A GENETIC APPROACH FOR THE AUTOMATIC ADAPTATION OF SEGMENTATION PARAMETERS

R. Q. Feitosa [a, c] *, G. A. O. P. Costa [a], T. B. Cazes [a], B. Feijo [b]

[a] Dept. of Electrical Engineering, [b] Dept of Informatics, Catholic University of Rio de Janeiro (PUC-Rio), Rua Marquês de São Vicente 225, Gávea, CEP 22453-900, Rio de Janeiro, RJ, Brasil
[c] Dept. of Computer Engineering, Rio de Janeiro State University (UERJ), Rua São Francisco Xavier 524, Maracanã, CEP 20550-900, Rio de Janeiro, RJ, Brasil
(raul, gilson, tcazes)@ele.puc-rio.br; bruno@inf.puc-rio.br

KEY WORDS: Segmentation, Optimization, Genetic Algorithm, Parameter Adaptation, Object-Oriented Classification.

ABSTRACT:

The key step in object-oriented image classification is the segmentation of the image into discrete meaningful objects. Generally the relation between the segmentation parameters and the corresponding segmentation outcome is far from being obvious, and the definition of suitable parameter values is usually done through a troublesome and time consuming trial and error process. This paper proposes a method for the automatic adaptation of segmentation parameters based on Genetic Algorithms. The intuitive and computationally uncomplicated fitness function proposed expresses the similarity of the segmentation result with a reference provided by the user. The method searches the solution space for a set of parameter values that minimizes this fitness function. A prototype including an implementation of a widely used segmentation algorithm was developed to assess performance of the method. A set of experiments on two pairs of LANDSAT and IKONOS images was carried out and the method was able in most cases to come close to the ideal solution.

1. INTRODUCTION

The remote sensing technology has experienced an amazing development over the last decades. Particularly with the advent of high spatial resolution space-borne sensors for commercial purposes, the amount of available data about the earth surface increased enormously. Moreover the high spatial resolution images exposed the limitations of the traditional pixel-wise classification techniques (Blaschke, 2001). This scenario boosted the demand for new automatic image interpretation methods.

One important advance in this field was the introduction of the object-oriented image classification approach (Blaschke, 2001). Methods based on this approach endeavor to explore all the semantic information contained in homogeneous image segments, not present in single pixels.

The key step in object-oriented image classification is the segmentation of the image into discrete meaningful objects. In fact, the performance of the whole interpretation depends essentially on the segmentation quality, and that depends on two major factors: the selected segmentation program and the segmentation parameter settings.

The first aspect is addressed by Meinel and Neubert in a recent publication (Meinel, 2004). The authors assess the quality of seven widely used segmentation programs over a pair of IKONOS images. The programs under analysis are ranked according to the adherence of their outcomes to a visually delineated reference.

The second quality conditioning aspect relates to the parameter adaptation. Most segmentation algorithms devised so far must be tuned in order to produce the desired outputs and cope with the varying characteristics (e.g. lightning conditions) of the images. Before starting the classification itself, the analyst must adapt the parameter values so that the segmentation produces meaningful objects. Generally the relation between the parameter values and the corresponding segmentation outcome is, however, far from being obvious, and the definition of suitable parameters is usually done through a troublesome and time consuming trial and error process.

The fact that most image processing (IP) algorithms and operators require some sort of tuning to perform properly in a given application has motivated the research on methods and tools to reduce the burden of IP parameter adaptation. As a result many semiautomatic approaches have been proposed, starting with simple graphic support tools, e.g. (Schneider, 1997), going through interactive systems, e.g. (Matsuyama, 1993), in which the user is required to rate the result after each adaptation iteration (Crevier, 1997), up to nearly automatic solutions that requires a minimum of human intervention.

The fully automatic approaches to adapt IP-parameters are usually based on a quality measure that is computed by comparing the outcome produced by the IP-operator with an available reference result. Quite often genetic algorithms (GA) (Davis, 1990) are applied to search the parameter-space for the solution that optimizes the selected quality measure (fitness function) (Bhanu, 1991; Bhanu, 1994; Kueblbeck, 1997). One important characteristic of GA is that they do not require any explicit model of the underlying process and can work with virtually any fitness function.

* Corresponding author.

In spite of the noticeable efforts reported in the literature, there is up till now no general solution to the problem of automatic adaptation of segmentation parameters. The performance of the GA based approaches depends on designing a fitness function that truly expresses the goodness of a segmentation outcome. Moreover, depending on the complexity of the process being optimized, the GA may involve unacceptable computational costs.

The present work addresses these topics and investigates the performance of a GA-based adaptation method working in conjunction with a particular segmentation algorithm. This work also proposes an intuitive and computationally uncomplicated fitness function for the GA.

A software prototype of the automatic adaptation method was built for performance assessment. Although the method can be easily extended to a variety of segmentation algorithms, experiments were limited to the algorithm proposed in (Baatz, 2000) and used in the eCognition software package (eCognition, 2005).

The subsequent text is organized in the following way. It begins with a brief overview of genetic algorithms. Next, a description of the segmentation algorithm used is made. A detailed description of the proposed adaptation method is then presented. The succeeding section reports the experimental evaluation carried out within this work. The final section contains the main conclusions of our work and suggests future research directions.

2. GENETIC ALGORITHMS

2.1 Genetic Algorithm's Principle

A genetic algorithm (GA) is a computational search technique to find approximate solutions to optimization problems. They are based in the biological evolution of species as presented by Charles Darwin (Darwin, 1859). The main principle of the Darwin's Theory of Evolution is that individual characteristics are transmitted from parents to children over generations, and individuals more adapted to the environment have greater chances to survive and pass on particular characteristics to their offspring.

2.2 Genetic Algorithm's Structure

In evolutionary computing context *individuals* represent potential solutions for a given problem, and their relevant characteristics with respect to the problem are called *genes*.

A *population* is a set of individuals in a particular *generation*, and individuals in a population are graded as to their capacity to solve the problem. That capacity is established by a *fitness function*, that indicates numerically how good an individual is as a solution to the problem (Michalewicz, 1998).

GAs propose an evolutionary process to search for solutions that maximize or minimize a fitness function. This search is made iteratively, over generations of individuals. For each generation the less fitted individuals are discarded, and new individuals are generated by the reproduction of the fittest. The creation of the new individuals is done by the use of *genetic operators*.

2.3 Genetic Operators

A genetic operator represents a rule for the generation of new individuals. The classical genetic operators are *crossover* and *mutation*. Mutation change gene values in a random fashion, respecting the genes' search space. Mutation is important to introduce a random component in the solution's search, in order to avoid convergence to local minima.

Crossover operators act by mixing genes between two individuals to create a new one that inherits characteristics of their parents. The general idea is that as a individual's fitness is a function of its characteristics, the exchange of good genes can produce better fitted individuals, depending on the genes inherited from their parents. Less fitted individuals can also be generated by this process, but they will have a low chance of being selected for reproduction.

There are many other genetic operators in the literature (Michalewicz, 1994). Most of them are variants of crossover and mutation, adapted for specific types of problems.

3. SEGMENTATION PROCEDURE

The segmentation procedure used in this work is based on the region growing algorithm proposed in (Baatz, 2000). The algorithm is a stepwise local optimization procedure that minimizes the average heterogeneity of the image objects.

Objects grow from single pixels, merging to neighboring objects. In each processing step an object can be merged to the neighbor that provides for the smallest growth of global heterogeneity. The merging decision is based on minimizing the resulting object's *weighted heterogeneity*, an arbitrary measure of heterogeneity weighted by object size.

The heterogeneity measure has a spectral and a spatial component. Spectral heterogeneity is defined over the spectral values of the pixels belonging to the object, and it is proportional to the standard deviation of the pixels' spectral values, weighted by arbitrary spectral band weights.

The spatial heterogeneity component is based on the deviation of the object's shape from a compact and a smooth shape. Compactness is defined as the ratio of the perimeter of the object and the square root of its area (the number of pixels it contains), and smoothness is defined as the ratio of the object's perimeter and the length of its bounding box (parallel to the image borders).

To simulate the parallel growth of the segments, objects are selected for merging only once in each iteration, in an evenly distributed fashion.

The merging decision mechanism is of key importance to this work, as it is where the external parameters of the segmentation procedure are employed. A *fusion factor* is calculated for each neighbor of the selected object, the neighbor for which this factor is minimum will be merged to the object, but only if the fusion factor is smaller than a certain threshold, defined as the square of the so called *scale parameter*. The procedure stops when no more objects can be merged.

As shown by equation 1, the fusion factor f contains a spectral heterogeneity component h_{color} and a spatial heterogeneity component h_{shape}. The relative importance of each type of heterogeneity is set by the color weight w_{color}.

$$f = w_{color} \cdot h_{color} + (1 - w_{color}) \cdot h_{shape} \quad (1)$$

Equation 2 shows the formulation of the spectral component of the fusion factor, where *Obj1* is the object selected for merging, *Obj2* is a neighbor object and *Obj3* is the result of the merging of *Obj1* and *Obj2*. In the equation c is a spectral band index and w_c is an arbitrary band weight; σ_c is the standard deviation of the pixel values for band c, considering all pixels belonging to an object; and n is the number of pixels of each object.

$$h_{color} = \sum_c w_c \left(n_{Obj3} \cdot \sigma_c^{Obj3} \left(n_{Obj1} \cdot \sigma_c^{Obj1} - n_{Obj2} \cdot \sigma_c^{Obj2} \right) \right) \quad (2)$$

The spatial component of the fusion factor has again two components (equation 3), a compactness component h_{cmpct} and a smoothness component h_{smooth}. The relative importance of each component is set by the weight w_{cmpct}.

$$h_{shape} = w_{cmpct} \cdot h_{cmpct} + (1 - w_{cmpct}) \cdot h_{smooth} \quad (3)$$

Equations 4 and 5 show how the spatial components h_{cmpct} and h_{smooth} are calculated. In the equations l stands for the perimeter of the objects and b for the perimeter of the objects' bounding box.

$$h_{cmpct} = n_{Obj3} \cdot \frac{l_{Obj3}}{\sqrt{n_{Obj3}}} - \left(n_{Obj1} \cdot \frac{l_{Obj1}}{\sqrt{n_{Obj1}}} + n_{Obj2} \cdot \frac{l_{Obj2}}{\sqrt{n_{Obj2}}} \right) \quad (4)$$

$$h_{smooth} = n_{Obj3} \cdot \frac{l_{Obj3}}{b_{Obj3}} - \left(n_{Obj1} \cdot \frac{l_{Obj1}}{b_{Obj1}} + n_{Obj2} \cdot \frac{l_{Obj2}}{b_{Obj2}} \right) \quad (5)$$

Throughout the segmentation procedure objects grow based on an adjustable criteria for heterogeneity. This adjustment can be made by setting the values of the *segmentation parameter*, the spectral band weights (w_c), the color weight (w_{color}) and the compactness weight (w_{cmpct}).

Adjusting the scale parameter influences the overall object size: the larger its value, the bigger the resulting segments. Additionally, the influence of each spectral channel, the influence of shape against color, and of compactness against smoothness in shapes can be set.

Given a particular image's spectral and spatial characteristics, the land use/land cover characteristics of the investigated site, and the relevance of certain classes of objects for the users' applications, those parameters can change considerably. And finding a good set of parameters for each case is by no means a trivial task.

4. ADAPTATION OF SEGMENTATION PARAMETERS USING A GENETIC ALGORITH

4.1 Processing Scheme

In this work a genetic algorithm evolves the segmentation parameters mentioned in the last section.

In the devised GA each individual consists of a set of segmentation parameters, each parameter representing a gene. The fitness of an individual is calculated by comparing the segmentation produced by the use of its genes with the target segmentation. The fittest individuals are the ones that provide for the best segmentations in terms of that comparison.

The gene values of the individuals in the initial population are generated randomly. As the evolutionary process advances, new generations of individuals are created by reproduction operations, in which the individuals exchange genes or are subjected to mutation. The selection of individuals for reproduction takes the fitness values into consideration, in a way that the fittest individuals have a larger probability of being selected. Furthermore, the reproduction process keeps the best individuals from one generation to the next.

The evolutionary process stops after a fixed number of generations, and the gene values of the fittest individual are taken as the final adapted segmentation parameters.

For computational efficiency, segmentation may be restricted to a small window around each target segment. This considerably reduces the processing time in comparison to segmenting the whole image at each fitness evaluation.

4.2 Fitness Evaluation

The fitness of an individual should indicate how good the segmentation of the input image is in comparison to the target segmentation. In mathematical terms, given a set of target segments S and a parameter vector P, a fitness function $F(S, P)$ that appropriately expresses the goodness of a segmentation outcome must be defined. Once the fitness function F is chosen, the task of the GA consists in searching for the parameter vector P_{opt}, for which the value of F is minimum:

$$P_{opt} = arg_P \left(min \left[F(S, P) \right] \right) \quad (6)$$

The fitness function devised in this work is defined as follows. Let S_i denote the set of pixels belonging to the i^{th} segment of the set S. Let $O_i(P)$ denote the set of pixels belonging to the segment with the largest intersection with S_i among the segments produced by using P as parameter values of the segmentation algorithm. The fitness function is then given by the equation below:

$$F(S, P) = \frac{1}{n} \sum_{i=1}^{n} \frac{\#(S_i - O(P)_i) + \#(O(P)_i - S_i)}{\#(S_i)} \quad (7)$$

in which '-' represents the set difference operator, '#( )' is the cardinality function, and n is the number of segments in the set S. Note that a perfect match between the target segmentation and the output segmentation with parameters P corresponds to $F=0$.

It is also important to point out that S does not need to represent a complete segmentation of the input image, where every pixel of the image would belong to a segment in S. In fact, in the experiments presented in this paper, S contains only 5 or 10 segments.

4.3 Reproduction Procedure

As stated before, the initial population, or the first generation of individuals, is created by setting random values for the genes of each individual. After fitness evaluation, a new population is created by substituting the M worst individuals of the prior population, being M a positive integer value smaller than the population size.

The new individuals are created by genetic operations over selected individuals of the prior population. The selection of individuals is done by a roulette mechanism, that takes into consideration normalized fitness values (Davis, 1990).

The following genetic operators were used (Davis, 1990; Michalewicz, 1994). *One point crossover*: two individuals exchange genes; *arithmetic crossover*: a linear combination of a set of genes of two individuals is performed; *mutation*: the value of a gene is modified by a random value; two types of *creep mutation*: gene values are adjusted (added or subtracted) by smaller or larger randomly generated values.

The selection of the reproduction operation is also done by a roulette mechanism, considering a predefined probability value for each operator. To prevent convergence to local minima, the operators' probabilities are interpolated during the evolution process (Davis, 1990), decreasing crossover probability while enhancing mutation and creep probabilities.

4.4 Implementation

A software prototype of the described automatic adaptation method was built for performance assessment. Although the method is not restricted to any particular segmentation algorithm, our experiments were limited to the algorithm proposed in (Baatz, 2000), and a C++ version of the aforementioned segmentation procedure was written specifically for that purpose. The genetic algorithm was also implemented in C++.

The parameters of the GA: number of generations, population size and genetic operations' initial and final probability values were set in a configuration file, so that the tuning of the GA could be made without the need for reprogramming.

Gene value domain, in terms of the maximum and minimum allowed values, as well as the decimal precision for each gene were also set in the configuration file.

The segmentation parameters: scale parameter, color and compactness weights were coded each into a single gene. The band weights (red, green and blue channel weights in the particular case of the experiments presented here) were coded into a single gene. A special coding method was devised so that this single value would be translated into a unique set of band weight values. This was done in order to avoid the possibility of multiple optimal solutions, as the band weights are normalized in our implementation of the segmentation algorithm.

5. PERFORMANCE EVALUATION

5.1 Input Images

Image data of two different sources were used: pansharped ETM Landsat and IKONOS images, produced in 2001 and 2002, and with spatial resolutions of approximately 15 and 1 meter respectively. From each scene two 256 by 256 pixel images were cut over sites with different land cover characteristics. Figures 1 and 2 show the images cut from the Landsat scene (images 1 and 2) and figures 3 and 4 show the images cut from the IKONOS scene (images 3 and 4).

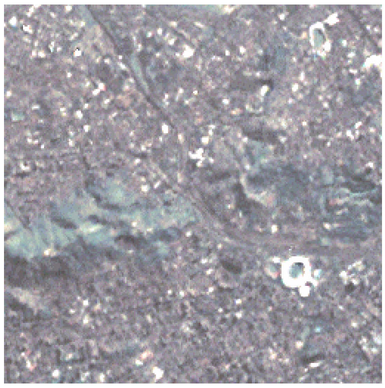

Figure 1. Image 1 (Landsat ETM)

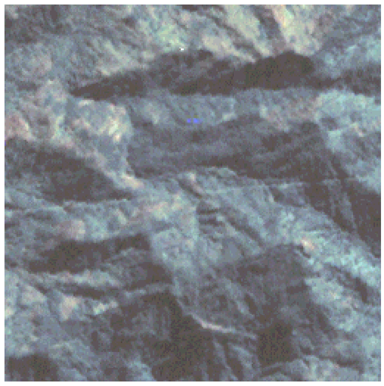

Figure 2. Image 2 (Landsat ETM)

Figure 3. Image 3 (IKONOS)

Figure 4. Image 4 (IKONOS)

5.2 Selection of Target Segments

In the practical use of the proposed method, a human operator will draw the target segments by hand. Generally there will be in such a case no guarantee that a set of parameter values for a perfect match exists. In fact, an eccentric choice of segment samples may hinder to attain a good fitness evaluation.

In order to purge this aspect from our analysis, we selected as target segments samples produced by the same segmentation algorithm working with known parameter values. So, in our experiments the only possible cause of a poor fitness evaluation is the departure from the optimum solution.

For each experiment the input images were segmented using different parameters. Sample segments were selected manually from the resulting segmentation to be used as the input target segmentation for the GA. The basic criterion for the selection of samples was to end up with a well spatially distributed set of segments that did not intercept the borders of the images.

At first, experiments were performed varying only the scale parameter, color weight and compactness weight, and maintaining the band weights fixed at the value 1. The results of those experiments are, however, not presented in this paper due to the limited space available. Table 1 shows the parameters used in the experiments presented in this paper. In the table, the parameter values of the target segmentation are shown in the top line of the rows that represent the experiments.

5.3 Genetic Algorithm Parameters

After several experiments executed to tune the GA, with the definition of the set of parameters that would facilitate convergence, avoiding local minima at the same time, the following parameters were defined for the GA: population size of 50 individuals, 40 generations, 90% of the individuals changed from one generation to the next.

The scale parameter could vary from 0 to 100, and the other segmentation parameters from 0 to 1. The decimal precision set for the scale parameter and for the color weight and compactness weight was 0.01, the precision for the band weights was set to 0.1.

5.4 Results

The experiment results are stated in Table 1. The columns *scale_param*, *w_color*, *w_cmpct*, *w_b1*, *w_b2* and *w_b3* show the values used in the target segmentation (on the top line), and the values of the fittest individual found by the GA in five runs of each experiment (on the bottom line). The column *evaluation* shows the fitness value of that individual.

experiment	*image*	*scale_param*	*w_color*	*w_cmpct*	*w_b1*	*w_b2*	*w_b3*	*evaluation*
1	1	30.0 27.7	0.80 0.65	0.50 0.33	0.6 0.6	0.3 0.3	0.1 0.1	0.09
2	1	30.0 25.4	0.80 0.69	0.50 0.32	0.3 0.3	0.1 0.1	0.6 0.6	0.00
3	1	60.0 44.2	0.80 0.82	0.50 0.57	0.6 0.6	0.3 0.3	0.1 0.1	0.02
4	1	60.0 44.2	0.80 0.71	0.50 0.32	0.3 0.3	0.1 0.1	0.6 0.6	0.01
5	2	30.0 27.8	0.80 0.76	0.50 0.38	0.6 0.6	0.3 0.3	0.1 0.1	0.01
6	2	30.0 29.0	0.80 0.70	0.50 0.35	0.3 0.3	0.1 0.1	0.6 0.6	0.02
7	2	60.0 41.6	0.80 0.56	0.50 0.14	0.6 0.6	0.3 0.3	0.1 0.1	0.06
8	2	60.0 53.7	0.80 0.79	0.50 0.47	0.3 0.3	0.1 0.1	0.6 0.6	0.00
9	3	30.0 29.7	0.80 0.77	0.50 0.42	0.1 0.1	0.3 0.4	0.6 0.5	0.00
10	3	30.0 30.2	0.80 0.81	0.50 0.50	0.6 0.6	0.3 0.3	0.1 0.1	0.01
11	3	60.0 58.8	0.80 0.80	0.50 0.51	0.1 0.3	0.3 0.2	0.6 0.5	0.02
12	3	60.0 57.6	0.80 0.74	0.50 0.28	0.6 0.7	0.3 0.2	0.1 0.1	0.11
13	4	30.0 26.2	0.80 0.62	0.50 0.25	0.1 0.1	0.3 0.3	0.6 0.6	0.03
14	4	30.0 17.2	0.80 0.34	0.50 0.34	0.3 0.2	0.6 0.5	0.1 0.1	0.27
15	4	60.0 44.5	0.80 0.65	0.50 0.24	0.3 0.3	0.6 0.6	0.1 0.1	0.21
16	4	60.0 53.4	0.80 0.79	0.50 0.57	0.1 0.1	0.3 0.3	0.6 0.6	0.26

Table 1. Experiment parameters and results.

The fitness value achieved for the Landsat images were all very close to zero, the ideal value. For the IKONOS images, slightly worst results were obtainded in the experiments 14, 15 and 16. Those results can be can be explained by the greater complexity of the shapes of the sample segments used in those experiments, related to the particular choice of band weights.

A visual inspection of the results shows that the resulting segments are very similar to the sample target segments. This indicates that the proposed fitness function is close related to the subjective human evaluation of the segmentation result.

It is interesting to notice the slight deviations of the scale parameter, color weight and compactness weights from the parameters used in the target segmentation. The, in average, largest deviation from the compactness weights can be explained by the largest importance given for spectral heterogeneity in the target segmentation, translated by the value of the color weight (0.8). Deviations from the scale parameter can be explained by the particular selection of the sample segments, as no particular effort was made to select the largest segments in the target segmentation. Visual inspection confirms those considerations once some of the largest segments, from the ones not selected as samples, correspond to sometimes a few segments in the resulting segmentation.

It is also worth mentioning that our experiments showed a similar performance for both sensors as well as for all test areas.

6. CONCLUSIONS AND FUTURE WORKS

The experimental results in terms of the evaluation of the best individuals found for the various experiments show the potential of the devised approach for the adaptation of segmentation parameters.

The GA was able in some cases to find more than one solution. While this can impose certain difficulties for the convergence of the GA, it shows the robustness of the developed methodology. This is endorsed by the similar performance observed in our experiments for both sensors and for distinct test areas.

A new experimental environment is currently under development, in which the user can draw segments over the image objects of interest and use those segments as the target segments for fitness evaluation. One objective of this investigation is to check if this approach can be used as an initial step in an adaptable object extraction technique.

Further developments of the implemented prototype are also under consideration, especially for its optimization in terms of reducing the time needed for the experiments. Currently, in a standard Pentium 4 1.8GHz processor, each experiment takes about 3,5 hours. Parallel computation of the evaluation of individuals in a generation is being considered. A further improvement of the GA can also help in that sense, the use of *cultural algorithm* concepts (Becerra, 2005) may help to accelerate the convergence to optimal solutions.

Moreover, the development of the computer technology will soon reduce the observed processing time to more satisfactory levels.

AKNOWLEGMENTS

The authors would like to acknowledge the support from the CAPES (Fundação Coordenação de Aperfeiçoamento de Pessoal Superior) and DAAD (Deutscher Akademischer Austauschdienst) in the context of the PROBRAL International Scientific Cooperation Program.

REFERENCES

Baatz, M., Schäpe, A. 2000. Multiresolution Segmentation – an optimization approach for high quality multi-scale image segmentation. In: *Strobl/Blaschke/Griesebner (editors): Angewandte Geographische Informationsverarbeitung XII*, Wichmann-Verlag, Heidelberg, pp. 12-23.

Becerra, R. L., Coello, C. A. C. 2005. Use of domain information to improve the performance of an evolutionary algorithm. In: Proceedings of the 2005 workshops on Genetic and Evolutionary Computation, Washington, D.C. pp. 362-365.

Bhanu, B., Ming, J. C., Lee, S. 1991. Closed-loop adaptive image segmentation. *In: Proceedings of the IEEE Conference on Computer Vision and patter Recognition (CVPR).*

Bhanu, B., Lee, S. 1994. *Genetic Learning for Adaptive Image Segmentation*. Luwer Academic Publishers, London.

Blaschke, T., Strobl, J., 2001. What is wrong with pixels? Some recent developments interfacing remote sensing and GIS. *GIS-Zeitschrift für Geoinformationssysteme*, 6, pp. 12-17.

Crevier D., Lepage, R. 1997. Knowledge-based image understanding system: A survey. *Computer Vision and Image Understanding*, 67(2), pp. 161-185.

Davis, L., 1990. *Handbook of Genetic Algorithms*, Van Nostrand Reinhold, New York.

Darwin, C. 1859. *On the Origin of Species by Means of Natural Selection or the Preservation of Favored Races in the Struggle for Life*. John Murray.

eCognition, 2005. *eCognition User Guide*, 2005. http://www.definiens.com (accessed 2 Dez. 2005)

Kueblbeck C., Wagner, T. 1997. Automatic configuration surface inspection systems. In: *A. Ravishankar Rao and Ning Chang (editors): Proceedings of the SPIE: Machine Vision Applications in Industrial Inspections V*, Vol. 3029, San Jose, California.

Matsuyama, T. 1993. Expert system for image processing, analysis, and recognition: declarative knowledge representation for computer vision. *Advances in Electronics and Electron Physics*, 86, pp. 81-171.

Meinel. G., Neubert, M. 2004. A comparison of segmentation programs for high resolution remote sensing data. *In: Proceedings of the ISRPS 2004 Annual Conference*, Istanbul, Turkey, pp. 19 - 23.

Michalewicz, Z., 1994. *Genetic Algorithms + Data Structures = Evolution Programs*, Springer-Verlag, New York.

Schneider, W., Eckstein, W., Steger, C. 1997. Real-time visualization of interactive parameter changes in image processing systems. *In: Georges G. Grinstein and Robert F. Erbacher (editors): Visual Data Exploration and Analysis IV, Proceedings SPIE*, Vol. 3017, pp. 286-295.