AUTOMATIC EVALUATION OF SEGMENTATION PARAMETERS

THIAGO BROERMAN CAZES
GILSON ALEXANDRE OSWALD PEDRO DA COSTA
RAUL QUEIROZ FEITOSA

Pontifícia Universidade Católica do Rio de Janeiro - PUC-RIO
Universidade do Estado do Rio de Janeiro - UERJ
{tcazes,gilson,raul}@ele.puc-rio.br

ABSTRACT – This paper evaluates a method, previously proposed by the authors, for the automatic adaptation of segmentation parameters based on Genetic Algorithms. An intuitive and computationally simple fitness function is used that expresses the similarity between the segmentation result and a reference provided by the user. The method searches the solution space for a set of parameter values that minimizes the fitness function. A prototype including two of the most widely used segmentation algorithms was developed to assess the performance of the method. A set of experiments using a high-resolution image was carried out. In experiments the method was able to come close to the ideal solution.

1 INTRODUCTION

The key step in object-oriented image interpretation is the segmentation (Blaschke 2001). In fact, the performance of the whole interpretation strongly depends on the segmentation quality. For that reason, proper segmentation parameters must be chosen, before starting the classification process. However, the relation between the parameter values and the corresponding segmentation outcome is generally far from being obvious, and the definition of suitable parameter values is usually done through a troublesome and time consuming trial and error process.

Many semiautomatic approaches have been proposed to reduce the burden of parameter adaptation, 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 automatic adaptation of segmentation parameters involves two main issues: the selection of an objective function that expresses adequately the quality of the segmentation (Bhanu 1995); and the choice of the optimization method for the search of parameter values that maximize the objective function. In supervised methods, the quality measure reflects the similarity among the segmentation output and reference segments, usually produced manually by a photo-interpreter (Zhang 1996). Unsupervised methods, on the contrary, use no references, and do not consider human induced subjectivity or application particularities (Espindola 2006).

Generally the relationship among the values of segmentation parameters and the quality measure can not be formulated analytically. In such cases calculus based optimization methods cannot be used. Genetic algorithms (GA) (Davis, 1990) do not require any explicit model of the underlying process and can work with virtually any objective function (Bhanu 1991; Bhanu 1994; Everingham 2002; Kueblbeck 1997).

The authors proposed in previous work (Feitosa 2007) an automatic GA-based adaptation method to estimate proper parameter values for segmentation algorithms. In the same work a fitness function was proposed that expresses the agreement between a set of user defined segment samples and the automatic segmentation outcome.

The present work addresses these topics and evaluate this method on two widely used region growing segmentation algorithms. One of them is the Baatz segmentation algorithm (Baatz 2000) based on object homogeneity criterion that is implement in the commercial software eCognition (eCognition 2005). The other one is the SPRING segmentation algorithm (Spring 1996) developed by the National Institute of Spatial Research (INPE) and distributed in the open source component library TERRALIB (Terralib 2007).

The subsequent text is organized in the following way. It begins with a brief overview of genetic algorithms. A detailed description of the 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.

T.B. Cazes; G.A.O.P. Costa; R.Q. Feitosa
2 GENETIC ALGORITHM

2.1 Basic Concepts and Terminology

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.

In evolutionary computing terms individual represent a potential solution 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 determined by a fitness function, which 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 performed 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.

2.2 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 an individual’s fitness is a function of its characteristics, and the exchange of good genes may 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 lower chance of being selected for reproduction.

Other genetic operators can be found in the literature (Michalewicz 1994). Most of them are variants of crossover and mutation, adapted for specific types of problems.

3 ADAPTATION OF SEGMENTATION PARAMETERS USING A GENETIC ALGORITHM

3.1 Processing Scheme

In this work a genetic algorithm evolves the values of segmentation parameters mentioned in the last section. In the devised GA each individual consists of a set of segmentation parameter values, each parameter representing a gene. The fitness of each solution (individual) is calculated by comparing the segmentation produced by the solution and reference segmentation (Figure 2).

The parameter values (genes) of the initial set of solutions (initial population) are generated randomly. As the evolutionary process advances, the best solution (fittest individuals) are selected and new solutions (generations) are created from them (reproduction).

The selection of individuals for reproduction takes the fitness values into consideration, so 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.

3.2 Reproduction Procedure

As stated before, the initial population (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 replacing 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, which 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 help preventing convergence to local minima, the operators’ application probabilities are interpolated during the evolution process (Davis 1990), decreasing crossover probability while enhancing mutation and creep probabilities.

### 3.3 Fitness Evaluation

The fitness of an individual should indicate the goodness of a segmentation result in relation to the reference segmentation. In mathematical terms, given a set of reference 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:

$$P_{opt} = \text{arg}_P \left[ \min \left( F(S, P) \right) \right] \quad (1)$$

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 $O(P)$ 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 Eq. 2, in which ‘,’ represents the set difference operator, ‘#( )’ is the cardinality function, and $n$ is the number of segments in the set $S$.

$$F(S, P) = \frac{1}{n} \sum_{i=1}^{n} \frac{\#(S_i - O(P)) + \#(O(P) - S_i)}{\#(S_i)} \quad (2)$$

Figure 3 shows graphically the components of the proposed fitness function. The region enclosed by the solid contour represents a referent segment $S_i$, and its area given in pixels corresponds to the denominator of Eq. 2. The region with the dashed contour represents $O(P)$. The shadowed area given in pixels corresponds to the numerator in Eq. 2. Notice that $F=0$ indicates a perfect match between the reference and the output segmentation.

It is 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 to 10 segments.

### 4 SEGMENTATION PROCEDURES

In this work two region growing based segmentation algorithms were tested.

The first algorithm (Bins et al. 1996) was implemented in SPRING and is available in TerraLib (TerraLib 2007). In essence, it works with two parameters; the most important one is a spectral threshold. Two adjacent regions may be merged if the Euclidean distance between the spectral mean values of them is lower than that threshold. The second parameter is an area threshold and establishes the minimum number of pixels that a segment must contain.

The second segmentation algorithm was proposed in (Baatz and Schäpe 2000) and is implemented in eCognition (Definiens 2005). It defines a uniformity criteria based on global image heterogeneity. The merging decision is based on minimizing the resulting object’s weighted heterogeneity, an arbitrary measure of heterogeneity weighted by object size. In each processing step one object can be merged to its neighbor that
provides for the minimum increase of global heterogeneity. 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). 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 further objects can be merged. In the evaluation of the fusion factor two additional parameters are consider. The first one represents the relative importance between spectral and spatial criteria. The second one expresses the relative weight of smoothness and compactness. Finally weights for each spectral image band may also be defined. For the segmentation of a 3 band image, the values of six parameters must be established.

5 EXPERIMENTS

In order to evaluate the performance of the proposed method, a software prototype was developed in C++. The prototype includes the implementation of the algorithm described in Baatz & Schäpe (Baatz 2000) and the SPRING segmentation routine available in TerraLib (TerraLib 2007).

In the proposed method, a human interpreter delimits manually a set of polygons over the image, which represents his expectation regarding the result of the segmentation. These polygons represent the reference segmentation, to be used during the segmentation parameters adaptation, and should capture the subjectivity of the interpreter in the segmentation evaluation.

The cognitive processes applied by the human interpreter is, especially for remote sensing data, very complex, and cannot be fully represented in the segmentation algorithm, as they lie almost totally below our threshold of consciousness (Crevier 1997).

Therefore, there is no guaranty that a set of parameter values able to produce a perfect match, with the reference defined by the interpreter, exists. That considered, the eventual GA incapacity to converge to a satisfactory solution can result from the limitations of the segmentation procedure, or from an eccentric choice of reference segments by the interpreter.

In experiments described hereafter, we evaluate the capacity of the method to capture the interpreter’s subjective perception about the segmentation quality.

5.1 Input Images

The test image used was taken from public resources on the Internet. And represent a parking area of Bus Company (see Figure 5) situated in Rio de Janeiro, Brazil. This image has 400x400 pixels, RGB format with 24 bits (8 bits per band).
5.3 Results

The experimental results are partially presented in Table 1. In Baatz algorithm GA converge to optimum solution after 40 iterations, in case of Spring algorithm this occur after 15 iterations. One reason to that is the fact that Spring method has less parameters to be estimate.

The fitness score on the validation set is close to the fitness score achieved on the training segment set. This is evidence that the method generalizes well with both segmentation algorithms, at least for objects of the same type.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Fitness score</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>training</td>
<td>validation</td>
</tr>
<tr>
<td>Baatz</td>
<td>0.310</td>
<td>0.360</td>
</tr>
<tr>
<td>Spring</td>
<td>0.301</td>
<td>0.296</td>
</tr>
</tbody>
</table>

Table 1 – Experiment result

Figure 6 shows the segmentation of the whole image produced by the Baatz algorithm using the parameter values selected by our method. The result is consistent for the busses on the images. For image objects other than busses, the segmentation does not perform so well. This must be credited to the segmentation algorithm itself. The Baatz algorithm tends to produce segments with similar size, which is determined by the scale parameter. This fact can be clearly observed in the large garage in the bus parking area. This object is much larger than the busses and was segmented in a number of much smaller objects. If the garage had been used along with a couple busses in the set of reference segments, our method would possibly find a suitable solution neither for the busses nor for the garage. This example shows that a consistent selection of training segment samples is a condition for our method to perform well. This statement applies to any segmentation algorithm.

The SPRING algorithm was able to nicely delineate buses as well as the garage with the same set of parameter values found by our method using only buses as training samples (see Figure 7). Contrary to the Baatz approach, SPRING does not consider segment size in the heterogeneity criteria, allowing for segments with quite different areas.
6 CONCLUSION AND FUTURE WORKS

In this paper a method for the automatic adaptation of segmentation parameter values based on Genetic Algorithms was analyzed in conjunction with two segmentation algorithms.

In this method, values of segmentation parameters are coded into genes of the individuals of the GA, and fitness evaluation is a measure of the similarity between a user defined set of segment samples and the segmentation result.

The results of experiments indicate that the adaptation method is able to express numerically, therefore, in an objective way, the subjective perception of the human interpreter of the segmentation quality. Through a small set of manually drawn segments, the interpreter could indicate the expected output of the segmentation procedure, and in all experiments the adaptation method was able to produce visually consistent solutions. The adaptation method performed equally well for both segmentation algorithms.

The software prototype implemented for this work is available upon request to the authors.

ACKNOWLEDGMENTS

The authors acknowledge the support provided by FAPERJ (Fundação de Amparo à Pesquisa do Estado do Rio de Janeiro) and FINEP (Financiadora de Estudos e Projetos).

REFERENCES


