Articles   School of Electrical and Electronic Engineering

A Surface Inspection Machine Vision System that Includes Fractal Texture Analysis

Jonathan Blackledge
Technological University Dublin, jonathan.blackledge@tudublin.ie

Dmitry Dubovitskiy
Oxford Recognition Limited

Recommended Citation

A Surface Inspection Machine Vision System that Includes Fractal Texture Analysis

Jonathan M Blackledge, Fellow, IET and Dmitry A Dubovitskiy, Member, IET

Abstract— The detection, recognition and classification of features in a digital image is an important component of quality control systems in production and process engineering and industrial systems monitoring, in general. In this paper, a new pattern recognition system is presented that has been designed for the specific task of monitoring the quality of sheet-steel production in a rolling mill. The system is based on using both the Euclidean and Fractal geometric properties of an imaged object to develop training data that is used in conjunction with a supervised learning procedure based on the application of a fuzzy inference engine. Thus, the classification method includes the application of a set of features which include fractal parameters such as the Lacunarity and Fractal Dimension and thereby incorporates the characterisation of an object in terms of texture that, in this application, has metallurgical significance.

The principal issues associated with object recognition are presented including a new segmentation algorithm. The self-learning procedure for designing a decision making engine using fuzzy logic and membership function theory is also presented and a new technique for the creation and extraction of information from a membership function considered.

The methods discussed, and the system developed, have a range of applications in 'machine vision' and automatic inspection. However, in this publication, we focus on the development and implementation of a surface inspection system designed specifically for monitoring surface quality in the manufacture of sheet-steel. For this publication, we include a demonstration version of the system which can be downloaded, installed and utilised by interested readers as discussed in Section VI.

Index Terms— Computer vision, Fractal image analysis, Segmentation, Object recognition, Decision making, Self-learning, Fuzzy logic, Image morphology, Surface inspection, Defectoscopy.

I. INTRODUCTION

FRACTAL geometry is the geometry associated with naturally occurring objects that have repeating patterns at different scales. The use of fractal geometry for simulation is well known and extensive [1]. Less well developed is the use of fractal geometry for characterizing objects in such a way that a machine dependent interpretation of the object can be made. A fractal geometric approach to computer vision is important in the interpretation and recognition of objects that are characterized by their texture and therefore difficult to interpret using conventional machine vision techniques. This

occurs in bio- and medical-imaging problems, non-destructive evaluation and materials science, for example.

The technology associated with the manufacture of high quality materials such as in the production of steel often includes the need for automatic surface inspection systems used for the purpose of quality control. Quality control systems are required for several tasks such as: screening defected products, monitoring manufacturing processes, sorting information for different applications and product certification for the end user. The system discussed in this paper has been developed for the Novolipetsk Iron and Steel Corporation in Russia. The system was tested with images captured at a standard rolling mill. A fast moving steel strip with speeds of up to six meters per second was inspected for several metallurgical class defects.

Object recognition in image analysis involves the use of image processing methods (e.g. [2], [3], [4]) that are often designed in an attempt to provide a machine interpretation of an image, ideally, in a form that allows some decision criterion to be applied [5]. An object is typically represented by some pattern matching template. The problem is to find the best representation for an object given the operating conditions under which recognition is to be achieved, the object type, its principal characteristics and the applications to which the vision system is to be applied.

Pattern recognition uses a range of different approaches that are not necessarily based on any one particular theme or unified theoretical approach. The main problem is that, to date, there is no complete theoretical model for simulating the processes that take place when a human interprets an image generated by the eye, i.e. there is no fully self-consistent model currently available for explaining the processes of visual image comprehension. Hence, machine vision remains a rather elusive subject area in which automatic inspection systems are advanced without having a fully operational theoretical framework as a guide. Nevertheless, numerous algorithms for understanding two- and three-dimensional objects in a digital image have and continue to be researched in order to design systems that can provide reliable automatic object detection, recognition and classification in an independent environment, e.g. [5], [6], [7] and [8]. Most of these 'systems' are based on using Euclidean type metrics and are therefore unable to be applied to the analysis of objects whose characteristics are of a non-Euclidean type, e.g. textured.

A. Machine Vision

Machine Vision can be thought of as the process of linking parts of the visual object's field with stored information or

'templates' with regard to a pre-determined significance for the observer. There are a number of questions concerning vision such as: (i) what are the goals and constraints? (ii) what type of algorithm or set of algorithms is required to effect vision? (iii) what are the implications for the process, given the types of hardware that might be available? (iv) what are the levels of representation required to achieve vision? The levels of representation are dependent on what type of segmentation, including edge detection, can and/or should be applied to an image [9], [10]. For example, we may be able to produce primal sketches from an image via some measure of the intensity changes in a scene which are recorded as place tokens and stored in a database. This allows sets of raw components to be generated, e.g. regions of pixels with similar intensity values or sets of lines obtained by isolating the edges of an image scene and computed by locating regions where there is a significant difference in the intensity. However, such sets are subject to inherent ambiguities when computed from a given input image and associated with those from which an existing data base has been constructed. Such ambiguities can only be overcome by the application of high-level rules, based on how humans interpret images, but the nature of this interpretation is not always clear. Nevertheless, parts of an image will tend to have an association if they share size, colour, figural similarity, continuity, shading and texture, for example. For this purpose, we are required to consider how best to segment an image and what form this segmentation should take.

The identification of the edges of an object in an image scene is an important aspect of the human visual system because it provides information on the basic topology of the object from which an interpretative match can be achieved. Some edges can be detected only through a representative view of a whole image and have no connection with local pixels. In other words, the segmentation of an image into a complex of edges is a useful pre-requisite for object identification, the solution sometimes requiring analysis of the entire scene. However, although many low-level processing methods can be applied for this purpose, the problem is to decide which object boundary each pixel in an image falls into and which high-level constraints are necessary. Thus, in many cases, a principal question is, which comes first, recognition or segmentation?

Compared to image processing, computer vision (which incorporates machine vision) is more than automated image processing. It results in a conclusion, based on a machine performing an inspection of its own. The machine must be programmed to be sensitive to the same aspects of the visual field as humans find meaningful. In this context, segmentation is concerned with the process of dividing an image into meaningful regions or 'segments'. It is used in image analysis to separate features or regions of a pre-determined type from the background and is the first step in automatic image analysis and pattern recognition. Segmentation is broadly based on one of two properties in an image: (i) similarity; (ii) discontinuity. The first property is used to segment an image into regions which have grey or colour levels within a predetermined range. The second property segments the image into regions of discontinuity where there is a more or less abrupt change in the values of the grey or colour levels.

In this paper, we consider an approach to object detection in an image scene that is based on a new edge recognition, edge tracing and edge following algorithm. The segmented object is then analysed in terms of metrics derived from both a Euclidean and fractal geometric perspective, the output fields being used to train a fuzzy inference engine (e.g. [11] and [12]) with a special developed supervised leaning technique [13]. The structure associated with this approach is based on some of the techniques for machine vision reported in [14], for example. The approach considered is generic in that it can, in principle, be applied to any type of imaging modality. There are numerous applications of this technique that include object recognition where self-calibration and leaning is often mandatory such as in remote sensing with Synthetic Aperture Radar [15], [17], medical imaging [16] non-destructive evaluation and testing, and other applications which specifically require the classification of objects that are textural [17].

In this paper, we focus on one particular application associated with quality control in the manufacture of sheet-steel and, in particular, the detection of certain type of (surface) defects. The early detection of such defects allow for corrections to be made in a manufacturing process. Further, although some features are not defects as such, information on their regularity of occurrence, for example, can help to establish the grade of sheet metal and provide a quality assurance. The system reported is, in principle, just one of a number of variations which can be used for object image analysis and classification in non-destructive evaluation. However, the system specifically includes features that are based on the textural properties of an image which is an important theme in object image analysis and of specific importance in the evaluation of a surface for the quality control of sheet-steel production.

B. Computer Vision using Fractals

The aims and objectives of the computer vision system reported in this paper concern the task of developing a methodology and implementing applications that focus on two key tasks: (i) the partial analysis of an image in terms of its fractal structure and the fractal properties that characterize that structure; (ii) the use of a Fuzzy Logic engine (and/or, more generally, Artificial Neural Networks) to classify an object based on both its Euclidean and Fractal geometric properties. The combination of these 'geometries' are used to define a processing and image analysis engine that is unique in its modus operandi but entirely generic in terms of the applications to which it can, in principle, be applied. The systems development reported reflects a wider investigation into the numerous applications of pattern recognition using fractal geometry as a central processing kernel leading to the development of a new library of pattern recognition algorithms including the fast computation of fractal parameters such as the Fractal Dimension, the Information Dimension, Correlation Dimension and multi-fractals [1], for example.

The literature on fractal geometry over the past thirty years reflects the wealth of articles that have and continue to use the principles of fractal geometry for simulation (e.g. [1]

and references therein). There is also a wealth of literature describing the 'fractal characteristics' of signals and images in such diverse fields as medicine, speech analysis, telecommunications, Internet traffic analysis and so on. However, there is significantly less published material reporting on the applications of these 'fractal characteristics' for the design and implementation of operational diagnostic systems, tailored to a specific application. However, accept for special cases, fractal based analysis alone is not sufficient in order to design a recognition and classification system. Both Euclidean and fractal parameters (and other statistical measures) need to be combined into a 'feature vector' in order to develop an operational vision system which can analyse objects that have textural properties. Medical images (including optical, ultrasound, CT and MR images, for example) are a natural field of interest because of their textural nature, complex structures and the difficulty of obtaining accurate diagnostic results which are efficient and time effective [16]. This paper focuses on an application in the area of non-destructive evaluation which may have other applications in robotics and materials science (e.g. [18], [19], [20], [21]). Indeed, in principle, any image analysis problem that can be enhanced using fractal geometry may find the results associated with his paper and the system developed to be of value.

C. Euclidean and Fractal Geometry

The underlying philosophy of Euclidean geometry is that we can combine primitive objects to build-up and construct complex ones. To do this we first need to analyze a complex object in terms of its 'elements' to construct a simple set of primitives. This is the basis for the construction of most man-made objects and computational Euclidean geometry including computer aided design, solid geometry, etc. It is also the basis which we tend to use for analysing a complex problem. Fractal geometry is based on looking at things in terms of the 'big picture' and observing the fact that the 'smaller pictures' look similar. It uses ideas, axioms, theorems and so on associated with complex objects with repeating patterns, and includes abstract concepts such as infinite repeatability. Hence, unlike Euclidean geometry, the philosophy of fractal geometry is to construct an object by classifying it in terms of its repetitive underlying structure and repeating this structure again and again. This is the basis for the 'geometry' of most natural objects [22]. In each case, the image is of an object that, at first sight, appears relatively complex with different textures. However, if we 'look' at the object imaginatively enough in terms of its repeating patterns at different scales, then this complexity starts to be seen for what it is - self-similar simplicity! This 'simplicity' is compounded in a range of different fractal parameters which have a variety of computational procedures associated with their accurate determination [1]. In the same way that a two-dimensional Euclidean object might be classified in terms of Euclidean parameters (such as the perimeter, area, 'centre of gravity' and so on) for the purpose of generating a matching template, so, fractal objects can be classified in terms of fractal geometric parameters such as the Fractal Dimension, Information Dimension and Lacunarity, for example [23], [22] [24].

The inclusion or otherwise of such 'fractal parameters' in terms of improving vision systems remains to be understood. However, from the research undertaken to date by the authors (e.g. [36], [37], [16]), it is clear that texture based analysis alone is not sufficient in order to design a recognition and classification vision system. Both Euclidean and fractal parameters need to be combined into a feature vector in order to develop an operational vision system which includes objects that have textural properties. The integration of Euclidean with fractal geometric parameters provides a more complete 'tool-kit' for pattern recognition in combination with supervised learning through fuzzy logic criteria, for example. The underlying goal is to attempt to classify the optimum number of metrics required for any given application in relation to the use of a relatively simple fuzzy inference decision engine and/or a more sophisticated Artificial Neural Network.

D. Texture Segmentation

Segmentation is the process by which image sub-units are assigned to objects in a scene. There are three main types of segmentation in practice: pixel based methods, edge based methods and region based methods. Such techniques often require a priori knowledge of the types of texture present and are typically applied to rectangular regions which are iteratively reduced in size until internal homogeneity is achieved. Neighbouring regions are then tested in an attempt to form aggregations of uniform texture.

Four commonly used techniques for classifying texture are: (i) Frequency space analysis; (ii) spatial grey level dependence (co-occurrence) matrices, a technique that computes a matrix of measures taken from a digital image and then defines features (such as the entropy, correlation, local homogeneity, etc.) as functions of the matrix; (iii) directional autocorrelations to determine periodicity in which an attempt is made to discover if there is any repeating pattern(s) in a given direction. This technique involves taking pixels adjacent in some direction and correlating them with themselves after shifting them by one pixel, two pixels, etc; (iv) fractal geometric analysis, e.g. [1], [4], [23], [22], [24], [25].

Fractal geometry is the geometry of self-similarity in which an object appears to look similar at different scales. The term fractal is derived from the Latin adjective *fractus*. The corresponding Latin verb 'frangere means 'to break', to create irregular fragments. In addition to 'fragmented' *fractus* can also mean 'irregular', both meanings being preserved in fragment. The geometry of nature appears to have a fundamental feature which is that the shapes of things look the same at different scales (self-similarity) or at least have an affinity at different scales (self-affinity). There is a fundamental relationship between texture and fractals especially random fractals (i.e. fractals that are statistically self-affine). The way in which we tend to perceive this 'geometry' is in terms of 'texture', 'an elusive notion which mathematicians and scientists tend to avoid because they can not grasp it... and... much of fractal geometry could pass as an implicit study of texture' - B Mandelbrot [22].

The self-affine characteristics of features occurring in so many images leads directly to the question as to how such

characteristics can be used to enhance the machine interpretation of such images. Fractal geometry provides a range of metrics which can be used to classify features that are characterized (not necessarily exclusively) by their texture. The research reported in this paper is based on a more general theme which is to develop computer tools to improve and/or automate image analysis using expert and/or Artificial Neural Networks, trained with metrics that include the fractal properties of an image.

The value of using fractal geometry for image analysis lies in its potential to classify an image into different regions of texture by using, for example, the Fractal Dimension as a measure for texture, e.g. [1], [24]. Significant differences in the texture can occur between two images or between two objects in the same image scene. In such cases, the Fractal Dimension can be used as a measure of this difference assuming that it is constant over the image scene, i.e. that the image is both fractal and stationary. However, in many practical situations, images are rarely stationary in the sense that the characteristics of the features change over the spatial support of the image. Moreover, these characteristics are rarely completely specified in terms of fractal geometric, Euclidean geometric or statistical measures but a combination of all three.

II. Object Recognition

Object recognition is concerned with a machine representation of a feature in an image which includes an interpretation such that a particular class can be assigned to the feature. For a typical object recognition system, the determination of the class is only one of the aspects of the overall task. In general, pattern recognition systems receive data in the form of some experimental variables which collectively form a stimuli vector [26], [27]. The determination of relevant attributes in the features that are present within the stimuli vector is an essential and central kernel in the design of any object recognition system. Typically, an ordered collection of relevant attributes which accurately and most completely represent the underlying features of an object is assembled into a feature vector.

Class is only one of the attributes that may or may not have to be determined depending on the nature of the problem. The attributes may be discrete values, Boolean entities or syntactic labels, for example. Learning in this context amounts to the determination of rules of associations between the features and attributes of an object. Practical image recognition systems generally contain several stages in addition to the recognition engine itself. Recognition represents information processing that is realised by some converter of the information (by some information channel), having an input and output. On input, such a system establishes information on the properties of an object. On output, the information relates to which class or feature of an object that has been assigned *a priori*.

The tasks of constructing and applying formal operations for numerical or character representation of objects in the real world is a principal basis for pattern recognition. The equivalence relations express a fit of an evaluated object to any class considered to have independent semantic units. The recognition classes of equivalence can be set by the user in the construction of an algorithm, which may use qualitative representations or external information on the likeness and differences of objects in the context of a solved task; the basis for the phrase 'recognition with the teacher'. When a computerized system decides on the task of classification without engaging external learning information, it is called automatic classification or 'recognition without the teacher'. The majority of algorithms for pattern recognition require the engagement of a number of algorithms, which can be provided only with high-performance computers [28].

There are two principal methods for object recognition which involve parametric and nonparametric approaches. Statistical, voting and alphabet propositions have been reviewed in [14], [29] and [7], for example. The main disadvantages of this approach is that the classes have to be clearly defined with no overlap. The methods based on the principle of separation and potential functions can be found in [7] and [30] which requires a large amount of training data or preliminary information about the system. This makes the recognition process rather clumsy and less flexible. Overall, there is no system which considers object recognition from superpositions in terms of global scenery. Thus, a principal problem remains, which is how to evaluate an object as a part of the 'bigger picture' without loss of the specific details and textures required for precision classification. In this paper, we focus on a combined approach which includes a method for multiple object location and classification but also introduces concepts from fractal geometry to evaluate the texture(s) of features associated with the metallurgical application considered. This is typical of many computer vision systems whose design is influenced by the application to which it is ultimately applied. Classification of the object(s) considered is undertaken using a Fuzzy Logic engine which requires expert training. In the following section the approach used to design the system is considered in terms of its application for surface inspection and quality control.

III. Object Location

Recognition is the process of comparing individual features against some pre-established template subject to a set of conditions and tolerances. This task can be reduced to the construction of some function determining a degree of proximity of the object to a sample - a template of the object. The process of recognition commonly takes place in four definable stages: (i) image acquisition and filtering; (ii) object location (using edge detection); (iii) measurement of object parameters; (iv) object class estimation and decision making.

A. Background to the Case

Suppose we have a digital image which is given by a (discrete) function $f[x, y]$ and contains some object described by a set of features $X = \{x_1, x_2, ..., x_n\}$. We consider the case when it is necessary to define a sample which is somewhat 'close' to this object in terms of a matching set. The system discussed in this paper is based on an object detection technique that includes a novel segmentation method and must

be adjusted and 'fine tuned' for each area of application. This includes those features associated with an object for which fractal models are well suited [1], [4]. The most basic approach consists of calculating some function of a pointwise coincidence between a map of the object and the image together with a search for a maximum correspondence indicating the closest match between an object pair or sample. In terms of a 'similarity function', this method can be viewed through measures such as a sum of square deviations, a sum of the modulus of deviations or as a paired sum of multiplications of values of brightness (a function of the greatest transparency), for example. The first two similarity functions compute the 'smallness' of a functional pair; instead of searching for a maximum (in the correlation surface after application of a matched filter, for example) it is necessary to search for a minimum.

Not all fragments of an object are equally important for recognition and a broadly distributed functional evaluation matched with weighted coefficients can therefore be undertaken. Appropriate similarity functions can be used as a sum of the weighted squares of deviations, a sum of the weighted modules of deviations and the sum of the weighted multiplication of pairs of brightness values. The correct selection of weight coefficients is important in the field of identification and can be calculated from a given set of samples. The common application for weighted comparisons occurs in the field of Artificial Intelligence and the design of Artificial Neural Networks. The advantage of using neural networks, bearing in mind their high efficiency, lies in the capability of introducing a flexible set of weights during operation (which, in practice, relates to the 'training' of the system using sample data). This property becomes especially important, if the data set is based on a non-stationary model, which varies in time while it is extended and updated.

The system described in this paper uses a knowledge database which generates a result (a decision) by subscribing different objects. The 'expert data' in the application field creates a knowledge database by using a supervised training system with a number of model objects [32]. The method is used with feedback relating to operations including object location and filtering. To illustrate this, we consider a typical example of an image taken of a metal surface with various 'defects' as given in Figure 1.

Figure 1 is the output after applying a Wiener filter and a Sobel edge detector [1]. Here, edge detection is not used to provide continuous edge contours but used to provide a rough location of each defect. The principal aim is to consider the whole image and not the 'detail' associated with the feature(s) in order to provide a rough guide as to the location of each object (defect) in the image scene. The object location algorithm described below is then applied which is based on a measure of the weight coefficients used to provide information about object connectivity, an example output of this procedure being given in Figure 2.
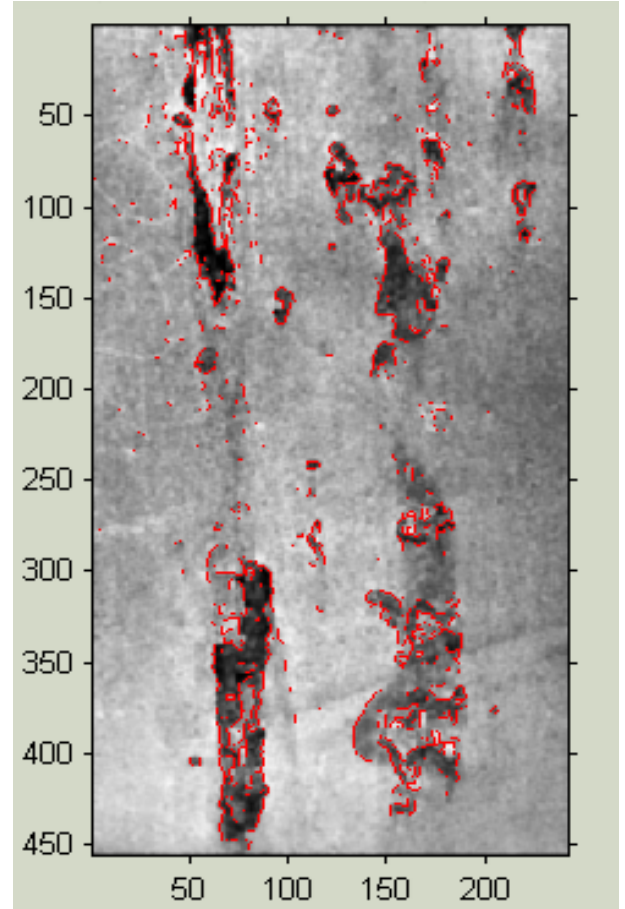

Fig. 1. Example of a digital image of a metal surface with typical defect object after application of a Wiener filter filtering to remove noise and a Sobel edge detector.

B. Object Location Algorithm

Object location is based on the weight coefficients for each pixel (located at x, y) defined as

$$f[x,y] = \begin{pmatrix} f[x-1,y+1] & f[x,y+1] & f[x+1,y+1] \\ f[x-1,y] & p_{\text{edge}}[x,y] & f[x+1,y] \\ f[x-1,y-1] & f[x,y-1] & f[x+1,y-1] \end{pmatrix}$$
$$\otimes \otimes \delta^2[x,y]p_{\text{obj}}[x,y]$$

where $\otimes\otimes$ denotes the two-dimensional (discrete) convolution and $\delta^2[x,y]$ is the two-dimensional Kronecker delta function. Here, $p_{\text{obj}}[x,y]$ is the probability that a pixel at x, y belongs to the object and $p_{\text{edge}}[x,y]$ is the probability that a pixel is close to an edge. These probability function are obtained from a fuzzy logic membership function which has a loop-back to the current object location. Thus, there is a local dependency of a current pixel $f[x,y]$ being recognised as a part of the object from the surrounding pixels on a 3×3 window basis but with a global evaluation based on the functions $p_{\text{obj}}[x,y]$ and $p_{\text{edge}}[x,y]$.

The probability function $p_{\text{obj}}[x,y]$ is a two dimensional matrix and recalculates local values dynamically with regard to $f[x,y]$. We now consider the process of constructing such a matrix. First, we compute the intensity L_{obj} of the object using only those pixels recognised as being part of the object.
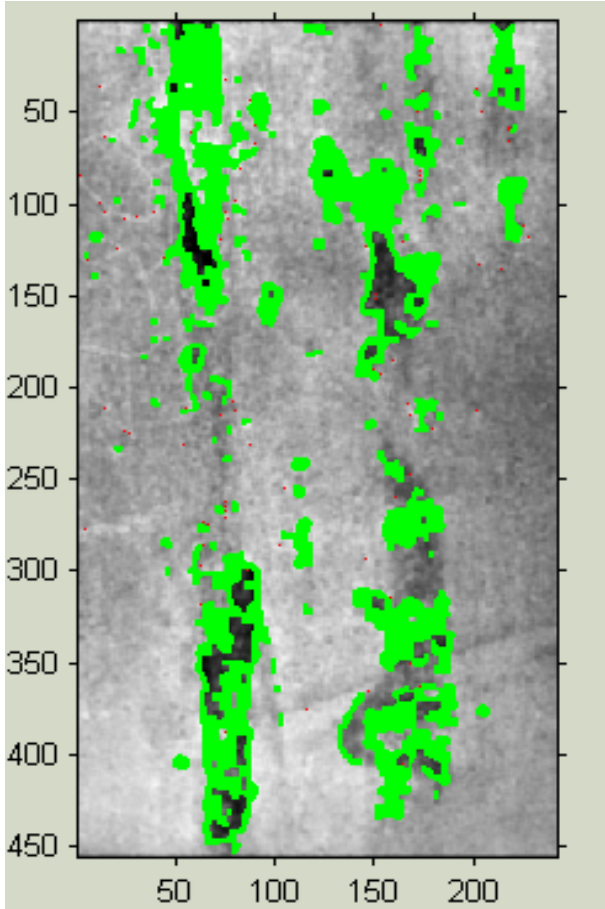
Fig. 2.   Determination of approximate object locations.

Initially L_{obj} has a higher background level L_{bgr} and as long as $L_{\mathrm{obj}} = L_{\mathrm{bgr}}$, it is taken that all objects have been indexed. In order to obtain L_{obj}, we use a probabilistic min-max equation developed through experimental tests that have proved to be robust for the majority of images associated with the application and given by

$$L_{\mathrm{obj}} = \begin{cases} L_x, & L_x \geq L_y; \\ L_y, & \text{otherwise.} \end{cases}$$

where

$$L_x = \frac{1}{2}\left(\min_y \left(\max_x f[x,y] \right) - \langle \max_x f[x,y] \rangle_y \right)$$
$$+ \langle \max_x f[x,y] \rangle_y,$$

$$L_y = \frac{1}{2}\left(\min_x \left(\max_y f[x,y] \right) - \langle \max_y f[x,y] \rangle_x \right)$$
$$+ \langle \max_y f[x,y] \rangle_x.$$

with $\langle \ \rangle$ used to denote the average value. The next stage is to compute a particular value of the membership function which is given by

$$p_{\mathrm{obj}}[x,y] = \sum_{x,y}(f[x,y]L_{\mathrm{obj}}[x,y] - L_{\mathrm{bgr}}[x,y] + L_{\mathrm{edge}}[x,y])$$

where $L_{\mathrm{edge}}[x,y]$ is an edge detection function. An iterative procedure is then applied and continued until a closed border is obtained. Depending upon the application, the iteration can include special filters designed to generate closed edges as given in Appendix I. The contour tracing and convex hull information is stored and used for the classification and decision making procedure. Each object is enumerated before proceeding to the next step of the object recognition process.

IV. Object classification and decision making

Once the object(s) have been isolated, object classification is required in which all possible features that best characterise the object are computed as accurately and efficiently as possible in order to provide a real time application. In the application considered here, we focus on a classification system that involves the textural characteristics of the objects. Some Euclidean and morphological measures are also considered. Using a combination of both Euclidean and fractal geometric measures provides a list of parameters which form the components of a feature vector $X = \{x_1, x_2, ..., x_n\}$. Details of the parameters $x_1, x_2, ..., x_n$ used as part of the systems development described in this paper are provided in Appendix II.

In general, the parameter set that is used depends on the application considered and requires an experimental procedure for optimising the system, details of which lie beyond the scope of this publication. Clearly, using an excessive number of parameters may not affect recognition accuracy but can reduce the computational efficiency of the operational system. The aim is to compute a feature vector that has an optimum number of Euclidean and Fractal geometric measures which can be used as an input to a Fuzzy Logic engine coupled with a suitable supervised learning procedure in order to affect an accurate and robust decision based on the approach described below.

A. Decision Making

Information on classes is stored in Knowledge DataBase (KDB) typically as a specified .kdb file which is loaded into the system depending upon the specific application. The data stored in a KDB represents the probability coefficients for the particular class associated with a particular object, i.e. the class probability is vector $\mathbf{p} = \{p_j\}$. It is estimated from the object feature vector $\mathbf{x} = \{x_i\}$ and membership functions $m_j(\mathbf{x})$ defined in the knowledge database. If $m_j(\mathbf{x})$ is a membership function, then the probability for each jth class and i^{th} feature is given by

$$p_j(\mathbf{x}_i) = \max\left[\frac{\sigma_j m_j(\mathbf{x}_{j,i})}{|\mathbf{x}_i - \mathbf{x}_{j,i}|} \right]$$

where σ_j is the distribution density of values $\mathbf{x}_j$ at the point $\mathbf{x}_i$ of the membership function. The next step is to compute the mean class probability given by

$$\langle p \rangle = \frac{1}{j}\sum_j \mathbf{w}_j p_j$$

where $\mathbf{w}_j$ is the weight coefficient matrix. This value is used to select the class associated with

$$p(j) = \min \left[(p_j \cdot \mathbf{w}_j - \langle p \rangle) \geq 0 \right]$$

providing a result for a decision associated with the jth class. The weight coefficient matrix is adjusted during the learning stage of the algorithm.

The decision criterion method considered here represents a weighing-density min-max expression, the estimation of the decision accuracy being achieved via the density function

$$d_i = |\mathbf{x}_{\sigma_{\max}} - \mathbf{x}_i|^3 + [\sigma_{\max}(\mathbf{x}_{\sigma_{\max}}) - p_j(\mathbf{x}_i)]^3$$

with an accuracy determined by

$$P = \mathbf{w}_j p_j - \mathbf{w}_j p_j \frac{2}{\pi} \sum_{i=1}^{N} d_i.$$

The overall accuracy depends on the level of confidence of an expert. In some cases, an expert may be unable to make a clear decision about which class to which an object belongs. In such cases, an 'overlap' occurs and further data is taken to be required in order to make a decision.

B. Supervised Learning Process

The supervised learning procedure is the most important part of the system for operation in automatic recognition mode. The training set of sample objects should cover all ranges of class characteristics with a uniform distribution together with a universal membership function. This rule should be taken into account for all classes participating in the training of the system. An expert defines the class and accuracy for each model object where the accuracy is the level of self confidence that the object belongs to a given class. The Graphical User Interface (GUI) designed for the training procedure is discussed later in Section VI(F). During this procedure, the system computes and transfers to a knowledge database a vector $\mathbf{x} = \{x_i\}$ which forms the membership function $m_j(\mathbf{x})$. The matrix of weight factors $w_{j,i}$ is formed at this stage accordingly for the ith parameter and j^{th} class using the following expression:

$$w_{i,j} = \left| 1 - \sum_{k=1}^{N} \left[p_{i,j}(\mathbf{x}_{i,j}^k) - \langle p_{i,j}(\mathbf{x}_{i,j}) \rangle \right] p_{i,j}(\mathbf{x}_{i,j}^k) \right|.$$

The result of the weight matching procedure is that all parameters which have been computed but have not made any contribution to the characteristic set of an object are removed from the decision making algorithm by setting $w_{j,i}$ to null.

V. SURFACE DEFECT RECOGNITION

Digital images are taken from real sheet metal surfaces in standard conditions using the same resolution as derived from a moving sheet using line scan cameras. These data are saved into a data base in standard grey-scale format with an 8-bit dynamic range. The data base is generated using a Microsoft SQL Server for several users. The diagram given in

3 shows the configuration of the system for the whole process associated with the surface inspection system. The line scan camera operates like a CCD camera but acquires an image of the surface on a line-by-line basis as it moves at a velocity v. The light source in not shown because it depends on camera requirements and the precise configuration of the image capture device given different ambient lighting conditions, for example. The 'Microscan Controller' is composed of a small frame and provides data on the surface in real time. The image and its parameters are transmitted to the 'Operator PC' which makes macro decisions for large objects. Ultimately, the images of defects are ported to a database (the 'Manufacturing process control software/database') where information on a current 'production run' is stored.
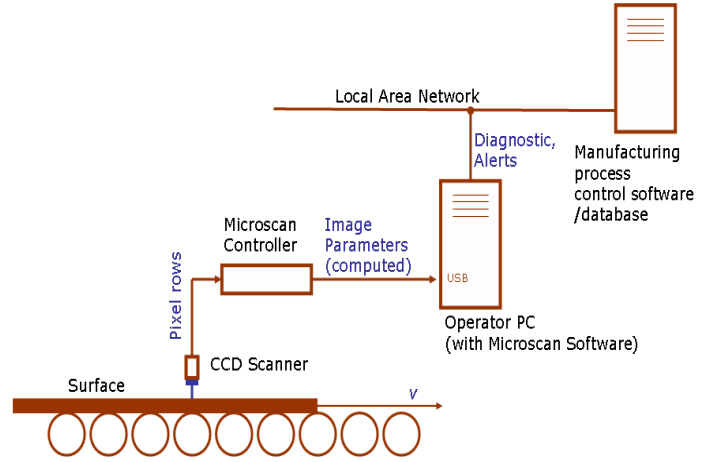

Fig. 3. Configuration of the surface inspection system.

After recognition has taken place, the result is stored is the database, the user checking the location of any defect and inspecting the location dimensions and results as required. The system is designed to recognise six classes of defect: (i) Non-metallic pats - scaling; (ii) Shrunken leaf; (iii) Cusping; (iv) File mark; (v) Clevage crack; (vi) Tear; clevage cracks and Tears being the most important defects in a given production run. Supervised training of the system utilizes upto 20 samples of each class. Approximately 15-20 objects are required for estimating the results associated with a particular class giving results with an accuracy of 85-95% under real operational conditions in real time.

We consider a typical task associated with the surface inspection system. For simplicity, we consider two parameters (two components of the feature vector) and three clearly identifiable defects that are typical of those used to train the system and generate the knowledge database with defect properties as given. Consider the surface defect images illustrating *Scale*, *Clevage crack* and *Cusping* as shown in a Figure 4 obtained using the system in an off-line mode. A Wiener filter is first applied to the image to remove background noise. Object location using a Sobel edge detector is then applied to determine the defects and the object location algorithm discussed in Section III applied. For exposition on the output associated with this example, two parameters are considered (as defined in Appendix II), namely, the fractal dimension and
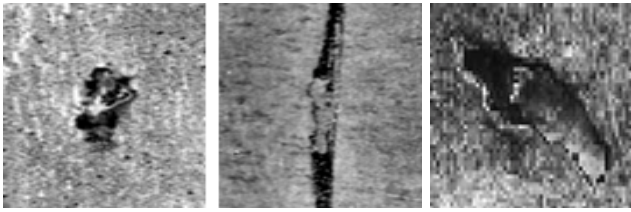
Fig. 4. Example surface defects illustrating *Scale* (left), *Clevage crack* (centre) and *Cusping* (right)

the convexity factor (one of the 'geometrical features' defined in Appendix II). The system is trained, the supervised learning process being described in Section IV(B). In the learning process, the system stores the membership functions as a KDB file, the membership functions for each parameter associated with the three defects considered being shown in Figure 5 and Figure 6.
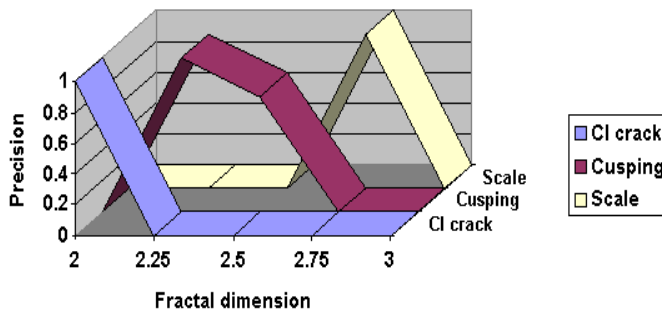

Fig. 5. Fractal dimensions associated with three surface defects.
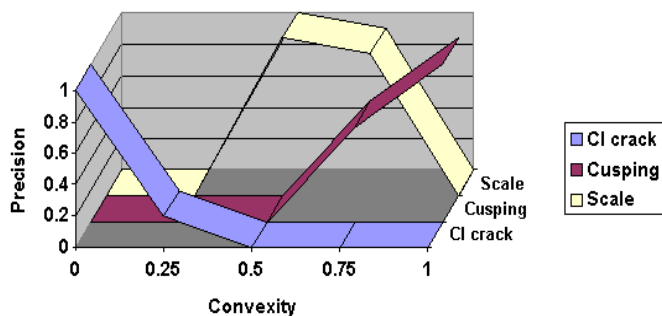

Fig. 6. Convexity associated with three surface defect.

Here, the horizontal axis displays the parameter values. On the vertical axis are displayed values relating to the precision of recognition established by an expert (in this cases, a metallurgist) during the training session on the basis of their (expert) knowledge on the nature of the defect. After training the system using a range of such images (all taken to belong to one of the three types: *Scale*, *Clevage Crack*, we consider a new image and undertake the same operations as those made during the training session. The system finds the object and computes its fractal dimension and convexity factor, the degree of precision associated with the fractal dimension, for example, being given in Figure 7. The degree of confidence
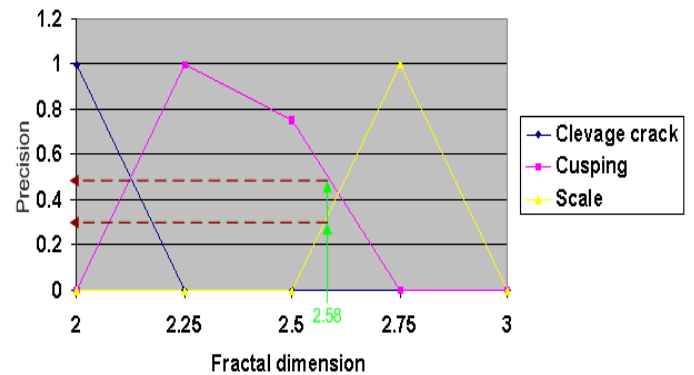

Fig. 7. Precision definition

for each class is computed and the maximum of these values taken to characterize that class to which the given image corresponds. In the example given, the output is *Scale*.

The example given above illustrates the principle upon which the system operates. In practice, the system has been designed using the parameters defined in Appendix II and the six defect classes defined above. By recording images of
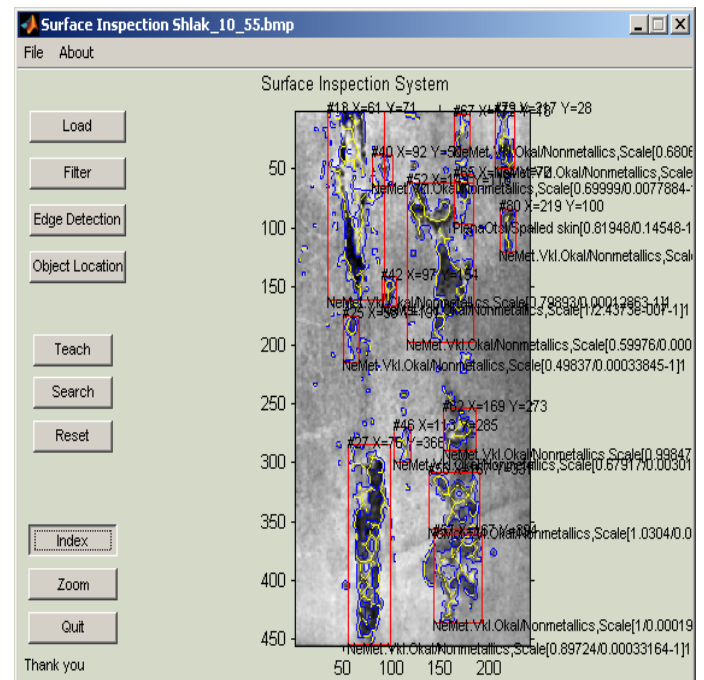
Fig. 8. Example output of the surface inspection recognition system in which the defect type, its location in the image frame and associated parameter values are displayed.

defects obtained through different frame grabbers and with different system display utilities, a range of results are obtained which are within the acceptable accuracy associated with an industrial quality control system for monitoring the production of sheet-steel, image standardization and correction being based on the use of Adobe Photoshop. Under identical systems conditions and image acquisition, the dispersion in accuracy of the system does not exceed 5. The system developed for this purpose is discussed further in the following section.

VI. Surface Inspection System Software: ORSIS

In this section, we describe the basis and operational performance associated with the Oxford Recognition Surface Inspection System (ORSIS) developed by Oxford Recognition Limited in collaboration with Dublin Institute of Technology and Novolipetsk Iron and Steel Corporation. The system developed has been designed for use with a standard PC with input from a good quality digital camera using Commercial-Off-The Shelf hardware. It analyses the structure of a defect, makes a decision using a knowledge database and outputs a result. Technical surface inspection experts create a knowledge database by training the system using a number of case-study images. This produces a KDB which 'improves' with the use of the system.

The current system is composed of the following basic steps:

1) Filtering: The image is Wiener filtered to reduce noise and remove unnecessary and obtrusive features such and light flecks.
2) Edge detection: The image is segmented to perform a separate analysis of each object.
3) Object Location: Implements an iterative algorithm for automatic localisation (as described in Section III).
4) Teaching: For each object, a set of recognition features are computed based on those discussed in Section IV. The features are numeric parameters that describe the objects in terms of a variety of Euclidean and fractal geometries and statistical features in one- and two-dimensions. The one-dimensional features correspond to the border of the defect and the two-dimensional features relate to the surface within the object boundary. In addition, a recognition algorithm is used to analyse the defect *structure* as illustrated in Figure 9. This provides information on the possible growth of the object when an inspection is undertaken over a period of time.
5) Search - Decision Making: The system uses fuzzy logic to combine features into a decision. A decision is the estimated class of the object and its accuracy. In this particular application, the output is designed to give six classes as described in Section V.
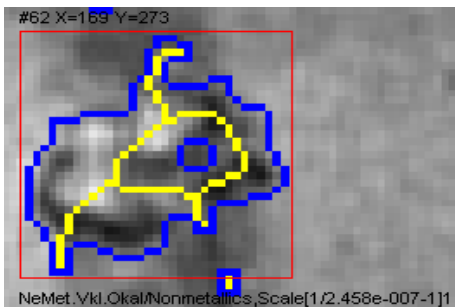

Fig. 9. Analysis of the structure of a defect for comparative features.

A. Key Advantages

The technology delivers high accuracy and automation which has been made possible by the following innovations:

1) Fractal analysis: The surface and boundary characteristics of the metallurgical defects considered in this application have natural fractal properties. Computation of these fractal properties provides for enhancement of an efficient and effective detection of defects that would not otherwise be possible.
2) Extended set of detectable features: High accuracy is achieved when multiple features are measured together and combined into a single result.
3) Advanced fuzzy logic engine: The knowledge-based recognition scheme used enables highly accurate diagnosis and offers significant improvements over current methods.

B. Knowledge Database

ORSIS is a knowledge-based system and requires extensive training before implementation in a manufacturing and/or process engineering environment. The training process includes a review and probabilistic classification of appropriate images by technical experts who can input results using the interface shown in Figure 10. The minimal number of training images depends on the number of classes and the diversity of objects within each class. The following sections describe how this application can be downloaded, installed and implemented. The demo version, which has been made available for this publication, is itemised in the following sections.

C. Platform Requirements

System Requirements

- Windows 98/ME/2000/XP
- >256 Mb RAM
- >30 Mb hard disk space

Image Requirements

- Input format: JPEG, BMP or TIF
- Image size: 640x480 to 1024x728 (higher image resolution requires RAM of 512 Mb or more)
- Good focus with no motion blur
- Uniform lighting
- Capture of the object which is well centered in the image frame and does not, for example, extend beyond the image boundaries

D. Installation

1) The downloadable demo version of ORSIS is available from hppt://www.oxreco.com/metalsetup.zip.
2) Installation is initiated through `setup.exe` from the root folder in which the downloaded application has been placed.
3) Follow the instructions on screen.

E. Recognition Mode

1) Click **Load Image** and select an image of a 'defect'. Samples can be found in folder `Pictures`, which, by default, reside in

$$...\backslash \texttt{ORSIS\_Demo}\backslash \texttt{Pictures}\backslash.$$

2) Click **Filter, Edge Detection, Object Location, Search**.
3) Recognition and class estimation takes approximately 20 seconds (for a typical modern PC operating under an XP windows environment) producing an output of the type given in (Figure 8).

F. Teaching

1) The default knowledge database is loaded from

$$...\backslash \texttt{ORSIS\_Demo}\backslash \texttt{bin}\backslash \texttt{def.kdb}.$$

To create a new database, select **New knowledge DB** from in the **File** menu (see Figure 11).

2) Click **Load Image** and select an image of a specific defect.
3) Click **Teach**.
4) ORSIS analyses the image for 10-20 seconds whereupon the Teaching Dialog (Figure 10) pops up for each object. The user is then required to enter an estimation:
   a) Class: 1 (Scaling), 2 (Non-metallic pats), 3 (Shrunken leaf), 4 (File mark), 5 (Clevage crack), 6 (Tear)
   b) Probability: a number between 0.0 and 1.0; 1.0 equates to 'absolutely sure', whereas zero should not normally be used. Typical values are 0.90-0.95.
5) Repeat Steps 1-4 to process all training images.
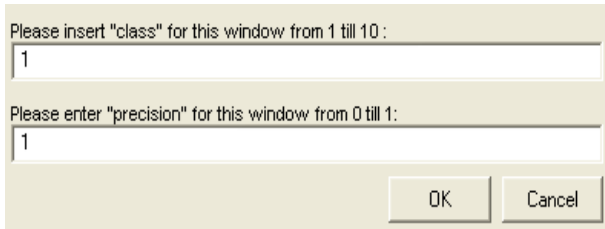6) Select **Save knowledge DB...** from **File** (see Figure 11) and enter a file name for the knowledge database.

Fig. 10.   Teaching dialog

G. User Interface

Main Window The commands available from the main window (see Figure 8) are summarised in Table I.
File Menu
  The file menu is given in Figure 11 whose menu items and actions are summarised in Table II.
Command Line Execution
  To launch the system in automatic mode type:

$$\texttt{ORSIS.exe\_"LoadGraf"\_\%1}$$

where `%1` is an image name (JPEG, BMP or TIFF formats are supported).

Button	Action
Load	loads image in JPEG, BMP or TIF formats
Filter	performs filtering of loaded image
Edge Detection	performs edge detection
Object Location	performs recursive object location algorithm
Teach	performs teaching in automatic mode
Search	performs recognition in automatic mode
Reset	performs reset Knowledge Data Base
Index	displays the defect boundary and features for structural analysis
Zoom	switches on/off the zoom mode (use the left mouse button to zoom in and the right button to zoom out)
Quit	closes the application

TABLE I
BUTTONS AND ASSOCIATED ACTIONS ON THE MAIN WINDOW OF ORSIS.

Menu Item	Action
New knowledge DB	resets knowledge database
Open knowledge DB...	loads knowledge database from a kdb file)
Save knowledge DB...	saves knowledge database to a kdb file
Open Image...	loads image in JPEG, BMP or TIF formats
Exit	closes the application

TABLE II
MENU ITEMS AND ASSOCIATED ACTIONS ON FILE MENU OF ORSIS.

VII. DISCUSSION

The methods discussed in this paper represent a novel approach to designing an object recognition system that is robust in classifying textured features, the application considered in this paper having required a symbiosis of the parametric representation of an object and its geometrical invariant properties. In comparison with existing methods, the approach adopted here has the following advantages:

Speed of operation. The approach uses a limited but effective parameter set (feature vector) associated with an object instead of a representation using a large set of values (pixel values, for example). This provides a considerably higher oper-

Fig. 11.   File menu

ational speed in comparison with existing schemes, especially with composite tasks, where the large majority of methods require object separation. The principal computational effort is that associated with the computation of the features as defined in Appendix II.

Accuracy. The methods constructed for the analysis of sets of geometrical primitives are, in general, more precise. Because the parameters are feature values, which are not connected to an orthogonal grid, it is possible to design different transformations (shifts, rotational displacements and scaling) without any significant loss of accuracy compared with a set of pixels, for example. On the other hand, the overall accuracy of the method is directly influenced by the accuracy of the procedure used to extract the required geometrical tags. In general, the accuracy of a method will always be lower, than, for example, classical correlative techniques, where, due to padding, errors can occur during the extraction of a parameter set. However, by using precise parametrization structures based on the features defined in Appendix II, remarkably good results are obtained.

Reliability. The proposed approach relies first and foremost on the reliability of the extraction procedure used to establish the geometrical and parametric properties of objects, which, in turn, depends on the quality of the image; principally in terms of the quality of the contours. It should be noted, that the image quality is a common problem in any visual system and that in conditions of poor visibility and/or resolution, all vision systems will fail. In other words, the reliability of the system is fundamentally dependent on the quality of the input data.

Aditional Features. An additional feature of the system discussed in this paper, is that the sub-products of the image processes can be used for tasks that are related to image analysis such as a search for objects in a field of view, object identification, maintaining an object in a view field, optical correction of a view point and so on. These can include tasks involving the relative motion of an object with respect to another or with respect to background for which the method considered can be applied - collision avoidance tasks, for example, in robotics.

Disadvantages. Among the characteristic disadvantages of the approach, it should be noted that: (i) The method requires a considerable number of different calculations to be performed and appropriate hardware requirements are therefore mandatory in the development of a real time system; (ii) the accuracy of the method is intimately connected with the required computing speed - an increase in accuracy can be achieved but may be incompatible with acceptable computing costs. In general, it is often difficult to acquire a template of samples under real life or field trial conditions which have a uniform distribution of membership functions. If a large number of training objects are non-uniformly distributed, it is, in general, not possible to generate accurate results.

Decision Processing steps. The approach to the decision process proposed includes the following important steps: (i) the estimation of the density distribution is accurately determined from the original samples in the membership function during a supervised learning phase which improves

the recognition accuracy under non-ideal conditions; (ii) the pre-filtering procedures provide a good response to the required features of the object without generating noise; (iii) the segmentation procedure discussed in Sections III efficiently select only those objects required; (iv) computation of fractal parameters (the Fractal Dimension and the Lacunarity) helps to characterize the textural features (in terms of texture classification) associated with an object (defect); (v) the integration of Euclidean with fractal geometric parameters provides a more complete suite of tools for pattern recognition in combination with supervised learning through fuzzy logic criteria.

VIII. CONCLUSION

This paper has been concerned with the task of developing a methodology and implementing applications that are concerned with two key tasks: (i) the partial analysis of an image in terms of its fractal structure and the fractal properties that characterize that structure; (ii) the use of a fuzzy logic engine to classify an object based on both its Euclidean and fractal geometric properties. The combination of these two aspects has been used to define a processing and image analysis engine that is unique in its modus operandi but entirely generic in terms of the applications to which it can be applied.

The work reported in this paper is part of a wider investigation into the numerous applications of pattern recognition using fractal geometry as a central processing kernel. This has led to the design of a new library of pattern recognition algorithms including the computation of parameters in addition to those that have been reported here such as the information dimension, correlation dimension and multi-fractals [1]. The inclusion or otherwise of such parameters in terms of improving vision systems such as the one considered here remains to be understood. However, from the work undertaken to date, it is clear that texture based analysis alone is not sufficient in order to design a recognition and classification system. Both Euclidean and fractal parameters need to be combined into a feature vector in order to develop an operational vision system which includes objects that have textural properties such as those associated with non-destructive evaluation, materials science, medical imaging, remote sensing and so on.

The creation of logic and general purpose hardware for artificial intelligence is a basic theme for any future development based on the results reported in this paper for the applications developed and beyond. The results of the current system can be utilized in a number of different areas although surface inspection imaging and defectoscopy, in general, would appear to be one of the most natural fields of interest because of the nature of the images available, their complex structures and the difficulty of obtaining accurate diagnostic results which are efficient and time effective. A further extension of our approach is to consider the effect of replacing the fuzzy logic engine used to date with an appropriate Artificial Neural Network. It is not clear as to whether the application of an ANN could provide a more effective system and whether it could provide greater flexibility with regard to the type of images used and the classifications that may be required.

Fig. 12.   C++ algorithm for contour generation (object edge recognition)

```
int ks=0;
ListDotsX[0]=StartX;
ListDotsY[0]=StartY;
long DotX[9]={0,-1,0,1,1,1,0,-1,-1};//Extend surface of
long DotY[9]={0,-1,-1,-1,0,1,1,1,0};//wheel from axle.
int Ox=StartX;// Set position of points 'A' and 'B'.
int Oy=StartY+1;
int ht,HaveToch=2;
do {   // Cycle while not returning to initial coordinates.
  for (nl=1;nl<=7;nl++){ // Cycle surface of wheel.
    ht=HaveToch+nl;
    if (ht>8) ht=ht-8;
    x1=Ox+DotX[ht]; //Calculate coordinates for
    y1=Oy+DotY[ht]; //surface of the wheel.
    if (*(pp + x1*h + y1)==0){ //If 0 then move the wheel
      Ox=x1; // axle and calculate the point of tangency
      Oy=y1; // of surface with object edge.
      if((ht==1)||(ht==3)||(ht==5)||(ht==7)) HaveToch=ht+5;
      if((ht==2)||(ht==4)||(ht==6)||(ht==8)) HaveToch=ht+6;
      if (HaveToch>8) HaveToch=HaveToch-8;
      break;
    }
    if (*(pp + x1*h + y1)==1){ //If 1 then check the
      if ((x1==StartX)&&(y1==StartY)) break;// initial
      ks=ks++;//conditions
      ListDotsX[ks]=x1;// and save the edge
      ListDotsY[ks]=y1;// coordinate of the object.
    }
  }
} while ((x1!=StartX)||(y1!=StartY));
```

Fig. 13.   C++ algorithm for Convex Hull

```
{NListDotsX[02*((maxX-minX)+(maxY-minY))] //Create dot list
 NListDotsY[02*((maxX-minX)+(maxY-minY))]}//of bound object.
ListDotsX[0]=StartX; // Set the initial coordinates
ListDotsY[0]=StartY; // for end of thread.
int nc=0,x4,y4,Mx4,My4;
double fi,cs,sn,step,r,RR,bz,sz;
for(nt=0;nt<(2*((maxX-minX)+(maxY-minY)));nt++){//Begin walk
  fi=atan2(NListDotsY[nt]-StartY,...          // around object.
  NListDotsX[nt]-StartX);
    RR=sqrt(pow((NListDotsX[nt]-StartX),2)+...
                          +pow((NListDotsY[nt]-StartY),2));
    cs=cos(fi);
    sn=sin(fi);
    if (fabs(sn)>fabs(cs)){ //Calculate the step length.
      bz=fabs(sn);
      sz=fabs(cs);
    }else{
      bz=fabs(cs);
      sz=fabs(sn);
    }
    step=sqrt(pow(((sz*(1-bz))/bz),2)+pow((1-bz),2))+1;
    for (r=0;r<=RR;r+=step){ // Search for all objects
      x4=round((double)StartX + r*cs);//in line of thread.
      y4=round((double)StartY + r*sn);
      if (*(ppg + x4*h + y4) == 1){
        Mx4=x4; // Save last coordinate
        My4=y4; // in temporary variables.
      }
    }
    if (((Mx4!=StartX)&&(My4!=StartY)) || //Last dot check.
 ((Mx4==StartX)&&(Mx4==NListDotsX[nt])&&(Mx4!=NListDotsX[nt+1]))||
 ((My4==StartY)&&(My4==NListDotsY[nt])&&(My4!=NListDotsY[nt+1]))){
      StartX=Mx4; // Assign new start coordinates.
      StartY=My4;
      nc=nc++;
      ListDotsX[nc]=StartX; // Save list of coordinates
      ListDotsY[nc]=StartY; // for polygon.
    }
}
```

APPENDIX I
SEGMENTATION ALGORITHMS

The algorithms presented in this Appendix are reproduced here for completeness due to their key importance with regard to designing an effective surface inspection system which requires an accurate and robust determination of an object boundary so that the feature elements defined in Appendix II can be applied effectively. After application of a Sobel edge detector to provide qualitative estimates on the location of each defect in the image plane and application of the object location algorithm given in Section III(B) the Contour Tracing Algorithm and Convex Hall Algorithm given below are applied to yield (respectively): (i) a quantitative estimate of the object(s) boundary from which the fractal dimension can be computed; (ii) estimates of the Euclidean features described in Appendix II.

A. Contour Tracing Algorithm

The C++ code for this algorithm is given in Figure 12 which computes the list of coordinates of the edge points of the segmented object.

The advantage of this algorithm over conventional edge detection techniques is that it considers not only the brightness gradient but also the spatial distribution in terms of the object as a whole. The benefit of this approach includes the computational costs that are reduced on average by a factor 2-3 and depend only on the complexity of the object.

B. Convex Hull Algorithm

This algorithm is used as a segmentation procedure for image recognition which is the basis for the MathWorks MATLAB function 'Qhull'. However, the algorithm presented here differs from that available in MATLAB in terms of its simplicity, reliability and computational speed. The reason for

this is that the number of cycles performed is limited and equal only to the total border length of the object. The main idea can be thought of in terms of walking around a contour while pulling a 'thread' which is attached to the object at a fixed point (initial condition). At any 'point of curvature', the thread stores the coordinates of the outer polygonal point. Thus, the path of the perimeter around the object provides the coordinates of all the outer polygonal points. The initial condition can be at any point along the object boundary. However, the direction of a detour does not depend on this conditions. The C++ code for this algorithm is given in Figure 13. The algorithm provides information on the basic Euclidean geometry of the object such the boundary area and perimeter used to define the convexity discussed in Section V, for example. These Euclidean metrics are used to derive Euclidean features as defined in Appendix II.

APPENDIX II
FEATURE VECTOR ELEMENTS

Features (which are typically compounded in a set of metrics - floating point or decimal integer numbers) describe the object state in an image and provides the input for a decision making engine. The issue of what type, and how many features should be used to develop a computer vision system, is critical in the design. The system considered here has been developed to include features associated with the texture of an object, i.e. features that are compounded in certain parameters associated with the fractal properties of a surface defect that are a measure texture, namely, the fractal dimension and the Lacunarity. The following features and their

derivatives have been considered (through experimentation) in the system reported.

Average Gradient G

describes how the intensity changes when scanning from the object center to the border. The object gradient is computed using the least squares method compounded in the following result:

$$g = \frac{N \sum\limits_{(m,n)\in S} r_{m,n}\tilde{f}_{m,n} - \sum\limits_{(m,n)\in S} r_{m,n} \sum\limits_{(m,n)\in S} \tilde{f}_{m,n}}{N \sum\limits_{(m,n)\in S} r_{m,n}^2 - \left(\sum\limits_{(m,n)\in S} r_{m,n}\right)^2},$$

where $\tilde{f}_{m,n}$ is the (digital) image of a defined object (after application of the object location algorithm), N is the number of pixels defining the object (which is of compact support S) and $r_{m,n}$ is the distance between (m,n) and the center (m',n'), i.e.

$$r_{m,n} = \sqrt{(m - m')^2 + (n - n')^2}.$$

The center coordinates (m',n') correspond to the local maximums of $\tilde{f}_{m,n}$ within the cluster. The cluster gradient is the average of object gradients,

$$G = \langle g_i \rangle_{i \in S}$$

where $i \in S$ is the object index.

Fractal Dimension D

determines the frequency characteristics of the object. It represents a measure of texture and characterises a random fractal image with a power spectrum of the form

$$P^2(k_x, k_y) = c|k|^{-2q},$$

where $|k| = \sqrt{k_x^2 + k_y^2}$ is the spatial frequency, c is a constant and $q = 4 - D$. Both D and c can be computed using a least squares method [1].

Lacunarity (Gap Dimension) Λ_k

characterizes the way the 'gaps' are distributed in an image [22], [1]. The gap dimension is, roughly speaking, a measure of the number of light or dark regions in an image. It is defined for a degree k by

$$\Lambda_k = \left\langle \left| \frac{f_{m,n}}{\langle f_{m,n} \rangle} - 1 \right|^k \right\rangle^{\frac{1}{k}},$$

where $\langle f_{m,n} \rangle = \frac{1}{N} \sum f_{m,n}$ denotes the mean value. For the system described in this paper, an average of local Lacunarities of the degree $k = 2$ is measured.

Symmetry Features S_n and M

are estimated by morphological analysis in a three-dimensional space, i.e. two-dimensional spatial coordinates and intensity. A symmetry feature S_n is measured for a given degree of symmetry n (currently $n = \{2,4\}$). This value shows the deviation from a perfectly symmetric object, i.e. S_n is close to zero when the object is symmetric and $S_n > 0$ otherwise. Feature M describes the fluctuation of the centre of mass for pixels with different intensities; $M = 0$ for symmetric objects and $M > 0$ otherwise.

Structure γ

provides an estimation of the 2D curvature of the object in terms of the following:

$$\gamma < 0, \text{if object bulging is less than a threshold};$$

$$\gamma = 0, \text{if the object has standard bulging};$$

$$\gamma > 0, \text{if object has a higher level of bulging}.$$

Geometrical Features

include the minimum $R_{\min}$ and maximum $R_{\max}$ radius of the object (or ratio $R_{\max}/R_{\min}$), object area S, object perimeter P (and the ratio S/P^2 - the 'convexity') and the coefficient of infill S/S_R, where S_R is the area of the bounding polygon which, in this application, is determined using the convex hull algorithm given in Appendix I.

The present solution detects objects by computer analysis using mixed mode features that are based on Euclidean and fractal metrics. The procedure of object detection is performed at the segmentation stage and needs to be adjusted for each area of application. The recognition algorithm then makes a decision using a knowledge database and outputs a result by subscribing objects based on the features defined above. The 'expert data' associated with a given application creates a knowledge database by using the supervised training system with a number of model objects as described in Section IV.

REFERENCES

[1] M. J. Turner, J. M. Blackledge and P. R. Andrews, *Fractal Geometry in Digital Imaging*, Academic Press, London, 1998.
[2] J. S. Lim, *Two-Dimensional Signal and Image Processing*, Prentice-Hall, 1990.
[3] J. M. Blackledge, *Digital Signal Processing*, 2nd Edition, Horwood Publishing, 2006.
[4] J. M. Blackledge, *Digital Image Processing*, Horwood Publishing, 2005.
[5] E. R. Davies, *Machine Vision: Theory, Algorithms, Practicalities*, Academic press, London, 1997.
[6] H. Freeman, *Machine vision. Algorithms, Architectures, and Systems*, Academic press, London, 1988.
[7] J. Louis and J. Galbiati, *Machine Vision and Digital Image Processing Fundamentals*, State University of New York, New-York, 1990.
[8] W. E. Snyder and H. Qi, *Machine Vision*, Cambridge University Press, England, 2004.
[9] D. Marr, *On the Theory of Edge Detection*, Proceedings of the Royal Society, London, B 207: 127-217, 1980
[10] V. S. Nalwa and T. O. Binford, *On Detecting Edges*, IEEE Trans. Pattern Analysis and Machine Intelligence, PAMI-8, 699-714, 1986.
[11] E. Cox, *The Fuzzy Systems Handbook*, AP Professional, 1994.
[12] V. B. Rao and H. V. Rao, *C++ Neural Networks and Fuzzy Logic*, MIS Press, 1993.
[13] N. K. Kasabov, *Foundations of Neural Networks, Fuzzy Systems and Knowledge Engineering*, MIT Press, 1996.

[14] R. Boyle, M. Sonka and V. Hlavac, *Image Processing, Analysis and Machine Vision*, PWS, USA, 1999.

[15] J. M. Blackledge, *On the Synthesis and Processing of Fractal Signal and Images*, Applications of Fractals and Chaos, Springer, 81-100, 1993

[16] J. M. Blackledge and D. A. Dubovitskiy, *Object Detection and Classification with Applications to Skin Cancer Screening*, ISAST Transactions of Intelligent Systems, Vol 1, No 1, 34-45, 2008 - http://www.isastorganization.org/ISAST_IS_2_2008.pdf

[17] J. M. Blackledge, B. Foxon and S. Mikhailov, *Fractal Dimensions Segmentation*, Image Processing: Mathematical Methods and Applications, (Ed. J. M. Blackledge), 249-289, 1997.

[18] H. M. Singer, I. Singer-Loginova, J. H. Bilgram and G. Amberg, *Morphology Diagram of Thermal Dendritic Solidification by Means of Phase-fold Models in Two and Three Dimensions*, Journal of Crystal Growth 296, 2006, 58-68.

[19] T. Mehran, A. D. C. Chan, A. Miri and R. M. Dansereau, *Fractal Analysis of Surface Electromyography Signals: A Novel Power Spectrum-Based Method*, Journal of Electromyography and Kinesiology, Elsevier, 2008 (accepted for publication: www.elsevier.com/locate/jelekin)

[20] D. Sanyal, P. Ramachandrarao and O. P. Gupta, *A Fast Strategy for Simulation of Phase Change Phenomena at Multiple Length Scale*, Computation Materials Science 37, 166-177, Elsevier, 2006.

[21] D. Sanyal, P. Ramachandrarao and O. P. Gupta, *A Fractal Description of Transport Phenomena in Dendritic Porous Network*, Chemical Engineering Science, 61, 307-315, Elsevier, 2006.

[22] B. B. Mandelbrot, *The Fractal Geometry of Nature*, Freeman, 1983.

[23] B. B. Mandelbrot, *Fractals: Form, Chance and Dimension*, Freeman, 1977.

[24] M. F. Barnsley, R. L. Dalvaney, B. B. Mandelbrot, H. O. Peitgen, D. Saupe and R. F. Mandelbrot, *The Science of Fractal Images*, Springer, 1988.

[25] M. F. Barnsley, *Fractals Everywhere*, Academic Press, 1988.

[26] W. E. L. Grimson, *Object Recognition by Computer: The Role of Geometric Constraints*, MIT Press, 1990.

[27] B. D. Ripley, *Pattern Recognition and Neural Networks*, Academic Press, Oxford, 1996.

[28] R. B. Macy and A. S.Pandya, *Pattern Recognition with Neural Networks in C++*, IEEE Press, Florida Atlantic University, 1995.

[29] E. L. Hall, *Computer Image Processing and Recognition*, Academic press, London, 1979.

[30] C. T. Leondes, *Image Processing and Pattern Recognition*, Academic Press, London, 1998.

[31] T. Pavlidis, *Algorithms for Graphics and Image Processing*, Computer Science Press, USA, 1982.

[32] L. A. Zadeh, *Fuzzy Sets and their Applications to Cognitive and Decision Processes*, Academic Press, New York, 1975.

[33] E. H. Mamdani, *Advances in Linguistic Synthesis of Fuzzy Controllers*, J. Man. Mach., 8, 669-678, 1976.

[34] E. Sanchez, *Resolution of Composite Fuzzy Relation Equations*, Int. Control, 30, 38-48, 1976.

[35] N. Vadiee, *Fuzzy Rule Based Expert System-I*, Prentice Hall, Englewood, 1993.

[36] J. M. Blackledge, *Diffusion and Fractional Diffusion based Models for Multiple Light Scattering*, ISAST Transactions on Electronics and Signal Processing, Vol 1, No. 1, 38- 60, 2007 - http://www.isastorganization.org/ISAST_ES_1_2007.pdf

[37] J. M. Blackledge, *Application of the Fractal Market Hypothesis for Macroeconomic Time Series Analysis*, ISAST Transactions on Electronics and Signal Processing, Vol 1, No. 2, 56-80, 2008 - http://www.isastorganization.org/ISAST_ES_2_2008.pdf

Jonathan Blackledge graduated in physics from Imperial College in 1980. He gained a PhD in theoretical physics from London University in 1984 and was then appointed a Research Fellow of Physics at Kings College, London, from 1984 to 1988, specializing in inverse problems in electromagnetism and acoustics. During this period, he worked on a number of industrial research contracts undertaking theoretical and computational research into the applications of inverse scattering theory for the analysis of signals and images. In 1988, he joined the Applied Mathematics and Computing Group at Cranfield University as Lecturer and later, as Senior Lecturer and Head of Group where he promoted postgraduate teaching and research in applied and engineering mathematics in areas which included computer aided engineering, digital signal processing and computer graphics. While at Cranfield, he co-founded Management and Personnel Services Limited through the Cranfield Business School which was originally established for the promotion of management consultancy working in partnership with the Chamber of Commerce. He managed the growth of the company from 1993 to 2007 to include the delivery of a range of National Vocational Qualifications, primarily through the City and Guilds London Institute, including engineering, ICT, business administration and management. In 1994, Jonathan Blackledge was appointed Professor of Applied Mathematics and Head of the Department of Mathematical Sciences at De Montfort University where he expanded the post-graduate and research portfolio of the Department and established the Institute of Simulation Sciences. In 2002 he was appointed Visiting Professor of Information and Communications Technology in the Advanced Signal Processing Research Group, Department of Electronics and Electrical Engineering at Loughborough University, England (a group which he co-founded in 2002 as part of his appointment). In 2004 he was appointed Professor Extraordinaire of Computer Science in the Department of Computer Science at the University of the Western Cape, South Africa. His principal roles at these institutes include the supervision of MSc and MPhil/PhD students and the delivery of specialist short courses for their Continuous Professional Development programmes. He currently holds the prestigious Stokes Professorship in Digital Signal Processing under the Science Foundation Ireland Programme based in the School of Electrical Engineering Systems at Dublin Institute of Technology, Ireland.

Dmitry Dubovitskiy received a BSc and Diploma in Aeronautical Engineering from Saratov Aviation Technical College in 1993, an MSc in Computer Science and Information Technology for the Moscow State Technical University 1999 and a PhD in Computer Science in 2005 under the supervision of Professor Blackledge. As a project leader in medical imaging at Microsharp Limited from 2002 to 2005, he specialized in information systems engineering, developing image recognition systems for medical applications for real time operational diagnosis. He is Executive Director and Principal Engineer at Oxford Recognition Limited which he founded in 2005 to undertaken research and development, specializing in the application of artificial intelligence for computer vision. He has developed a range of computer vision systems for industry including applications for 3D image visualisation, face recognition, information security and digital rights management and is currently principal coordinator for the INTAS project in distributed automated systems for acquiring and analysing eye tracking data.