A Texture Classification System with Automatic Feature Vector Optimization using Genetic Algorithms

Alexander Stolpmann*[†], Laurence S. Dooley[‡](formerly[†])
* Fachhochschule Braunschweig/Wolfenbüttel, Fachbereich Informatik
Salzdahlumer Straße 46/48, D-38302 Wolfenbüttel, Germany
Phone +49 5331 939 656, Fax +49 5331 939 602
e-mail stolpmann@fh-wolfenbuettel.de
† University of Glamorgan, School of Electronics
Pontypridd, Mid Glamorgan, Wales, United Kingdom
‡ Monash University, Faculty of Information Technology
Gippsland Campus, Churchill, Victoria 3842, Australia
Phone +61 3 9902 6200, Fax +61 3 9902 6300
e-mail laurence.dooley@infotech.monash.edu.au

ABSTRACT This paper describes the use of genetic algorithms in a complex modular image processing system for texture classification. This is part of a system developed within a research project concerning the classification of genuine texture. An attempt is made to underline why an automatic feature vector optimization module is a useful part of the texture classification system. Furthermore the way of including the genetic algorithms into the system and the necessary feedback structure is explained.

KEYWORDS Genetic Algorithms, Pattern Recognition and Image Processing, Texture

INTRODUCTION

The real world does not supply the laboratory conditions that are necessary for the majority of the existing image processing systems. Therefore all those ill structured or coarse grained objects and especially objects without clear boundaries are a major problem in the field of object recognition. In some areas it is even problematical to speak of objects in the classical sense. One only has to think of a cornfield, a lawn, ripples on water, clouds in the sky or leaves on a tree to comprehend the problems.

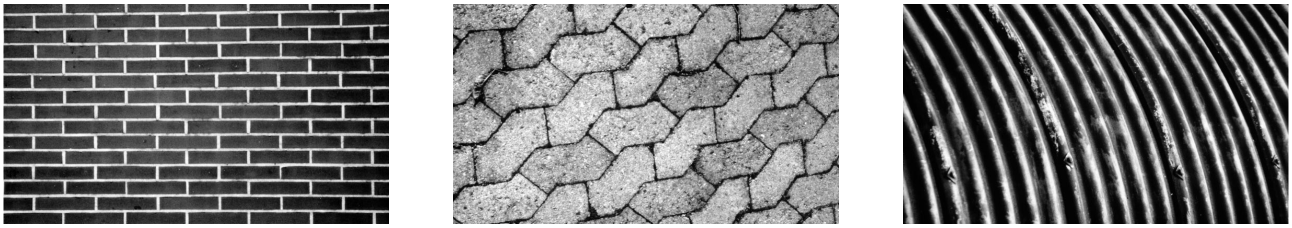
The aim of the research work of which this paper describes a part is to find a solution to the problem of classifying genuine texture, as texture represents the unstructuredness very well. Some alternative and completely new methods are being developed and tested. Furthermore existing approaches for sub-problems are included in the system. This system shall not be adapted for a special problem but shall be universal. Therefore it can be used for multiple tasks in many fields, e. g. medicine, remote sensing, machine vision, etc.

This paper describes the part that brings self optimization into the system. Genetic algorithms have been chosen to effectively minimize the set of features statistically extracted in prior modules of the system. Such minimization is necessary to reduce the dimension of the following classification modules.

A SHORT DESCRIPTION OF TEXTURE

The word *texture*, which descends from the Latin word *texere* ("to weave"), is not uniquely defined, but rather described context dependently. The WEBSTER [18] definition closest to the use in image analysis is *similar qualities dependent on the nature and arrangement of the constituent particles of a substance.*

In general it can be said that a texture describes the surface composition of an object. Texture can be divided into regular texture, composed of repeated texture primitives which are large against the pixel resolution and could be described in further detail (cf. fig. 1(a), 1(b), 1(c)), and statistical texture with texture primitives that near unity and have a random distribution (cf. fig. 1(d), 1(e), 1(f), 1(g), 1(h), 1(i)). Very often texture is of a hierarchical kind where the macro texture is regular and the micro texture of statistically describable primitives (cf. fig. 1(j), 1(k), 1(l)).



(a) Brick wall with white mortar.

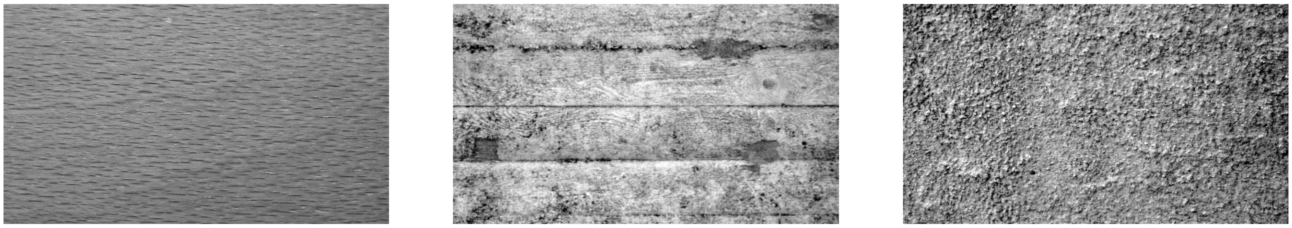
(b) Interlocking paving-stones.

(c) Curved corrugated iron sheets.

(d) Bark of an old birch tree.

(e) Blades of grass covered with frozen dew.

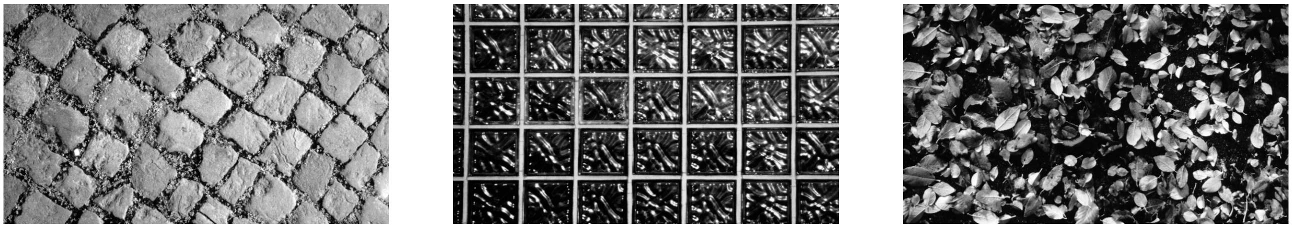
(f) Field of clouds against blue sky.



(g) Veneer of a beech tree.

(h) Wall of concrete with wooden plank structure.

(i) A roughcasted wall.



(j) Cobblestones with unstructured edges.

(k) Glass bricks with sunlight reflections.

(l) Autumn leaves lying on the ground.

Figure 1: Pictures of natural, half-natural and man-made texture.

For testing the system only genuine texture is being used, which comprises all those textures that are not artificially constructed, i.e. made on and for a computer. The group can be subdivided into natural, half-natural and man-made texture, a classification introduced by the author in [14].

THE SYSTEM

Figure 2 shows a simplified layout of the Texture Classification System. It is composed of modules and layers. The centerpiece is the Core-System which consists of three modules: the preprocessing methods, the statistical methods and the soft-computing methods. The Core-System classifies the input images and is itself the center module of the Enhanced-System with the image preparation module before and the postprocessing module after it. The Enhanced-System is capable of identifying single texture regions within a larger image. The Optimization System is made up of the genetic algorithm module and the fitness evaluation module. This subsystem can be used for optimization tasks throughout the complete system. The paper only describes the use for the feature vector size optimization.

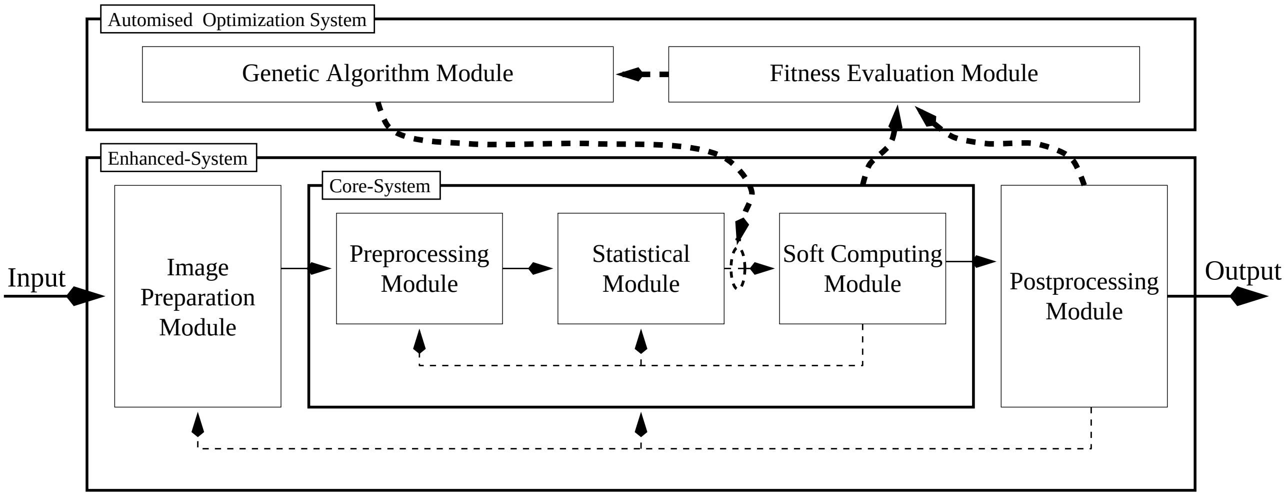


Figure 2: The Texture Classification System.

STATISTICAL FEATURE EXTRACTION

Using statistical methods for feature extraction is one way of classifying texture. Those methods are used in the texture classification system described in this paper.

Preprocessing methods are used to reduce redundant information contained in the texture images. Thus the relevant information can be extracted more easily by using statistical methods.

Apart from using standard filter methods the images are processed through wavelets filters [5] and LAWS-measures [8]. The results of these computational steps are not used in the classical sense—it is for example not of interest where edges can be found—but are statistically evaluated.

As the neighbourhood relations of image pixels are of importance high order statistics are used additionally to first order statistical calculations—like mean, variance, skewness, kurtosis. Very good results have been obtained concerning the orientation of similar grey-level pixels within an image. The spatial grey-level dependence (SGLD) matrices used yield potential features, among which are the entropy, correlation, inertia and homogenity [3].

The combination of all these filters and statistical calculations result in a vast amount of data. But only a fraction of the extracted features provide non-redundant information that is unique to a specific texture. Therefore the number of features has to be reduced to avoid wasting computational resources.

FEATURE SELECTION USING GENETIC ALGORITHMS

As the number of features generated in the previous modules of the system is very large—easily exceeding 10^3 or even 10^4—it is necessary to select relevant features for the classification which takes place in the following modules. Doing this manually is not an option as the dimension of the feature plane is by far too large to be visualizable and the possible interconnections between features too complex. Therefore an automatic feature selector has to be included into the system.

Genetic algorithms have the capability of finding very good local or even global optimal solutions in complex data-planes [11], [15]. Therefore every feature is associated to one *gene*—a boolean element—and all genes compose the equivalent of a *DNA*. If the gene is set to zero the associated feature is not used in the following modules of the system and it is used if the gene is set to one. One half of the starting population is created randomly, the other half consists of the negated first half. The negation provides an even distribution of zeros and ones for every gene throughout the starting population. In the next step all or some members of the population are used to create a new generation by exchanging parts of the DNA-string. This is called *crossover*. Depending on the way of selection and production of DNAs the population can grow rapidly. Additionally some DNAs can be mutated to avoid getting stuck in a local optimum.

SINGLE-CUT AND TWO-CUT CROSSOVER

Figure 3 shows graphically how the single-cut crossover works. The spot where the DNA-string is cut is chosen by random. Actually the single-cut crossover is a special two-cut crossover with the second cut always at the end of the DNA-string. The general version of the two-cut crossover is shown in figure 4.

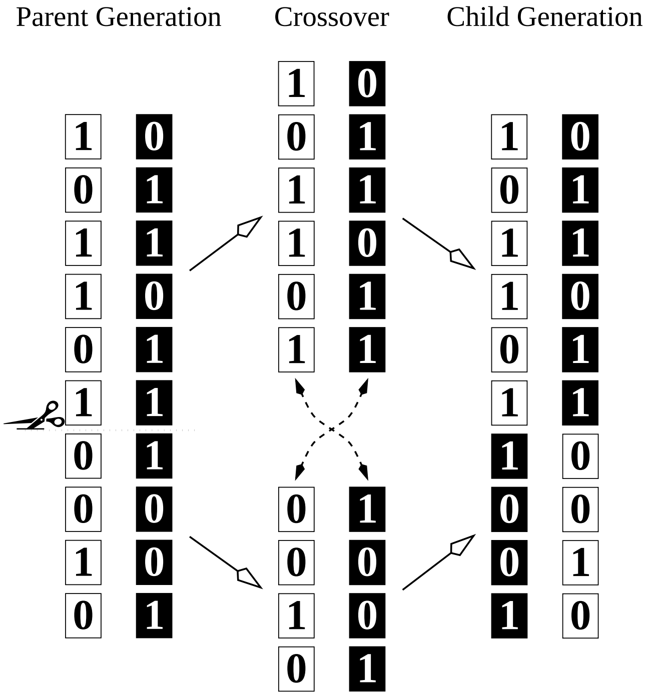


Figure 3: Crossover with single cut.

The two-cut crossover promotes the idea of a ring-DNA—cf. fig. 5—which has no first or last element. The position of all genes are equal.

SHUFFLE CROSSOVER

The problem with the formerly mentioned crossover methods is that neighbouring genes will almost always stay together, especially if the DNA-string is very long as in the case of the application described in this paper.

A way of avoiding this is by using the shuffle crossover method. A special shuffle-DNA-string of the same size as the other DNA-strings is created randomly for every new generation. The genes of the parent DNA-strings are exchanged if the according gene of the shuffle-DNA-string is set to one, not exchanged if set to zero. Figure 6 will clarify this procedure.

MUTATION

Mutation is needed to avoid finding only local optimal solution. This can happen if throughout the population a certain gene is set to one, or zero accordingly, like the third, sixth or eighth gene in the figures 3 to 6. By means of crossover the influence of this gene will never again change, thus only half of the possible solutions can be addressed. Mutation now toggles the influence of a randomly chosen gene from one to zero and vice versa.

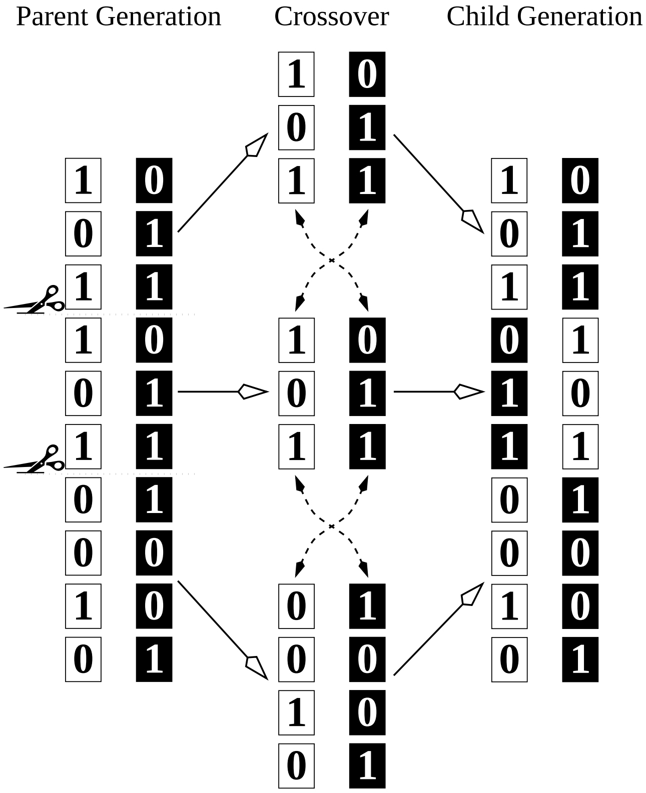


Figure 4: Crossover with two cuts.

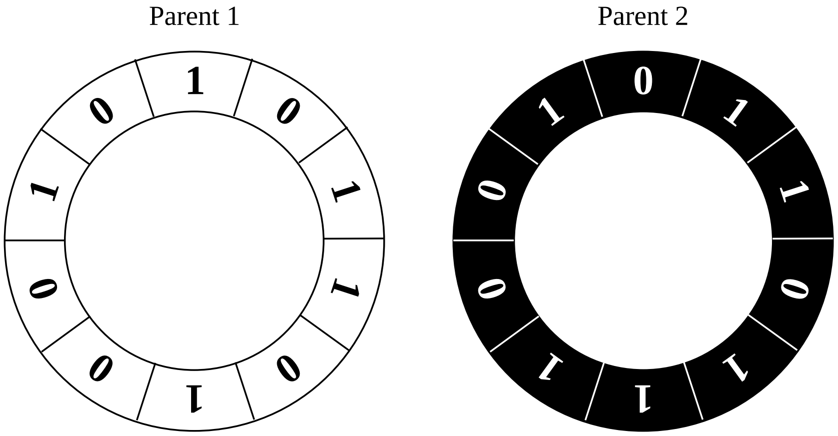


Figure 5: DNA ring.

CHOOSING THE PARENTS

A number of different mating procedures can be adopted. The *brute force* method is to mate every individual of the population to all the others. A parent population size of m will produce $m^2 - m$ children, all of which have to be rated for their individual fitness. A way of producing less children—and thus fewer fitness tests—is mating by rank. Possibilities that produce m children are

$$\{[1 \infty 2], [3 \infty 4], [5 \infty 6], \cdots, [m-1 \infty m]\}$$

and

$$\{[1 \infty m], [2 \infty m-1], [3 \infty m-2], \cdots, [m/2 \infty m/2-1]\} \quad .$$

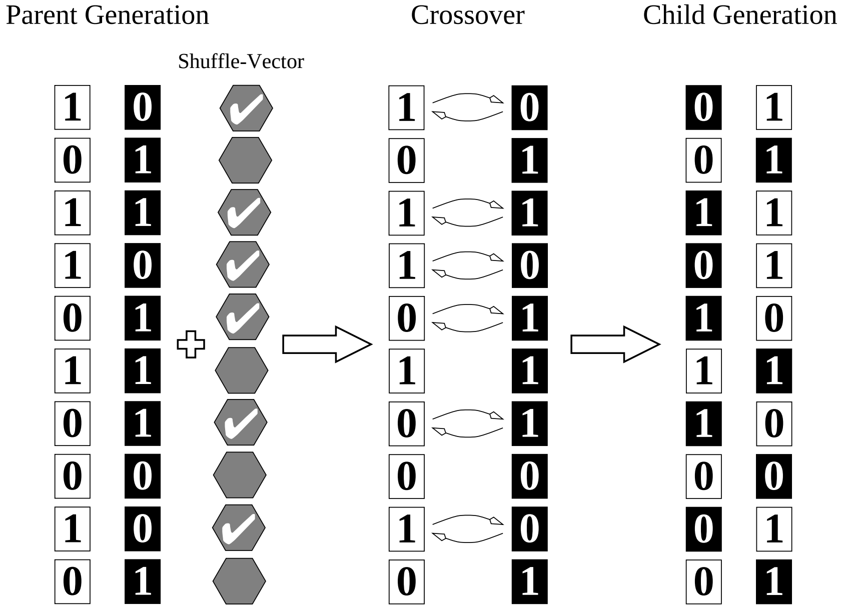


Figure 6: Crossover method with shuffle-DNA-string.

A variant common in the animal kingdom is to mate the fittest to all others, producing $2(m-1)$ children:

$$\{[1 \mathbin{⚭} 2], [1 \mathbin{⚭} 3], [1 \mathbin{⚭} 4], \cdots, [1 \mathbin{⚭} m]\} \quad .$$

A further variant would be to use the two, three or p fittest individuals for mating with all others:

$$\{[i \mathbin{⚭} j]\} \quad \forall \quad i \in \{1, p\}, j \in \{i, m\}, p < m \quad .$$

Here the census will count $p(2m-p-1)$ children. For $p = m-1$ this method is identical to the *brute force* method mentioned above.

FITNESS EVALUATION

Next to the crossover process itself the fitness evaluation is most important. Fitness evaluation is the performance test of the system using every DNA of the population and thus a number of sets of selected features. After those tests only the better DNA-strings stay in the population and the production of a new population starts again. To keep the population from growing indefinitely only the m fittest individuals are kept. At that point there is no distinction between parents and children.

This process continues as long as the fitness differences between parent and child population are significant for a specified number of generations.

In the case of this texture classification system the fitness describes the ability to distinguish between different textures. Figure 7 gives an idea how the features that describe a certain texture build a cluster in the feature plane. The aim is to find such a selection of features for which the cluster do not intersect with one another. The better the clusters are kept apart, the easier the classification is.

CLASSIFICATION

As the aim is to find the relevant features to classify the textures a system had to be developed that handles the multi-dimensional feature vectors. Additionally it had to provide the possibility to judge the relevancy of the features.

FUZZY CLUSTERING

One possibility of classifying the feature vectors is using a clustering system [6], [9], [17]. For each (relevant) feature a new dimension is created. This results in an n-dimensional plane. All vectors that belong to a specific texture make up a set, or cluster, which is diverse to all other texture clusters in this plane. With increasing numbers of dissimilar textures the chance of clusters overlapping in one or more dimension increases. Therefore new methods had to be adopted to overcome this problem.

Fuzzy logic methods have a potential for handling uncertain knowledge [13], [19], [21]. Thus it is possible to classify texture of which the feature vectors do not point to the center of a certain cluster but into an overlapping area of two or more clusters [1], [2], [4], [16]. Such a texture has a membership value of a certain height for every cluster of the plane. Usually this value equals zero for almost all clusters and obtains a high value for the fault cluster in question. Texture images that have membership values of about equal height have to be treated in a postprocessing system.

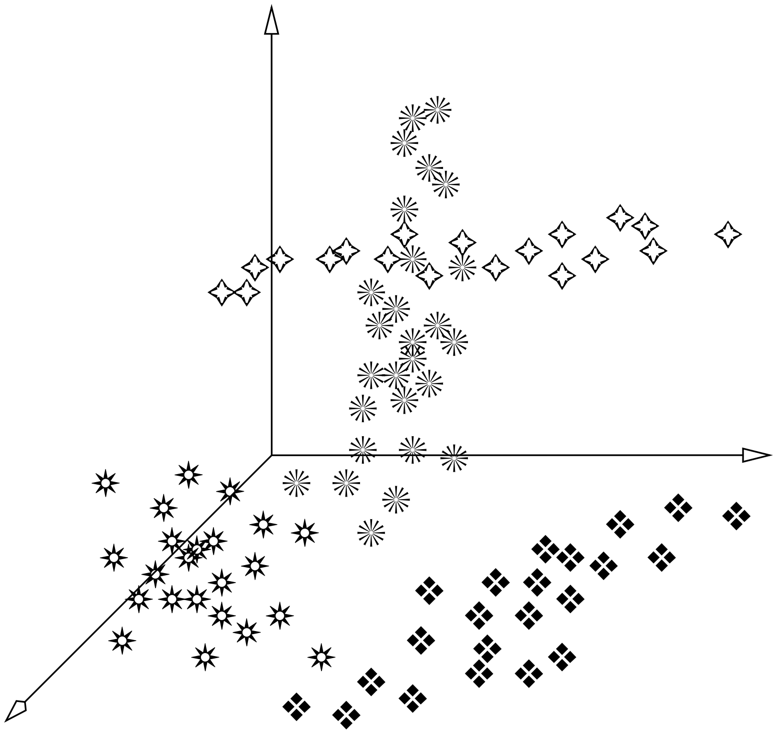


Figure 7: Cluster in a 3D-feature-plane.

NEURAL NETWORKS

An alternative to fuzzy clustering is to use neural networks. Widely known backpropagation (BP) networks have been successfully tested for this task [7], [10]. One problem with BP networks is that they are not retrainable, i.e. new classes can not be added without destroying the trained classes. This makes it more difficult and time consuming to expand the number of classes.

A neural network type which overcomes these problems is the adaptive resonance theory (ART) family [20]. Networks of this type can be retrained at any time. These networks are currently included into the system.

A general problem with neural networks is the rather long training time and the great need for computational resources, especially when input (number of features) and output (number of classes) vectors are large in size. In such cases the use of special neural network hardware is almost unavoidable, especially if the network has to be trained again and again to find the optimal feature vector as in the case of this texture classification system.

CONCLUSIONS

This paper has introduced the use of a complex modular image processing system for texture classification. The necessity and advantages of this approach have been delineated and discussed. Problems associated with the actual textures themselves and other modules of the system have also been addressed.

REFERENCES

[1] J C Bezdek. *Fuzzy mathematics in pattern classification.* PhD Thesis, Cornell University, 1973.

[2] J C Dunn. *A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters.* Journal of Cybernetics, 3/3:32–57, 1973.

[3] I M Elfadel, R W Picard. *Gibbs Random Fields, Co-occurrences, and Texture Modeling.* Technical Report # 204, MIT Media Laboratory Perceptual Computing Group, Januar 1993.

[4] I Gath, A B Geva. *Unsupervised Optimal Fuzzy Clustering.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 11/7:773–781, 1989.

[5] A Graps. *An Introduction to Wavelets.* IEEE Computational Science and Engineering, 2(2), 1995.

[6] L Kaufmann, P J Rousseeuw. *Finding Groups in Data: An Introduction to Cluster Analysis.* John Wiley and Sons, New York, 1990.

[7] C Klevenhusen. *Texture Analysis with Artificial Neural Networks.* Diplomarbeit, Fachhochschule Braunschweig/Wolfenbüttel / University of Glamorgan, 1996.

[8] K I Laws. *Rapid texture identification.* SPIE Image Processing for Missile Guidance, 238:376–380, 1980.

[9] J MacQueen. *Some methods for classification and analysis of multivariate observations.* Proceedings of the 5th Berkeley Symposium, 1:281–297, 1967.

[10] S Malon. *Entwicklung eines Systems zur Texturanalyse mittels Wavelet-Transformation und neuronalem Netz für das Bildverarbeitungssystem WiT.* Diplomarbeit, Fachhochschule Braunschweig/Wolfenbüttel, 1997.

[11] E Schöneburg, F Heinzmann, S Feddersen. *Genetische Algorithmen und Evolutionsstrategien.* Addison-Wesley, Bonn/Paris/Reading (Mass.), 1994.

[12] M Sonka, V Hlavac, R Boyle. *Image Processing, Analysis and Machine Vision.* Chapman & Hall, London, 1993.

[13] A Stolpmann. *Fuzzy Logik: Eine Einführung ins Unscharfe.* Diplomarbeit, Fachhochschule Braunschweig/Wolfenbüttel, 1993.

[14] A Stolpmann. *Automatic Recognition of Coarse-Grained Non-Regular Structures using Non-Parametric Processing and Soft-Computing Methods – A Texture Classification System.* MPhil/PhD Transfer Document, University of Glamorgan, 1997.

[15] A Stolpmann, L S Dooley. *Genetic Algorithms for Automised Feature Selection in a Texture Classification System.* Proceedings of the 4th International Conference on Signal Processing, Beijing, 1998.

[16] T Tilli. *Mustererkennung mit Fuzzy-Logik: Analysieren, klassifizieren, erkennen und diagnostizieren.* Franzis-Verlag, München, 1993.

[17] R C Tryon. *Cluster Analysis.* Edwards Bros., Ann Arbor, 1939.

[18] *Webster's New Encyclopedic Dictionary.* Black Dog & Leventhal Publishers, New York, 1993.

[19] L A Zadeh. *Fuzzy Sets.* Information Control, 8:338–353, 1965.

[20] A Zell. *Simulation Neuronaler Netze.* Addison-Wesley, Bonn/Paris/Reading (Mass.), 1994.

[21] H-J Zimmermann. *Fuzzy Set Theory – And Its Applications.* Kluwer Academic Publishers, Boston/Dordrecht/London, 1991.