5th Int. Conference Structural Integrity of Welded Structures (ISCS2007), Timisora, Romania, 20-21 Nov 2007
- Testing & risk assessment in the development of advanced materials and joints -
For all papers of this publication click: www.ndt.net/search/docs.php3?MainSource=56

Microstructural imaging algorithms – Design and implementation

Sandu Crâsteți
National R&D Institute for Welding and Material Testing, Timişoara, Romania
Phone: 0040256 491828, Fax: 0040256 492797; e-mail: sandu@isim.ro

Abstract
Based on prior results, obtained in the fields of microstructural imaging, the paper focuses on aspects regarding classic and modern paradigms.
The main emphasis is given to define the objectives and goals that might characterize microstructural imaging applications, in order to decide upon the feasible approaches used to obtain reliable evaluation results. Based on a step-by-step process of defining and deciding upon the algorithms to be designed and implemented, the potential offered by the soft computing paradigms is analysed.
The results consist of a set of digital image processing algorithms, designed and implemented on the basis of both the potential of the general purpose classic imaging algorithms and that of the soft computing approach, as well as an analysis regarding the experimental results obtained by using this set of algorithms in cases of practical microstructural evaluation.
The paper also discusses some significant results in the field, presented in the literature.
Keywords: microstructure, imaging, digital image processing, algorithm, soft computing, fuzzy logic

1. Introduction

Some of the most useful image processing operations in microstructural applications are represented by pre-processing (enhancement) operations, represented by the adjustment of brightness, contrast or grey levels in an image [2]. One of the main reasons for approaching these processing methods is represented by the need to compensate the difficulties in image acquisition (underexposure, lack of contrast, narrow histograms). Using image pre-processing one can increase the overall brightness of the details of interest in an digital image acquired from an optical microscope, and magnify the tiny residual variations in contrast, revealing the necessary details to allow further processing aiming both qualitative and quantitative evaluations of the indications of interest.

Further steps of the image processing methodology are represented by segmentation techniques [1], [2], [5], developed as the first stages in the attempt to analyse and evaluate the digital images. by segmentation, an image is partitioned into distinct regions, meant to strongly correlate with features of interest. It can also be regarded as the process of grouping together meaningful image regions, containing pixels having similar attributes.

Segmentation might be considered the link between low-level processing, represented by the manipulation of pixel grey levels, to enhance certain characteristics of interest, and high-level processing, involving manipulation and analysis of groups of pixels that represent features of interest. Such techniques are part in any complex application destined to solve problems involving object (image indication) detection, recognition and quantitative evaluation.

In the case of both image processing techniques it is possible - in some applications it becomes necessary, and in most applications it proves useful - to develop and implement soft computing approaches, especially fuzzy ones [1], [3], [4], due to the fact that images, generated as sets of pixels characterised by grey levels, are prone to an imprecise characterization regarding the separation of meaningful indications, as well as to incertitude regarding the best processing parameters.

2. Image enhancement algorithms

Grey level (and colour) mapping techniques are widely used as digital image pre-processing and enhancement methods. By using *linear mapping* [2] one can adjust the overall brightness of a greyscale image simply by adding a constant bias b to pixel grey-level values:

$$g(x,y) = f(x,y) + b \tag{1}$$

If b >0, overall brightness is increased; if b < 0, it is decreased. In a similar way one can adjust contrast in a greyscale image through multiplication of pixel values by a constant gain, a:

$$g(x,y) = af(x,y) \tag{2}$$

If a >1, contrast is increased whereas if a<1 it is reduced.

Equations (1) and (2) can be combined to give a general expression for brightness and contrast modification:

$$g(x,y) = af(x,y) + b \tag{3}$$

Often one does not specify a gain and a bias, but would map a particular range of grey levels, [f_1, f_2], onto a new range, [g_1, g_2], using the following equation:

$$g(x,y) = g_1 + \left(\frac{g_2 - g_1}{f_2 - f_1}\right)\left[f(x,y) - f_1\right] \tag{4}$$

Equations (3) and (4) are equivalent to a linear mapping pf pixel grey levels. When dealing with 8-bit digital images, the mapping generates values in the range 0 – 255.

In similar ways are defined other mapping equations: the logarithmic mapping, the exponential mapping, the non-linear mapping.

A set of pre-processing and enhancing algorithms, designed by using these equations, were implemented using the Java2D API (Application Programming Interface), following the suggestions encountered in literature. The resulting images were further applied segmentation and quantitative evaluation modules, also implemented on the basis of Java^{TM} technology, [2], [6], [7], [8].

3. Segmentation algorithms

Segmentation techniques can be classified as either contextual or non-contextual [2]. Non-contextual techniques ignore the relationship that exist between features in an image, pixels being just grouped together on the basis of some global attributes, such as grey level. Contextual techniques additionally make use of the relationships that exist between image features. Such a technique might group together pixels having similar grey levels and are close to one another.

One of the non-contextual techniques applied for the segmentation modules, which were designed and implemented (using Java^{TM} technology), was the *thresholding*. This technique transforms a dataset containing values that vary over some range into a new dataset containing only two values, by applying a threshold to the to the input data. Input values that fall bellow the threshold are replaced by one of the output values; input values at or above the threshold are replaced by the second output value.

The most common form of image thresholding, for which an algorithm was designed and implemented, applies to every pixel the following rule:

$$g(x,y) = \begin{cases} 0, & f(x,y) < T \\ 1, & f(x,y) \geq T \end{cases} \tag{5}$$

A variation of this technique is represented by thresholding using two threshold values to define a range of useful grey levels:

$$g(x,y) = \begin{cases} 0, & f(x,y) < T_1 \\ 1, & T_1 \leq f(x,y) \leq T_2 \\ 0, & f(x,y) > T_2 \end{cases} \tag{6}$$

The success of thresholding depends critically on the selection of an appropriate threshold value. A general approach to threshold selection involves histogram analysis for the image of interest. This is based on the assumption, true only in certain situations, that different features in an image give rise to distinct peaks in its histogram. If this assumption is true, one might distinguish between two features of different grey levels by thresholding at a point between the histogram peaks corresponding to those two features.

The approaches to contextual segmentation are based on the concept of discontinuity or the concept of similarity. The techniques based on discontinuity attempt to partition the image by detecting abrupt changes in grey level (like edge detection methods).

Techniques based on similarity attempt to create uniform regions directly, by grouping together connected pixels that satisfy predefined similarity criteria. The results of segmentation may depend critically on these criteria and on the definition adopted for connectivity.

Both approaches, based on discontinuity and similarity, mirror one another, in the sense that boundary completion is equivalent to breaking one region into two.

One of the contextual methods for which a segmentation algorithm was designed and implemented is the region growing procedure. This is a complex bottom-up procedure that starts with a set of seed pixels, aiming to grow a uniform, connected region from each seed. A pixel is to be added to a region iff:

- it has not been assigned to any other region
- it has a neighbour of that region
- the new region created by addition of the pixel is still uniform

A complete segmentation of an image must satisfy the following criteria:

i. all pixels must be assigned to regions
ii. each pixel must belong to a single region only
iii. each region must be a connected set of pixels
iv. each region must be uniform
v. any merged pair of adjacent regions must be non-uniform

Region growing is not a stable operation. It satisfies the third and fourth criteria, but not the others.

The segmentation results, obtained by applying the modules obtained by implementing thresholding algorithms, are presented in section 5, in comparison with the results obtained by applying fuzzy concepts, presented in the next section.

4. Fuzzy sets concepts applied for image processing

In [1] there are presented considerations regarding contrast enhancement using paradigms that are specific for the theory of fuzzy sets. As an example, the process of enhancing the contrast of a grey scale image uses the following rules:

IF a pixel is dark, THEN make it *darker*
IF a pixel is grey, THEN make it *grey*
IF a pixel is bright, THEN make it *brighter*

Taking into account the fact that these are fuzzy terms, the concepts of *dark*, *grey* and *bright* can be expressed by *membership functions*.

Because the entire (and complex) process of fuzzification, processing the antecedents of all rules, implication, aggregation, and defuzzification must be applied to every pixel in the input image, fuzzy image processing represents a computationally intensive approach.

Practical applications and algorithmic considerations, based on fuzzy sets theory, are also presented in [3], and were used for obtaining comparative segmentation data.

In an attempt to use fuzzy sets in image segmentation, using a measure of fuzziness (a distance between the original grey level image and the thresholded image, the first step is to determine the membership function, representing the (subjective) degree by which a pixel is classified to belong to an object (foreground) or to the background. Consider that the average grey level of the background is μ_0 and that of the objects is μ_1. The smaller the difference between the level of any pixel x and the appropriate mean for its class, the greater will be the value of the membership function $\mu_x(x)$. A good membership function is considered to be the following:

$$u_x(x) = \begin{cases} \dfrac{1}{1+|x-\mu_0|/C} & if \quad x \le p \\ \dfrac{1}{1+|x-\mu_i|/C} & if \quad x > p \end{cases} \qquad (7)$$

for a given threshold t. C is a constant, representing the difference between the maximum and minimum grey levels. Any pixel x belongs either to the background set or the to the object set depending on the relationship between the grey level of the pixel and the threshold t. For an object pixel (x>t), the degree to which it belongs to the object set is given by $u_x(x)$, which should be a value between ½ and 1.

Given the membership function, the degree of fuzziness of the segmentation, measured for a given t, is based on the entropy of a fuzzy set, determined by using Shannon's function:

$$H_f(x) = -x \log(x) - (1 - x)\log(1 - x) \tag{8}$$

and so the entropy of the entire fuzzy set (the image) would be:

$$E(p) = \frac{1}{MN} \sum_g H_f(\mu(g))h(g) \tag{9}$$

for all grey levels g, where N and M are the number of rows and columns, and h is the grey level histogram. This is a function of t because u_x is. Where E(t) is a minimum, t is the appropriate threshold that minimizes fuzziness.

For a given threshold t, we have [3]:

$$\mu_0(t) = \frac{\sum_{g=0}^{t} g \cdot h(g)}{\sum_{g=0}^{t} h(g)} \tag{10}$$

as the estimate of the background mean, and:

$$\mu_0(t) = \frac{\sum_{g=t+1}^{254} g \cdot h(g)}{\sum_{g=t+1}^{254} h(g)} \tag{11}$$

as the estimate of the object mean, where both values depend on the threshold.
To minimize the fuzziness, one has to try all possible thresholds t and select the one that that yields the minimum value of the fuzziness measure.

5. Results

The following figures represent the experimental results obtained for two types of thermo resistant welded steels, by applying the following image processing algorithms, designed on the basis of the theoretic considerations presented prior in the paper, and implemented as software modules using Java™ technology:

- image enhancing (pre-processing) based on logarithmic grey level mapping
- image segmentation based on thresholding using one threshold value
- image segmentation based on thresholding using two threshold value
- image segmentation based on fuzziness minimization

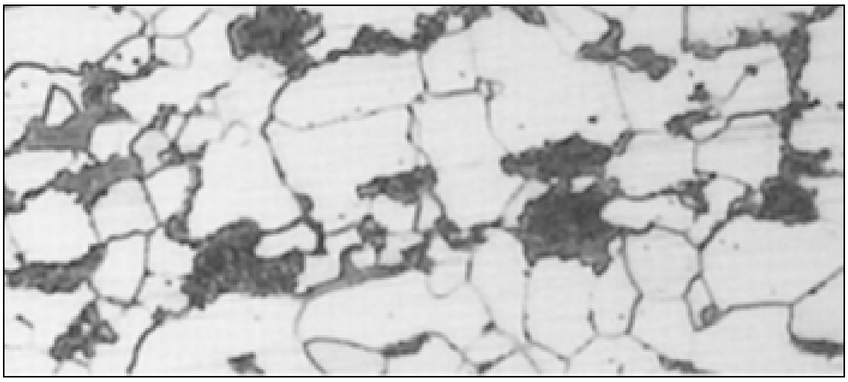

Figure 1. OLT 45 steel: base material, magnification: 100x, etching agent: Nital 2%; pre-processed

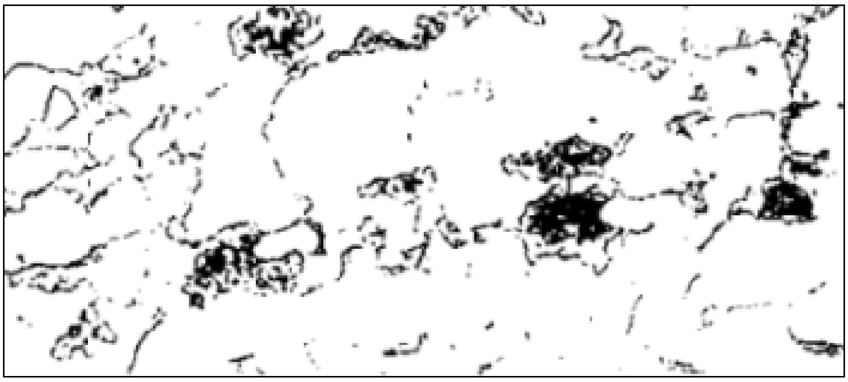

Figure 2. The segmented image in Figure 1, using one threshold value

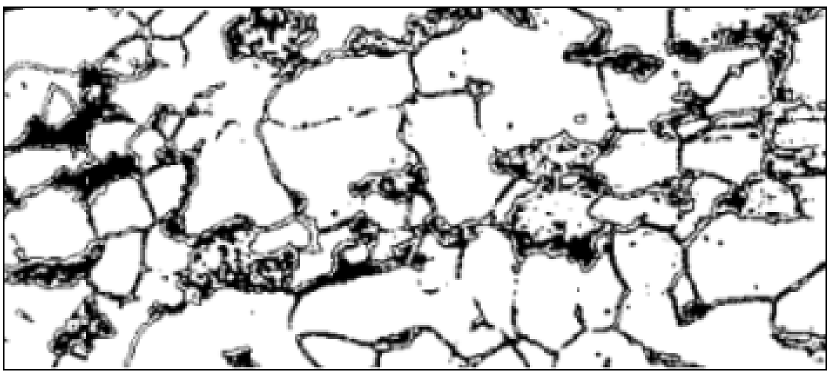

Figure 3. The segmented image in Figure 1, using two threshold values

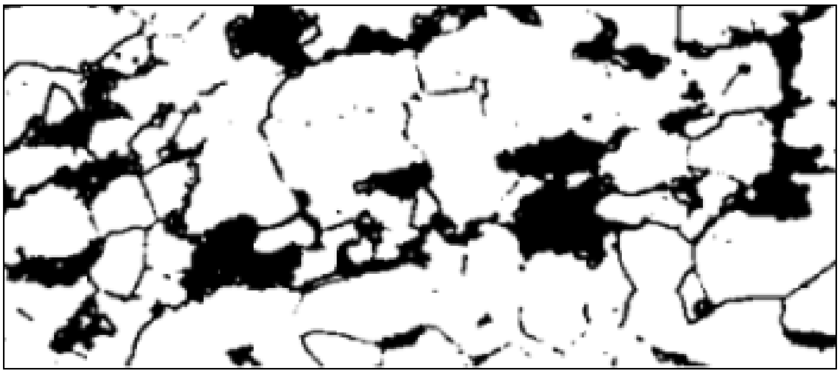

Figure 4. The segmented image in Figure 1, using fuzziness minimization (t = 179)

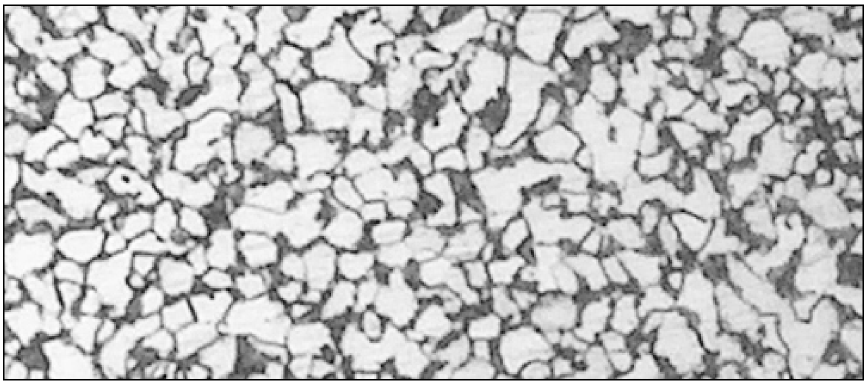

Figure 5. OLT 45 steel: HAZ, magnification: 100x, etching agent: Nital 2%; pre-processed

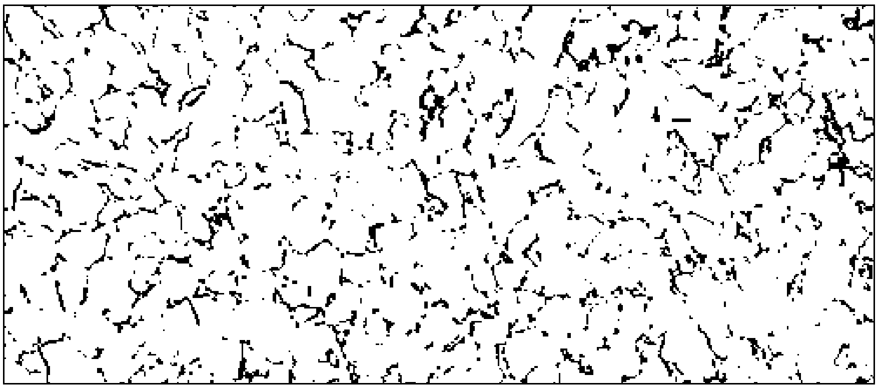

Figure 6. The segmented image in Figure 5, using one threshold value

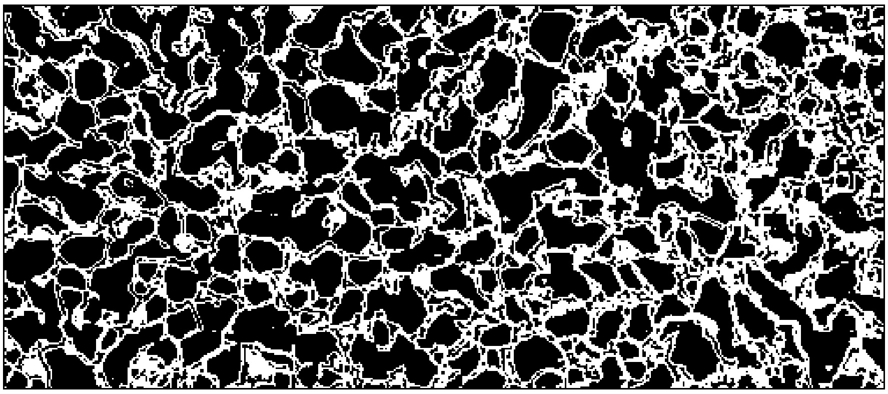

Figure 7. The segmented image in Figure 5, using two threshold values

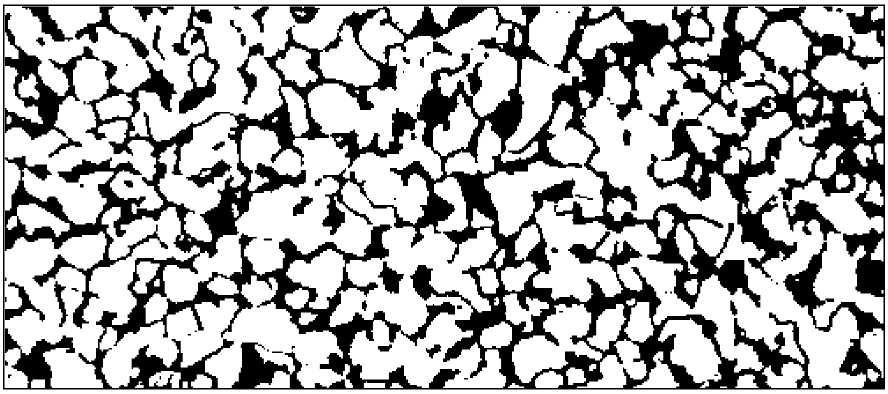

Figure 8. The segmented image in Figure 5, using fuzziness minimization (t = 180)

6 Conclusions

6.1 The results obtained are relevant for the possibilities offered by the digital image processing techniques, in the case of microstructure analysis;

6.2 The results of the tests carried out on specimens cut from thermo resistant welded steels, analysed using an optical microscope, offered the possibility (in further stages of the experimental programme) for quantitative evaluations and measurements (for a magnification of 100x – constituents' percentage evaluation, and for the magnification of 500x – measurements of grains' dimensions);

6.3 The use of Java^{TM} technology (Java2D API), including some of the facilities offered by the ImageJ package [8], made it possible to use supplementary methods, like computer graphics, to apply a large range of processing tools (for selecting regions of interest, for example);

6.4 The experimental results obtained so far by implementing both the classic digital image processing paradigms, and the newer fuzzy paradigms, offered meaningful reasons for further developments, with a more pronounced emphasis regarding the fuzzy, and soft computing in general (including neural networks), approaches.

References

1. R.C. Gonzalez and R.E. Woods, 'Digital Image Processing', 3rd Edition, Pearson Prentice Hall, 2007
2. N. Efford, 'Digital Image Processing, a practical introduction using Java', Pearson Education Limited, 2000
3. J.R. Parker, 'Algorithms for Image Processing and Computer Vision', John Wiley & Sons, Inc., 1997
4. S.E. Umbaugh, 'Computer Imaging: Digital Image Analysis and Processing', Prentice Hall, 2005
5. M. Seul, L. O'Gorman and M.J. Sammon, 'Practical Algorithms for Image Analysis', Cambridge University Press, 2000
6. D.A. Lyon, 'Image Processing in Java', Prentice Hall. 1999
7. L.H. Rodrigues, 'Building Imaging Applications with Java™ Technology: Using AWT Imaging, Java 2D™, and Java™Advanced Imaging (JAI)', Prentice hall, 2001
8. http://rsb.info.nih.gov/ij/