

ON THE USE OF NEURAL NETWORKS AND
GEOMETRICAL CRITERIA FOR LOCALISATION
OF HIGHLY IRREGULAR ELLIPTICAL SHAPES

P. Carvalho, N. Costa, B. Ribeiro, A. Dourado

CISUC - Centro de Informática e Sistemas

Dep. de Eng. Informática da Univ. de Coimbra

Pinhal de Marrocos, Pólo II, P-3030 Coimbra, Portugal

Tel: (+351) 39 790000 Fax: (+351) 39 701266

E-mail: {carvalho, nfilipe, bribeiro, dourado}@dei.uc.pt

Perception of irregular elliptical shapes with neural networks

ON THE USE OF NEURAL NETWORKS AND GEOMETRICAL CRITERIA FOR LOCALISATION OF HIGHLY IRREGULAR ELLIPTICAL SHAPES

ABSTRACT:

Detection of elliptical shapes is of extreme importance in several computer vision applications. In this paper a new method for irregular elliptical shapes localization in multi-connected regions is described. This method first computes a set of elementary arc segments which are then aggregated using geometrical decision criteria and *a posteriori* aggregation probabilities obtained from a neural network for Bayes classification. To identify and characterize the elementary arc segments a cluster identification, a contour grouping strategy and some extensions to Fitzgibbon's ellipse fitting method are introduced. These methods are applied successfully in the setup of an automatic lime granule inspection system. The algorithm has proven to be very robust since it is able to correctly detect elliptical shapes even when noisy data are present.

KEYWORDS:

Automatic Visual Inspection; Ellipse Detection; Neural Networks; Perceptual organization; Bayes Classification; Identification of Clusters; Pulp and Paper Industry

I. INTRODUCTION

Segmentation and localisation of elliptical shapes is a fundamental operation in several computer vision applications which comprehend many different areas, such as the detection of tumors in medical image analysis or the identification of components in industrial applications. Localisation of these shapes is a relative simple task when their regions are found to be isolated in the treated images. However, whenever in the presence of partial juxtaposition or superposition of adjacent regions, there is a substantial increase in the problem's complexity, since, in these circumstances, it is not possible to identify the contours of the several distinct regions in their entire extension. In these situations, the problem solution usually involves some model search procedure. In this category of methods, techniques such as the Hough transform, least square and least median square fit algorithms are included.

The Hough transform, in spite of being robust, exhibits several limitations, as due to its high memory requirements, as due to its low computational efficiency. Namely, it is verified that sequential Hough transform implementations for elliptical shapes search exhibit behaviors of order $O(n^6)$. If the gradients direction is known, then the complexity is reduced by a factor of n [1].

There exist basically two type of methods for ellipse fitting [2] [3]: (i) the least square fitting methods and (ii) the least median fitting methods. It is verified that, in principle, these methods exhibit efficient localisation behaviors whenever the intended shapes are regular. However, in the presence of irregular shapes, as is the case for lime granules, the results obtained by these strategies tend to decline rapidly. Usually it is seen that (i) several ellipses are identified for each segment and that (ii) it is not straightforward to determine the ellipses in multi-connected regions due to the difficulty of deciding which boundary segments of the region belong to each elliptical structure.

It is important to the pulp and paper industry to recycle intermediate reagents for both economical and environmental reasons. One of these paths is used for bleaching liquor recovering which is performed by a causticizing chemical reaction. The velocity of this reaction and consequently its efficiency is highly conditioned by the quality of the lime used. This fact leads to the need for the analysis of lime granules properties, usually left to human inspection. However, the high costs of manual inspection and the fatigue of human operators, which introduce high variances in the resulting data, are stimulating the development of computer-based systems for inspection tasks. Lime granules are more or less elliptical in shape and therefore their automatic visual inspection can be seen as an elliptical object localization and measurement problem.

In this paper a new and robust method, based on geometrical properties and a neural network used as a Bayes estimator, is introduced for localisation of irregular elliptical shapes in multi-connected regions. This method is applied in the setting of an automatic lime granule visual inspection system. The algorithm is inspired in human visual perception theory, namely on the Gestalt's principals for perceptual organisation. According to this theory, perceptual organisation of visual information is governed by a set of properties, which are partially quantified in this paper for arc segments by several geometric measures. Learning, based on past experience is also recognised to play an important role in perception by the Gestalt theory [4]. In the developed method, learning is encompassed by a neural network. Several work exist on perception in general (see [5] for a survey) and on perception using neural networks in particular by exploring their optimisation capabilities [6][7] and dynamics [8]. We use a neural network as a noise tolerant tool to compute non parametrically aggregation probabilities between arc segments, i. e., to compute accurate probability values even when large biases occur in some of the applied features. Using the classification

structure proposed in [5], this method may be considered a 2D structural level perception organisation algorithm, since 2D features are involved in the reconstruction of 2D shapes. In [4], Kanizsa suggests that perception of shapes is performed by first identifying in the visual field regular regions in space or time. After separation of the visual field the process continues with the perception inference of completion of whatever is absent or occluded [5]. This strategy is followed in this paper. Namely, the algorithm applies a two-stage process: (i) in a first stage elementary arc segments (arc segments which are bounded by contact points between distinct regions) are identified and some measures, which quantify some of the properties mentioned in the Gestalt psychology, are extracted. This stage is described in section II.. (ii) In a second stage, described in section III., the obtained elementary arcs are grouped together using geometrical properties and a neural network classifier to reconstruct the elliptical shapes present in the image. In section A. a new contour grouping method based on the Green's discrete theorem is introduced. This method enables the identification of all clusters of elliptical regions in the image. In section B. the methodology applied throughout the identification and characterisation of elementary arc segments is described. In this section some extensions to Fitzgibbon's ellipse estimation algorithm are presented. Namely, a strategy to avoid inversion problems is developed, due to near singularity definitions of the scatter matrix, which is needed for measure extraction. Section III. describes the aggregation algorithm used for elliptical shape reconstruction from the elementary arc segments. This method is based on geometrical properties and uses a Bayes classification approach implemented with a neural network. Finally, in section IV. the main conclusions and results are presented.

II. LOCALIZATION OF IRREGULAR ELLIPTICAL SHAPES

Let F be the original digital image and F^B the binary image obtained from F . In this paper it is required that all inner areas of the elliptical regions in the image be filled with the same gray level. For discussion purposes, it will be assumed that these regions are filled with the maximum gray level (white) and that the image's background is filled with the minimum gray level (black). This is very easy to accomplish whenever a threshold selection algorithm (see [9] for a survey) can be used for image segmentation, which, for industrial applications, is relatively straightforward, since light and acquisition conditions are usually controllable. In the outlined inspection problem we use a minimum cross-entropy threshold selection method [10] [11].

Once the binary image is obtained, the localization procedure for clusters of irregular elliptical shapes is initiated. The method described in this paper is composed by two-stages: (i) elementary arc segment identification and characterisation - these operations will be described in the following sections, and (ii) elementary arc segment grouping - the grouping strategy will be outlined in section III..

A. Contour grouping using the Green's theorem

Elementary arc segments (EAS) are obtained from the contours of the clusters of regions identified in the binary image. In this step, the contour tracing algorithm must be able to identify each independent group of contours belonging to each cluster of multi-connected elliptical regions in the image (a cluster of multi-connected regions may include in its interior other independent clusters of regions).

Before describing the cluster's identification strategy, let us first introduce some definitions which will be used along the paper.

Definition 1 *A contour point is a point that does not exhibit 4-connectness.*

Definition 2 *For external contours, the cluster's region lies always inside the area defined by the contour.*

Definition 3 *For internal contours, the cluster's region lies always outside the area defined by the contour.*

Definition 4 *A cluster of irregular multi-connected elliptical regions is considered to be any set of multi-connected irregular elliptical regions, such that there exists a continuous path, completely included inside the cluster's region, connecting any two points of the set. (See figure 1 (a). Contours $Cont_{1,4}$ and $Cont_{1,5}$ define one of the several clusters exhibited in the image.)*

Let $SC_j \stackrel{def}{=} \{Cont_{j,1}, \dots, Cont_{j,n}\}$ be the set of contours of the j th cluster of regions in the image and $Cont_{j,i}$ be its i th contour. Whenever a new contour starting point $p \stackrel{def}{=} (x, y)$ is detected (new contour points are detected by an image horizontal, from top to bottom, scanning procedure), it is determined whether it belongs to an outer contour of a new cluster of regions or to an internal contour of an already identified cluster. This task is performed using the Green's discrete theorem [12] [13] (validation of equation 2). Namely, point p belongs to an internal contour of an already identified cluster if and only if point p is included inside an external contour (most outer contour) of an already scanned cluster, i. e.(1),

$$\exists j = 1, \dots, K : p \subset Cont_{j,1} \tag{1}$$

$Cont_{j,1}$ – contour number 1 of cluster j

K - number of identified clusters

If equation 1 is verified for some j , then the new contour is appended to set SC_j , else a new cluster has been detected and, therefore, K is incremented by one and a new set $SC_K = \{Cont_{K,1}\}$ is created.

The inclusion test in equation 1 is easily performed with the Green's discrete theorem [12] [13]: let $Cont_{j,1} \stackrel{def}{=} \{p_1, p_2, \dots, p_n\}$, n - number of points in the contour, be the set of points that form the contour, where each point $p_z \stackrel{def}{=} (x_z, y_z, d_z)$ is described by its co-ordinates (x_z, y_z) and by its chain code direction d_z . If equation 2 is corroborated, then point p is included inside $Cont_{j,1}$. Since equation 2 has to account for the contour's direction, functions C_y and D_y should be redefined as in equations 5 and 6 (see proof 1 in appendix).

$$\sum_{z=1}^n [\Delta(x_z - x, y_z - y) D_y(d_{z-1}, d_z) + J(x_z - x, y_z - y) C_y(d_{z-1}, d_z)] = 1 \quad (2)$$

$$J(x, y) = \begin{cases} 1 \Leftarrow x \neq 0 \text{ or } y \neq 0 \\ 0 \Leftarrow x = 0 \text{ and } y = 0 \end{cases} \quad (3)$$

$$\Delta(x, y) = \begin{cases} 1 \Leftarrow x > 0 \text{ and } y = 0 \\ 0 \Leftarrow x \leq 0 \text{ or } y \neq 0 \end{cases} \quad (4)$$

$$C_y(a, b) = \begin{cases} 0 \Leftarrow a \leq (b + 3) \bmod 8 \\ t \Leftarrow else \end{cases} \quad (5)$$

$$D_y(a, b) = \begin{cases} t \Leftarrow b \in \{1, 2, 3, 4\} \wedge a \in \{0, 1, 2, 3\} \\ -t \Leftarrow b \in \{0, 5, 6, 7\} \wedge a \in \{4, 5, 6, 7\} \\ 0 \Leftarrow else \end{cases} \quad (6)$$

$$t = \begin{cases} 1 \Leftarrow \text{counterclockwise direction contours} \\ -1 \Leftarrow \text{clockwise direction contours} \end{cases} \quad (7)$$

Given that equation 2 depends on the contour's direction, it is imperative that it is clearly determined prior to the use of the Green's theorem. For this purpose equation 12 can be applied (see proof 2 in appendix): let $Cont$ be a closed contour, $\sum_{k=1}^n h_x(d_k) = 0$, $h_x(d)$ - horizontal displacement for chain code d , such that there are no common points in $Cont$, that is, (8) holds,

$$\nexists i, j \in \{1, \dots, n\} : p_i = p_j \wedge i \neq j \quad (8)$$

Further, let $p_i \stackrel{def}{=} (x_i, y_i, d_i) \in Cont$, such that p_i is not a local extreme point, that is,

$$p_i \in Cont : h_x(d_i) h_x(d_{i-1}) > 0 \quad (9)$$

and let S be the set of all points $p_j \stackrel{def}{=} (x_j, y_j, d_j) \in Cont$, such that $x_j + h_x(d_j) = x_i$ and points p_i and p_j exhibit inverse chain code directions, that is

$$\sum_{k=i}^j h_x(d_k) = 0, \quad j + 1 \neq i \wedge h_x(d_j) h_x(d_i) < 0 \quad (10)$$

or

$$x_j = x_i, \quad i \neq j, \quad h_x(d_i) h_x(d_j) > 0 \quad (11)$$

then, it is seen that:

$$Dir = \begin{cases} \text{Clockwise} \Leftarrow \Delta_1(i, S) < 0 \\ \text{CounterClockwise} \Leftarrow \Delta_1(i, S) > 0 \end{cases} \quad (12)$$

$$\Delta_1(i, S) = \sum_{j \in S} h_x(d_j) \left[\sum_{k=i}^{j-1} h_y(d_k) \right] \quad (13)$$

Points p_j obtained with equation 11 may not correspond to local extreme points. If, however, this happens, then a new point p_i and a new set S must be computed. Valid points must, therefore, verify

$$h_x(d_{j-1}) h_x(d_j) > 0 \quad (14)$$

Functions $h_x(d)$ and $h_y(d)$ are, respectively, the horizontal and the vertical pixel displacement for chain code d .

This contour grouping strategy identifies the whole set of independent clusters (clusters not included inside any other cluster) of multi-connected regions in the image. All external contours are always the first $Cont_{j,1}$ in their cluster SC_j . Furthermore, it is verified that the inclusion order of the contours in each SC_j is increasing, that is, there does not exist a pair of contours $\{Cont_{j,i}, Cont_{j,w}\} \in SC_j$ such that

$$i < w : Cont_{j,i} \subset Cont_{j,w} \quad (15)$$

It often happens that some clusters are completely included inside other clusters (see figure 1 (a)). In these cases, the grouping algorithm only identifies one external contour - the most outer contour of the cluster that is not included inside any other cluster - and, therefore, only one set SC_j will be created by the outlined grouping strategy (see figure 1 (a) - only set SC_1 exists). To solve these situations, all contours' sets are inspected to identify all levels of clusters. Solving these situations at this stage has several advantages: (i) computing requirements are reduced. Given that the region of one ellipse is always defined in one independent cluster, it is useless to inspect adjacent clusters to identify the ellipse. (ii) For neighbour clusters it is further avoided to perform physically non-causal ellipse identification, which could arise under highly noisy situations. These false identifications could of course be detected by performing a convexity test on the identified structures. However, this would

increase unnecessarily the algorithm's complexity and computational requirements.

To identify the whole set of clusters, first an inclusion tree is constructed. In these trees each node represents a contour belonging to the same set SC_j . Whenever a node exhibits a link to a child node, it is implied that the child node contour is included inside the father node contour. Note that the tree's *root* is $Cont_{j,1}$.

To perform the inclusion test the Green's discrete algorithm is once again applied. Note that, given the property expressed in equation 15, any contour $Cont$ finds always its correct order in the inclusion tree. In figure 1 (b) the obtained inclusion tree from the set of contours of figure 1 (a) is shown. These inclusion trees exhibit some useful properties:

1. If a child node of an external contour node has links to child nodes, then it represents an internal contour of a cluster.
2. Those nodes which do not exhibit any links to child nodes may represent internal or external contours
3. Those child nodes whose father nodes represent internal contours can represent internal or external contours.

The last step of the grouping procedure is to identify all clusters present in each inclusion tree. This is performed with a recursive algorithm based on the above stated properties. For each inclusion tree the following steps are applied (the initial argument is the tree's root; note that only those contours which correspond to the several $Cont_{j,1}$, $j = 1, \dots, K$, are initially marked as external contours):

Inputs:

node - a node of the tree; initial value: the tree's root

Outputs:

Identified sets of clusters

Algorithm of **Determineclusters** (*node*)

Begin

For (*each childnode of node*) *do*

If (*[childnode has childnodes and node represents an external contour] or [childnode represents an internal contour]*) *then*

Mark childnode as an internal contour

Determineclusters (*childnode*)

Else

Mark childnode as an external contour

Disconnect child node and all its descendant nodes from the tree

Increment K and construct a new cluster SC_K with them

Delete all Cont included in the new tree from the previous SC_j

End

External and internal contours are easily identified using the following function Ψ (see proof 3 in appendix): choose point $p_i = (x_i, y_i, d_i) \in Cont$ such that $(x_i, y_i - 1), (x_i, y_i + 1) \notin Cont$ and $d_i, d_{i-1} \in \{0, 1, 7\}$ or $d_i, d_{i-1} \in \{3, 4, 5\}$ (function $\Delta_2(d_1, d_2)$ is defined in table 1):

$$Cont = \begin{cases} Extern \Leftarrow \Psi(x_i, y_i, d_i, d_{i-1}) = +1 \\ Intern \Leftarrow \Psi(x_i, y_i, d_i, d_{i-1}) = -1 \end{cases} \quad (16)$$

$$\Psi(x, y, d_1, d_2) = \frac{f^B(x, y + \Delta_2(d_1, d_2)) - f^B(x, y - \Delta_2(d_1, d_2))}{f^B(x, y)} \quad (17)$$

This test enables the detection of any level of inclusions of clusters which might exist in the image. At the end of this step a set $SC = \{SC_1, \dots, SC_K\}$ of contours is obtained (K - number of clusters in the image).

Finally, external contours of clusters must be stored in counterclockwise direction and internal ones with the inverse direction.

B. Computing elementary arc segments

Once all clusters have been determined, the elementary arc segments are identified. An elementary arc is defined as a set of sequential points of a contour $Cont_{k,j}$ that are delimited by two adjacent contact points (a subset of corner points) between distinct elliptical regions.

There are currently two categories of methods that can be applied for corner point detection: (i) polygonal approximation methods [14] [15] and (ii) curvature based methods [16]. While the first ones try to fit a polygon, by minimizing the fitting error, to the curve, the second type of methods use a curvature measure (direct estimation, curvature estimation after gaussian filtering or scale-space curvature estimation) to compute maximum and minimum curvature points. Curvature based methods exhibit the advantage of their accuracy in corner point localisation. However, it is generally seen that their computing requirements are higher than for polygon approximation methods.

In this work, corner points are obtained by the incremental splitting polygonal approximation method [15] which is computationally very efficient. Let $p_i, p_j \in Cont$ be the two points in a contour with the maximum distance between them. To compute the corner points of $Cont$ the splitting method proceeds as follows:

1. Find point $p_z \in Cont$, such that p_z is a point between p_i and p_j on the contour, with

maximum distance to the chord defined by p_i and p_j .

2. If the distance between p_z and $\overline{p_i p_j}$ is greater than T (predefined threshold), then p_z is taken as a corner point and Goto step 3 else Stop.
3. Apply recursively the algorithm to segments $\widehat{p_i p_z}$ and $\widehat{p_z p_j}$.

The distance between $\overline{p_i p_j}$ and any point p_z is obtained with the use of the incremental distance: let $\{p_i, p_{i+1}, \dots, p_j\}$ be the ordered sequence of points of the contour between p_i and p_j , and $dist_z$ be the distance from p_z to $\overline{p_i p_j}$, then:

$$dist_z = dist_{z-1} + \delta(d_{z-1}) \quad (18)$$

$$dist_0 = 0$$

$\delta()$ - function defined in table 2. Note that $\delta(d) = -\delta((d+4) \bmod 8)$

d_i -chain code direction of point p_z to point p_{z+1}

After the calculation of all corner points, the contact points between the elliptical shapes (corner points which delimit convex regions) are identified with the application of the following criterion:

Convexity Criterion: elementary arcs are convex. Kim and Rosenfeld [17] have shown that an arc segment is convex if every point of a line segment, defined by any two points of that arc segment, lie inside the region of support.

Let $p_i \stackrel{def}{=} (x_i, y_i)$ be the i th corner point obtained by the incremental splitting method in contour $Cont_{k,j}$. Point p_i is considered to be a contact point if and only if its region of

support is non convex (it delimits two distinct convex regions), that is:

$$f^B(P_m) = black \quad (19)$$

where $f^B(P_m)$ represents the pixel gray level in the binary image at coordinates $P_m \stackrel{def}{=} (x_m, y_m)$; P_m represents the middle point of the line segment defined Δ points to the right and to the left of p_i (see figure 2 (a)). By this way, for each contour $Cont_{k,j}$ a set of contact points $CPS_{k,j} \stackrel{def}{=} \{CP_1^{k,j}, \dots, CP_{ncp}^{k,j}\}$ ($CP_i^{k,j}$ - i th contact point of $Cont_{k,j}$, ncp - number of contact points in contour $Cont_{k,j}$) is obtained. In figure 5 some results obtained with this method are shown. Thresholds T and Δ could be avoided by using a scale-space approach for contact point determination as in [18] (actually these values define indirectly the scale of each curve) or an adaptive scale method [19][20]. On the one hand, it is seen that the former methods are quite computationally intensive and that the later ones exhibit high noise sensibility [21]. On the other hand, it is verified that the aggregation algorithm is able to cope with incorrectly identified contact points, i. e., identification of contact points which are physically non existent. The algorithm is however not able to treat situations where physically relevant contact points have been missed. We, therefore, use threshold values which define a scale representation where all significant, and probably some non-significant, contact points are identified. Since contact points are defined by accentuated concave neighbourhoods it is verified that the number of identified contact points varies smoothly with scale definition.

An elementary arc segment (EAS) is formed by a subset of contiguous contour points belonging to the same contour that are bounded by two adjacent contact points, that is, for each $Cont_{k,j}$ ($p_z \in Cont_{k,j}$):

$$EAS_i^{k,j} \stackrel{def}{=} \{CP_i^{k,j}, \dots, p_z, \dots, CP_{i+1}^{k,j}\} \quad (20)$$

1. Characterization of elementary arc segments

Once all elementary arcs have been identified, a set of ellipse related measures (center, eccentricity) are computed for each *EAS*. These measures are later used in the elliptical shapes reconstruction procedure. In this section an extension to Fitzgibbon, Pilu and Fisher's [22] algorithm, used for the above mentioned measures estimation, is presented.

Given the general conic represented by an implicit second order polynomial

$$F(\mathbf{x}, \mathbf{a}) = \mathbf{x}\mathbf{a} = ax^2 + bxy + cy^2 + dx + ey + f = 0 \quad (21)$$

where $\mathbf{a} = \begin{bmatrix} a & b & c & d & e & f \end{bmatrix}$ and $\mathbf{x} = \begin{bmatrix} x^2 & xy & y^2 & x & y & 1 \end{bmatrix}^T$ the fitting is generally approached by minimizing the algebraic distance error:

$$Er = \sum_{i=1}^{np} F(\mathbf{x}_i, \mathbf{a})^2 = \|\mathbf{D}\mathbf{a}\|^2 \quad (22)$$

$$\mathbf{D} = \begin{bmatrix} \mathbf{x}_1 & \mathbf{x}_2 & \mathbf{x}_3 & \dots & \mathbf{x}_{np-1} & \mathbf{x}_{np} \end{bmatrix}^T \quad (23)$$

np – number of points in the curve

For highly irregular arc segments it is seen that the scatter matrix $\mathbf{S} = \mathbf{D}^T\mathbf{D}$ is very often near singularity which leads to inversion problems, since the matrix's condition number (defined as the ratio between the largest singular value of S to the smallest) is usually higher than 10^{16} . For double precision floating point representation to prevent a roundoff error the condition number should be less than 10^{12} . Therefore, direct estimation using equation 22 is avoided. In our approach the ellipse center is first estimated using a gradient descent method. At this step the arcs are assumed to be circular. Each estimated center is then corrected using a set of geometrical properties. Using the estimated center as a constraint, the singularity source in S is reduced (a redefinition of S which exhibits much smaller condition numbers,

usually in the order of 10^6 , is introduced) and equation 22 is applied to compute the intended parameters.

The initial estimation of the centers is performed with Landau's algorithm [23] [24]. Given an elementary arc segment $EAS_i^{k,j} \stackrel{def}{=} \{(x_1, y_1), \dots, (x_z, y_z), \dots, (x_{np}, y_{np})\}$ ($CP_i^{k,j} \stackrel{def}{=} (x_1, y_1)$; $CP_{i+1}^{k,j} \stackrel{def}{=} (x_{np}, y_{np})$; np - number of contour points in the elementary arc segment), for each point a radial vector \vec{r}_i is defined. If a circular arc of radius R and center $\vec{a} \stackrel{def}{=} (x_a, y_a)$ is outlined, then its radius and center may be computed iterating equations 24 and 25.

$$\vec{a}_{v+1} = \frac{1}{np} \sum_{w=1}^{np} \left[\vec{r}_w - R_v \vec{l}_{\vec{r}_w} - \vec{a}_v \right] \quad (24)$$

$$R_v = \frac{1}{np} \sum_{w=1}^{np} |\vec{r}_w - \vec{a}_v| \quad (25)$$

$\vec{l}_{\vec{r}_w}$ - unity vector with the same direction as \vec{r}_w

\vec{a}_i - estimated center vector in the i th iteration

R_i - estimated radius in the i th iteration

For the initial center and radius the method described in [24] is applied: let $P_1 \stackrel{def}{=} (x_1, y_1)$ be the middle point of the $EAS_i^{k,j}$, and $P_2 \stackrel{def}{=} (x_2, y_2)$ the middle point of the chord defined by points $CP_i^{k,j}$ and $CP_{i+1}^{k,j}$ (see figure 2 (b)), the initial center \vec{a}_0 and radius R_0 are obtained by

$$\varsigma = \tan^{-1} \left(\frac{y_2 - y_1}{x_2 - x_1} \right) \quad (26)$$

$$R_0 = \frac{\left(\frac{D}{2}\right)^2 + h^2}{2h} \quad (27)$$

$$\vec{a}_0 = (x_2 + (R_0 - h) \cos(\varsigma), y_2 + (R_0 - h) \sin(\varsigma)) \quad (28)$$

D, h - computed as shown in figure 2 (b)

In the proposed method equations 24 and 25 are iterated until the required precision Tp (we use $Tp = 0.1$) is obtained at a maximum number ϵ of iterations (we use a fixed value of $\epsilon = 100$). Therefore the stop condition is given by:

$$|\vec{a}_{v+1} - \vec{a}_v| < Tp \text{ or } v > \epsilon \quad (29)$$

Tp - predefined precision

These threshold values are not critical, since centers are further corrected using a set of geometrical properties as will be explained below.

It is seen that Landau's method produces satisfactory results if the *EAS* are smooth and approximately circular. However, given the irregular shapes which characterize most of the regions, the number of points of each elementary arc segment is often reduced. It is also common to find several *EAS* with almost linear shapes. These two factors are responsible for abnormal centers identification (see figure 5 (a)). To overcome this problem, a set of correcting steps is introduced. These steps are based on the following geometrical properties:

Property 1: Let θ be the internal angle between \overline{OP} and \overline{PW} (see figure 3). Let \overline{PW} be the line segment normal to the tangent line segment \overline{PK} at point p and η be the length relation between the ellipse's axes as defined in (92). It is seen that the maximum value of θ for $p \in ellipse$ is given by (30) (see proof 4 in appendix):

$$\theta_{\max} = \max_{\alpha \in [0, \frac{\pi}{2}]} \{\theta(\alpha)\} = \tan^{-1} \left(\frac{\eta - 1}{2\sqrt{\eta}} \right) \quad (30)$$

For lime granules usually $\eta \in [0.5, 2]$, hence, $\theta_{\max} \in [-19.5, 19.5]$ degrees. However, given the irregular shapes of most EAS , a maximum deviation of ζ degrees (it is suggested to take $\zeta = \pm 30^\circ$) is allowed. The fulfillment of this property is imposed by the following procedure: if $\theta > \zeta$ then vector $\overrightarrow{P_1 a}$ is rotated, with respect to point P_1 , by $\varphi = -\theta$ degrees, that is:

$$a = \begin{bmatrix} \cos \varphi & -\sin \varphi \\ \sin \varphi & \cos \varphi \end{bmatrix} [a - P_1] + P_1 \quad (31)$$

Property 2: The center of an ellipse lies always inside the region bounded by its border contour. Therefore, the estimated center a must lie inside its region of support R^k (see figure 4) of segment $EAS_t^{k,j}$. This property is verified and its fulfillment is imposed to the identified elementary arc segments through the following steps: (i) determine point P_5 ($P_5 = P_1 - \overrightarrow{P_1 a}$) as shown in figure 4; (ii) if $|\overrightarrow{P_1 a} \cap R^k| < |\overrightarrow{P_1 P_5} \cap R^k|$ then apply equation 31 with $\varphi = \pi$; (iii) if $f^B(x_a, y_a) \notin R^k$ (outside the object) then determine line segment $\overline{P_1 P_3}$ such that $\angle(\overline{P_1 P_3}) = \angle(\overline{P_1 a})$ and $\overline{P_1 P_3} \subset R^k$, that is, determine point P_3 such that the obtained line segment $\overline{P_1 P_3}$ (same slope as $\overline{P_1 a}$) lies entirely inside the region of support R^k (see figure 4); (iv) obtain the new center a as in equation 32 ($|\overrightarrow{r} \cap R|$ represents the length of vector \overrightarrow{r} that lies inside region R).

$$a = \frac{1}{2}(P_1 + P_3) \quad (32)$$

To compute point P_3 the following method can be applied: if line segment $\overline{P_1 P_4}$ intercepts a region R_k , then it is seen that there exists at least one point of R_k 's contour ($Cont_k$) which belongs to line segment $\overline{P_1 P_4}$. A point belongs to a line segment if two conditions are verified: (i) the point lies on the line defined by the line segment, and (ii) the point is situated between that line's segment start and end points. If R_k 's contour is considered a

continuous parametric function, condition (i) implies that:

$$\exists p_z \in Cont_k : \rho - f(t) \cos(\psi) - g(t) \sin(\psi) = 0 \quad (33)$$

$$Cont_k = \begin{cases} x = f(t) \\ y = g(t) \end{cases}$$

$$p_z = (f(t), g(t))$$

$$\rho = x \cos(\psi) + y \sin(\psi) \text{ - polar equation of the line defined by } \overline{P_1P_4}$$

Since, however, R_k 's contour is discrete (finite set of pixels of coordinates $p_z = (x_z, y_z)$), the equality of equation 33 is only obtained under certain conditions. Therefore, instead of equation 33 it is considered that point p_z lies on the line segment defined by $\overline{P_1P_4}$ if $\overline{P_1P_4}$ intercepts pixel of coordinates p_z , that is:

$$|\Delta\rho_z| = |\rho - x_z \cos(\psi) - y_z \sin(\psi)| \leq \sqrt{0.5} \quad (34)$$

An efficient implementation of equation 34 is suggested in [15]: let $\Delta\rho_z$ be the Euclidean distance between the line defined by $\overline{P_1P_4}$ and pixel p_z of contour $Cont_k$, then:

$$\Delta\rho_1 = \rho - x_1 \cos(\psi) - y_1 \sin(\psi) \quad (35)$$

$$\Delta\rho_{z+1} = \Delta\rho_z + \delta(d_z) \quad (36)$$

$\delta()$ - function defined in table 2

d_z - chain code direction of pixel $p_z \in Cont_k$

As for condition (ii) it is seen that p_z lies between the endpoints of line segment $\overline{P_1P_4}$ if equation 37 is verified [25] (function $\Gamma()$ is defined in equation 62).

$$\Gamma(p_{\gamma+}, p_{\gamma-}, a) \Gamma(p_{\gamma+}, p_{\gamma-}, P_4) < 0 \quad (37)$$

$$p_{\gamma\pm} = \left(x_z \pm \gamma, \frac{(x_z \pm \gamma) \sin(\psi) - \rho}{\cos(\psi)} \right), \gamma \in \mathbb{R} \setminus \{0\} \quad (38)$$

Property 3: For those arc segments which are perfectly elliptical, it is verified that the modulus of the identified radius is exactly half the modulus of the diameter with the direction of the vector defining the radius. Given the irregular shapes of regions in natural scenes, it is frequently seen that abnormal values are identified. To correct this problem, minimum values for radius (R) are imposed, that is, $R \geq \delta |\overline{P_3a}|$, $\delta \in]0, 1[$. The several identified R values are limited in the upper bound by a function of the diameter's modulus with the direction of $\overline{P_1a}$. These conditions are imposed by the following procedure: (i) determine line segment $\overline{P_1P_3}$ as explained in property 2; (ii) rotate vector $\overrightarrow{aP_1}$ (vector whose origin is the identified center) by $\alpha = \pi$ radians with respect to point a . In this way point P_4 is determined (see figure 4); (iii) if $R < \delta |\overline{P_3a}|$ or $P_4 \in \overline{P_3a}$ and $|\overline{P_3a}| < \delta |\overline{P_1a}|$ then obtain the new center as in equation 32 (in our implementation $\delta = 0.1$; all shown results where obtained with this value).

In figure 5 several results obtained by Landau's method and with the described correction algorithm are presented. As can be observed, this simple algorithm enables the corrections of the centers in those situations where Landau's method fails.

To compute the intended eccentricity measure, the function $F(\mathbf{x}, \mathbf{a})$ has to be estimated for each elementary arc segment. In order to fit an ellipse to each of the elementary arc segments we use a least square approach with the following constraints (constraint 3 assures that the estimated function corresponds to an ellipse [22]; (x_a, y_a) - identified center):

1. Constraint 1: $x_a = \frac{-2cd+bc}{4ac-b^2}$

2. Constraint 2: $y_a = \frac{-2e+bd}{4ac-b^2}$

3. Constraint 3: $4ac - b^2 = 1$

Substituting constraints 1 and 2 into equation 21 it is seen that:

$$\begin{cases} x_1 = x - x_a \\ y_1 = y - y_a \end{cases} \quad (39)$$

$$F(\mathbf{x}_1, \mathbf{a}_1) = \mathbf{a}_1 \mathbf{x}_1 = ax_1^2 + bx_1y_1 + cy_1^2 + f_1 = 0 \quad (40)$$

where $\mathbf{a}_1 = \begin{bmatrix} a & b & c & f_1 \end{bmatrix}$ and $\mathbf{x}_1 = \begin{bmatrix} x^2 & xy & y^2 & 1 \end{bmatrix}^T$. Following Fitzgibbon, Pilu and Fisher [22], the fitting problem is:

$$\min(Er) = \min(\|\mathbf{D}\mathbf{a}_1\|^2) \text{ subjected to the quadratic constraint } \mathbf{a}_1^T \mathbf{C}\mathbf{a}_1 = 1 \quad (41)$$

$$\mathbf{C} = \begin{bmatrix} 0 & 0 & 2 & 0 \\ 0 & -1 & 0 & 0 \\ 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (42)$$

Applying the Lagrange multiplier λ , the solution of equation 41 is equivalent to:

$$2\mathbf{D}^T \mathbf{D}\mathbf{a}_1 - 2\lambda \mathbf{C}\mathbf{a}_1 = 0 \quad (43)$$

$$\mathbf{a}_1^T \mathbf{C}\mathbf{a}_1 = 1$$

which may be rewritten as:

$$\mathbf{S}\mathbf{a}_1 = \lambda \mathbf{C}\mathbf{a}_1 \quad (44)$$

$$\mathbf{a}_1^T \mathbf{C}\mathbf{a}_1 = 1 \quad (45)$$

$$\mathbf{S} = \mathbf{D}^T \mathbf{D} \quad (46)$$

The solution to this system may be computed using the generalized eigenvectors. The solution which yields the lowest residual $\widehat{\mathbf{a}}_i^T \mathbf{S} \widehat{\mathbf{a}}_i = \lambda_i$ should be taken, where $\widehat{\mathbf{a}}_i = \mu_i \mathbf{u}_i$ and

$$\mu_i = \sqrt{\frac{\lambda_i}{\mathbf{u}_i \mathbf{S} \mathbf{u}_i^T}} \quad (47)$$

It should be noted that the scatter matrix \mathbf{S} is now invertible, since the singularity problems (see figure 8 for a comparison between condition numbers) are avoided by taking out of \mathbf{D} the x and the y columns. In figure 6 the centers and ellipses estimated with the proposed method are shown and in figure 7 a comparison of center identification using this approach and the methods of Fitzgibbon, Pilu and Fisher [22] and Wu and Wang [26] is presented.

III. RECONSTRUCTION OF ELLIPTICAL REGIONS USING A NEURAL NETWORK AND GEOMETRICAL DECISION CRITERIA

After characterization of elementary arc segments, a new stage is initiated to reconstruct the elliptical shapes from the identified *EAS*. This task is accomplished through an aggregation procedure between *EAS* based on *a posteriori* probability analysis, computed with a neural network, and a set of geometrical decision criteria. These criteria are used both for measures definition (applied to compute a feature vector which is used as the input to the neural network) and as decision criteria which are applied in the aggregation method.

2. Geometrical decision criteria

According to Gestalt psychologists, visual perception is highly influenced by properties such as proximity, continuity, similarity, closure and symmetry [5]. Through the analysis of some of the geometrical properties of the elementary arc segments a subset of these properties are used to establish aggregation validation criteria and several measures which allow to quantify some of these properties. Namely the aggregation point and the shared adjacency

criteria are inspired on the continuity property. These criteria enable the method to assert if continuity is physically plausible. The proximity property is measured by the proximity and the concentricity criteria, while closure influenced the definition of the completeness and the center position criteria. Similarity is quantified by the curvature criterion.

Aggregation Point Criterion: the aggregation between two arc segments $EAS_i^{k,j}$ and $EAS_q^{k,l}$ must preserve the contour direction. When this operation is performed, it should affect the pair of contact points which are closest to each other. Since internal contours are stored in the clockwise direction and external contours are stored in the inverse direction, to fulfil this criterion, it is seen that, the aggregation may only be performed between contact points $(CP_i^{k,j}, CP_{q+1}^{k,j})$ or $(CP_{i+1}^{k,j}, CP_q^{k,j})$. If other combinations were chosen, it would not be possible to assure the contour direction preservation of the resulting segment.

Proximity Criterion: contact points of sequential elementary arc segments belonging to the same object tend to be localized at smaller distances from each other than to other contact points. This criterion leads to the definition of an aggregation distance measure: let $EAS_i^{k,j}$ and $EAS_q^{k,l}$ be two aggregation candidates, their aggregation distance measure is defined by:

$$D_{i,q}^{1k,j,l} = \min \left(\left| \overline{CP_i^{k,j} CP_{q+1}^{k,j}} \right|, \left| \overline{CP_{i+1}^{k,j} CP_q^{k,j}} \right| \right) \quad (48)$$

The aggregation candidate of $EAS_i^{k,j}$ is $EAS_q^{k,l}$ if and only if (N - number of EAS in cluster k):

$$q = \text{MinArg}_{z=1,\dots,i-1,i+1,\dots,N} \left\{ D_{i,z}^{1k,j,l} \right\} \quad (49)$$

Concentricity Criterion: let $EAS_i^{k,j}$ and $EAS_q^{k,l}$ be two convex arc segments with centers a_1 and a_2 , respectively. They belong to the same elliptical object if they are concentric, that is, their centers lie in relative proximity. This criterion leads to the definition of second

aggregation distance measure:

$$D_{i,q}^{2k,j,l} = |\overline{a_1 a_2}| \quad (50)$$

Completeness Criterion: Almost complete arc segments tend to aggregate with less complete arc segments. A completeness measure for an $EAS_i^{k,j}$ arc can be defined by (see figure 2 (b)):

$$C_i^{k,j} = \frac{Radius}{Chord\ Height} = \frac{|\overline{P_1 a}|}{|\overline{P_1 P_2}|} \quad (51)$$

It is seen that $C_i^{k,j} \in]0, \infty[$.

The introduced proximity, concentricity and completeness criteria enable the introduction of an “energy measure” for the resulting aggregation between two EAS . This measure should be inversely proportional to the probability of both EAS belonging to the same elliptical object. It is seen that this probability decreases whenever the measures of equations 48, 50 and the inverse of equation 51 increase. Therefore, this measure is defined by:

$$E_{i,q}^{k,j,l} = \left(D_{i,q}^{1k,j,l} + D_{i,q}^{2k,j,l} \right) * \left(1 + \frac{1}{C_i^{k,j}} + \frac{1}{C_q^{k,l}} \right) \quad (52)$$

In [27] and [21] the perceptual grouping problem is formulated as an energy minimisation problem, where the energy function includes, as in this case, terms related to closure and proximity. In our method the energy definition is given by the sum between the distance measures plus the sum of the closure measures weighted by the sum of the distance measures. In practice it is observed that the proximity measures are much higher than the closure measures. Therefore, this weighted sum is used to avoid almost complete arcs to exhibit small energy measure. Although, theoretically, almost complete arcs may exhibit values of C which may tend to 0, in practice it is observed that these values are usually greater than 0.5. If the simple summation would be applied, E would be dominated mainly by the proximity

terms. However, by taking the weighted terms, E is increased, in these cases, between 2 and 4 times its values due to the proximity measures alone. As for the less complete arcs, it is seen that the sum of the inverse closure measures is usually less than 1, leading to a domination of E by the proximity measures. Therefore, E will increase for almost complete arcs and will decrease (for similar proximity values) whenever in the presence of combinations between incomplete arcs or between almost complete and very incomplete arcs.

Curvature Similarity Criterion: If two arcs belong to the same elliptical object, then they should exhibit similar curvatures. The similarity measure should account for geometrical similarity and for scale similarity. A suitable measure is $(b, c, f_1$ as computed in equation 40)

$$\Omega = \Theta Ec \quad (53)$$

$$Ec = \begin{cases} \sqrt{\eta^2 - 1} \Leftarrow \eta > 1 \\ \sqrt{1 - \eta^2} \Leftarrow \eta < 1 \end{cases} \quad \text{-- eccentricity} \quad (54)$$

$$\Theta = \begin{cases} A \Leftarrow \eta > 1 \\ B \Leftarrow \eta < 1 \end{cases}, \eta = \frac{A}{B} \quad (55)$$

$$A = \sqrt{\frac{-f_1}{a'}} \quad (56)$$

$$B = \sqrt{\frac{-f_1}{c'}} \quad (57)$$

$$a' = \cos^2 \xi + b \cos \xi \sin \xi + c \sin^2 \xi \quad (58)$$

$$c' = \sin^2 \xi - b \cos \xi \sin \xi + c \cos^2 \xi \quad (59)$$

$$\xi = \frac{1}{2} \cot^{-1} \left(\frac{1-c}{b} \right) \quad (60)$$

Let $EAS_i^{k,j}$ and $EAS_q^{k,l}$ be two aggregation candidates with Ω defined respectively by Ω_1 and Ω_2 , then their similarity measure is defined by:

$$CES_{i,q}^{k,j,l} = \max \left\{ \frac{\Omega_1}{\Omega_2}, \frac{\Omega_2}{\Omega_1} \right\} \quad (61)$$

Reciprocity Criterion: if arc segment $EAS_i^{k,j}$ is adjacent to arc segment $EAS_q^{k,l}$, then the inverse is also true.

In the proposed aggregation algorithm, this criterion is used to eliminate aggregations whenever the aggregation choice is not mutual. That is, the aggregation between arcs $EAS_i^{k,j}$ and $EAS_q^{k,l}$ only takes place if the first chooses the second as its preferred aggregation candidate and vice-versa.

Shared Adjacency Criterion: if two arc segments $EAS_i^{k,j}$ and $EAS_q^{k,l}$ are adjacent with aggregation points $(CP_i^{k,j}, CP_{q+1}^{k,j})$, then it is also verified that $EAS_{i-1}^{k,j}$ and $EAS_{q+1}^{k,l}$ are adjacent by the same set of points and should therefore be aggregated together. However, if the aggregation set of points is given by $(CP_{i+1}^{k,j}, CP_q^{k,j})$, then the same property is verified but in relation to arcs $EAS_{i+1}^{k,j}$ and $EAS_{q-1}^{k,l}$. This criterion enables the correction of some aggregation situations, namely, when less correct identified centers lead to less well defined outputs from the neural network.

Note that whenever an aggregation is performed it is implied that a physical connection exists between the two EAS as well as between their adjacent EAS which share the same contact points used for aggregation. Therefore, there always exists a redundancy in data for performing the aggregation decision. As can be observed in figure 6, EAS with almost linear shapes may lead, under certain circumstances, to biased estimations of centers. These biases result in less correct D^2 , CES and E measures, which in turn may saturate the output of the

neural network classifier. This effect is minimised by training the neural network explicitly with all the defined features to enable it to learn a noise tolerant behaviour. However, classification errors may occur. In these cases, data redundancy is used, as will be explained later, to recover from these situations.

Center Position Criterion: if two *EAS* (let them be EAS_1 and EAS_2 with centers a_1 and a_2 , respectively) aggregate together, then their centers must lie on the same side of the line segment defined by the set of contact points obtained with the aggregation point criterion (let it be $\{CP_1, CP_2\}$). This is mathematically expressed by:

$$\Gamma (CP_1, CP_2, a_1) \Gamma (CP_1, CP_2, a_2) > 0 \quad (62)$$

$$\Gamma (p_1, p_2, p_3) = (x_2 - x_1)(y_3 - y_1) - (x_3 - x_1)(y_2 - y_1) \quad (63)$$

This is another closure motivated criterion. Note that whenever (62) is not verified, the two *EAS* are not sufficiently "faced" to each other, in order to be considered for aggregation.

3. The neural network classifier

It is intuitively known that if two *EAS* exhibit appropriate values of the measures in equations 48, 50, 51, 52 and 61, they belong to the same elliptical region. However, the exact nature of the relationship among the above mentioned measures is unknown. Neural networks are well known for their capability to learn non linear relationships between input and output data. In this work we use a neural network to compute the *a posteriori* aggregation probability between all pairs of *EAS* of a cluster of elliptical regions. Basically, the purpose is to assure that objects with similar characteristics may be assigned and cataloged accordingly to a predefined set of classes.

In recent years, neural networks have been used as pattern classifiers in several fields

of application. Their massively parallel distributed architecture and their ability to learn from experience makes them adequate for classification and function approximation problems which are tolerant to noise but to which hard rules cannot be easily applied. These classifiers often provide reduced error rates when compared to conventional statistical approaches and are a powerful and flexible means for mapping a fixed number of inputs into a set of discrete classes. It has been proven by several researchers [28][29] that given infinite training data, consistent neural classifiers trained using least square error minimisation approximate the Bayesian decision boundaries to arbitrary precision. If they are reasonably well trained, network outputs are expected to approximate the corresponding *a posteriori* class probabilities and can be summed to one. Estimation accuracy depends on the network's complexity, the amount of training data, and the degree to which training data reflect true likelihood distributions and *a priori* class probabilities.

Usually, classifiers are seen as a three block system, where the first two blocks are respectively responsible for creating a measures vector and a feature vector which can then be applied in the classification phase (third block). The first two blocks have been described in the preceding section. We will now focus our analysis on the last block. Its task is to map an input vector $\mathbf{x}_C \in \mathbb{R}^G$ into a symbol $C(\mathbf{x}_C) \in \mathcal{K} \stackrel{def}{=} \{\omega_1, \omega_2, \dots, \omega_{N_\kappa}\}$, which divides the G -dimension Euclidean space in N_κ regions, so that each region corresponds to a class ω_i . Since our goal is to identify the *EAS* that aggregate from those who do not, we simply have $N_\kappa = 2$. For the Bayes optimum decision, a vector \mathbf{x}_C is assigned to class i if $P(\omega_i|\mathbf{x}_C) > P(\omega_j|\mathbf{x}_C), \forall j \neq i$, so the Bayes optimal boundary is the loci of all points $\mathbf{x}_C^* : P(\omega_i|\mathbf{x}_C^*) - P(\omega_j|\mathbf{x}_C^*) = 0$ for a two-class problem. If we define the neural classifier as a function $\Pi : \mathbb{R}^G \rightarrow \mathbb{R}^V$, where \mathbf{x}_C^i are the input or feature vectors of G dimensions and \mathbf{y}^i the target vectors of V dimensions, such that $\mathbf{y}^i \in \mathcal{T} \stackrel{def}{=} \{\Theta^1, \Theta^2, \dots, \Theta^{N_\kappa}\}$, each one

codifying one of the $N_{\mathcal{K}}$ possible classes, then we can estimate \mathbf{y} , such that $\hat{\mathbf{y}} = \Pi(\mathbf{x}_C)$. Each component of $\hat{\mathbf{y}}$ can be obtained in such a way that

$$\hat{y}_i \approx \Pi_i^*(\mathbf{x}_C) = \sum_{j=1}^{N_{\mathcal{K}}} \Theta_i^j P(y_i = \Theta_i^j | \mathbf{x}_C) \quad (64)$$

where Π^* is the estimator that minimizes the Euclidean mean squared error, *i.e.*, the optimum estimator. If the $\Theta^i \in \mathcal{T}$ vectors are of $N_{\mathcal{K}}$ dimensions and if its component i clearly identifies the ω_i class, that is,

$$\Theta_i^i \neq \Theta_i^k, \quad \forall i \in \{1, \dots, N_{\mathcal{K}}\}, \forall k \in \{1, \dots, N_{\mathcal{K}}\} \setminus \{i\} \quad (65)$$

then the last term of equation 64 can be simplified and

$$\hat{y}_i \approx \Theta_i^1 P(\omega_1 | \mathbf{x}_C) + \dots + \Theta_i^{N_{\mathcal{K}}} P(\omega_{N_{\mathcal{K}}} | \mathbf{x}_C), \quad \forall i \in \{1, \dots, N_{\mathcal{K}}\} \quad (66)$$

that is, the outputs are the estimates of the linear combinations of *a posteriori* probabilities of the $N_{\mathcal{K}}$ classes.

One of the main problems one can encounter when designing a multilayer feedforward neural network is related to the number of hidden layers to use and the number of neurons per layer. Several thoroughly discussions can be found in [30][31][32]. In order to reduce the probability of getting stuck in a local minimum the network was tuned using the *momentum* with an adaptive learning rate algorithm from [33] with a training set composed by 841 pairs of *EAS*. It was considered, for training purposes, that

$$AGP_{i,q}^{k,j,l} = \begin{cases} 0.9 \Leftarrow EAS_i^{k,j}, EAS_q^{k,l} \in \text{same ellipse} \\ 0.1 \Leftarrow EAS_i^{k,j}, EAS_q^{k,l} \notin \text{same ellipse} \end{cases} \quad (67)$$

where $AGP_{i,q}^{k,j,l}$ is the *a posteriori* aggregation probability between $EAS_i^{k,j}$ and $EAS_q^{k,l}$. We assume that the application maps the classes into target vectors such that $\omega_1 \rightarrow 0.1$ and $\omega_2 \rightarrow 0.9$ (0 and 1 could be used instead; however, as pointed out in [34] this should be

avoided since sigmoids will exhibit these values only when their inputs are very large), and therefore we have $V = 1$. Replacing these results in equation 66 comes

$$\hat{y}_1 \approx 0.1P(y_1 = 0.1|\mathbf{x}_C) + 0.9P(y_1 = 0.9|\mathbf{x}_C) = 0.1P(\omega_1|\mathbf{x}_C) + 0.9P(\omega_2|\mathbf{x}_C) \quad (68)$$

Since the classes are mutually exclusive and since they use up all classification possibilities, it is seen that $P(\omega_1|\mathbf{x}_C) = 1 - P(\omega_2|\mathbf{x}_C)$, and therefore we can rewrite this equation as:

$$\hat{y}_1 \approx 0.1 + 0.8P(\omega_2|\mathbf{x}_C) \quad (69)$$

The neural network inputs \mathbf{x}_C reflect several geometrical measures of each *EAS* as well as joined measures of pairs of *EAS* (figure 9). The input vector for each pair of *EAS* ($EAS_i^{k,j}, EAS_q^{k,l}$) is composed by six measures: $D_{i,q}^{1k,j,l}$, $D_{i,q}^{2k,j,l}$, $E_{i,q}^{k,j,l} \cdot \ln(2 + ord(D_{i,q}^{1k,j,l}))$, $CES_{i,q}^{k,j,l}$, $C_i^{k,j}$ and $C_q^{k,l}$ ($ord(D_{i,q}^{1k,j,l})$ is a function which returns the number of possible aggregation combinations with $EAS_i^{k,j}$ with less D^1 measure than $E_{i,q}^{k,j,l}$). These measures are all normalized to the interval $[0, 1]$.

Several neural networks with different architectures have been trained and tested: networks with one and two hidden layers, with different numbers of neurons as well as different activation functions in the hidden layer, although all from the *sigmoid* type since we needed to learn the nonlinearities in the data. The *sigmoid* activation function of the output layer has been kept constant and of type *logsig*, so that the network outputs would be in the interval $[0; 1]$. Table 3 shows the best training results obtained with different networks.

The simplest architecture was chosen, composed by 6 input, 9 hidden and 1 output neurons, all with *logsig* type activation functions. Although, in theory, equation 49 should hold, in practice, if we have noisy pairs of *EAS* then it is seen that the $AGP_{i,q}^{k,j,l}$ may exhibit values different from the ideal ones. Therefore we say that an output n is wrong if $A(n) > 0.3$ and $T(n) = 0.1$, or $A(n) < 0.6$ and $T(n) = 0.9$, where A is the output matrix and T the target

matrix. The interval $[0.3;0.6]$ is, therefore, considered to be indefinite. Table 3 shows the results obtained with the chosen neural network ($N(6, 9, 1)$ with *logsig* activation functions) as well as other tested networks. An error of 0.433% was obtained with the validation set (231 *EAS*) for a network trained only with artificial *EAS* (this was performed to test the neural network’s generalisation capabilities). This error became smaller by tuning the neural network with a training set of *EAS* of real lime granules.

4. Aggregation of elementary arc segments using a posteriori aggregation probabilities and geometrical decision criteria

Although the *a posteriori* aggregation probabilities obtained by the neural network are reliable, it is seen that, for highly irregular pairs of *EAS* incorrect values may be obtained. To solve these problems a method based on a subset of the mentioned decision criteria and a voting strategy for critical boundary situations (when an aggregation candidate set of *EAS* exhibit conflicting *AGP* values among some of them) is developed. In this method, the aggregation point and the proximity criteria are always applied to determine the pairs of aggregation candidates. These pairs are validated using the reciprocity (this criterion is applied using the neural networks output values) and the center position criteria. Whenever an aggregation is performed between a pair of *EAS* the shared adjacency criterion is applied to their neighbors.

Using the neural network outputs, we assume, for decision purposes, that a pair of *EAS* belongs to the same elliptical region if its *a posteriori* aggregation probability degree is greater than 0.5. Since we only have two classes, each *EAS* should be classified in such a way that verifies $P(\omega_j|\mathbf{x}_C) > 0.5, \forall \omega_j \in \mathcal{K}$, that is, it should be assigned to the class with

the highest *a posteriori* probability. Replacing this result in equation 69 comes

$$\frac{\hat{y}_1 - 0.1}{0.8} > 0.5 \iff \hat{y}_1 > 0.5 \quad (70)$$

For each cluster k of identified elliptical regions a matrix $A_{N \times N}$ (N - number of *EAS* in cluster k)

$$A_{N \times N} [i, q] = AGP_{i,q}^{k,j,l} \quad (71)$$

is constructed. It should be noted that this matrix is symmetrical with respect to its diagonal, since $AGP_{i,q}^{k,j,l} = AGP_{q,i}^{k,j,l}$. From $A_{N \times N}$ the absolute maximum value $A_{N \times N} [i, q]$ is obtained, which represents the strongest *a posteriori* probability between two *EAS*. For each of identified *EAS*, their relative maximum values are searched in $A_{N \times N}$, that is, if pair $\{EAS_i^{k,j}, EAS_q^{k,l}\}$ has been found to have the absolute maximum *AGP* value in $A_{N \times N}$ then segments $EAS_w^{k,z}$ and $EAS_y^{k,x}$ are searched, such that:

$$\begin{cases} A_{N \times N} [i, w] > 0.5 \\ A_{N \times N} [q, y] > 0.5 \end{cases} \quad (72)$$

where $w, y \in \{1, 2, \dots, N\}$, $w \neq i$ and $y \neq q$. Whenever a new relative maximum has been found, the described search procedure is repeated with the obtained *EAS*. All *EAS* obtained with (72) are organised in a tree structure. The tree's root is formed by $\{EAS_i^{k,j}, EAS_q^{k,l}\}$ and the first level of nodes is composed by all *EAS* such that (72) is verified. Nodes are appended to the tree only once, even when they are chosen multiple times. The second level of tree nodes are obtained using the same procedure and by substituting q or i in (72) by the node's *EAS* number. These trees will be used to implement an hierarchical voting scheme to assert majority. Whenever a set of *EAS* are candidates for a given elliptical structure it may occur that, due to misclassification, there exist conflicting *AGP* values among them. In these cases, the aggregation will be considered with such an *EAS* if the majority of the candidate

EAS exhibit correct *AGP* values regarding this arc segment. However, it is seen that if the arc does not physically belong to the ellipse under analysis it will probably add several *EAS*, i. e., in the case it defines a different ellipse which is constituted by several *EAS*, to the tree. Therefore, the misclassified *EAS* could easily win majority. To avoid these situations, majority for a node of level z is computed only with the tree's nodes of level $w \leq z$ if $z > 1$ or $w \leq 1$ if $z \leq 1$, $w \in Z_0^+$. Each tree defines a candidate set, noted here by *CAS*, of *EAS* for an elliptical structure. Once a tree has been computed the aggregation algorithm is initiated. A flowchart of the aggregation method is exhibited in figure 11. From figure 11 it can be observed that if the number of *EAS* in *CAS* is 2 (only one pair of candidates for aggregation exists), then the aggregation point and the proximity criteria are applied to compute the aggregation points. With these points the center position criterion is tested and, if the test is positive, both *EAS* are aggregated and the shared adjacency criterion is applied to establish the aggregation between their neighbors; else the aggregation decision is postponed (it may happen that another *EAS* chooses one of this candidates for aggregation). However, if the number of *EAS* \in *CAS* is greater than 2, then the reciprocity criterion must be verified for all possible pairs of *EAS* \in *CAS*. If this condition holds, then the aggregation point and the proximity criteria are used to establish the best aggregation pair (with the smallest D^1 measure between each pair). The aggregation operation is only performed if the center position criteria is verified for the identified pair $\{EAS_j, EAS_i\} \in CAS$, in which case the new pair forms a new *EAS*. This procedure is repeated until only one *EAS* exists in *CAS*. Once again, if the aggregation conditions are met then the shared adjacency criterion must be executed. However, if, at least, one *EAS* of the aggregation set *CAS* does not fulfil the reciprocity criterion with respect to another *EAS* \in *CAS*, then the aggregation cannot be performed without further information. To solve these situations, a voting strategy is used.

Each $EAS_i \in CAS$ votes once for each $EAS_q \in CAS$ of level $w \leq z$ if $z > 1$ or $w \leq 1$ if $z \leq 1$ whenever $AGP_{iq}^{kjl} < 0.5$. After the voting operation all EAS which have received more than 50% of the votes are excluded from the aggregation set. For each EAS_i with less than 50% of the votes, the proximity criterion is applied to obtain its aggregation candidate EAS_q . If its candidate exhibits 0 votes then it is a strong aggregation candidate and, therefore, the center position criterion should be applied to confirm the aggregation. Otherwise, if EAS_q has received at least 1 vote, then it is an unreliable candidate. In this case, its neighbors, defined by their aggregation points (see the shared adjacency criterion), must be verified for all the criteria and if the conditions are met, then, by applying the shared adjacency criterion, segments EAS_i and EAS_q are aggregated. Note that whenever it is not possible to assure a correct aggregation decision between a pair of EAS , the decision process is carried out with their neighbor information. This redundancy has proven to be very robust, as noisy data can be rejected with the voting and neighborhood analysis strategies.

Figure 12 (a) shows a cluster of elliptical regions. For simplicity, only the granule formed by segments $\{4, 9, 10, 14, 19\}$ will be considered. The AGP values for these segments are listed in figure 13 (a). Three misclassification values have been intentionally introduced to test the aggregation algorithm's performance with noise. Assume that the new absolute maximum value has been computed for segments 4 and 9. According to the algorithm, their relative maximum values are computed. From figure 13 (a) it is seen that the reciprocity criterion does not hold for all segments. In consequence, we proceed with the voting system. Segments 4 and 10 have 1 vote each, segment 19 has 2 votes and segments 9 and 14 have 0 votes, as can be derived from figure 13. The first segment of CAS is segment 4. Since it has less than 50% of the votes, we apply the aggregation point criterion to its contact points. Segment 14 is chosen as a possible candidate. Since it has 0 votes, we apply the center

position criterion and if it holds we aggregate the two *EAS*. The shared adjacency criterion is executed and segments 3 and 15 are aggregated. Since the contour direction must be preserved, the next *EAS* chosen is segment 19, despite its 2 votes against. This implies that we must check its neighbours (segments 5 and 18). Since they aggregate, segments 4 and 19 must also aggregate (shared adjacency criterion). The process is then repeated for all the other segments of the *CAS* set (figure 13 (b)) and, finally, a new absolute maximum value is computed for the remaining *EAS*. A special case exists between segments 9 and 10. In this case the shared adjacency can not be performed since the two aggregation points coincide. In these cases the shared adjacency criterion is skipped. The whole process is repeated for all *EAS* in the cluster. Figure 12 (b) shows the results.

The described algorithm was able to correct several noisy informations introduced in the neural network output during the test procedures. On the other hand, it is seen that the output information obtained with the neural network is highly reliable. This is partially verified by the small errors obtained during its learning stage with respect to the learning set as well as to the validation set.

5. Discussion and results

In this section several results obtained with the described method will be shown and its robustness discussed.

The algorithm employs several threshold values which, however, are not critical to its successful behaviour. In our implementation most of these values are taken constant. For contact point computation two threshold values are applied (T and Δ). In figure 14 the variation effects of these parameters are illustrated. As can be observed the number of identified contact points changes smoothly with T and Δ . The aggregation method will only

breakdown if physical contact points are missed by the identification procedure, i. e., the scale parameters are taken to high, as shown in figure 14 (e). In these cases, several *EAS* may be identified as one, being the algorithm unable to split them, since it operates only with *EAS* aggregation. When the scale parameters are taken too low, several physical non-existent contact points will be identified as in figure 14 (a). The aggregation algorithm is able to solve these situations, since the analysis departs from *EAS* properties. Therefore, if these *EAS* exhibit appropriate feature measures they will be correctly classified and aggregated by the method. As will be shown, even when feature values lead to incorrect classification, the aggregation method solves these situations in most cases using the redundancy in data. The main disadvantage of detecting non significant contact points is the increased computational load. In figure 14 (c), 14 (f) and 14 (i) the aggregation results of the identified *EAS* in figure 14 (a), 14 (d) and 14 (g) are shown. As expected, in figure 14 (i), not all physically existent granules have been obtained, since, as can be observed in figure 14 (h), due to incorrect scale definition some contact points have been missed. These situations may, however, be partially avoided if a scale-space approach is used for contact point determination.

With regard to threshold T_p , it is observed from (29) that it only serves as an early stop condition for the iterative center estimation method, that is, if the *EAS* is sufficiently circular then the intended precision for center estimation may be obtained in a few iterations. For less circular arcs, usually the maximum iteration condition serves as stop condition. The obtained precision is not critical, since the identified centers are further treated with the enumerated geometrical correction principals (see section II.).

Threshold ϵ is always taken as $\epsilon = 100$, since if convergence is not obtained after 100 iterations then it is seen that the arc segment is highly irregular and, therefore, it probably requires several hundreds of iterations for its center to converge (not necessarily to a correct

value). However, since further correction steps are taken, we stop the estimation process after ϵ iterations.

For center estimation correction another threshold (δ) is applied. This threshold controls maximum and minimum values of the radius. In our implementation, $\delta = 0.1$ and all results shown in this paper were obtained with this value.

The described aggregation method uses extensively data redundancy to recover from classification failure situations. This data redundancy is expressed in the shared adjacency criterion. Namely, whenever a given *EAS* exhibits less than 50% of rejection votes from its *CAS* members, the shared adjacency criterion is applied to establish the aggregation decision. This mechanism only fails when there are insufficient number of votes for a given *EAS* (this condition may be relaxed and be performed for any number of votes) or when the adjacent *EAS* used with the criterion exhibit noisy feature values that prevent their aggregation. Both situations are unlikely to happen simultaneously given the good responses of the neural network and the center estimation method. Potential failures are mainly due to large biased center identification, which usually may occur for very small or for very linear *EAS*. Even in these situations, biases are largely reduced by the correction method's property 3. As for the voting mechanism, it is seen that when more than 50% of votes are obtained there is still a significant chance of aggregation between these *EAS* if they are rejected due to misclassification. This occurs if their adjacent *EAS* are analysed before them (note that an aggregation establishes a physical connection).

In figure 15 the robustness of the neural network is tested by varying the D^1 and D^2 measures (and indirectly the E , CES and C_1) for pair $\{EAS_1, EAS_3\}$ of figure 12 (a). D^2 has been varied by shifting EAS_1 's center. For this pair of *EAS* the normalised measures are $D_1 = 0.068$ and $D_2 = 0.055$. Possible D_2 values for centers of EAS_1 (such that the

center position is physically plausible) are $D_2 \in [0; 0.188]$ and $D_2 \in [0; 0.274]$, respectively for centers positioned above and under EAS_3 's center. As can be observed from figure 15 both intervals are slightly above the correct classification ranges. This means that the neural network would output correct classification results even under misidentified centers for EAS_1 as long they are physically plausible (included inside the granule). In this particular case, improper classification would be achieved when EAS_1 radius (R) approximates 0 ($D_2 > 0.148$) or for large values of R ($D_2 > 0.18$), i. e., R is almost outside the granule. This behaviour is obtained, since noisy data sets have been included in the neural network's training database. All feature values are explicitly shown to the network to enable it to learn a noise tolerant behaviour.

Typical results obtained with the outlined method are shown in figures 16 to 19. Note that in these results several non physical contact points have been detected.

IV. CONCLUSIONS

A new and robust algorithm for highly irregular elliptical object localization in multi-connected regions, applied to a lime granule inspection system, is presented in this paper. The method first identifies all clusters of granules in the image. For this propose a new inclusion method based on the Green's theorem is described. For each cluster, the algorithm decomposes all regions into a set of EAS and several measures are extracted from each EAS . Since some of these measures are ellipse related measures, an extension to Landau's method is described for center estimation, and Fitzgibbon's direct ellipse estimation technique is reformulated to avoid inversion problems of the scatter matrix. An ordered set of EAS for each elliptical region is then constructed upon *a posteriori* probabilities, which are computed by a neural network and a search path method based on geometrical properties analysis. All

critical situations (where classification results alone may lead to contradictions) are solved with the aid of a voting method and a neighborhood analysis scheme to confirm aggregation. This system has proven to be very reliable. Namely, it was able to correct several noisy informations introduced in the neural network output during the test procedure. Given the noise rejection ability of the path search method and the reduced errors obtained by the neural network, this method exhibits a robust behavior in lime granule localization. Other advantages come from the use of a neural network to solve the problem. It is well known that its generalization capabilities allow us to deal with noisy data situations, as it was the case when, for test purposes, a neural network, trained only with artificial *EAS*, was tested with real *EAS*.

The method employs several threshold values. However, it is seen that only two of these values - T and Δ , which code indirectly the curves' scale for contact point estimation-influence the algorithm's performance, since all the other threshold values are constant in the system's implementation. Further, T and Δ may be avoided by using a scale-space approach for contact point detection.

Appendix

Proof 1:

In this section, we derive (7), the direction invariant discrete Green theorem implementation.

Proof. Given the Green's theorem

$$\oint_{\Xi} [Mdx + Ndy] = \iint_R \left[\frac{\partial N}{\partial x} - \frac{\partial M}{\partial y} \right] dxdy \quad (73)$$

where

Ξ - closed, piecewise smooth and counterclockwise plane curve

R - region bounded by Ξ

M, N - continuous functions with continuous first order derivatives in R

it can be shown that, for a clockwise closed curve, the Green's theorem may be rewritten as

$$\oint_{\Xi'} [Mdx + Ndy] = - \iint_R \left[\frac{\partial N}{\partial x} - \frac{\partial M}{\partial y} \right] dxdy \quad (74)$$

where

Ξ' - closed, piecewise smooth and clockwise plane curve

leading to

$$\oint_{\Xi'} [Mdx + Ndy] = - \oint_{\Xi} [Mdx + Ndy] \quad (75)$$

Tang [13] has shown that

$$\oint_{\Xi} [Mdx + Ndy] \approx \sum_L [F_x(x, y) D_y(x, y) + f(x, y) C_y(x, y)] \quad (76)$$

where

$$f(x, y) = \frac{\partial N}{\partial x} - \frac{\partial M}{\partial y}$$

$$F_x(x, y) = \sum_{i=0}^x f(i, y)$$

L - counterclockwise discrete contour path of R

Combining equation 75 and 76, it is verified that

$$\begin{aligned} \oint_{\Xi'} [Mdx + Ndy] &\approx - \sum_L [F_x(x, y) D_y(x, y) + f(x, y) C_y(x, y)] = \\ &= \sum_{L'} [F_x(x, y) D'_y(x, y) + f(x, y) C'_y(x, y)] \end{aligned} \quad (77)$$

L' - clockwise discrete contour path of R

and therefore

$$\begin{cases} D_y = -D'_y \\ C_y = -C'_y \end{cases} \quad (78)$$

This result is expressed in equation 7.

■

Proof 2:

In this section we will prove equation 12 for counterclockwise traced contours. For clockwise traced contours the prove is similar.

Proof.

Property 1: If $Cont$ is a closed contour ($\sum_{k=1}^n h_x(d_k) = 0$), then it is verified that for all points $p_j \in Cont$, that do not correspond to local extreme points, that is, $h_x(d_{j-1}) h_x(d_j) > 0$, there exists at least one $p_k \in Cont$, $k + 1 \neq j$, such that $\sum_{z=j}^k h_x(d_z) = 0$. That is, there exists at least one point $p_{k+1} \in Cont$ with the same horizontal displacement as point p_j . It can be shown that under these conditions the following equality holds $h_x(d_j) + h_x(d_k) = 0$.

For counterclockwise traced curves it is verified that the enclosed region's area may be computed using the Green's theorem (equation 73) by taking $M = -y$ and $N = 0$, that is

$$Area = \iint_R dx dy = - \oint_{\Xi} y dx > 0 \quad (79)$$

which, for our coordinate system (inverted y axes), has to be rewritten to

$$Area = - \iint_R dx dy = \oint_{\Xi} y dx > 0 \quad (80)$$

For discrete curves $dx \approx h_x(d)$, therefore equation 80 is approximated by

$$\oint_{\Xi} y dx \approx \sum_{i=1}^n y_i h_x(d_i) \quad (81)$$

Integrating equation 81 along a slice of width $h_x(d_i)$ at point p_i (see figure 20) it is seen that (note that for vertical point transitions $h_y() = 0$):

$$Area = y_i h_x(d_i) + \sum_{j \in S} y_j h_x(d_j) > 0 \quad (82)$$

Rewriting equation 82 with $y_j = y_i + \sum_{k=i}^{j-1} h_y(d_k)$ we get

$$h_x(d_i) y_i + \sum_{j \in S} h_x(d_j) \left[y_i + \sum_{k=i}^{j-1} h_y(d_k) \right] > 0 \quad (83)$$

Grouping the terms with common y_i equation 84 is obtained.

$$y_i \left[h_x(d_i) + \sum_{j \in S} h_x(d_j) \right] > - \sum_{j \in S} h_x(d_j) \left[\sum_{k=i}^{j-1} h_y(d_k) \right] \quad (84)$$

From property 1, it is seen that for closed contours, under constraint of equation 14:

$$h_x(d_i) + \sum_{j \in S} h_x(d_j) = 0 \quad (85)$$

Hence, combining equations 84 and 85 we finally get:

$$\sum_{j \in S} h_x(d_j) \left[\sum_{k=i}^{j-1} h_y(d_k) \right] > 0 \quad (86)$$

■

Proof 3:

In this section function $\Psi(x_i, y_i, d_i, d_{i-1})$ is derived.

Proof.

Closed contours exhibit the following properties:

1. For counterclockwise traced contours it is seen that for all points with code directions $d_i \in \{0, 1, 7\}$ ($h_x(d_i) = 1$) the bounded region is found to be in the decreasing y direction to the point and it is found to lie in the increasing y direction if $d_i \in \{3, 4, 5\}$ ($h_x(d_i) = -1$). **Proof.** we have already shown that for counterclockwise traced contours (see *proof 2*)

$$\sum_{j \in S} h_x(d_j) \left[\sum_{k=i}^{j-1} h_y(d_k) \right] > 0 \quad (87)$$

Substituting $\sum_{k=i}^{j-1} h_y(d_k) = y_j - y_i$ it is seen that

$$\sum_{j \in S} h_x(d_j) [y_j - y_i] > 0 \quad (88)$$

Without loss of generality, let us consider that there exists only one point $p_j \in S$ (for more points it can be proven by grouping pairs of points in set S). In this case equation 88 resumes to

$$h_x(d_j) [y_j - y_i] > 0 \quad (89)$$

From equation 85, it is seen that

$$h_x(d_j) = -h_x(d_i) \quad (90)$$

Therefore, combining equations 89 and 90, 91 is obtained:

$$\begin{cases} h_x(d_i) = -1 \iff y_j > y_i \\ h_x(d_i) = 1 \iff y_j < y_i \end{cases} \quad (91)$$

■

- 2 For clockwise traced contours all points with code directions $d_i \in \{0, 1, 7\}$ ($h_x(d_i) = 1$) the bounded region is in the increasing y direction to the point and it is found to lie in the decreasing y direction if $d_i \in \{3, 4, 5\}$ ($h_x(d_i) = -1$). This can be proven in a similar way as was done above, taking $h_x(d_j) [y_j - y_i] < 0$.

Combining these result with definition 1, it is verified that, if a point $p_i = (x_i, y_i) \in Cont$ is taken such that $(x_i, y_i - 1), (x_i, y_i + 1) \notin Cont$ then, from property 1, for external contours traced in counterclockwise direction, it is verified that points $(x, y - 1) \notin Object$, $(x, y + 1) \in Object$ if $d_i \in \{3, 4, 5\}$ and that points $(x, y + 1) \notin Object$, $(x, y - 1) \in Object$ if $d_i \in \{0, 1, 7\}$. However, if the contour was traced in the inverse direction then, for external contours, $(x, y - 1) \in Object$, $(x, y + 1) \notin Object$ if $d_i \in \{3, 4, 5\}$ and $(x, y + 1) \in Object$, $(x, y - 1) \notin Object$ if $d_i \in \{0, 1, 7\}$. Therefore, it is seen that $\Psi(x_i, y_i, d_i, d_{i-1}) = +1$. For internal contours, equation 16 can be proven using the same strategy. ■

Proof 4:

In this section, (30) is derived.

Proof. Let the ellipse be described by its polar equation (for simplicity no translation and rotation effects are considered):

$$\begin{cases} x = \eta B \cos(\alpha) \\ y = B \sin(\alpha) \end{cases} \quad \alpha \in [0, 2\pi], \eta \in \mathbb{R}^+ \quad (92)$$

From figure 3 it is seen that $\theta = \beta - \gamma$. Since $\overline{PZ} \perp \overline{OP}$ it is verified that

$$\tan(\gamma) = -\frac{1}{\tan(\alpha)} \quad (93)$$

and from equation 92

$$\tan(\beta) = \frac{dy}{dx} = -\frac{1}{\eta \tan(\alpha)} \quad (94)$$

Therefore,

$$\tan(\theta) = \frac{(\eta - 1) \tan(\alpha)}{\eta \tan^2(\alpha) + 1} \quad (95)$$

Without lost of generality let us analyse equation 95 for $\alpha \in [0, \frac{\pi}{2}]$. The maximum value of

$\tan(\theta)$ will occur when:

$$\left\{ \begin{array}{l} \frac{d \tan(\theta)}{d\vartheta} = 0 \\ \frac{d^2 \tan(\theta)}{d\vartheta^2} > 0 \\ \vartheta = \tan(\alpha) \end{array} \right. \iff \vartheta = \frac{1}{\sqrt{\eta}} \quad (96)$$

Substituting 96 in 95 it is seen that

$$\tan(\theta) = \frac{\eta - 1}{2\sqrt{\eta}} \quad (97)$$

From 97 it is seen that function $\tan(\theta)$ is monotonously increasing, therefore θ will be maximum when $\tan(\theta)$ reaches its maximum value, that is:

$$\theta_{\max} = \tan^{-1} \left(\frac{\eta - 1}{2\sqrt{\eta}} \right) \quad (98)$$

■

Acknowledgments

The authors thank the anonymous reviewers for their useful comments and suggestions which have improved this paper.

The authors express their gratitude to Eng. José Luis Amaral (Portucel-Viana) and to Eng. Carlos Gomes da Costa (Portucel-Cacia) for the used lime samples and to FCT/Praxis XXI which partially financed this project.

REFERENCES

- [1] Kumar S, Ranganathan N, and Goldof D. Parallel algorithms for circle detection in images. *Pattern Recognition*, 27(8):1019–1028, 1994.
- [2] Rosin P. Analysing error of fit functions for ellipses. *Pattern Recognition Letters*, 17:1461–1470, 1996.

- [3] Cui Y, Weng, and Reynolds H. Estimation of ellipse parameters using optimal minimum-variance estimators. *Pattern Recognition Letters*, 17(3):309–316, 1996.
- [4] Kanizsa G. *Organization in Vision*. New York Praeger, 1979.
- [5] Sarkar S and Boyer K. Perceptual organization in computer vision: a review and a proposal for a classificatory structure. *IEEE Transactions on Systems, Man, and Cybernetics*, 23(2):382–399, 1993.
- [6] Mohan R and Nevatia R. Using perceptual organization to extract 3-d structures. *IEEE Transactions on Pattern Analysis Machine Intelligence*, 11(11):1121–1139, 1989.
- [7] Mohan R and Nevatia R. Perceptual organization for scene segmentation and discription. *IEEE Transactions on Pattern Analysis Machine Intelligence*, 14(6):616–635, 1992.
- [8] Grossberg S and Mingolla E. Neural dynamics of perceptual groupings: texture, boundaries, and emergent segmentation. *Perception Psychophys*, 38(2):141–171, 1985.
- [9] Pal N and Pal S. A review on image segmentation techniques. *Pattern Recognition*, 26(9):1277–1294, 1993.
- [10] Brink A and Pendock N. Minimum cross-entropy threshold selection. *Pattern Recognition*, 29(1):179–188, 1996.
- [11] Sahoo P, Wilkins C, and Yeager J. Threshold selection using renyi’s entropy. *Pattern Recognition*, 30(1):71–84, 1997.
- [12] Tang Yand Len B. Region filling with the use of the discrete green theorem. *Computer Vision and Image Processing*, 42(3):297–305, 1988.

- [13] Yang L and Albregsten F. Fast and exact computation of cartesian geometric moments using discrete green's theorem. *Pattern Recognition*, 29(7):1061–1073, 1996.
- [14] Tsai D. Boundary-based corner detection using neural networks. *Pattern Recognition*, 30(1):85–97, 1997.
- [15] Esplid R and Jonassen I. A comparison of splitting methods for the identification of corner points. *Pattern Recognition Letters*, 12(2):79–83, 1991.
- [16] Zhu P and Chirlian P. On critical point detection of digital shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(8):737–748, 1995.
- [17] Kim C and Rosenfeld A. Digital straight lines and convexity of digital regions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 4(2):149–154, 1982.
- [18] Rattarangsi A and Chin R. Scale-spce detection of corners of planar curves. *IEEE Transactions on Pattern Analysis Machine Intelligence*, 14(4):430–450, 1992.
- [19] Teh C and Chin R. On detection of dominant points in digital curves. *IEEE Transactions on Pattern Analysis Machine Intelligence*, 11(8):859–872, 1989.
- [20] Ray B and Ray K. An algorithm for detection of dominant points and poligonal approximation of digitized curves. *Pattern Recognition Letters*, 13(12):849–856, 1992.
- [21] Carvalho P. Contribuição para o desenvolvimento de técnicas de inspeção visual automática em dois casos industriais - in portuguese. Master's thesis, FCTUC - Universidade de Coimbra, 1995.
- [22] Fitzgibbon A, Pilu M, and Fisher R. Direct least squares fitting of ellipses. In *IEEE Int. Conf. On Image Processing*, 1996.

- [23] Landau U. Estimation of circular arc center and its radius. *Computer Vision, Graphics and Image Processing*, 38:317–326, 1987.
- [24] Wu Q and Rodd M. Boundary segmentation and parameter estimation for industrial inspection. *IEE Proceedings*, 137(7), 1990.
- [25] Blinn J. Fractional invisibility. *IEEE Computer Graphics and Applications*, 8(6):77–84, 1988.
- [26] Wu W.-Y. and Wang M.-J. Elliptical object detection by using its geometrical properties. *Pattern Recognition*, 26(10):1499–1509, 1993.
- [27] McCafferty J. *Human and machine vision: computing perceptual organization*. West Sussex, England: Ellis Horwood, 1989.
- [28] Wan E. Neural network classification: a bayesian interpretation. *IEEE Transactions on Neural Networks*, 1(8):303–305, 1990.
- [29] Ruck D, Rogers S, Kabrisky M, Oxley M, and Suter B. The multilayer perceptron as an approximation to a bayes optimal discriminant function. *IEEE Transactions on Neural Networks*, 1(4):296–298, 1990.
- [30] Bishop C. *Neural networks for pattern recognition*. Oxford University Press, 1995.
- [31] Chester D. Why two hidden layers are better than one. In *IEEE Int. Joint Conf. On Neural Networks (IJCNN90)*, pages 265–268, 1990.
- [32] Haykin S. *Neural networks: a comprehensive foundation*. Prentice Hall, 1994.
- [33] Demuth H and Beale M. *Neural Network Toolbox User's Guide*. The MathWorks, Inc., 1994.

[34] Masters T. *Practical neural network recipes in C++*. Academic Press, 1993.

Author Biographies

- Paulo de Carvalho received a BSc. degree in Informatics Engineering and a MSc. degree in Systems and Automation both from the Department of Electrical Engineering, University of Coimbra, Portugal, in 1992 and 1996, respectively.

Since 1995 he is an Assistant in the Department of Informatics Engineering at the University of Coimbra, where he is currently conducting his PhD works. His main research interests are colour vision, automatic visual inspection, pattern recognition and neural networks.

- Nuno Costa received a BSc. degree in Informatics Engineering from the Department of Informatics Engineering, University of Coimbra, Portugal, in 1996. Since 1996 he is PhD student in the Department of Informatics Engineering at the University of Coimbra. His main research interests are neural networks and their applications to engineering systems, intelligent process control and fault detection and system diagnosis.

- Bernardete Ribeiro received a BSc. degree in Chemical Engineering from the Chemical Engineering Department, University of Coimbra, Portugal, in 1975, the MSc. degree in Computer Science and the PhD degree in Electrical Engineering, speciality in Informatics, both from the Electrical Engineering Department, University of Coimbra.

She is currently Assistant Professor of the Department of Informatics Engineering at the University of Coimbra. Her main research interests are neural networks and their applications to engineering systems, fault detection and diagnosis, and intelligent process control.

- António Dourado is Associate Professor in the Department of Informatics Engineering, Faculty of Science and Technology, University of Coimbra. He got the Electrical En-

gineering Degree by University of Coimbra in 1977, the Diplome de Docteur-Ingénieur (Automatique) by the Université Paul Sabatier and LAAS du CNRS, Toulouse, France in Dec 1983 and the Doctor Degree (PhD) by the University of Coimbra in 1984, and the pos-doc degree Aggregation in 1997. He is teaching Digital Instrumentation and Automatic Control. His present research interests are Soft Computing techniques for intelligent instrumentation and control, with emphasis on real-time applications in industry. He is President of the Portuguese Association of Automatic Control (the IFAC NMO) during 1999/2000.

Figure legends

Figure 1: a) Example of several agglomerates which are included inside other agglomerates of regions; b) Inclusion tree of the contours in (a) and its splitting indication.

Figure 2: a) Contact points determination; b) Initial centre estimation.

Figure 3: Computing angle θ .

Figure 4: Center correction procedure.

Figure 5: Center and radius identification results obtained by (a) Landau's method and (b) by Landau's method with correction steps. Contact points are marked with \mathbf{x} on the contours.

Figure 6 : Estimated centers and ellipses by the proposed method.

Figure 7: Comparison of centre identification results (rectangles - the described method; triangles - Wu's method; circles with horizontal lines - Fitzgibbon's method; filled circles - contact points; dashed lines - identified ellipses by the described method)

Figure 8: Condition numbers of S for each segment in figure 7. a) Proposed method; b) direct estimation.

Figure 9: Neural network general scheme.

Figure 10: Training results of the neural network.

Figure 11 Aggregation algorithm's flowchart.

Figure 12: a) Arc segment representation; b) Reconstructed image.

Figure 13: a) Aggregation matrix; b) *CAS* tree..

Figure 14: Variation effects of T and Δ . Contact points are represented by circles. Left: a) $T = 2, \Delta = 4$; d) $T \in [3, 5], \Delta \in [4, 12]$; g) $T = 6, \Delta = 4$. Center: center identification using contact points in a), d) and g), respectively. Right: Aggregation results from b), e) and h).

Figure 15: $AGP_{1,3}$ results by varying EAS_1 's center position (D^2) and localisation (D^1). a) When EAS_1 's center is above EAS_3 's center; b) When EAS_1 's center is below EAS_3 's center.

Figure 16: a) Identified centers and contact points; b) Aggregation results;

Figure 17: a) Identified centers and contact points; b) Aggregation results;

Figure 18: a) Identified centers and contact points; b) Aggregation results;

Figure 19: a) Identified centers and contact points; b) Aggregation results;

Figure 20: Integrating along a slice $h_x()$.

<i>Contour Direction</i>	d_1, d_2	Δ_2
<i>Clockwise</i>	0, 1, 7	1
<i>Clockwise</i>	3, 4, 5	-1
<i>Counterclockwise</i>	0, 1, 7	-1
<i>Counterclockwise</i>	3, 4, 5	1

Table 1: Definition of function $\Delta_2(d_1, d_2)$.

d_i	0	1	2	3
$\delta()$	$-\cos(\psi)$	$\sin(\psi) - \cos(\psi)$	$\sin(\psi)$	$\sin(\psi) + \cos(\psi)$

Table 2: Definition of function $\delta()$ ($\psi = \angle \overline{p_i p_j} + \frac{\pi}{2}$).

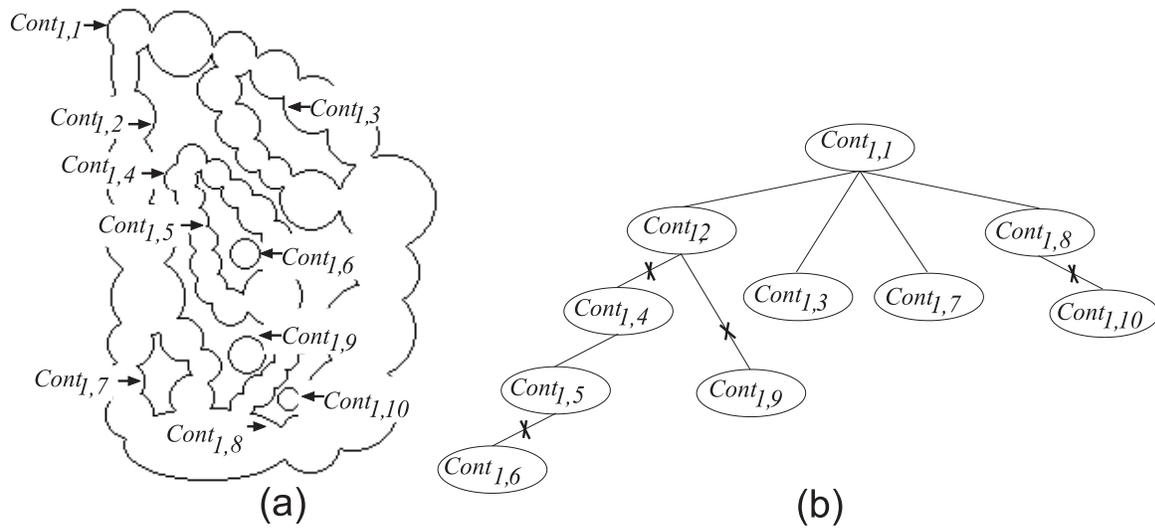


Figure 1: a) Example of several agglomerates which are included inside other agglomerates of regions; b) Inclusion tree of the contours in (a) and its splitting indication.

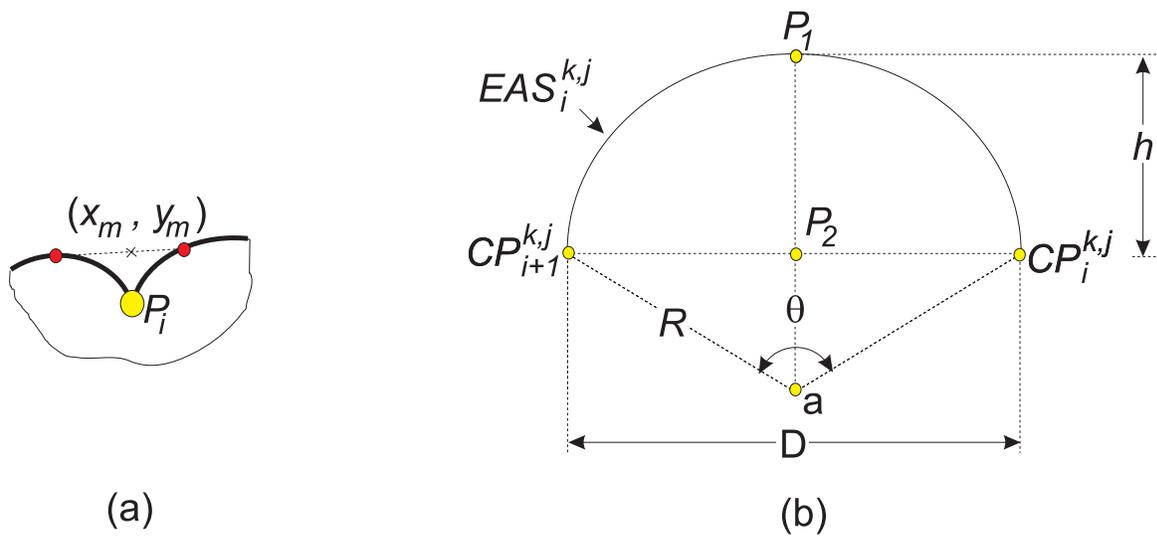


Figure 2: a) Contact points determination; b) Initial centre estimation.

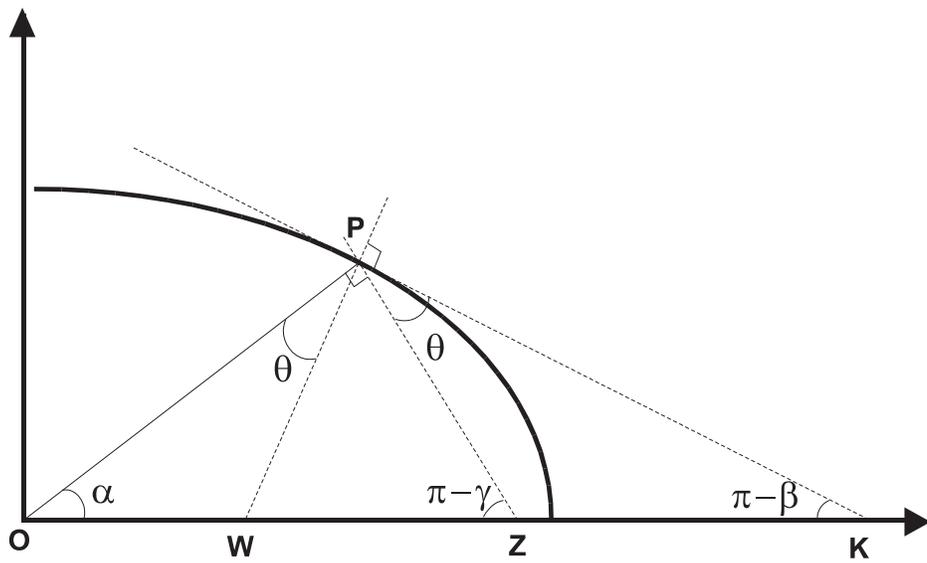


Figure 3: Computing angle θ .

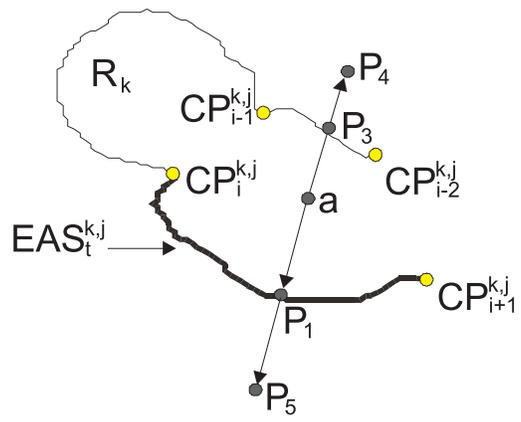


Figure 4: Center correction procedure.

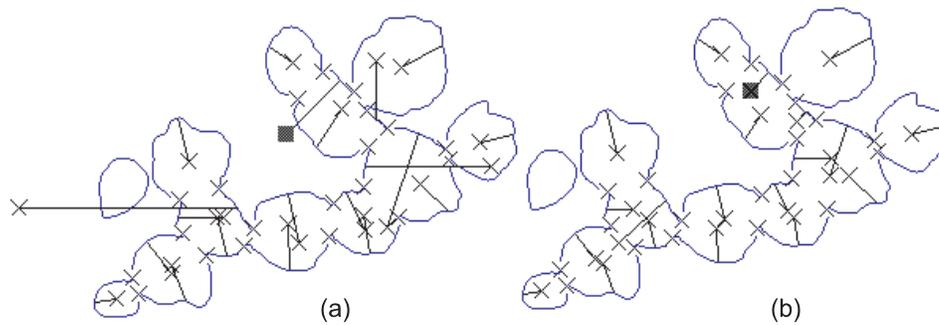


Figure 5: Center and radius identification results obtained by (a) Landau's method and (b) by Landau's method with correction steps. Contact points are marked with \mathbf{x} on the contours.

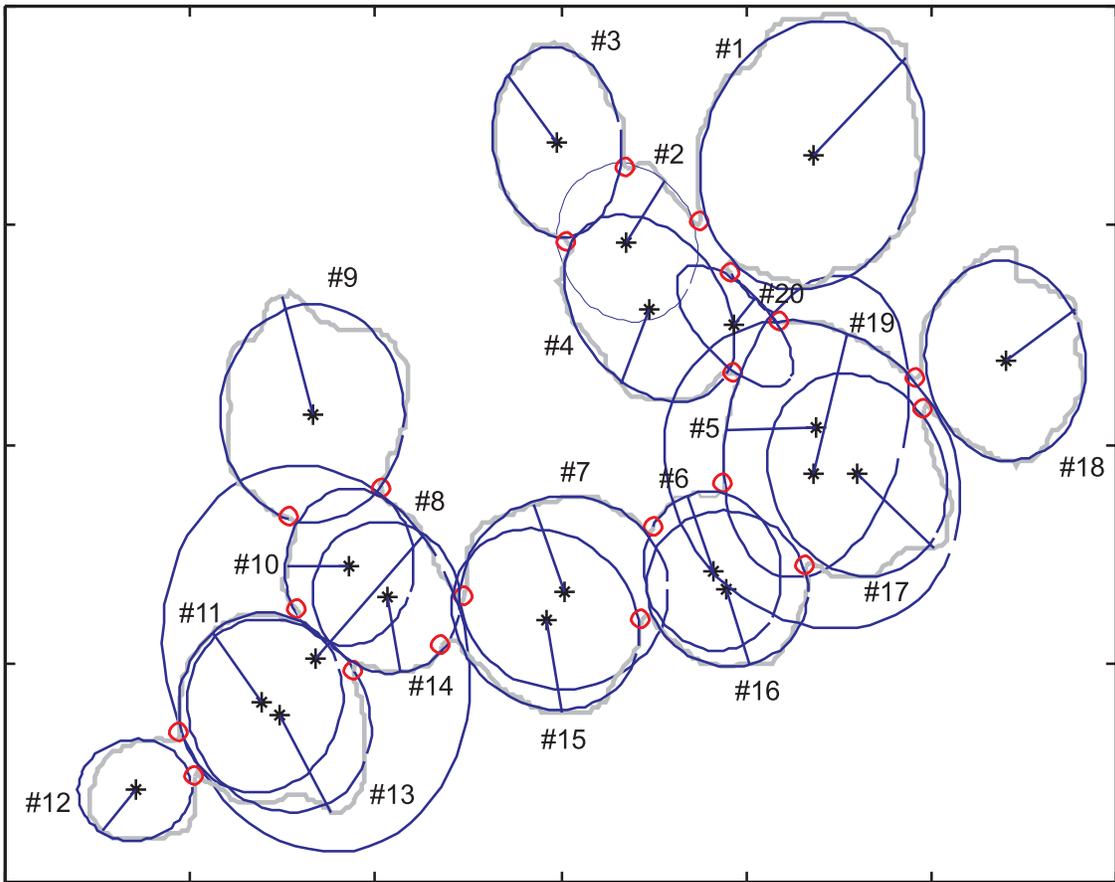


Figure 6: Estimated centers and ellipses by the proposed method.

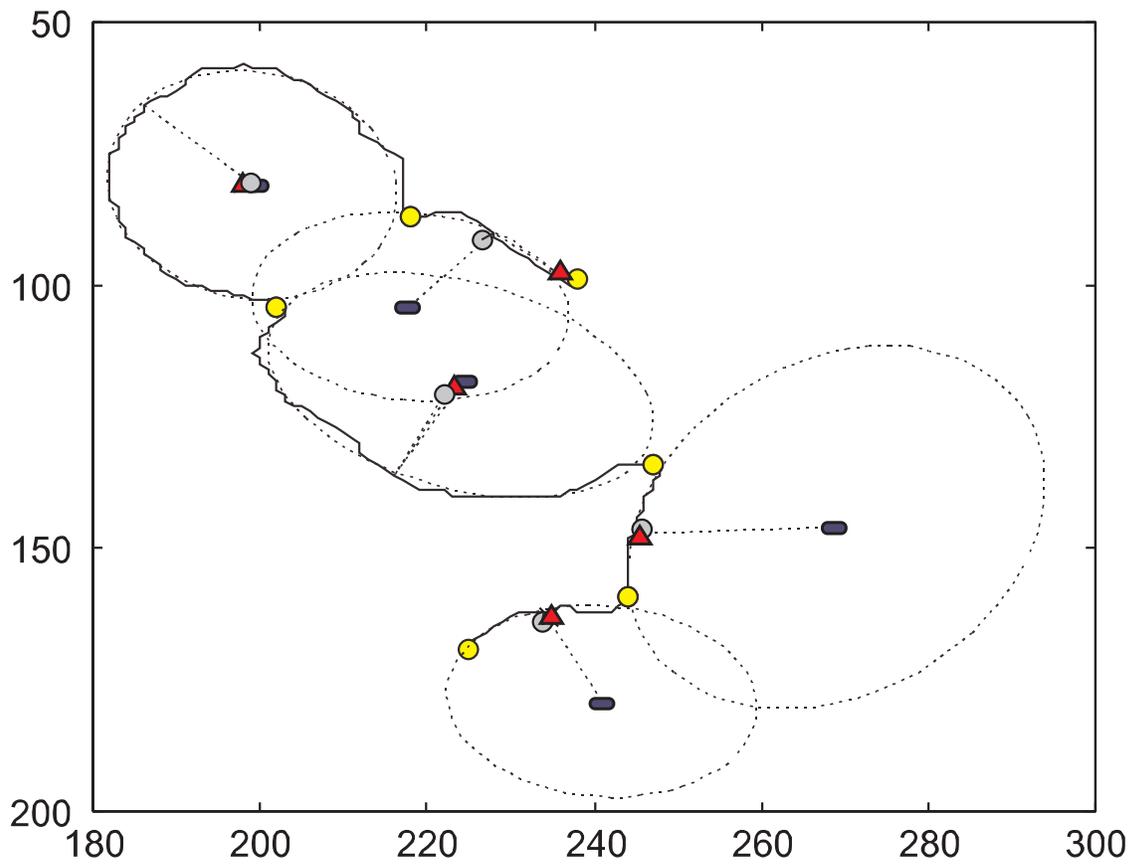


Figure 7: Comparison of centre identification results (rectangles - the described method; triangles - Wu's method; circles with horizontal lines - Fitzgibbon's method; filled circles - contact points; dashed lines - identified ellipses by the described method)

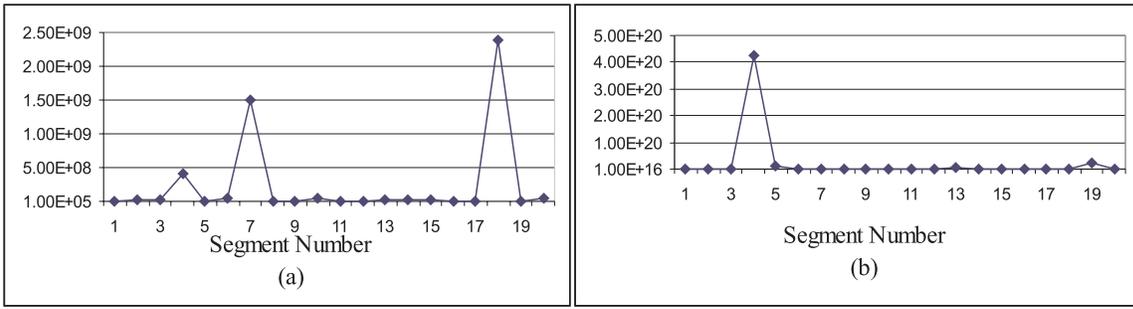


Figure 8: Condition numbers of S for each segment in figure 7. a) Proposed method; b) direct estimation.

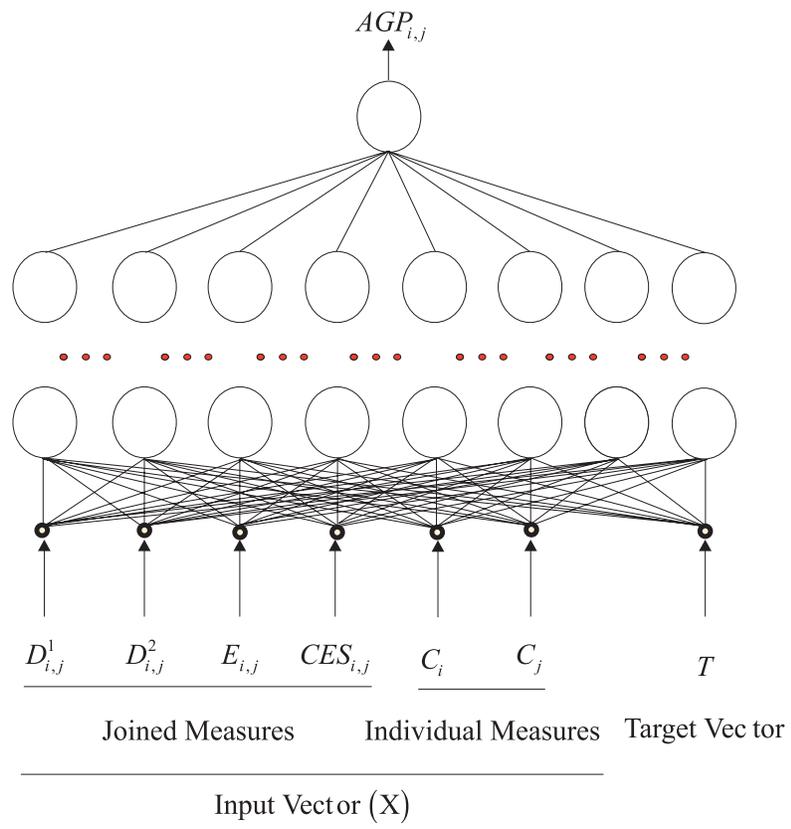


Figure 9: Neural network general scheme.

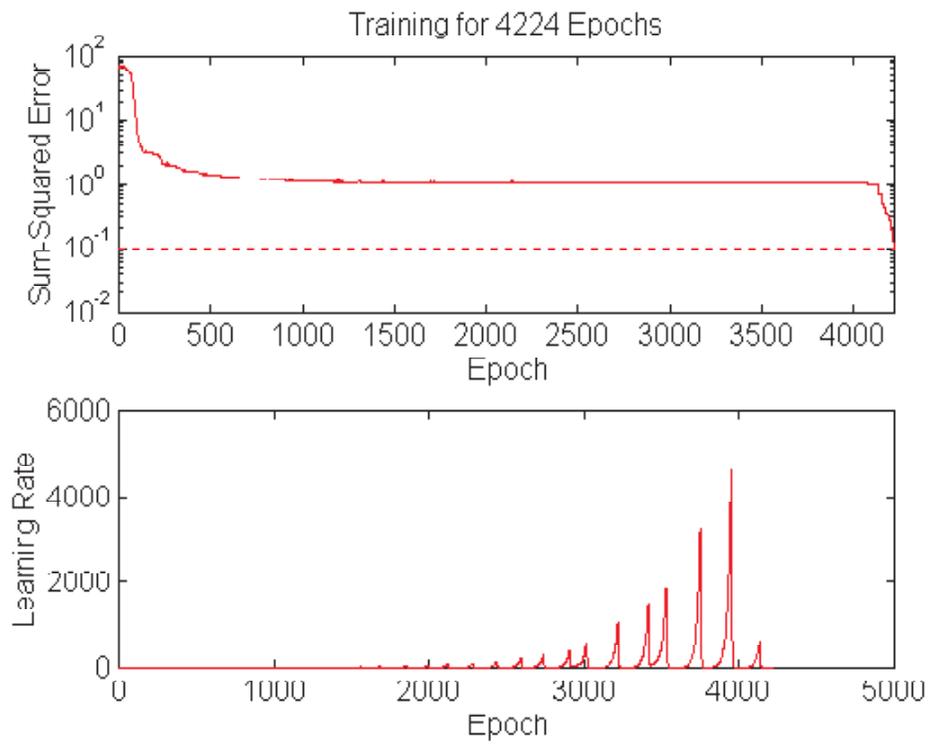


Figure 10: Training results of the chosen neural network.

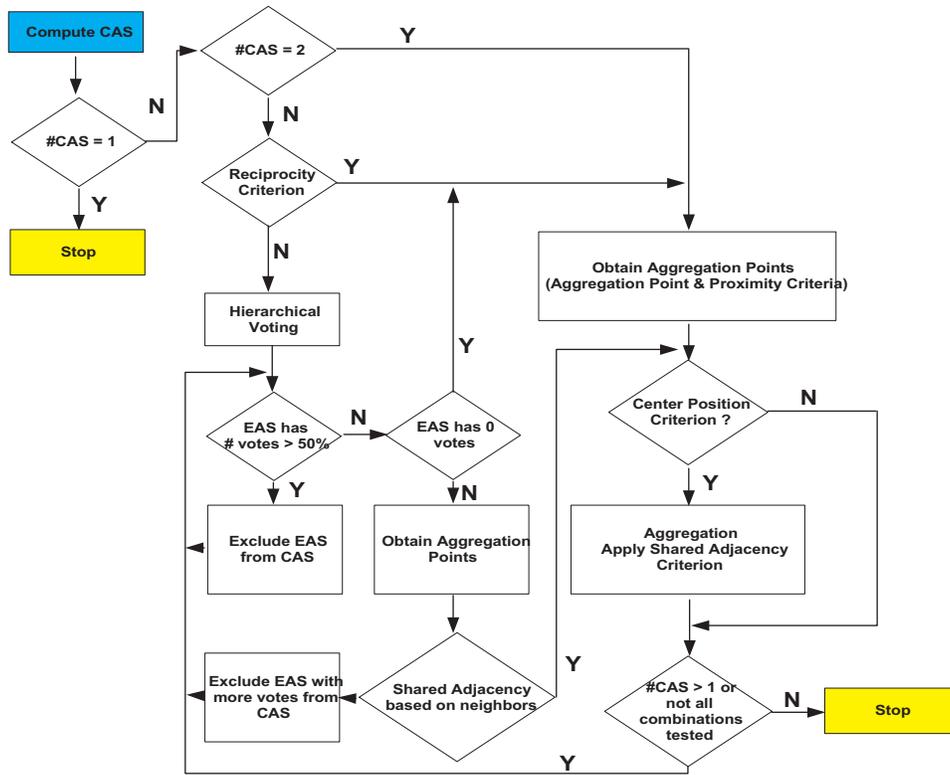


Figure 11: Aggregation algorithm's flowchart.

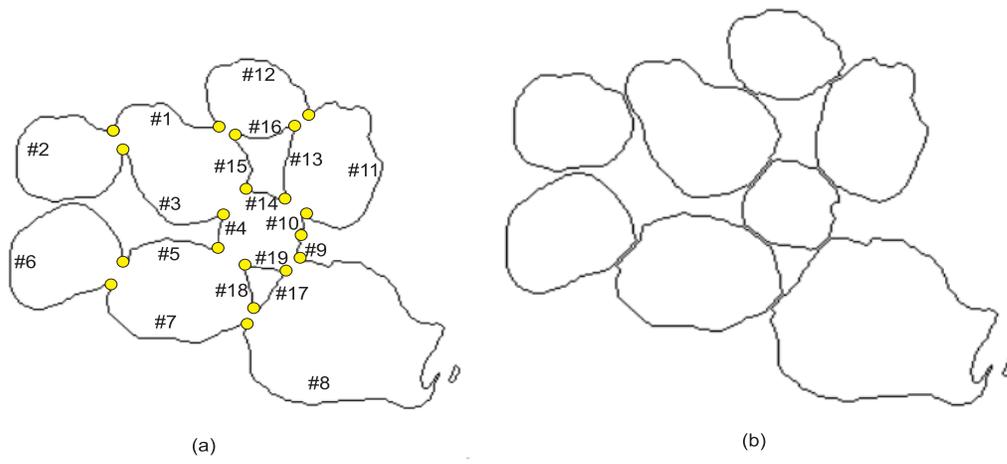
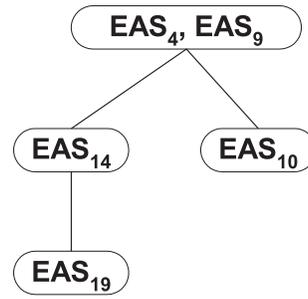


Figure 12: a) Arc segment representation; b) Reconstructed image.

	4	9	10	14	19
4	X	0.9	0.1	0.9	0.1
9	0.9	X	0.9	0.9	0.1
10	0.1	0.9	X	0.9	0.9
14	0.9	0.9	0.9	X	0.9
19	0.1	0.1	0.9	0.9	X

(a)



(b)

Figure 13: a) Aggregation matrix; b) CAS tree.

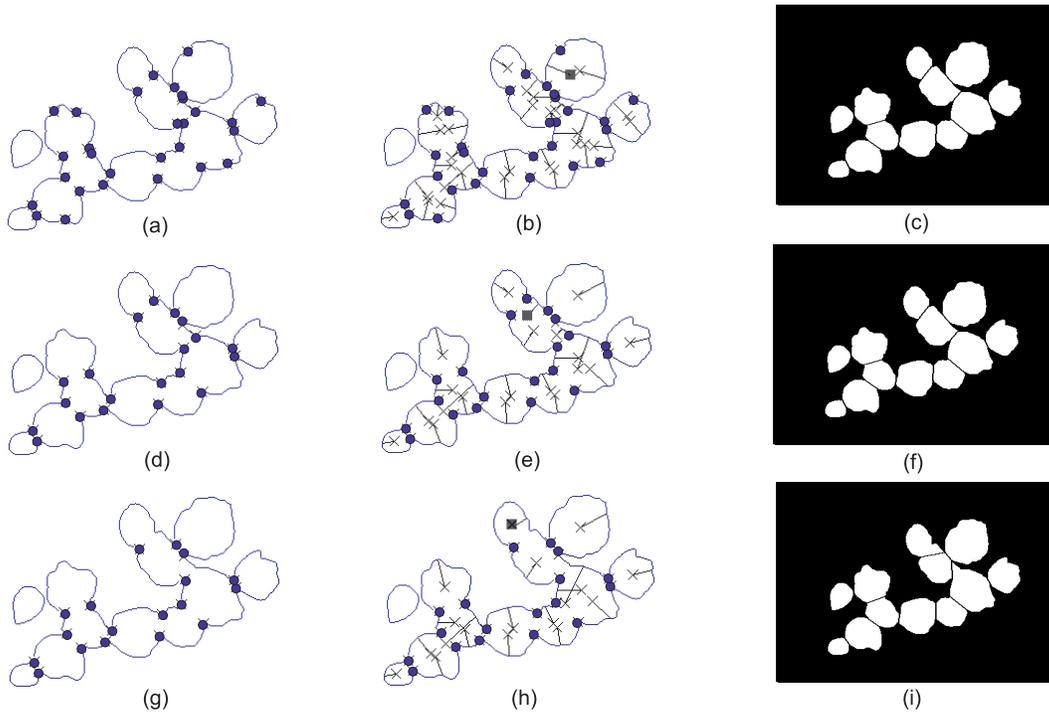


Figure 14: Variation effects of T and Δ . Contact points are represented by circles. Left: a) $T = 2$, $\Delta = 4$; d) $T \in [3, 5]$, $\Delta \in [4, 12]$; g) $T = 6$, $\Delta = 4$. Center: center identification using contact points in a), d) and g), respectively. Right: Aggregation results from b), e) and h).

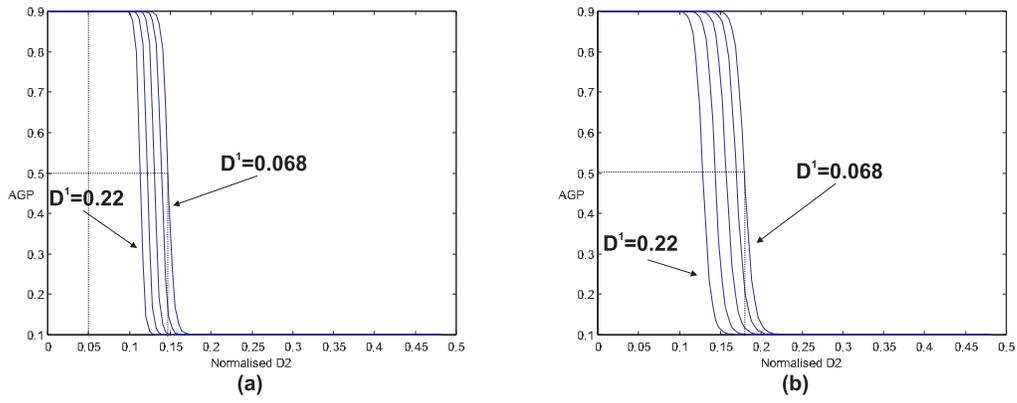
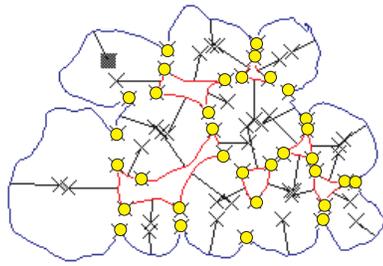
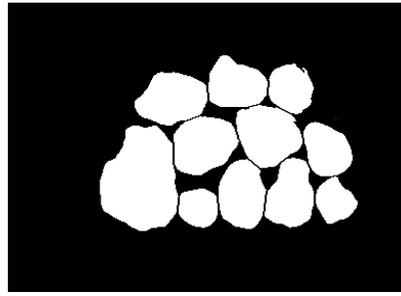


Figure 15: $AGP_{1,3}$ results by varying EAS_1 's center position (D^2) and localisation (D^1).

a) When EAS_1 's center is above EAS_3 's center; b) When EAS_1 's center is below EAS_3 's center.



(a)



(b)

Figure 16: a) Identified centers and contact points; b) Aggregation results.

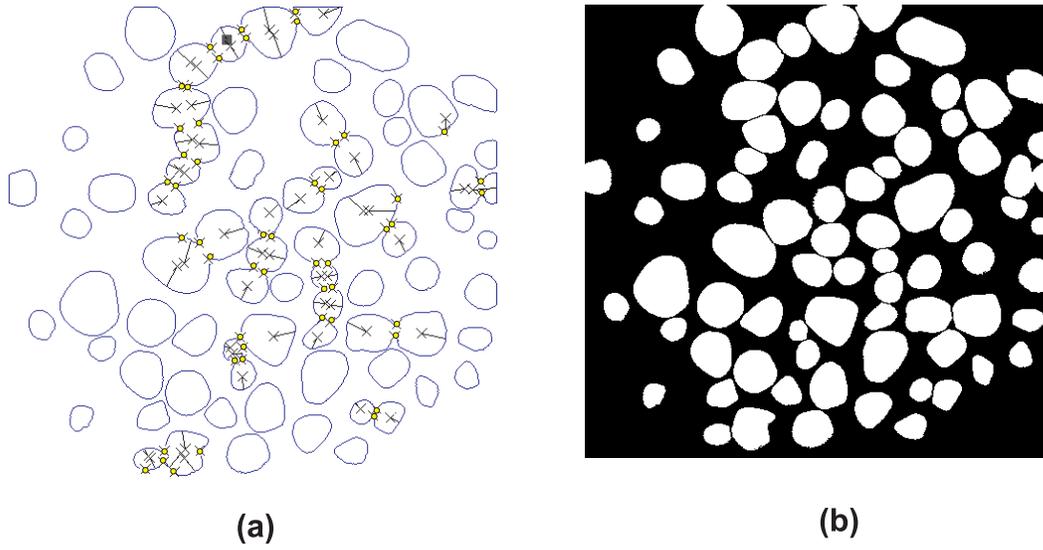


Figure 17: a) Identified centers and contact points; b) Aggregation results.

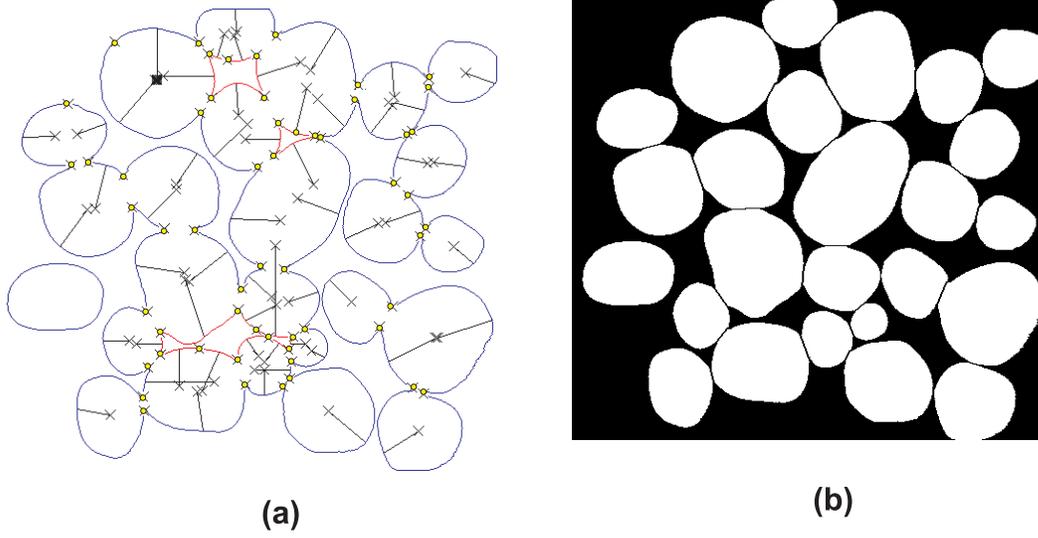


Figure 18: a) Identified centers and contact points; b) Aggregation results.

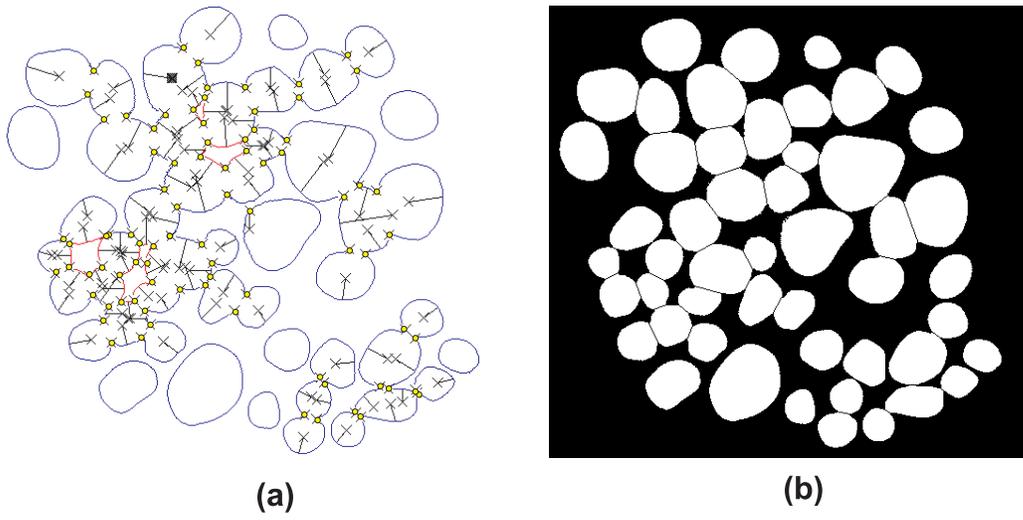


Figure 19: a) Identified centers and contact points; b) Aggregation results.

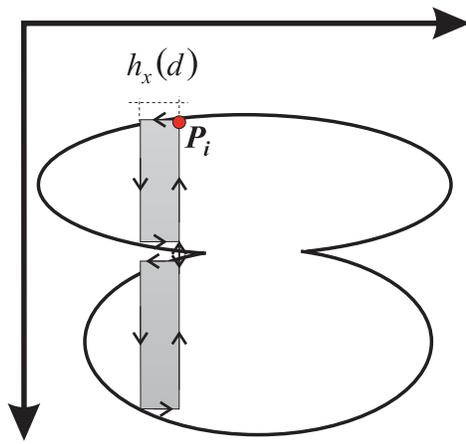


Figure 20: Integrating along a slice $h_x()$.

	Training Set		Validation Set	
	(841 samples)		(231 samples)	
	Misclassif. Percentage	# Classif. Errors	Misclassif. Percentage	# Classif. Errors
logsig/logsig				
N(6,5,1)	0.000%	0	0.433%	1
N(6,9,1)	0.000%	0	0.000%	0
N(6,12,1)	0.000%	0	0.433%	1
N(6,15,1)	0.000%	0	0.433%	1
logsig/logsig/logsig				
N(6,6,3,1)	0.000%	0	0.433%	1
N(6,8,4,1)	0.000%	0	0.000%	0
N(6,12,6,1)	0.000%	0	0.433%	1
tansig/logsig				
N(6,5,1)	0.238%	2	0.000%	0
N(6,9,1)	8.790%	74	10.400%	24
N(6,12,1)	0.000%	0	0.000%	0
N(6,15,1)	8.790%	74	10.400%	24
tansig/tansig/logsig				
N(6,6,3,1)	0.120%	1	0.433%	1
N(6,8,4,1)	8.790%	74	10.400%	24
N(6,12,6,1)	0.000%	0	0.000%	0

Table 3: Best neural network classification performance.