical morphology has been widely used in bone-fracture detection [17]. A computer-aided system for long-bone fracture detection had been proposed by Donnelley et al. [10], where a scale-space approach is used for edge detection, Hough transform for parameter approximation, and gradient analysis for diathesis segmentation and fracture detection. A frequently used data mining technique called *classification*, had also been widely used to detect the presence of fracture. These systems combine various image features such shape, texture, and colour, and deploy machine learning algorithms to identify fractures [19].

The contour of a fractured-bone often reveals a significant change in curvature in the neighborhood of the injury. Hence, this local property of a fracture can be successfully utilized for its identification, and for the purpose of analysis in this

work, we will deploy some geometric concepts such as digital straight line segment (DSS), digital arc, and digital curve (DC). In the last few decades, many researchers in the digital-geometry community have extensively studied the underlying mathematical properties [4], [27]. Nguyen et al. have reported a technique for the decomposition of a digital curve into digital arcs and straight line segments, which also aids in locating dominant points in the arc-regions [23]. Fairly straight contours can be represented by a set of DSS alone.

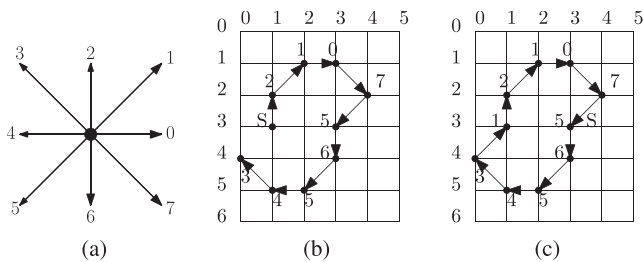
In previous work [3], an entropy-based thresholding approach was used for segmenting a bone X-ray image from its surrounding flesh region. Elementary analysis of fractured-bone contours and correction of discontinuities therein, also appeared in prior work [1,2].

The goal of this work is to integrate some of the earlier approaches with a new geometry-based technique of fracture identification and elimination of segmentation errors so as to build a complete automated analysis tool. The rest of the paper is organized as follows. Section 3 discusses various design considerations. Section 4 describes an overview of the flow and details the methodologies used in this work. Section 5 focuses on implementation issues, and reports experimental results on several test cases. Section 6 provides a demonstration of the proposed tool.

### 3. Design considerations

In the domain of digital geometry, the contour of a digital object (or a binary image) can be approximated with a sequence of DSS's [27], or with approximate digital straight line segments (ADSS) [4]. The underlying geometric properties of an image, captured by such representations, have been shown to be very powerful for their shape extraction and analysis. In the proposed work, we have introduced a new concept called *relaxed digital straight line segment* (RDSS) that is specially tailored for the purpose of fast bone-contour correction and fracture detection. A few definitions related to this work are given below.

**Chain code:** The direction around the border between pixels is encoded using chain code. If  $m(i, j)$  is a grid point, then the grid point  $(i', j')$  is a neighbor of  $m$ , provided that  $\max(|i - i'|, |j - j'|) = 1$ . The chain code [11] of  $m$  with respect to its neighboring grid point in  $P$  can have a value in 0, 1, 2, ..., 7 as shown in Fig. 1(a). The chain code of an open-ended digital curve shown in Fig. 1(b), is (1,3)(210756543) when traversed from the start-point  $S = (1, 3)$ ; Fig. 1(c) shows a close-ended digital curve,



**Fig. 1 – (a) Chain code for 8-neighbors around a grid point, (b) digital curve (1, 3)(210756543), (c) digital curve (3, 3)(6543121075).**

whose chain code is (3,3)(6543121075) with the start point  $S = (3, 3)$ .

**Digital straight line segment (DSS):** The properties of a DSS are defined in terms of its chain code as follows [27,4]:

- (F1) The runs in the chain code should have at most two directions, differing by  $45^\circ$ , and the run length in one of them must be 1.
- (F2) The runs can have only two lengths, which are consecutive integers.
- (F3) One of the run lengths can occur only once at a time.
- (F4) For the run length that occurs in runs, these runs can themselves have only two lengths, which are consecutive integers, and so on.

In order to reduce the complexity of contour decomposition, the concept of ADSS was introduced earlier, which is defined as follows [4]:

**Approximate digital straight line segment (ADSS):** ADSS has used (F1) along with certain modifications in (F2) but dropped condition (F3) and condition (F4) imposed on DSS. Such a modification of (F2) allows the run-length to vary by more than unity depending on the minimum run length. Thus, an ADSS will satisfy the following properties:

(F1') The runs can have at most two directions ( $n$  and  $s$ ), differing by  $45^\circ$ , and for one of these directions (singular element  $s$ ), the run length must be 1. (F2') The runs can have a range of possible lengths ( $[p, q]$ , excepting  $l$  and  $r$ ) such that,

$$(i) \quad q - p \leq d = \lfloor (p + 1)/2 \rfloor$$

(ii)  $(l - p), (r - p) \leq e = \lfloor (p + 1)/2 \rfloor$  where  $l$  is the length of the leftmost run of  $n$ ,  $r$  is the length of the rightmost run of  $n$ , and  $e$  represent the allowance of approximation.

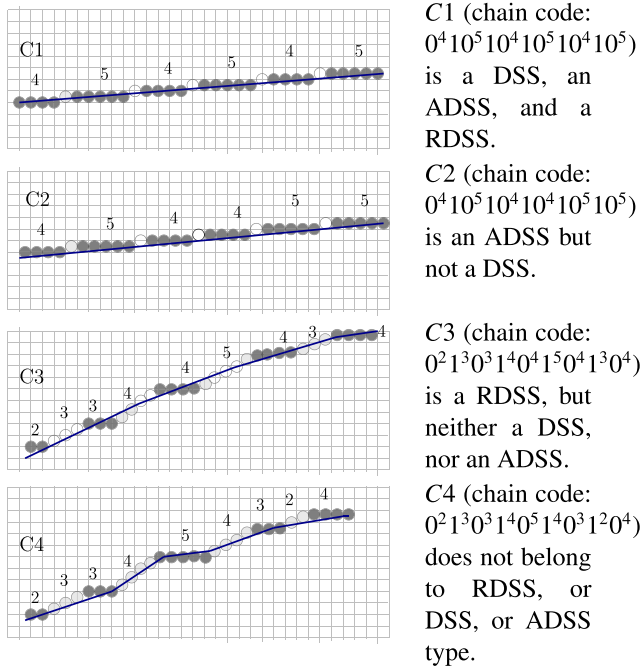
**Relaxed digital straight line segment (RDSS):** For the proposed work, we further relax the conditions on digital straightness to define a relaxed digital straight line segment (RDSS) by modifying the first two properties of ADSS as follows:

- (R1) The runs have at most two directions, differing by  $45^\circ$ .
- (R2) Both directions can have at-most three run lengths, which are consecutive integers.

Rule (R1) is a truncated version of the first property of both DSS and ADSS [4], [27], and Rule (R2) is a modified version of the second property (F2') of ADSS.

Property (R2) mandates that if a RDSS is generated with two directions  $C_1$  and  $C_2$  such that  $l_i$ 's represent run length of  $C_1$  and  $m_i$ 's represent run length of  $C_2$ , then  $|l_i - l_j| \leq 2$  for  $i \neq j$  and  $i, j \in \{1, 2, 3\}$  and  $|m_i - m_j| \leq 2$  for  $i \neq j$  and  $i, j \in \{1, 2, 3\}$ . This definition of RDSS allows us to approximate a visibly straight-looking digital curve having a small curvature with a single straight line segment.

While defining RDSS, we have retained one component of the first property of DSS (F1). Otherwise, if more than two directions are considered in a single line segment, then at least two among the three consecutive directions in the chain code differ by  $90^\circ$ . As a result, a curved line with a sharp curvature may also be falsely identified as a straight line thus destroying the basic shape of the image.



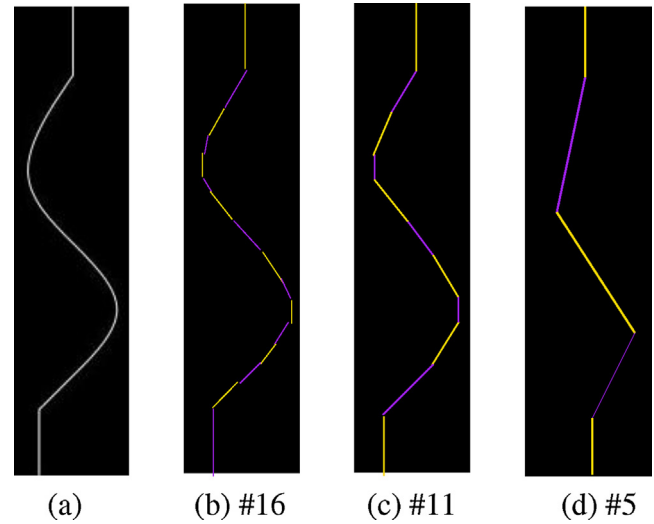
**Fig. 2 – Illustration of DSS, ADSS, and RDSS.**

RDSS relaxes the constraint on single occurrence of runs. The major advantages of this relaxation includes:

- approximation of a curved line with a small curvature as a straight line segment.
- avoidance of stringent constraints required by DSS and ADSS so as to identify small curves as a single straight line segment.

The second property of RDSS is included so as to restrict the number of run lengths in each direction. If the run-lengths of the chain code are consecutive integers, then a short curved segment therein can also be approximated with a single RDSS.

Fig. 2 shows various examples of DSS, ADSS, and RDSS with chain-code directions ‘0’ and ‘1’. In this example, C1 satisfies all properties of DSS, ADSS, and RDSS. In C2, both run lengths have multiple occurrences. Thus, the property of single occurrence of one run length (3rd property of DSS [27]), is not satisfied. Hence, C2 cannot be treated as a DSS; however, by definition, it will be recognized as an ADSS [4]. The digital curve C3 has multiple occurrences of both directions. Therefore, it violates the basic property of both DSS and ADSS. Hence, C3 will neither be recognized as a DSS nor as an ADSS. Curve C3 has three run lengths for each of the two directions which are consecutive integers (‘2’, ‘3’, ‘4’ and ‘3’, ‘4’, ‘5’, respectively). Hence, C3 satisfies both properties of RDSS. The digital curve C4 also has multiple occurrences of run-lengths with each direction. However, the run lengths in direction ‘0’ are not consecutive integers and the maximum difference between two run lengths of ‘0’ is 3. Thus, C4 violates the first condition of DSS and ADSS; furthermore, it violates the second condition (R2) of RDSS as well. Hence, it does not fulfil the necessary conditions for being categorized as DSS, ADSS, or RDSS.



**Fig. 3 – (a) A digital curve and the corresponding number of (b) DSSs, (c) ADSSs, and (d) RDSSs.**

As mentioned earlier, RDSS allows us to represent a single-pixel curve having a small curvature with a single straight line segment. For example, in Fig. 3(d), the curved line (shown in Fig. 3(a)) can be approximated with five consecutive RDSS’s. Since the conditions for being a DSS or an ADSS are tighter than those needed by a RDSS, the decomposition of a digital curve into DSS or ADSS components will, in general, produce a larger number of straight line segments compared to RDSS-based decomposition (as shown in Fig. 3(b) and (c)). An increase in the number of straight line segments renders the approximation of a curve with straight segments overly accurate than what is needed for solving our problem; further, such instances become more complex and costly to analyze. Hence, in the proposed algorithm, we decompose a bone-contour into a number of RDSS’s, and this strategy helps us to remove most of the false discontinuities that may arise because of segmentation errors.

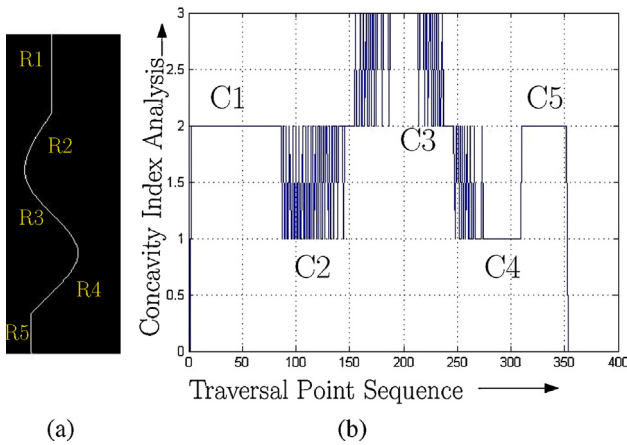
In our analysis, we have used another important geometric property of a digital curve called concavity index, as defined below.

**Concavity index ( $\alpha$ ).** While traversing the contour of an image, each pixel point  $m_i$  on the contour is assigned a concavity index, ( $\alpha_i$ ) where  $\alpha_0$  and  $\alpha_1$  are initialized to 0 and 1, respectively. To obtain  $\alpha_{i+1}$ ,  $\alpha_i$  is incremented (decremented) by the difference of the directions,  $d_i$  and  $d_{i+1}$ , if the contour is traversed in clockwise (counter-clockwise) direction from  $m_i$  to  $m_{i+1}$ , where  $d_i$  is the incident direction at  $m_i$ .

Note that if the curve-traversal advances in the clockwise (anti-clockwise) direction, the concavity index increases (decreases). In the latter case, it may assume a negative value. On the other hand, if the traversal continues in the same direction, the concavity index remains unchanged. Thus, a large variation of  $\alpha$  meaning some sudden and unexpected change in the directionality, indicates a possible location of a fracture.

Fig. 4(b) shows the plot for concavity index as obtained by traversing the curve of Fig. 4(a) from the top side. An anti-clockwise movement on segments R2 and R4 leads to low-concavity values in C2 and C4, whereas, a clockwise





**Fig. 4 – Concavity index (a) a line with straight and curved region, (b) concavity indices of different regions of the curve shown in (a).**

movement on segment R3 shows a high-concavity as in C3. No change of directionality on the curve is indicated by C1 and C5.

#### 4. Description of the system

An overview of different stages of the proposed method is depicted in Fig. 5. In this block diagram, the preprocessing phase is represented by Stage (b). Stage (c) and Stage (d) represent the blocks for contour correction and fracture identification, respectively. Once the presence of a fracture is confirmed, its exact location and fracture points are identified on the bone-contour. The main contribution of this work includes (i) identification of probable fracture points (Stage (e), Stage (f), and Stage (g)), (ii) detection of fracture locations (Stage (h)), and (iii) categorizing the nature of fracture (Stage (i)).

##### 4.1. Preprocessing of bone X-ray images

In an X-ray image, the bone parts appear along with the surrounding tissues or muscles (i.e., flesh). Hence, segmentation of the bone region and generation of exact bone-contour are needed for automated fracture detection. The proposed tool

(Fig. 5) has two preprocessing steps: (a) segmentation of bone region from surrounding flesh, and (b) extraction of contour from the segmented bone image.

##### 4.1.1. Bone region segmentation from an X-ray image

Extraction of bone region from overlapping muscles and tissue shadow is a challenging task in digital X-ray image segmentation. Each X-ray image has three distinct regions - a black background region, a low-intensity flesh region, and a high-intensity bone region. We have observed that traditional thresholding-based approaches such as 'OTSU' [25] often result in over-segmentation or under-segmentation for most of the X-ray images. Also, an edge-detection approach such as 'Canny' method [7] is observed to generate multiple false edges because of the presence of flesh-shadow surrounding the bone components. Another widely used segmentation method is k-means clustering [15,16] that segments the region of interest (ROI) more efficiently than classical segmentation approaches such as 'Canny' or 'OTSU'. We have evaluated k-means clustering and an entropy-based method for bone-region segmentation.

**4.1.1.1. Segmentation of bone components using k-means clustering.** A k-means clustering based segmentation approach can be used to compute pixel-cluster memberships for segmenting the image [15,16]. This method classifies the input data points into multiple classes on the basis of their inherent distances from each other. The points are clustered around centroids  $\mu_i \forall i = 1, 2, \dots, k$ , which are obtained by minimizing the objective function, known as squared error function V:

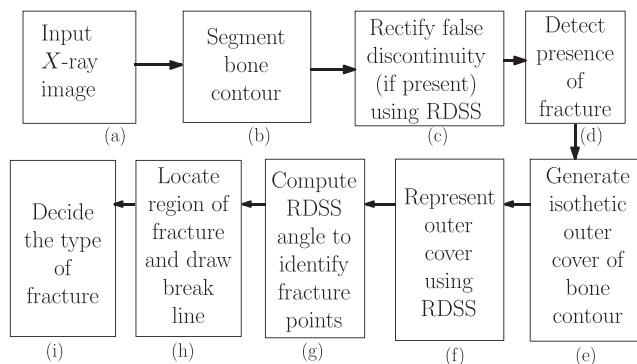
$$V = \sum_{i=1}^k \sum_{x_j \in S_i} (x_j - \mu_i)^2 \quad (1)$$

where there are  $k$  clusters  $S_i, i = 1, 2, \dots, k$ , and  $\mu_i$  is the centroid or mean point of all points  $x_j \in S_i$ . Based on the intensity distribution in the input image, segmentation is performed using k-means clustering. In this approach, the centroid is initialized with  $k$  random intensities, and pixels are clustered according to their distances from the centroid intensity.

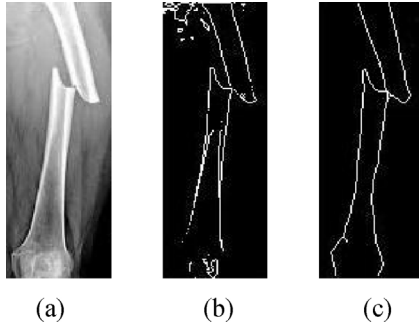
It has been observed that k-means clustering may produce over-segmented bone regions in many X-ray images. Thus, we have evaluated another method based on entropy-standard deviation [2,3] for bone-region segmentation.

##### 4.1.1.2. Segmentation of bone using an entropy-based approach.

Entropy-based segmentation techniques [2,3] can also be used for bone-image segmentation. In this method, local-entropy represents the grayscale intensity variation in the neighborhood of a pixel. As the uncertainty associated with the belongingness of a pixel in the flesh-to-bone transition region is much higher compared to smooth regions, the local entropy value of a pixel increases at the transition region. In this segmentation approach, we compute the local entropy for each pixel (for a  $9 \times 9$  window) of the input image and multiply it with local standard deviation value for that pixel (for the same window size). The image generated from the product matrix shows relatively high intensity values at bone-boundary regions (as shown in Fig. 9(b) Fig. 6(b) shows the



**Fig. 5 – Block diagram of the proposed method for automated fracture detection.**



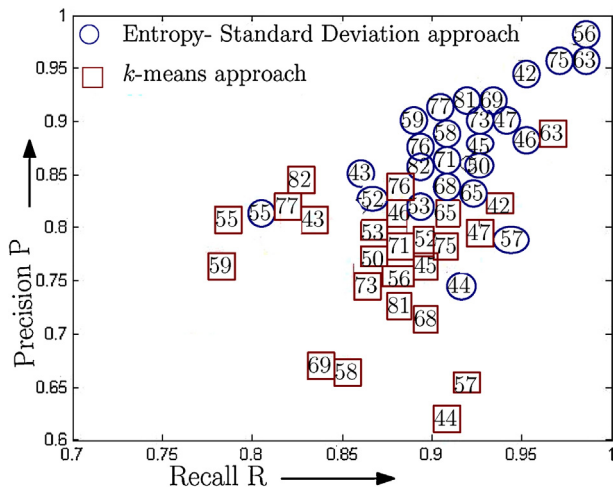
**Fig. 6 – Segmentation and contour generation (a) Input digital X-ray image, (b) bone-contour using  $k$ -means segmentation, (c) counter using entropy-standard deviation based segmentation.**

segmentation of input X-ray image obtained by the  $k$ -means segmentation, and Fig. 6(c) shows the contour of the segmented image obtained by the entropy-based segmentation and adaptive thresholding approach.

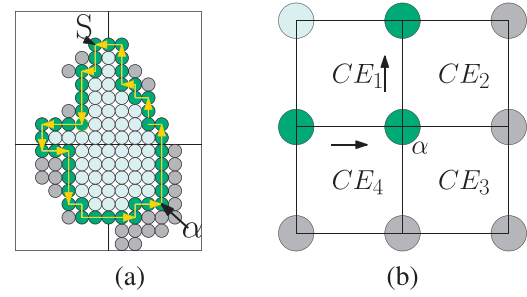
We have compared the entropy-based segmentation approach [2] with  $k$ -means clustering [16] using the precision-recall metric [21]. It is observed that the entropy-based method exhibits higher precision-recall values compared to the  $k$ -means approach for all test-cases from the medical image database compiled by us (<http://oldwww.iests.ac.in/component/content/article/155-itcategory/3282-medical-image-database>) (shown in Fig. 7). Thus, we have adopted the segmentation method based on entropy-standard deviation for bone-contour generation.

#### 4.1.2. Contour generation using adaptive thresholding

The proposed approach generates the bone-contour from a segmented bone image (entropy-standard-deviation image) using adaptive thresholding [6]. The segmented bone image  $J$  is traversed through every pixel  $\alpha$ , and a small window around



**Fig. 7 – Precision-recall curve (image number represents the respective image Id in the medical image database (<http://oldwww.iests.ac.in/component/content/article/155-itcategory/3282-medical-image-database>)).**



**Fig. 8 – (a) Object-contour pixel selection, (b) 8-neighbours of pixel  $\alpha$ .**

it with 8-neighboring pixels are examined (see Fig. 8(b)). The window is divided into four cells, top-left ( $CE_1$ ), top-right ( $CE_2$ ), bottom-left ( $CE_3$ ) and bottom-right ( $CE_4$ ). All these four cells are incident on  $\alpha$ . The contour traversal algorithm analyses the intensity values of neighboring pixels and selects the next pixel position whenever it identifies a pixel with an intensity value higher than the adaptive-threshold value of the present pixel (Fig. 8(a)). If the pixel is visited for the first time, it adds the current pixel in the visited list and decides the direction for the next move [2].

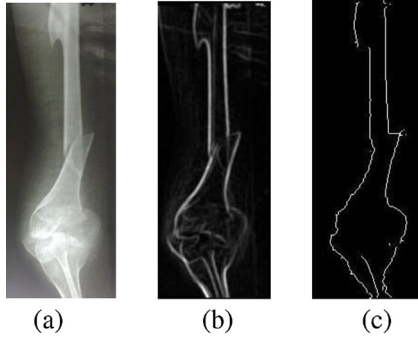
**4.1.2.1. Threshold computation in adaptive thresholding.** Let  $\max_i$  and  $\min_i$  denote the respective maximum and minimum gray level values in  $CE_i$ . The global maximum ( $\max_g$ ), is defined as the largest among all  $\max_i$ 's for  $i = 1, 2, 3, 4$ , corresponding to the four cells incident on  $\alpha$ . Similarly, the global minimum ( $\min_g$ ) is defined as the smallest among all  $\min_i$ 's for  $i = 1, 2, 3, 4$ . Based on  $\max_g$  and  $\min_g$ , we define the local threshold value as  $\tau_{loc} = \frac{1}{2} (\max_g - \min_g)$ . The adaptive value of the threshold  $\tau_{adpt}$  is now computed as:

$$\tau_{adpt} = \gamma \tau_{loc} + (1 - \gamma) (\tau_{loc} + \tau_{prev}) \quad (2)$$

where  $\gamma \in [0,1]$  represents the smoothing factor for the exponential moving average, and  $\tau_{prev}$  represents the adaptive threshold calculated for the previous point lying on the boundary of the cellular region. For computational simplicity, we consider  $\gamma = \frac{1}{2}$  [6]. Hence, the computation of  $\tau_{adpt}$  can be effected by a right-shift operation only. Once the adaptive threshold  $\tau_{adpt}$  for a given common point is computed, a cell  $C_i$  is said to contain a portion of bone boundary, provided  $\max_g - \min_g \geq \tau_{adpt}$ .

**4.1.2.2. Selection of the initial threshold value.** The initial value ( $\tau_0$ ) is selected from the histogram of the intermediate entropy-standard-deviation image. As the bone boundary consists of a few pixels of an intermediate image, we consider the intensity value that corresponds to 99.95 percentile of the intermediate image as  $\tau_0$ .

Based on the computed adaptive threshold value, the algorithm decides the direction of next movement. Fig. 8(b) shows an object with high- and low-intensity pixels. Green pixels belonging to the contour are identified using the adaptive thresholding approach. During each traversal, the object remains on the right side (left side) of the contour for a



**Fig. 9 – Segmentation and contour generation (a) Input digital X-ray image, (b) Entropy-standard deviation image of (a), (c) bone-contour of (a).**

counter-clockwise (clock-wise) move. Fig. 9(c) shows the final contour of the bone generated from an input X-ray image.

#### 4.2. Contour correction using relaxed digital straight line segments

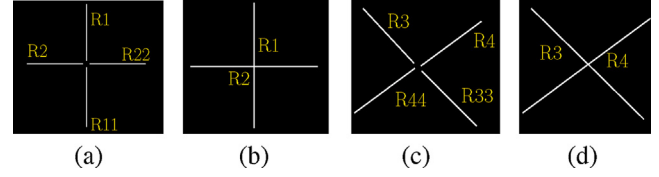
A bone fracture may cause a disconnected or an uneven bone-contour. Hence, any discontinuity in the contour that appears due to segmentation or contour generation error (see region ‘A’ of Fig. 11(a)), can mislead the fracture-detection process. In order to overcome this problem, we have proposed a novel approach based on RDSS.

In the proposed method, the bone-contour generated using adaptive thresholding approach is traversed and the corresponding chain code is constructed. Next, the code is analyzed to approximate the underlying curve with a sequence of RDSS's. Any discontinuity in a long-bone contour should appear as two disconnected RDSS's, aligned in the same direction. Two segments of a contour may not be traversed in one pass because of discontinuity. Hence, the end-points of the two fragmented contours can have the same or complemented direction in the corresponding chain code. Either  $R_1$  or  $R_2$  can be extrapolated to generate a new RDSS ( $R_3$ , with chain code  $C_{31}$  and  $C_{32}$ ) that can correct the contour discontinuity. If RDSS  $R_1$  has chain code  $C_{11}$  and  $C_{12}$ , and  $R_2$  has chain code  $C_{21}$  and  $C_{22}$ , then the extrapolated RDSS  $R_3$  with chain code  $C_{31}$  and  $C_{32}$  should satisfy the following relation:

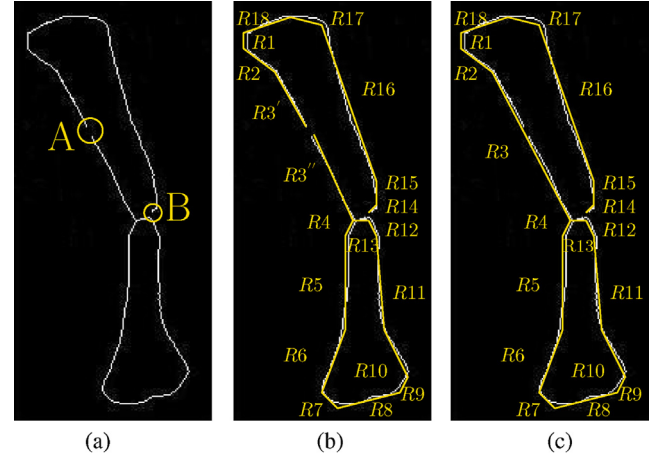
$$R_3 = R_1 \parallel R_2 \quad (3)$$

where  $\{C_{21}, C_{22}\} \in \{C_{11}, C_{12}, C'_{11}, C'_{12}\}$ , and  $\{C_{31}, C_{32}\} \in \{C_{11}, C_{12}\} \cup \{C_{21}, C_{22}\}$ ;  $C'_i = (C_i + 4) \bmod 8$ , and ‘ $\parallel$ ’ denotes the merging of  $R_1$  and  $R_2$  after extrapolation of any one of them.

The proposed algorithm searches for all such closely positioned RDSS pairs that cover the bone-contour with its complementary chain code string. For example, a RDSS with a chain code consisting of  $\{6, 7\}$  can be paired with another having the code  $\{6, 7\}$  or  $\{2, 3\}$  since the RDSS covering the two line segments across the break can be traversed in same or opposite direction, with the pixels of one end of the line segments are lying in 16 neighborhood of each other (see RDSS  $R_2$  and  $R_{22}$  of Fig. 10(a)). After finding such RDSS pairs, if they are sufficiently close enough, any one of these two respective



**Fig. 10 – Contour correction (a) Input plus image (incomplete), (b) corrected plus image using RDSS, (c) input cross image (incomplete), (d) corrected cross image using RDSS.**



**Fig. 11 – Contour correction (a) bone-contour with false discontinuity arises at ‘A’, (b) contour with RDSS, (c) corrected contour with RDSS.**

contour segments is extended in the direction of traversal to connect them into a single contour segment, which in turn, can be represented by a single RDSS.

In Fig. 10(a), the two lines  $R_1$  and  $R_2$  that intersect like a ‘+’ symbol, are disconnected at the middle intersection point. So four RDSS pairs ( $R_1$ ,  $R_{11}$ ,  $R_2$  and  $R_{22}$ ) will be generated during the traversal of the contour. The proposed RDSS matching algorithm rectifies the contour by connecting each RDSS with its respective counterpart (see Fig. 10(b)). Similar effects will be observed when lines  $R_3$  and  $R_4$  intersect like a ‘x’ symbol, but do not touch at the middle intersection point (see Fig. 10(c)). Here, RDSS  $R_1$  and  $R_{11}$  have the same chain code (chain code value 6) as they have been traversed in the same direction, whereas, RDSS  $R_4$  and  $R_{44}$  have the complementary chain code (chain code 1 and 5, respectively). The proposed algorithm uses Relation (3) to connect them into a single RDSS. Finally, the corrected contour lines  $R_1$ ,  $R_2$ , and  $R_3$ ,  $R_4$  are generated (Fig. 10(b) and (d)).

A discontinuity appearing in a contour may also correspond to a fracture, which will indeed show up as two different RDSS's. Any such discontinuity in the bone-contour caused by a fracture (see region ‘B’ of Fig. 11(a)) will have contour segments that cannot be aligned as a single straight line segment. So the chain-code-string of consecutive RDSS pairs will not satisfy the relation given in Eq. (3) ( $R_{13}$  and  $R_{14}$  of Fig. 11(c)). Therefore, the sequence of RDSS's that approximate the contour cannot be merged into fewer RDSS's. On the other hand, the contour discontinuities caused by segmentation

or contour-generation errors can be corrected by projecting two neighboring RDSS's towards each other and merging them.

Approximation of bone-contours can also be performed using DSS or ADSS; however, the number of segments required to cover the contour would become higher because in a single-pixel wide long-bone image, the contour usually changes the direction at an interval of every 5 to 10 pixels. Thus, the use of RDSS helps to reduce the number of approximating straight line segments, and to rectify the false breaks in the contour, while reporting the correct fracture locations.

#### 4.3. Fracture detection

The proposed technique identifies the presence of fracture by analyzing the changes in concavity-indices of the pixels on the bone-contour. Once the presence of fracture is confirmed, the break-points and the region of break are identified. Finally, the line-of-break is reconstructed by joining the break-points to mark the region of fracture.

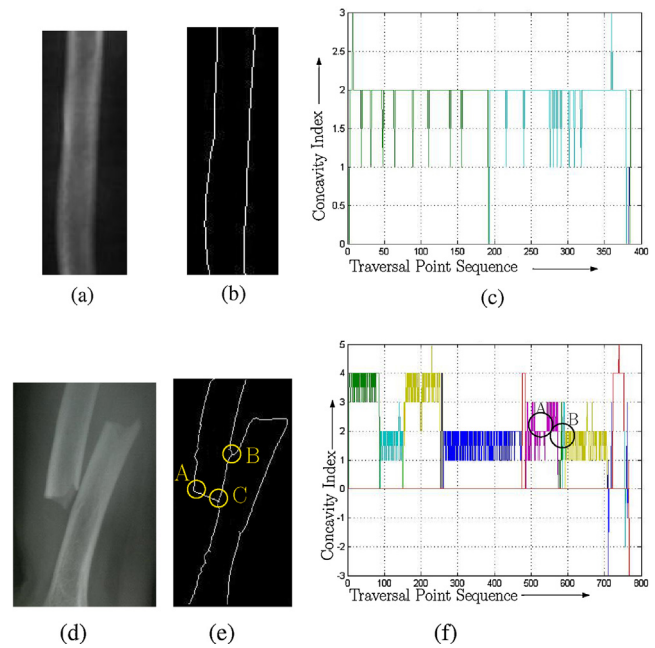
##### 4.3.1. Fracture detection using concavity index

The boundary of a fracture-region in a long-bone appears as an irregular curve on its contour. The chain-code directions are observed to change sharply when the contour around these regions, is traversed pixel-by-pixel. Note that the concavity index remains unchanged over a long straight contour; however, it increases (decreases) if the curve turns around in clock-wise (anti-clockwise) direction. This property of concavity index is utilized here to detect the presence of fracture in a long-bone.

Experimental evidence shows that during contour traversal in the neighborhood of a fractured region, the chain code changes rapidly in both clock-wise and anti-clockwise directions within a short range of 5–10 pixels. As a consequence, the concavity index changes rapidly over a short span during pixel traversal. A plot of concavity-indices thus shows sharp rise and fall when the contour around a fractured region is traversed. Fig. 12(c) shows the nature of variation of concavity index during the traversal of a healthy long-bone. Note that the direction of traversal remains unchanged from top-left to bottom-right and from top-right to bottom-left. Thus, the concavity indices remains steady throughout the traversal. On the other hand, contour traversal around a fractured region results in frequent increments and decrements in concavity-indices. Region 'A' and Region 'B' of Fig. 12(f) show the rise and fall of concavity-indices within a small range of 5–10 pixels. These regions correspond to fractured-locations 'A', 'B' and 'C' shown in Fig. 12(e) respectively. However, a sharp variation of concavity-index may also be falsely observed in certain instances. This situation may occur when a curved contour of a normal bone is traversed. Such cases of false-acceptance can be minimized by detecting the line-of-break in the bone-contour.

##### 4.3.2. Probable fracture point identification using RDSS

In order to generate the outer isothetic cover of a bone-contour, a prior algorithm [5] is used. Next, we decompose the outer cover into a sequence of RDSS components, and compute the angle between each pair of consecutive components.



**Fig. 12 – Concavity curve (a) Input X-ray image of humerus, (b) bone-contour of (a), (c) concavity Index plot of (b), (d) input X-ray image of fractured humerus, (e) bone-contour of (d), and (f) concavity Index plot of (e).**

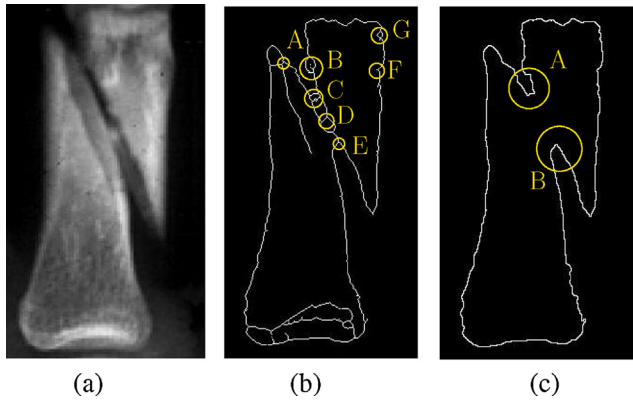
The probable fracture points are then identified on the basis of these angles.

4.3.2.1. *Construction of isothetic cover [5].* Given a specific pixel-grid size that is used for digitization, an isothetic (axes-parallel) cover of a digital object is a contour that tightly encloses the object [5]. This concept is utilized here for efficient localization of fracture-regions. The contour around a fracture appears with an irregular boundary. However, the contours for normal bone-joints also show certain amount of irregularity. While concavity analysis of bone-contour aids in identifying fracture-regions, it often becomes difficult to localize the line-of-break correctly. Since a fracture in a long-bone contour shows up symmetric abnormality on the contour along the line-of-fracture, the analysis of an isothetic cover that tightly encloses the fractured-region helps us to locate the proper position of line-of-fracture in the bone.

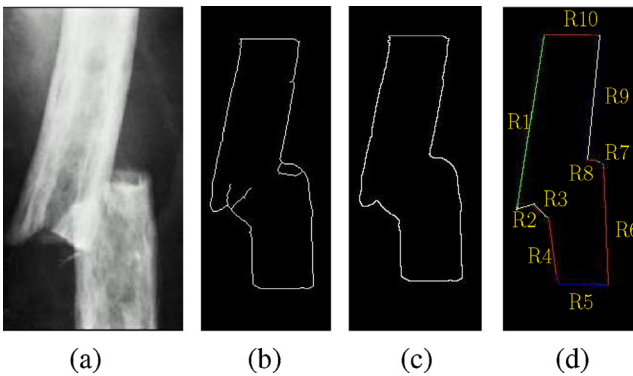
Fig. 13 (b) shows the fractured-bone contour with the probable regions of abnormalities identified during contour traversal ('A', 'B', 'C', 'D', 'E', 'F', and 'G'). Fig. 13 (c) shows the isothetic cover of the bone-contour. Region 'A' and Region 'B' of the isothetic cover represent the symmetric-abnormality regions. The line-of-break passes through these regions. Note that a traversal along the periphery of the isothetic cover aids in identifying the line-of-break correctly, and at the same time, it minimizes the effects of false-positive breaks that were identified during concavity analysis.

4.3.2.2. *RDSS representation of isothetic cover.* The symmetric-abnormality locations in a fractured-bone contour can be identified by computing the angle between consecutive RDSS's





**Fig. 13 – Outer cover: (a) input X-ray image of fractured hand, (b) bone-contour with probable fracture points, (c) isothetic outer cover of (b) with probable fracture points.**



**Fig. 14 – Outer cover: (a) input X-ray image of fractured hand, (b) bone-contour of (a), (c) isothetic outer cover of (b), (d) RDSS representation of the boundary of the outer cover (c).**

in these regions. In order to estimate the angles at all curved-contour regions the proposed algorithm checks the chain-code around the boundary of the isothetic cover of the contour, and generates the sequence of RDSS's that approximates the boundary. Two consecutive RDSS's with different chain-codes, indicate the presence of some abnormal curvature in the corresponding region.

Fig. 14 (d) shows the RDSS-approximation for the boundary of the isothetic cover of the contour shown in Fig. 14(a). In this figure, two consecutive RDSS's have been colored differently. The change in chain-code between the RDSS-pair (R1, R2), and (R8, R9), indicates the presence of certain abnormality in the contour.

**4.3.2.3. Angle between RDSS.** The proposed algorithm identifies the consecutive RDSS-pairs that have different chain-codes and computes the internal angle between them (excluding the top and bottom corners). A training set of thirty X-ray images (including both fractured and normal healthy long-bones) are used in our experiments. Table 1 shows the typical values of the angle subtended by two consecutive RDSS components. The X-ray images used in Table 1 are taken from the <http://www.becs.ac.in/component/content/>

**Table 1 – Angle between consecutive RDSS pairs for the training set.**

Image no. (MD no.)	Image type	RDSS $\angle$
1 (42)	Fractured	75°, 66°
2 (43)	Fractured	45°, 20°
3 (44)	Fractured	56°, 33°
4 (46)	Fractured	58°, 27°
5 (47)	Fractured	81°, 45°
6 (50)	Fractured	70°, 71°
7 (52)	Fractured	82°, 68°
8 (53)	Fractured	77°, 42°
9 (56)	Fractured	63°, 60°
10 (55)	Fractured	40°, 26°
11 (57)	Fractured	45°, 27°
12 (10)	Healthy	7°, 4°
13 (5)	Healthy	6°, 5°
14 (18)	Healthy	4°, 3°
15 (12)	Healthy	9°, 4°
16 (59)	Fractured	45°, 27°
17 (63)	Fractured	45°, 45°
18 (65)	Fractured	74°, 64°
19 (71)	Fractured	68°, 55°
20 (73)	Fractured	72°, 21°
21 (75)	Fractured	45°, 45°
22 (77)	Fractured	45°, 45°
23 (81)	Fractured	72°, 50°
24 (82)	Fractured	80°, 45°
25 (76)	Fractured	76°, 71°
26 (25)	Healthy	5°, 4°
27 (23)	Healthy	6°, 3°
28 (19)	Healthy	7°, 5°
29 (22)	Healthy	8°, 3°
30 (11)	Healthy	6°, 4°

[article/155-itcategory/3282-medical-image-database](http://www.becs.ac.in/component/content/article/155-itcategory/3282-medical-image-database) medical image database (MD) (<http://www.becs.ac.in/component/content/article/155-itcategory/3282-medical-image-database>).

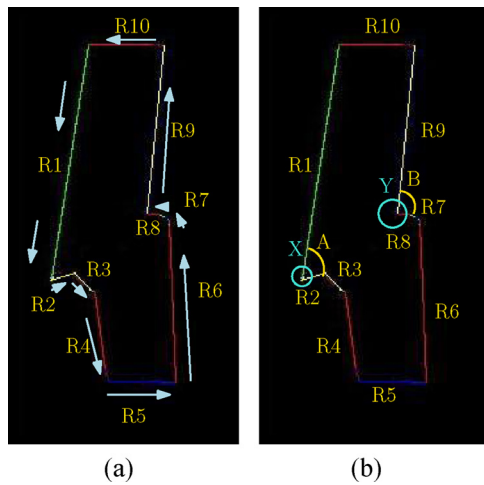
We observe that all angles are acute and in the case of a fracture, such an angle is usually above 10°. For healthy long-bones, these angles are normally below 10°. Based on such data, we conclude that an acute angle (>10°) between RDSS pairs is likely to indicate the presence of contour abnormality in the corresponding regions. Once the angles between all consecutive RDSS-pairs are estimated, the pair that subtends the maximum acute angle is used to identify the possible fracture site on the bone-contour. It is also observed that when two fracture points are identified on the same or opposite side of the bone-contour, the value of one angle always becomes greater than >30°. Fig. 15 (b) shows the acute angle created between consecutive RDSS pairs ('R1'-'R2' and 'R8'-'R9') in break-regions. Table 2 shows the results for each test image at the points of fracture.

**4.3.2.4. Determination of probable fracture points.** The point of intersection of two consecutive RDSS components that subtend an acute angle (10°) at the intersection point, represents a fracture point on bone-contour. However, for a complex type of fracture, multiple RDSS-pairs may appear having an acute angle between them. In such a situation, the selection of probable fracture points is made based on the following criteria.

- The directions along which the two fracture points are visited during contour-traversal should be opposite. This

**Table 2 – Fracture identification parameters.**

Image no.	# Connected components	RDSS $\angle$ (break points)	Concavity index (at break points)	# RDSSs
Fig. 16(a1)	1	45°, 18°	3, 4	12
Fig. 16(b1)	3	26°, 45°	6, 8	26
Fig. 16(c1)	1	56°, 45°	4, -3	14
Fig. 16(d1)	2	70°, 50°	2, -2	18
Fig. 16(e1)	2	80°, 68°	3, -4	22
Fig. 17(a1)	1	65°, 77°	6, -5	10
Fig. 17(b1)	1	60°, 45°	-8, -3	13
Fig. 17(c1)	1	80°, 56°	-3, 4	10
Fig. 17(d1)	1	77°, 20°	2, -1	17
Fig. 17(e1)	1	80°, 75°	2, 2	4

**Fig. 15 – Outer cover: (a) RDSS representation of bone-contour with direction of traversal, (b) angle of intersection of RDSS pair at the point of fracture.**

ensures the presence of two points in the opposite side of the bone-contour.

- (ii) The acute angle subtended at a probable fracture should be significantly larger compared to that at other points on the bone-contour traversed in the same direction.
- (iii) If all probable points have the same direction of traversal, then the two largest angles are selected to indicate the end points of a line-of-break.

After selection of two probable fracture points on the bone-contour, the line-of-fracture is generated. Fig. 15(a) shows the direction of traversal at the point of fracture. Note that the directions of traversal at the two fracture points 'X' and 'Y', are opposite to each other. Fig. 15(b) shows the angle of intersection between two consecutive RDSS-pair (R1, R2) at the fracture-point 'X', and the angle between RDSS-pair (R8, R9) at the fracture-point 'Y'.

#### 4.3.3. Detection of fracture type

The proposed method categorizes a fracture on the basis of the severity of injury. Evidences from the medical literature [12] indicate that a simple fracture results in a single break with loosely-connected fragments, whereas, in the case of complex or compound fracture, multiple bone splinters appear

with external skin injury. We perform connected component analysis [13] on the segmented bone-contour to count the number of disconnected components in the fractured bone. A fracture that shows up a single connected component is classified as being simple and that with multiple or disjoint components is classified as being complex. Further, a fracture may appear in the top (proximal), middle (diaphyseal), or bottom (distal) region of the bone. The proposed approach identifies the location of fracture-points and categorizes the fracture accordingly. If the points appear within the upper 30% of an bone image, then it is termed as proximal region fracture. Fracture points within the lower 30% of the image indicate distal fracture, and those in the middle part of the bone (middle 40%) represent diaphyseal region fracture. Table 3 reports the type of fractures for several test images.

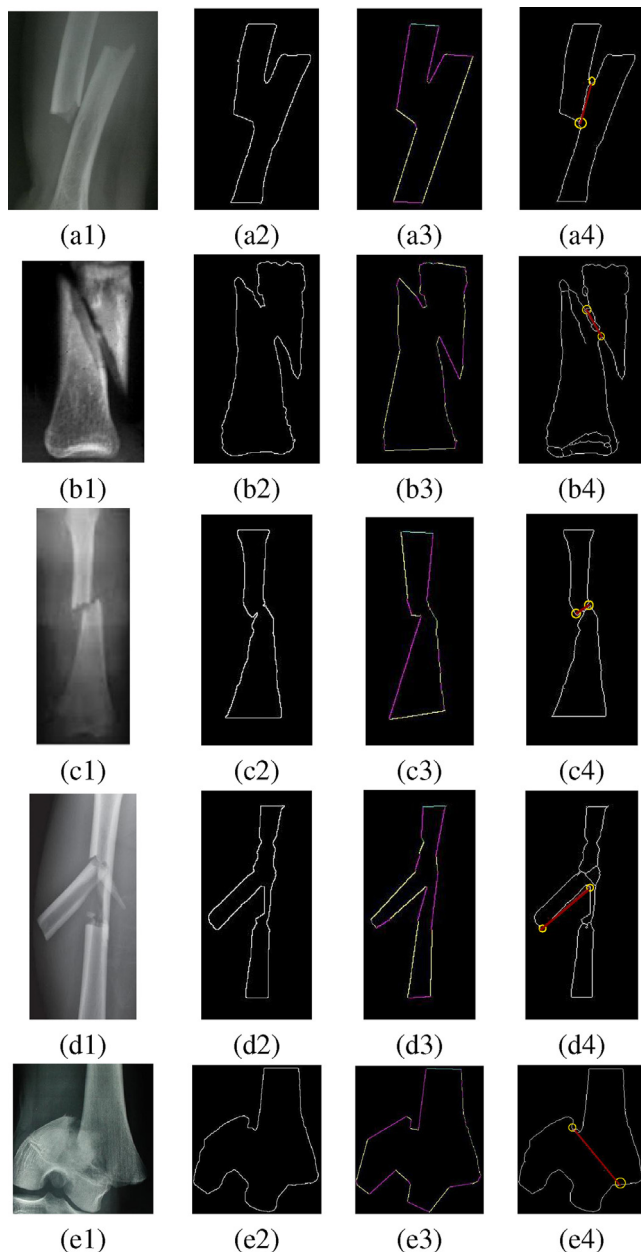
## 5. Case studies and experimental experience

The proposed algorithm is tested on several X-ray images with long-bone fracture. In most of the cases, identification of the presence of fracture and detection of the line-of-fracture are performed correctly. The results generated in different phases of the proposed method are shown in Figs. 16 and 17. The first column in each of Figs. 16 and 17 shows the input X-ray image. The second and third columns show the isothetic cover of the bone-contour and its RDSS representation where each colored-line represents an RDSS. Column 4 displays the detected line-of-fracture observed on the bone-contour. The bone-image in Fig. 16(a1) shows a fracture in the diaphyseal region of Humerus. Fig. 16(a4) shows the line-of-fracture where the end-points are marked. The fracture is correctly classified as being simple, and occurring in the diaphyseal region (as shown in Table 3).

Fig. 16(b1) shows an X-ray image of a fractured tibia bone in which two components are connected loosely. The isothetic cover of the bone-contour is shown in Fig. 16(b2). The image has three connected components, and hence, it has been categorized as a complex fracture (Table 3). The line-of-fracture and its end points are marked in Fig. 16(b4). In this X-ray image, the line-of-fracture extends from the top part of the image to the middle region. The algorithm has identified the region of fracture correctly as having occurred in the proximal region (see Table 3). A femur-fracture image is shown in Fig. 16(c1),

**Table 3 – Fracture identification summary.**

Image no.	Type	Location	Time (ms)	Comments
Fig. 16(a1) (225 × 305)	Simple	Diaphyseal	0.61	Correct
Fig. 16(b1) (263 × 334)	Complex	Proximal	0.80	Correct
Fig. 16(c1) (176 × 291)	Simple	Diaphyseal	0.39	Correct
Fig. 16(d1) (186 × 306)	Complex	Diaphyseal	0.5	Correct
Fig. 16(e1) (313 × 403)	Complex	Diaphyseal	1.76	Wrong (Distal)
Fig. 17(a1) (161 × 323)	Simple	Diaphyseal	0.44	Correct
Fig. 17(b1) (172 × 334)	Simple	Distal	0.51	Correct
Fig. 17(c1) (141 × 252)	Simple	Diaphyseal	0.29	Correct
Fig. 17(d1) (158 × 332)	Simple	Diaphyseal	0.65	Wrong (Distal)
Fig. 17(e1) (105 × 217)	Simple	Diaphyseal	0.65	Correct



**Fig. 16 – Fracture identification with line-of-break (a) Fracture identification with line-of-break (a) Column-1: Input X-ray image, (b) Column-2: Isothetic outer cover of bone-contour, (c) Column-3: RDSS image of isothetic cover, (d) Column-4: Line of break marked in bone-contour.**

where the fracture has affected a small area in the diaphyseal part of the bone. The proposed method has identified the region of fracture correctly as being simple and diaphyseal. In some situations, a fracture may affect only one side of the bone-contour as shown in Fig. 16(d1). In this case, all probable fracture points will appear having the same direction of traversal.

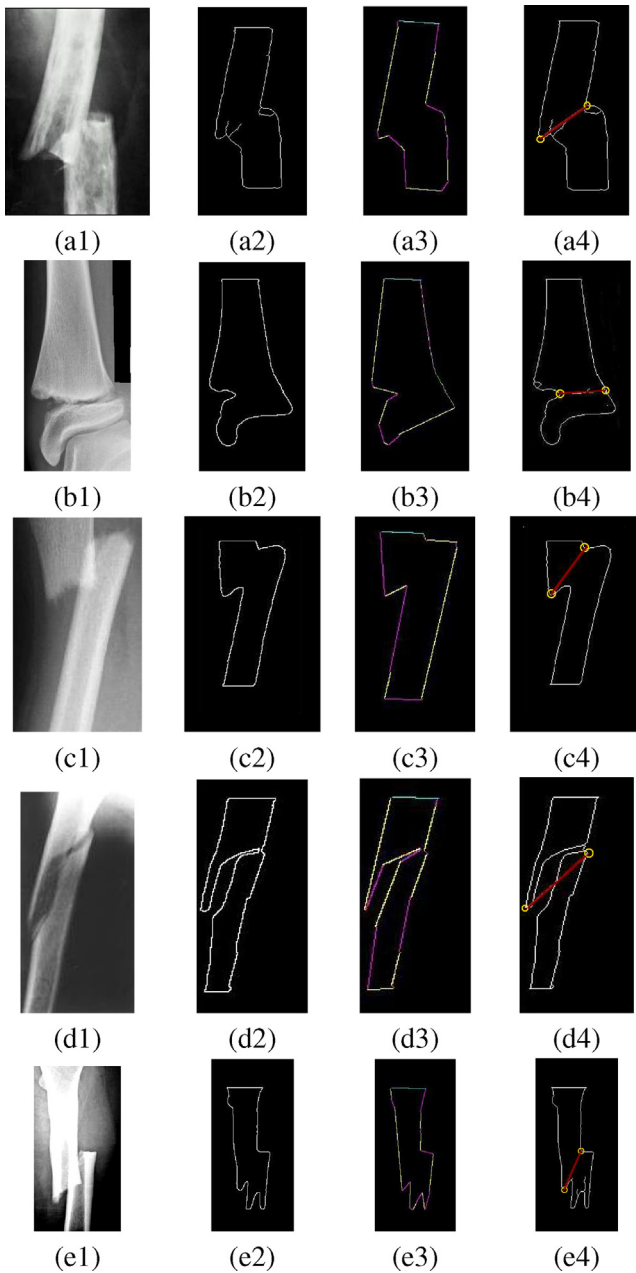
The fracture points with maximum acute angle values are selected as the end points of the line-of-break (Fig. 16(d4)). As the line-of-break has appeared in the middle part of the image with multiple connected components, the fracture is classified as being complex and having occurred in the diaphyseal region.

Long-bone fractures near a bone-joint may consist of multiple possible fracture points in a curved region as shown in Figs. 16(e1) and 17(b1). In such situations, the nearby values of the acute angles between consecutive RDSS pairs at probable fracture points, help to identify the line-of-fracture accurately (see Figs. 16(e4) and 17(b4)). All X-ray images shown in Fig. 17(a1), (c1), (e1), (d1), indicate simple fracture. Fig. 17(c1), (e1) have a fracture in the diaphyseal part of the bone whereas Fig. 17(d1) has a fracture in the distal part of the bone. In Fig. 17(d3), an acute angle is also subtended by the RDSS'S at the bottom part; this is observed because of the natural curvature of the bone. The proposed algorithm only examines the long-bone part and ignores such boundary cases. The line-of-fracture along with end points is marked (see Fig. 17(d4)) based on RDSS-approximation. However, the fracture-region in Fig. (c1) is wrongly classified as being diaphyseal (see Table 3).

The efficiency of the proposed method depends on the quality of segmentation and the sharpness and clarity of bone-contour. Over-segmentation or under-segmentation may lead to incorrect identification of fracture points. Hence, the choice of a suitable segmentation tool is important for successful implementation of this technique.

Table 2 shows the decisions taken regarding the type and location of the fracture in an X-ray. A fracture with a single connected component is categorized as being simple and that with more than one connected components is categorized as being complex (multi-fragmentary). The location of a fracture is decided on the basis of the average position of break-points in the bone-image. In some cases, it shows incorrect results because of the absence of a complete X-ray image.

Table 3 reports the CPU time (in ms) needed by the proposed fracture detection technique on various types of fractured bone X-ray images. It is observed that for an image of size



**Fig. 17 – Fracture identification with line-of-break (a) Column-1: Input X-ray image, (b) Column-2: Isothetic outer cover of bone-contour, (c) Column-3: RDSS image of isothetic cover, (d) Column-4: Line of break marked in bone-contour.**

around 10KB, the proposed algorithm takes less than 2 ms time (on Intel Core 560 PC with clock speed 3.33 GHz, 64 bit OS and 4 GB RAM) to complete the entire process.

## 6. Description of the software tool

The proposed tool has been developed using Matlab R2013b with Windows 7 OS on Intel Core 560 PC with clock speed 3.33 GHz, 64 bit OS, and 4 GB RAM. We have built a graphic user's interface (GUI) with which a demonstration with

four different images can be viewed. The display-boxes for viewing the output image are also provided. The tool allows the user to select a demonstration option among four available examples. After selection, the user may click on the 'show demonstration' button. This will initiate the background process and display the input image in the left-most box. As the execution continues, the entropy-standard deviation image, RDSS-approximation of the bounding isothetic cover, and the line-of-fracture are shown on the GUI platform. Finally, the location of the fracture in the given X-ray image is determined and displayed on the screen. In the case of a normal healthy bone, the 'No Fracture Detected' message will appear in the dialogue box. For details, one may click 'Demonstration' in the linked page <http://oldwww.iests.ac.in/it-abiswas-research> Demonstration link (<http://oldwww.iests.ac.in/it-abiswas-research>).

## 7. Conclusion

We have proposed an integrated tool for contour correction, fracture detection, line-of-break identification and for the classification of long-bone fractures based on digital-geometric analysis of an input X-ray image. We have shown that certain geometric features can be successfully utilized for providing fast and accurate solutions to the automation of medical image analysis. Our experiments on several X-ray image databases demonstrate the suitability and effectiveness of the proposed approach for the purpose of fracture detection in human long-bone structures. This work can further be extended in the future for automated fracture classification and for virtual simulation of orthopaedic surgery.

## Acknowledgements

Authors would like to acknowledge Department of Science & Technology, Government of India for financial support vide reference no. SR/WOS-A/ET-1022/2014 under the Woman Scientist Scheme to carry out this work. The laboratory and computing facilities used at the Nanotechnology Research Triangle, ACMU, Indian Statistical Institute, Kolkata, are also thankfully acknowledged.

## REFERENCES

- [1] O. Bandyopadhyay, A. Biswas, B.B. Bhattacharya, Long-bone fracture detection in digital X-ray images based on concavity index, in: *Proceedings, International Workshop on Combinatorial Image Analysis*, vol. LNCS 8466, 2014, pp. 212–223.
- [2] O. Bandyopadhyay, A. Biswas, B. Chanda, B.B. Bhattacharya, Bone contour tracing in digital X-ray images based on adaptive thresholding, in: *Proceedings of the 5th International Conference on Pattern Recognition and Machine Intelligence*, vol. LNCS 8251, 2013, pp. 465–473.
- [3] O. Bandyopadhyay, B. Chanda, B.B. Bhattacharya, Entropy-based automatic segmentation of bones in digital X-ray images, in: *Proceedings of the 4th International Conference on Pattern Recognition and Machine Intelligence*, vol. LNCS 6744, 2011, pp. 122–129.



- [4] P. Bhowmick, B.B. Bhattacharya, Fast polygonal approximation of digital curves using relaxed straightness properties, *IEEE Trans. Pattern Anal. Mach. Intell.* 29 (9) (2007) 1590–1602.
- [5] A. Biswas, P. Bhowmick, B.B. Bhattacharya, Construction of isothetic covers of a digital object: a combinatorial approach, *J. Visual Commun. Image Represent.* 21 (2010) 295–310.
- [6] A. Biswas, S. Khara, P. Bhowmik, B.B. Bhattacharya, Extraction of region of interest from face images using cellular analysis, in: *Proceedings of the 1st Annual Compute Conference, ACM Compute*, 2008, pp. 1–8 (Article No. 15).
- [7] J. Canny, A computational approach to edge detection, *IEEE Trans. Pattern Anal. Mach. Intell.* 8 (6) (1986) 679–698.
- [8] J. Cecil, P. Ramanathan, M. Pirela-Cruz, M.B.R. Kumar, A virtual reality based simulation environment for orthopedic surgery, in: *Proceedings of the On the Move to Meaningful Internet Systems*, vol. LNCS 8842, 2014, pp. 275–285.
- [9] J. Cecil, P. Ramanathan, V. Rahneshin, A. Prakash, M. Pirela-Cruz, Collaborative virtual environments for orthopedic surgery, in: *Proceedings of the IEEE International Conference on Automation Science and Engineering*, 2013, pp. 133–137.
- [10] M. Donnelley, G. Knowles, T. Hearn, A CAD system for long-bone segmentation and fracture detection, *Image Signal Process.* 5099 (2008) 153–162.
- [11] H. Freeman, On the encoding of arbitrary geometric configurations, *IRE Trans. Electron. Comput.* 10 (1961) 260–268.
- [12] C. Garnavos, N.K. Kanakaris, N.G. Lasanianos, P. Tzortzi, R.M. West, New classification system for long-bone fractures supplementing the AO/OTA classification, *Orthopedics* 35 (5) (2012) 709–719.
- [13] R. Gonzalez, R.E. Woods, *Digital Image Processing*, 2008.
- [14] I. Hacıhaliloglu, R. Abugharbieh, A.J. Hodgson, R.N. Rohling, P. Guy, Automatic bone localization and fracture detection from volumetric ultrasound images using 3-D local phase features, *Ultrasound Med. Biol.* 38 (2012) 128–144.
- [15] M. Hung, J. Wu, J. Chang, D. Yang, An efficient k-means clustering algorithm using simple partitioning, *J. Inf. Sci. Eng.* 21 (2005) 1157–1177.
- [16] T. Kanungo, D.M. Mount, N.S. Netanyahu, C.D. Piatko, R. Silverman, A.Y. Wu, An efficient k-means clustering algorithm: Analysis and implementation, *IEEE Trans. Pattern Anal. Mach. Intell.* 24 (7) (2002) 881–892.
- [17] J. Liang, B. Pan, Y. Huang, X. Fan, Fracture identification of X-ray image, in: *Proceedings of the IEEE International Conference on Wavelet Analysis and Pattern Recognition*, 2010, pp. 67–73.
- [18] P.L. Lin, P.Y. Huangemail, P.W. Huang, H.C. Hsuemail, C.C. Chen, Teeth segmentation of dental periapical radiographs based on local singularity analysis, *Comput. Methods Progr. Biomed.* 113 (2014) 433–445.
- [19] V.L.F. Lum, W.K. Leow, Y. Chen, Combining classifiers for bone fracture detection in X-ray images, in: *Proceedings of the IEEE International Congress on Image and Signal Processing*, vol. 1, 2005, pp. 1149–1152.
- [20] Z. Ma, J.M. Tavares, R.N. Jorge, T. Mascarenhas, A review of algorithms for medical image segmentation and their applications to the female pelvic cavity, *Comput. Methods Biomed. Eng.* 13 (2) (2010) 235–246.
- [21] D. Martin, C. Fowlkes, J. Malik, Learning to detect natural image boundaries using local brightness, color and texture cues, *IEEE Trans. Pattern Anal. Mach. Intell.* 26 (2004) 530–549.
- [22] A. Mehrabi, C. Glckstein, A. Benner, B. Hashemi, C. Herfarth, F. Kallinowski, A new way for surgical education - development and evaluation of a computer-based training module, *Comput. Biol. Med.* 30 (2) (2000) 97–109.
- [23] T.P. Nguyen, I. Debled-Rennesson, A discrete geometry approach for dominant point detection, *Pattern Recognit.* 44 (2011) 32–44.
- [24] U.A.A. Niroshika, R.G.N. Meegama, A deformable model to segment discontinuous boundaries of bone fractures in X-ray images, in: *Proceedings of the IEEE International Conference on Industrial and Information System*, 2013, pp. 309–314.
- [25] N. Otsu, A threshold selection method from gray-level histograms, *IEEE Trans. Syst. Man Cybern.* 9 (1979) 62–66.
- [26] F. Paulano, J.J. Jimnez, R. Pulido, 3D segmentation and labeling of fractured bone from CT images, *Visual Comput. (CGI 2014 issue)* 30 (2014) 939–948.
- [27] A. Rosenfeld, Digital straight line segments, *IEEE Trans. Comput.* 23 (1974) 1264–1269.
- [28] T.P. Tian, Y. Chen, W.K. Leow, W. Hsu, T.S. Howe, M.A. Png, Computing neck-shaft angle of femur for X-ray fracture detection, in: *Proceedings of the International Conference on Computer Analysis of Images and Patterns*, 2003, pp. 82–89.
- [29] M. Tsai, M. Hsieh, S. Jou, Virtual reality orthopedic surgery simulator, *Comput. Biol. Med.* 31 (5) (2001) 333–351.
- [30] D.W. Yap, Y. Chen, W.K. Leow, T.S. Howe, M.A. Png, Detecting femur fractures by texture analysis of trabeculae, in: *Proceedings of the IEEE International Conference on Pattern Recognition*, vol. 3, 2004, pp. 730–733.
- [31] G. Zheng, X. Dong, P.A. Grutzner, L. Nolte, Automated detection and segmentation of cylindrical fragments from calibrated C-arm images for long bone fracture reduction, *Comput. Methods Progr. Biomed.* 87 (2007) 1–11.