

# Local Scale for Boundary Shape Description: Application in Locating Landmarks Automatically

Sylvia Rueda  
CMIAG Group  
School of Computer Science  
University of Nottingham  
Jubilee Campus  
Wollaton Road, Nottingham  
NG8 1BB, United Kingdom  
sxr@cs.nott.ac.uk

Jayaram K. Udupa  
Medical Image Processing Group  
Department of Radiology  
University of Pennsylvania  
Fourth Floor, Blockley Hall  
423 Guardian Drive, Philadelphia  
PA 19104-6021, USA  
jay@mipg.upenn.edu

Li Bai  
CMIAG Group  
School of Computer Science  
University of Nottingham  
Jubilee Campus  
Wollaton Road, Nottingham  
NG8 1BB, United Kingdom  
bai@cs.nott.ac.uk

## Abstract

*A good shape descriptor is necessary for automatically identifying landmarks on boundaries. Our method of boundary shape description is based on the notion of  $c$ -scale, which is a new local scale concept, defined at each boundary element. From this representation we can extract special points of interest such as convex and concave corners, straight lines, circular segments, and inflection points. The results show that this method gives a complete description of shape and allows the automatic positioning of mathematical landmarks, which agree with our intuitive ideas of where landmarks may be defined. This method is applicable to spaces of any dimensionality, although we have focused in this paper on 2D shapes.*

## 1 Introduction

Shape description plays a fundamental role in computer vision and pattern recognition, especially in the fields of automatically defining landmarks, shape analysis, image segmentation, and registration. The concept of shape can be defined as information that is invariant to the geometrical transformations of rotation, scaling, and translation [5]. Shape representations must be unique, complete, and should be able to reflect the differences between similar objects while abstracting from detail and keeping the basic features. Although many methods for shape description exist, there is none that could be labeled as a general method, in the sense that they are usually application dependent. According to [20] and [15], the existing methods can be classified into contour-based and region-based approaches. The contour-based class is based on the use

of the shape boundary whereas region-based methods include the inside of the shape to do the analysis. Most common techniques for contour-based approaches consist of shape signatures [18] [4], boundary moments [16], polygonal and curve decomposition [11] [12] [1], Fourier descriptors [18] [2], chain code representation [6] [8] [7], syntactic analysis [9] [10], and shape invariants using boundary primitives [14] [17]. This paper describes a new contour-based descriptor.

Let the shape of a 2D object be represented by its external boundary  $\mathcal{B}$ . The elements  $b_i$  of  $\mathcal{B}$  will be called *bels* (an abbreviation for “boundary elements”) or *points* from now on. An important class of methods for characterizing shapes uses the curvature estimated at each bel. To estimate the curvature of a digital boundary, three main approaches have been used [19]: orientation-based, derivative-based and osculating circle-based. In orientation-based methods, the curvature is estimated by the change of slope of the tangent at each point of the boundary in a certain direction, using chain codes to approximate the boundary and different tangent estimations. The derivative-based approach involves calculating the curvature based on derivatives. To achieve this, the boundary is approximated by second order curves such as splines to obtain a parametric definition of the curve. Then, the curvature is estimated using

$$\kappa = \frac{x''y' - x'y''}{(x'^2 + y'^2)^{\frac{3}{2}}}. \quad (1)$$

In this case, the curvature is calculated on sampled points of the approximated boundary, and the results should be mapped again to the digital boundary using interpolation or averaging [13]. For the osculating circle approach, after smoothing, the boundary is fitted by a circular disc of a certain radius. Then, the curvature is computed using the

inverse of these radii at each bel. To avoid digital effects, in all these curvature estimators, some continuous approximation is attempted when estimating the tangents at the boundary, or when the boundary is approximated to obtain a parametric definition, or when a circle is fitted to a smoothed boundary. This may introduce errors in the boundary or even in the shape. In this paper, we present a method of estimating curvature without modifying the boundary, taking into account the digital effects and noise, and considering scale in curvature calculation, to obtain a complete description of shapes with different levels of detail. Furthermore, a byproduct of our method is that a shape can be represented by characteristic points or landmarks. Landmarks can be anatomical, mathematical, or pseudo-landmarks [5]. Manual description of shapes is a tedious and error prone task. The curvature at each bel can be a good indicator for positioning these landmarks. In this paper, we aim to describe shapes automatically using mathematical landmarks obtained after a scale-based curvature estimation process.

We first present a theory for local curvature scale estimation in Section 2. In Section 3, we propose a method for shape description using local curvature scale. In Section 4, we extend this method for landmark selection. In Section 5, we demonstrate the utility of these methods based on shape examples drawn from different areas. Our Conclusions and future work are stated in Section 6.

## 2 Local Curvature Scale: $c$ -scale

We define local curvature scale segment or  $c$ -scale segment at any point  $b$  on a boundary  $\mathcal{B}$  as the largest connected set of points of  $\mathcal{B}$  connected to  $b$  such that no point in that set is farther than  $t$  from a line connecting the two end points of the connected set of points.

Let  $b_1, \dots, b_n$  be the points (or bels) defining a boundary  $\mathcal{B}$ . We will associate with each point  $b = b_i$  its  $c$ -scale

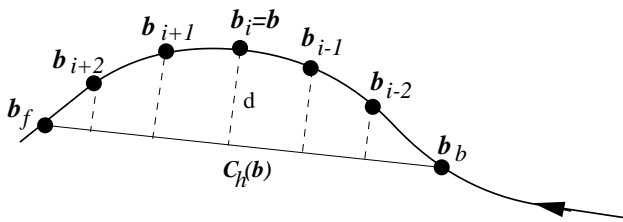


Figure 1.  $c$ -scale estimation.

segment  $C(b)$ . This set is an indirect indicator of the curvature at  $b$ . To determine  $C(b_i)$ , we progressively examine the neighbors, first the set of points  $b_{i-2}, b_i, b_{i+2}$ , then the set  $b_{i-3}, b_{i-2}, b_i, b_{i+2}, b_{i+3}$ , and so on (Fig. 1). At each examination, we calculate the distance of the points in the set from the straight line connecting the two end points of the

set. If the maximum distance of these points from the line is greater than a threshold  $t$ , we define the  $c$ -scale segment  $C(b)$  of  $b$  as the last set of connected points found for which the distance was still within the threshold. The  $c$ -scale value we assign to  $b_i$ , denoted  $C_h(b)$  is the chord length corresponding to  $C(b)$ , which is the length of the straight-line segment between the end points  $b_b$  and  $b_f$  of  $C(b)$ . If  $C_h(b)$  is large, it indicates small curvature at  $b$ , and if it is small, it indicates high curvature.  $c$ -scale values are very helpful in estimating actual segments and their curvature independent of digital effects.

## Relation Between $c$ -scale Value $C_h(b)$ and Radius $r$

We will now arrive at the actual arc length  $A(b)$  at  $b$  corresponding to the  $c$ -scale segment  $C(b)$  from a knowledge of  $C_h(b)$  and by assuming that  $C(b)$  locally represents a circular arc. Let  $b$  be a point on a circular arc  $\mathcal{A}$ . Any segment joining the two ends of the circular arc is a chord  $P_2P_3$  of a circle  $\mathcal{C}$  with radius  $r$  and centre  $o$ . The radius and centre of the circle can be obtained using the following chord property: *the perpendicular bisector of a chord passes through the centre of a circle*. If we trace the perpendicular bisector

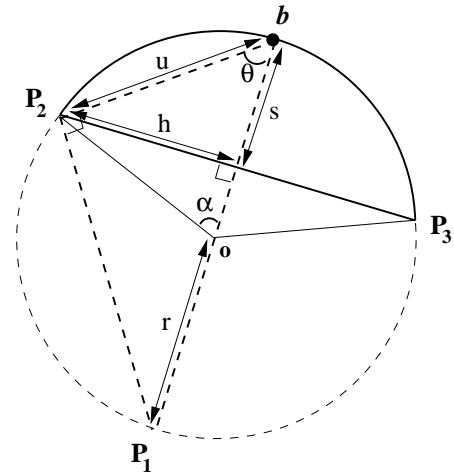


Figure 2. Geometric properties of the chord in a circle.

of the chord, we divide the chord into two equal segments. Let  $h$  be one of these segments and  $s$  the distance between  $b$  and the middle point of the chord. A right triangle of sides  $h, s$  and  $u$  can then be defined (Fig.2), so  $u = \sqrt{s^2 + h^2}$ . If we trace a right triangle  $P_1P_2b$ , then, we can calculate  $r$  using Thales theorem:

$$\cos \theta = \frac{s}{u} = \frac{u}{2r}, \quad (2)$$

and

$$r = \frac{u^2}{2s} = \frac{s^2 + h^2}{2s}. \quad (3)$$

In our case, we have  $h = \frac{C_h(b)}{2}$ . If we substitute  $h$  in (3), we obtain the expression we were looking for  $r$ .

$$r = \frac{4s^2 + C_h(b)^2}{8s}. \quad (4)$$

### Relation Between $c$ -scale $C_h(b)$ and Arc Length $A(b)$

The perimeter of a circle is defined as:  $P = 2\pi r$ . Therefore, the length of a circular arc  $A(b)$  with an angle  $2\alpha$  at the centre (Fig. 2) is given by

$$A(b) = \frac{2\pi r}{360} \cdot (2\alpha) = \frac{\pi r \alpha}{90}. \quad (5)$$

From  $\tan \alpha = \frac{h}{r-s} = \frac{C_h(b)}{2(r-s)}$ , the relation between the chord and the arc length is

$$A(b) = \frac{\arctan\left(\frac{C_h(b)}{2(r-s)}\right)\pi r}{90}. \quad (6)$$

### Relation Between $c$ -scale $C_h(b)$ and Curvature $\kappa(b)$

The curvature  $\kappa$  of a circular segment of radius  $r$  is  $\kappa = \frac{1}{r}$ . Therefore, the relationship between curvature and  $c$ -scale value is:

$$\kappa(b) = \frac{8s}{4s^2 + C_h(b)^2}, \quad (7)$$

where  $C_h(b)$  represents the length of the chord of the osculating circle matching the  $c$ -scale segment  $C(b)$  at  $b$ , and  $s$  is the distance between the mid-point of the arc and the mid-point of the chord.

### Orientation

The orientation at each point  $b$  is generally defined as the angle  $\Psi$  in degrees between the tangent to the boundary at  $b$  with respect to the  $x$ -axis. In our digital setting, we assume the tangent at  $b$  to be a line at  $b$  parallel to the chord connecting the end points  $b_b$  and  $b_f$  of the  $c$ -scale segment  $C(b)$  at  $b$ . The boundary is followed anticlockwise, the inside of the object being on the left of the boundary at any bel  $b$ . Using the chords found previously for each bel, we calculate the unit vector  $\mathbf{u}$  of the chords in the direction of the boundary (Fig. 3).  $\mathbf{u}$  has two components: the direction cosines  $u_x, u_y$ . From the direction cosines, we can write down the orientation at  $b$  as

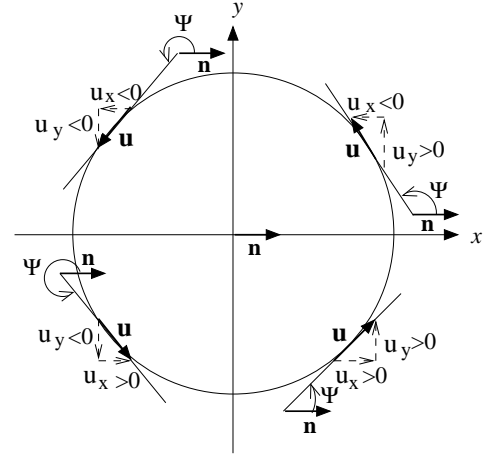


Figure 3. Orientation using direction cosines.

$$\Psi(b) = \begin{cases} 360 - \arccos(u_x(b)) & \text{if } u_y < 0 \\ \arccos(u_x(b)) & \text{if } u_y \geq 0. \end{cases} \quad (8)$$

For a variety of reasons, it is better to measure the orientation with respect to the starting point  $b_1$  as

$$O(b) = \Psi(b) - \Psi(b_1). \quad (9)$$

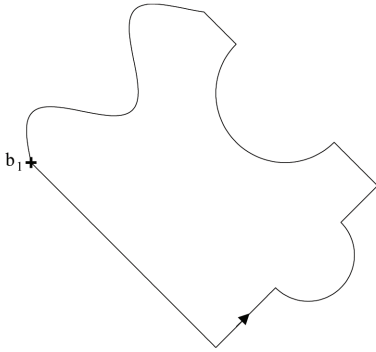
To have a continuous orientation along the boundary, we need to detect the discontinuities caused if we change from  $360^\circ$  to  $0^\circ$  and vice versa, keeping track of how many turns the chord continuously makes. This way we can also describe shapes such as spirals that have several turns and represent the real angle at each point of the boundary that may be greater than  $360^\circ$ .

### Relation Between Orientation $O(b)$ and Curvature

Curvature is defined as the rate of change of the slope of the tangent at each point on the boundary. Therefore, this is the exact definition of the first derivative  $O'(b)$  of the orientation  $O(b)$ . Using  $O'(b)$  we find that the orientation of the boundary is also reflected in the curvature obtained, and therefore, we have more information than using only the magnitude of the curvature. Using this approach we can locate special points of interest. Local positive maxima of  $O'(b)$  will correspond to convex corners, local negative minima to concave corners, constant zero curvature to straight lines, constant non-zero curvature to circular segments, and zero crossings to inflection points. This way we have a whole description of the boundary using  $O'(b)$ .

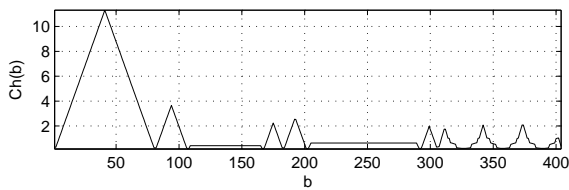
## Example

The above theory is now illustrated using an example shape in Fig.4. This shape is constructed from theoretical functions. It includes different parts such as a rotated rectangle, circular arcs of different radii (convex and concave), and a sine wave. The starting point of the boundary  $b_1$  is represented in the figure by a cross. The boundary is oriented and follows the direction of the arrow, leaving always the inside of the object to the left. The order of the bells is defined using this direction, and the coordinates of the bells are computed by using the functions that define the different sections. To this shape, we apply our  $c$ -scale calculation



**Figure 4. A general shape formed by straight lines, circular arcs, and a sine wave.**

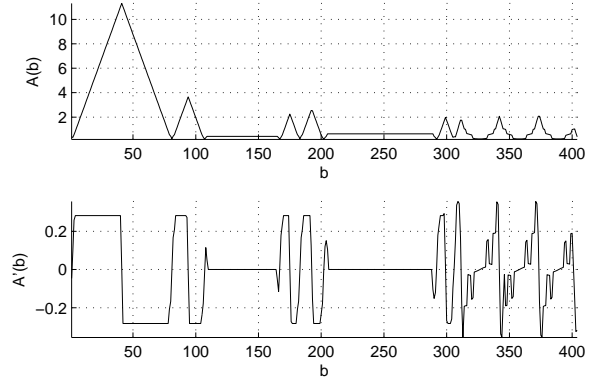
method with  $t = 0.02$ . This parameter controls the level of detail or global scale. For digital boundaries, we usually set  $t \approx 3$ , which works well for all the shapes we tested. This value of  $t$  is able to preserve appropriate boundary details and at the same time ward against digitization noise. The values of the chord lengths (i.e.,  $c$ -scale) for each bel are represented in Fig.5.



**Figure 5. The  $c$ -scale value  $C_h(b)$  along the boundary for the shape in Fig. 4 starting with  $b_1$  at origin.**

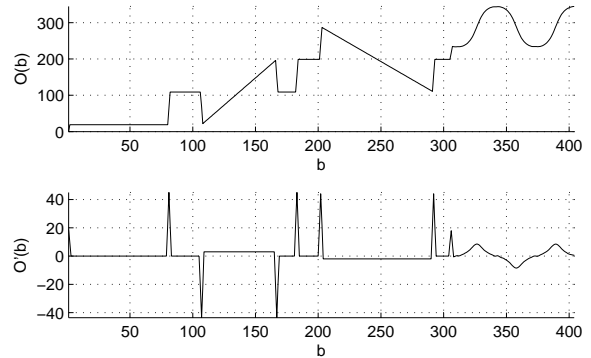
Due to the symmetry of the  $c$ -scale method, each straight segment on the boundary corresponds to a peak in the  $C_h(b)$

plot, and the location of the peak corresponds to the midpoint of the segment. Chords of same length are obtained when we have a circular shape with constant curvature. And valleys on the  $C_h(b)$  plot represent curved regions of the boundary. Using Equation (6), a representation of the arc length  $A(b)$  ( as shown in Fig.6) can be made. In this case,



**Figure 6. Arc length  $A(b)$  and its derivative  $A'(b)$  for the shape in Fig. 4.**

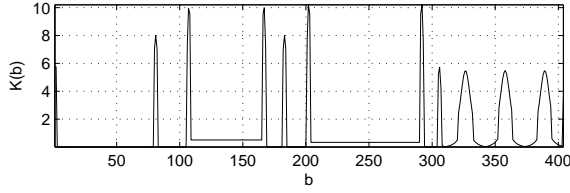
$A(b)$  and  $C_h(b)$  are very similar, but this may not be the case for digital boundaries. The orientation of each bel is defined in degrees. The first element of the boundary will be set to  $0^\circ$  and the rest will be defined with respect to this first element. The orientation  $O(b)$  for each bel of the shape of Fig.4 is shown in Fig.7, as well as its derivative,  $O'(b)$ .



**Figure 7. Orientation  $O(b)$  and its derivative  $O'(b)$  for the shape in Fig.4.**

$O'(b)$  will be very useful for shape description as it constitutes a good estimation of curvature, allowing additionally to distinguish between convex and concave regions in

the boundary. According to Equation (7), we represent the curvature of the shape under consideration in Fig.8.

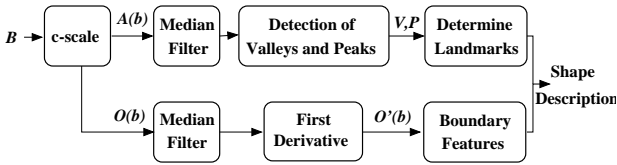


**Figure 8. Curvature  $\kappa(b)$  for the shape in Fig. 4.**

Straight lines on the boundary have a curvature equal to zero, circular regions have constant curvature and sine waves and corners have high curvature values.

### 3 A Method of Shape Description Using $c$ -scale

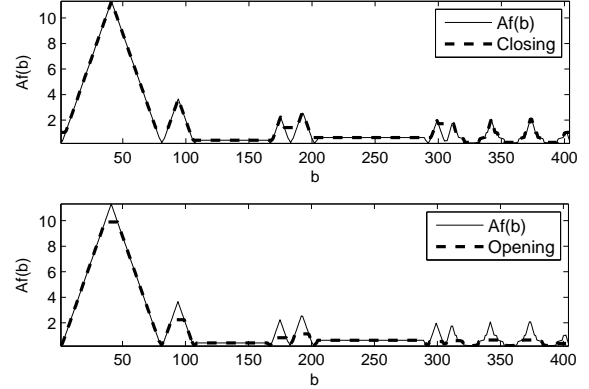
In this section we present the method of boundary shape description using  $c$ -scale. Given a (digital) boundary  $\mathcal{B}$  and a scale parameter  $t$ , our goal is to obtain a partition  $P_{\mathcal{B}}$  of  $\mathcal{B}$  into segments, a set  $s_L$  of landmarks, and a shape description assigned to every element of  $P_{\mathcal{B}}$ . The method is summarized in Figure 9. First, we determine at each point



**Figure 9. The method of boundary shape description.**

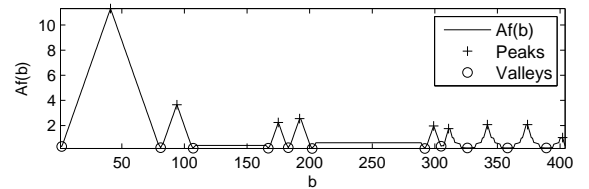
$b$  of  $\mathcal{B}$  the  $c$ -scale value  $C_h(b)$  from which we estimate arc length  $A(b)$  via Equation 6 and orientation  $O(b)$  via Equation 9. Second, we smooth  $A(b)$  and  $O(b)$  using a median filter of width  $2w + 1$  centered at every element  $b$ , where  $w$  is the half width of the window used for filtering, specified in terms of the number of points considered on either side of  $b$ . We repeat this process  $n$  times to get a smoothed version of  $A(b)$ , called  $A_f(b)$ . This is necessary only for digital boundaries. Then, we detect automatically the peaks and the valleys of  $A_f(b)$  by using mathematical morphology. The peaks correspond to straight line segments in the boundary and the valleys to curved segments. Mathematical morphology [16] is based on set theory and provides powerful tools for image analysis. Fundamental operations are

erosion, dilation, opening and closing. An opening consists of an erosion followed by a dilation. A closing is defined as a dilation followed by an erosion. A structuring element defines the size and shape of the transformation to be done. In our case, we will use a structuring element of size  $se$  applied to the signal  $A_f(b)$ . Opening and closing are the transformations we need to detect the peaks and the valleys of  $A_f(b)$  (see Fig.10). In particular, to find the valleys,



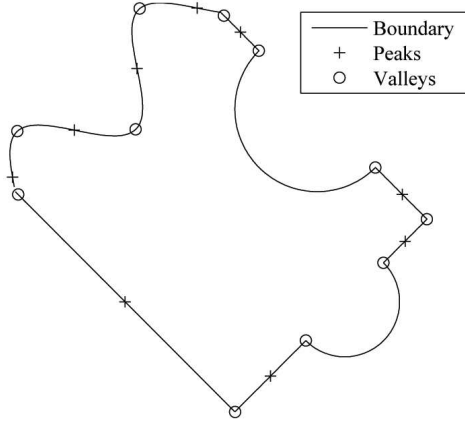
**Figure 10. Closing and opening for valley and peak detection in  $A_f(b)$  for the shape in Fig. 4.**

we apply to  $A_f(b)$  a bottom-hat filtering operation, which is the difference between  $A_f(b)$  and its closing. This filter will extract only the valleys of  $A_f(b)$ . Once we have detected all the valleys in  $A_f(b)$  we find the minimum value for each valley detected. These local minima correspond to the valleys in  $A_f(b)$  and represent points with high curvature in  $\mathcal{B}$ . Similarly, to find the peaks, we need a top-hat filtering operation, which is the difference between  $A_f(b)$  and its opening. This will extract only the peaks of  $A_f(b)$ . These maxima correspond to the peaks in  $A_f(b)$ , which are the middle points of the straight segments in  $\mathcal{B}$ . By selecting a different size for the structuring element, we can vary the number of valleys and peaks detected. We can avoid spurious valleys and peaks by keeping only peaks and valleys bigger than a certain value  $el$ .



**Figure 11. Detection of peaks and valleys.**

For the example of Fig.4, we used a structuring element of size 5 for both valley and peak detection. No element selection was necessary in this case by specifying  $el$ . After the bottom-hat and the top-hat filtering operation, we detect the peaks and valleys as shown in Fig.11. Once we have detected the peaks  $P$  and the valleys  $V$  in  $A_f(b)$ , we can locate the landmarks  $s_L$  on the boundary as shown in Fig.12 by simply identifying the points where local minima/maxima occurred. We can select the level of detail we



**Figure 12. Landmarks denoting the peaks and valleys of  $A_f(b)$  for the shape in Fig.4.**

want to consider by using two threshold values  $nP$  and  $nV$ .  $nP$  will be the smallest value in  $A_f(b)$  from which we want to consider peaks.  $nV$  will be the highest value in  $A_f(b)$  from which we want to consider valleys. This is used in the case that we want to consider only more prominent valleys (corresponding to higher curvature regions) and peaks (corresponding to lower curvature regions). From  $O'(b)$  we can extract a complete description of each bel in  $\mathcal{B}$ . Local positive maxima will correspond to convex corners, local negative maxima to concave corners, constant zero curvature to straight line segments, constant non-zero curvature to circular segments, and zero crossings to inflection points.

#### 4 A Method of Landmark Selection

In this section, we present a method of landmark selection with different levels of detail. The number of landmarks identified depends on the scale parameter  $t$ , the size of the structuring elements  $se$  considered in bottom-hat and top-hat filtering operations, the elements selected  $el$  when selecting valleys and peaks in  $A_f(b)$ , and on the level of detail considered using the thresholds  $nP$  and  $nV$ . We have several options to select as landmarks on a boundary. We can consider only the points corresponding to peaks in

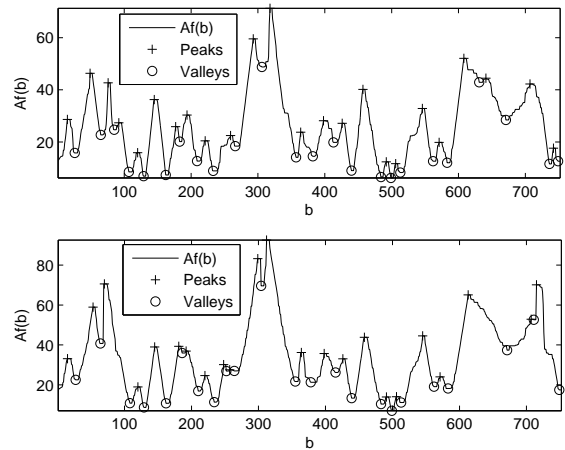
$A_f(b)$ , only to valleys, or both. This can vary depending on the application. In our case, we are interested in automatic landmark selection to build a Point Distribution Model for Active Shape Models [3]. We will use both peaks and valleys of  $A_f(b)$  to describe the shapes. This will be illustrated in the next section.

### 5 Results

We have illustrated in the previous section the curve description process on a mathematical object (Fig.4). In this section, we will focus on automatic landmark selection and present the results for digital boundaries of a natural object and a medical object. We assume that these shapes are extracted after segmentation of appropriate images.

#### Natural Object: a Rabbit

The object we want to characterize with landmarks is presented in Fig.15. In this case, the shape is extracted from a binary image of a rabbit. To show how we can describe a shape considering different levels of detail, we will vary first the scale parameter  $t$ . The greater the  $t$  value considered the less number of details we pick from the digital boundary. But  $t$  has to be large enough to avoid noise and digital effects. A value of  $t$  around 3.1 works well for most shapes we have tested.

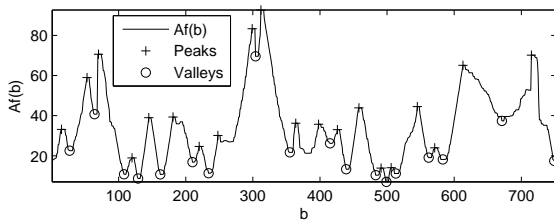


**Figure 13. Detection of Peaks and Valleys in  $A_f(b)$  for  $t = 2$  (top) and  $t = 3.1$  (bottom) for the rabbit shape.**

In Fig.13, we show the plots of  $A_f(b)$  for the scale values  $t = 2$  and  $t = 3.1$ , both smoothed with the median filter with parameters  $w = 3$  and  $n = 2$ . In these curves, we have

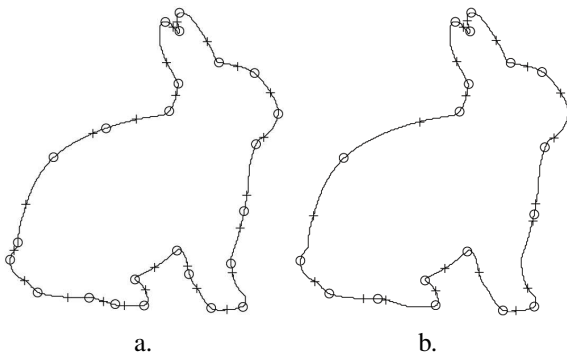
detected the valleys and peaks using a structuring element of size  $se = 5$  for both peaks and valleys.

From Fig.13 we observe that both curves are roughly the same and that they differ in a way that enables us to select more or less number of details on the boundary shape. By selecting structuring elements of smaller size, we can detect only sharp valleys or peaks. We notice from Fig.13 that in some parts we have redundant landmarks because of the detection of small peaks or valleys. For example, for  $t = 3.1$ , we can see at  $b$  around 710 that a spurious peak has been detected. Similarly for  $t = 2$ , around the value 625 we have a peak and a valley corresponding to small variations. To avoid this, we can use the element selection method (cf. Section 4) and consider only the peaks and valleys obtained that are greater than a value  $el$ . In this case, if we set  $t = 3.1$ ,  $el = 1$  for peak and valley detection, we will keep all the valleys and peaks bigger than this value and the result will be as in Fig.14.



**Figure 14. Detection of Peaks and Valleys in  $A_f(b)$  for  $t = 3.1$ , with  $el = 1$  and  $se = 5$  for the rabbit shape.**

We can also use different element selections for peaks and valleys.



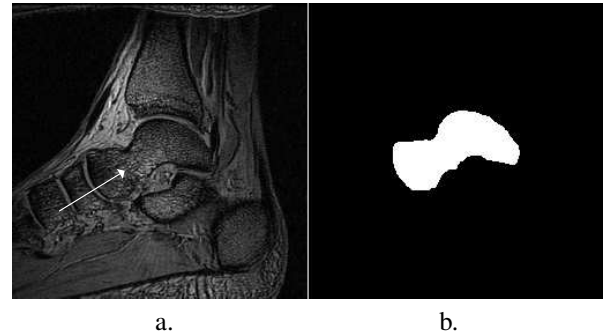
**Figure 15. Landmarks on the boundary of a rabbit shape for different  $t$ : a.  $t = 2$ , b.  $t = 3.1$ .**

Fig.15.a shows the landmarks detected for  $t = 2$ ,  $el = 1$

and  $se = 5$ . In Fig.15.b, the landmarks are located using  $t = 3.1$ ,  $el = 1$  and  $se = 5$ . We can select the appropriate configuration of parameters depending on the application for the automatic selection of landmarks of shapes.

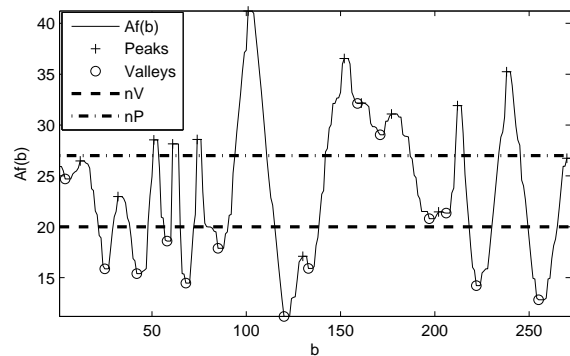
## Medical Object: the Talus Bone of the Foot

We have applied our method to the talus bone (Fig.16.b) segmented from an MR foot image (Fig.16.a). From this



**Figure 16. The talus bone of the foot in a MR image.**

segmented image we have extracted the contour to locate landmarks automatically. In this case we want to illustrate how to select details keeping only higher peaks, which correspond to the middle point of straight segments in the boundary, and lower valleys, which correspond to regions of the boundary that have higher curvature. This can be

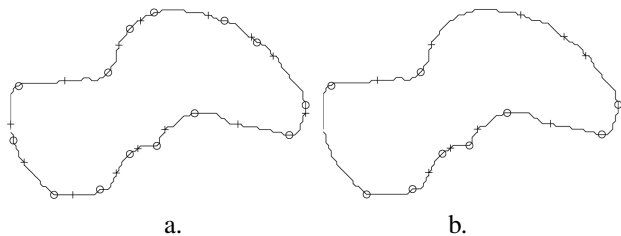


**Figure 17. Peaks and Valleys in  $A_f(b)$  for the talus bone of Fig.16.**

selected using the parameters  $nV$  and  $nP$  presented in Section 3. To see the effect of this selection, we calculate  $A_f(b)$

using  $t = 3.1$ ,  $w = 2$  and  $n = 2$ . Then, we detect valleys and peaks using a structuring element of size  $se = 5$  and without doing any element selection ( $el = 0$ ). The peaks and valleys detected can be seen in Fig.17.

We select  $nV = 20$  and  $nP = 27$  as thresholds (cf. Fig.17). Fig.18.a represents the shape without selection of



**Figure 18. Landmarks on the talus shape for  $t = 3.1$ : a. without and b. with selection via  $nP$  and  $nV$ .**

landmarks whereas Fig.18.b shows the selection of landmarks using  $nV$  and  $nP$ . Only high curvature regions and very low curvature regions were selected.

## 6 Conclusions

In this paper, a new theory and method for shape description and automatic landmark selection based on the novel concept of  $c$ -scale are presented. This approach produces a complete, simple and new boundary shape description with numerous potential applications. For each boundary element  $b$ , the arc length of homogeneous curvature regions is estimated as well as the orientation of the tangent at  $b$ . This method is different from previous methods of curvature estimation and can be directly applied to digital boundaries without requiring prior approximations of the boundary. It is also capable of handling different levels of shape detail. We have shown that this method is useful for shape description as well as automatic landmark selection. This work focuses on 2D shapes but it can be extended to shapes of any dimensionality. In future work, we will use this method to build automatically Point Distribution Models for creating Active Shape Models (ASMs) [3] toward the goal of organ modeling and segmentation.

## Acknowledgments

This research is funded by the European Commission Fp6 Marie Curie Action Programme (MEST-CT-2005-021170).

The medical images were provided by the Medical Image Processing Group, Department of Radiology, University of Pennsylvania, Philadelphia, USA.

## References

- [1] S. Berretti, A. D. Bimbo, and P. Pala. Retrieval by shape similarity with perceptual distance and effective indexing. *IEEE Trans. Multimedia*, 2(4):225–239, 2000.
- [2] R. Chellapa and R. Bagdazian. Fourier coding of image boundaries. *IEEE Trans. PAMI*, 6(1):102–105, 1984.
- [3] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham. Active shape models - their training and application. *Computer Vision Image Understanding*, 61:38–59, 1995.
- [4] E. R. Davies. *Machine Vision: Theory, Algorithms, Practicabilities*. Academic Press, 1997.
- [5] I. L. Dryden and K. V. Mardia. *Statistical Shape Analysis*. Wiley, 1998.
- [6] H. Freeman. On the encoding of arbitrary geometric configurations. *IRE Trans. on Electronic Computers*, 10:260–268, 1961.
- [7] H. Freeman. Shape description via the use of critical points. *Pattern Recognition*, 10:159–166, 1978.
- [8] H. Freeman and J. Saghri. Comparative analysis of line-drawing modeling schemes. *Computer Graphics and Image Processing*, 12:203–223, 1980.
- [9] K. S. Fu. *Syntactic Methods in Pattern Recognition*. Academic Press, 1974.
- [10] K. S. Fu. *Syntactic Pattern Recognition and Applications*. Prentice Hall, 1982.
- [11] W. I. Groskey and R. Mehrotra. Index-based object recognition in pictorial data management. *Computer Graphics and Image Processing*, 52:416–436, 1990.
- [12] W. I. Groskey, P. Neo, and R. Mehrotra. A pictorial index mechanism for model-based matching. *Data Knowledge Eng.*, 8:309–327, 1992.
- [13] S. Herman and R. Klette. A comparative study on 2d curvature estimators. *IC on Computer Theory and Applications*, pages 584–589, 2007.
- [14] S. Z. Li. Shape matching based on invariants. in: *O. Omidvar(Ed.), Shape Analysis, Progress in Neural Networks*, 6:203–228, 1999.
- [15] S. Loncaric. A survey of shape analysis techniques. *Pattern recognition*, 38(8):983–1001, August 1998.
- [16] M. Sonka, V. Hlavac, and R. Boyle. *Image Processing, Analysis, and Machine Vision*. Brooks/Cole, 1999.
- [17] D. M. Squire and T. M. Caelli. Invariance signature: characterizing contours by their departures from invariance. *Computer Vision and Image Understanding*, 77:284–316, 2000.
- [18] P. J. van Otterloo. *A Contour-Oriented Approach to Shape Analysis*. Prentice Hall, 1991.
- [19] M. Worring and A. Smeulders. The accuracy and precision of curvature estimation methods. *IC on Pattern Recognition*, III:139–142, 1992.
- [20] D. Zhang and G. Lu. Review of shape representation and description techniques. *Pattern recognition*, 37(1):1–19, January 2004.