INTEGRATING TEXTURE FEATURES INTO A REGION-BASED MULTI-SCALE IMAGE SEGMENTATION ALGORITHM

Angelos Tzotsos*, Christos Iosifidis, Demetre Argialas

Laboratory of Remote Sensing, Department of Surveying, School of Rural and Surveying Engineering, National Technical University of Athens, Greece – tzotsos@gmail.com, chiossif@gmail.com, argialas@central.ntua.gr

Commission IV, WG IV/4 – Commission VIII, WG VIII/11

KEY WORDS: Object Based Image Analysis, Remote Sensing, Multiresolution, Co-Occurrence Matrix, Angular Second Moment

ABSTRACT:

The objective of this research was the design and development of a region-based multi-scale segmentation algorithm with the integration of complex texture features, in order to provide a low level processing tool for object-oriented image analysis. The implemented algorithm is called Texture-based MSEG and can be described as a region merging procedure. The algorithm is composed of two profiles. In the simple profile, the main part of the segmentation algorithm was included. The first object representation is the single pixel of the image. Through iterative pair-wise object fusions, which are made at several iterations, called passes, the final segmentation is achieved. The criterion for object merging is a homogeneity cost measure, defined as object heterogeneity, and computed based on spectral and shape features for each possible object merge. The heterogeneity is then compared to a user defined threshold, called scale parameter, in order for the decision of the merge to be determined. The processing order of the primitive objects is defined through a procedure (Starting Point Estimation), which is based on image partitioning, statistical indices and dithering algorithms. The advanced profile was implemented as an extension of the simple profile and was designed to include multi-resolution functionality and a global heterogeneity heuristic module for improving the segmentation capabilities. As part of the advanced profile, an integration of texture features to the region merging segmentation procedure was implemented through an Advanced Texture Heuristics module. Towards this texture-enhanced segmentation method, complex statistical measures of texture had to be computed based on objects, however, and not on orthogonal image regions. For each image object the grey level co-occurrence matrices and their statistical features were computed. The Advanced Texture Heuristics module, integrated new heuristics in the decision for object merging, involving similarity measures of adjacent image objects, based on the computed texture features. The algorithm was implemented in C++ and was tested on remotely sensed images of different sensors, resolutions and complexity levels. The results were satisfactory since the produced primitive objects, were comparable to those of other segmentation algorithms. A comparison between the simple profile derived primitive objects and the texture based primitive objects also took place showing that texture features can provide good segmentation results in addition to spectral and shape features.

1. INTRODUCTION

1.1 Recent developments in Remote Sensing

Remote sensing has recently achieved great progress both in sensors and image analysis algorithms. Due to very high resolution imagery, such as IKONOS and Quick Bird, traditional classification methods, have become less effective given the magnitude of heterogeneity appearing in the spectral feature space of such imagery. The spectral heterogeneity of imaging data has increased rapidly, and the traditional methods tend to produce classification errors such as multiple spectral signatures within a semantic object. Those multiple signatures, cannot be effectively dealt with standard methods and tend to produce "salt and pepper" classification results, when one semantic object is composed of multiple spectral signatures.

Another disadvantage of traditional classification methods is that they do not use information related to shape, site and spatial relation (context) of the objects of the scene. Context information is a key element to photo-interpretation, and a key feature used by all photo-interpreters because it encapsulates expert knowledge about the image objects. Such knowledge, however is not explicit, and needs to be extracted, represented and used for image analysis purposes. In order to improve classification results from image analysis, it is of high importance to be able to use key features of photo-interpretation, such as shape and texture.

1.2 Texture Image Segmentation and Object-based Image Analysis

Approaches have been developed the fields of Computer Vision and Remote Sensing, for texture analysis and image segmentation. In addition to simple texture features, such as standard deviation and variance, Haralick proposed more complex texture features computed from co-occurrence matrices (Haralick et al 1973, Haralick 1979). These second order texture features were used in image classification of remote sensing imagery with good results (Materka and Strzelecki 1998). Even more complex texture models have been used for texture modelling, classification and segmentation, such as Hidden Markov Models, Wavelets and Gabor filters (Materka and Strzelecki 1998) with very good results in remote sensing and medical applications. Several methods have been proposed for texture image segmentation, taking advantage of the latest texture modelling methods (Chen et al 2002, Fauzi

* Heroon Polytechneiou 9, 15780, Zografou, Athens, Greece. Tel. +302107722684

and Lewis 2003, Havlicek and Tay 2001, Liapis et al 1998). At the same time, image classification also moved towards computational and artificial intelligence methods (Sukissian et al 1994, Benz et al 2004).
During the last few years, a new approach, called Object-Oriented Image Analysis, integrated low level image analysis methods, such as segmentation procedures and algorithms (Baatz & Schäpe 2000), with high level methods, such as Artificial Intelligence (knowledge-based expert systems and fuzzy systems) and Pattern Recognition methods. Within this approach, the low level image analysis produces primitive image objects, while the high level processing classifies these primitives into meaningful domain objects (Benz et al 2004).

1.3 Research Objectives

The main objective of this research was the integration of texture features into an object-oriented image segmentation algorithm. It was desired that the modified segmentation algorithm could be used as a low level processing part of an object-oriented image analysis system so that to be applied at multiple image resolutions and to produce objects of multiple scales (sizes), according to user-customizable parameters.
A further objective was the ability of the produced algorithm to be generic and produce good and classification-ready results from as many remote sensing data as possible. Remote sensing data are, in general, difficult to process, with complex textural and spectral information. Therefore, there was a need for the algorithm to be able to handle texture information and context features in order to produce better segmentation results.

2. METHODOLOGY

2.1 MSEG algorithm – Simple Profile Overview

The MSEG algorithm (Tzotsos and Argialas 2006) was designed to be a region merging technique, since region merging techniques are fast, generic and can be fully automated (without the need of seed points) (Sonka et al 1998, Pal and Pal 1993). Given that existing Object-Oriented Image Analysis systems (eCognition User Guide 2005) have used such methods was also a strong argument for the effectiveness of the region merging techniques.
After the data input stage (Figure 1), an image partitioning procedure (Macroblock Estimation) was applied to the dataset resulting into rectangular regions of variable dimensions, called macroblocks. Image partitioning was applied for computing local statistics and for computation of starting points. Starting points were then used for initialization of the algorithm and for reproducibility of segmentation results.
After the Macroblock Estimation, the SPE module (Starting Point Estimation) computed local statistics and provided starting points for initialization of the region merging process. It should be stretched that **starting points were not used as seed points** (as in region growing techniques) but are used to keep track of the order in which **all pixels were initially processed** (Tzotsos and Argialas 2006).
MSEG is basically defined as a region merging technique. Like all algorithms of this kind, it was based on several local or global criteria and heuristics, in order to merge objects in an iterative procedure, until no other merges can occur (Sonka et al 1998). In most cases, a feature of some kind (mean spectral values, texture, entropy, mean square errors, shape indices etc.) or combination of such features computes the overall "energy" of each object. Various definitions of homogeneity (energy minimization measures or measures of similarity within an object) have been defined (Sonka et al 1998, Pal and Pal 1993). Recently, a very successful segmentation algorithm, embedded in the Object Oriented Image Analysis Software eCognition (Baatz & Schäpe 2000), implemented such measures of homogeneity, for making the merging decision between neighbouring objects, with very good results. Some spectral and spatial heuristics were also used to further optimize the segmentation. In the proposed segmentation algorithm, similar homogeneity measures were used, and then complex texture features were implemented in later stages.

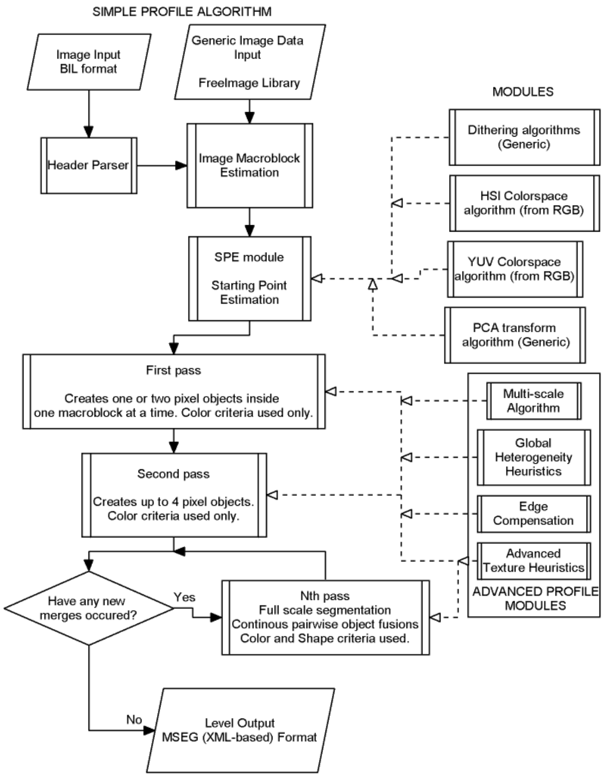

Figure 1: Flowchart of the MSEG algorithm

In order for the MSEG algorithm to provide primitive objects, several steps of region merging (passes) were followed. The purpose of the first segmentation pass (Figure 1) was to initialize image objects and to provide the first over-segmentation, in order for the algorithm to be able to begin region merging at following stages. Initially, the objects of the image are the single pixels. During first pass, the algorithm is merging single pixels-objects pair wise, inside each macroblock. For the second pass of the algorithm (Figure 1), the objects created by the first pass were used in a new pair wise merging procedure. Again, the same strategy of merging was used, finding the best match for each object, and then checking if there is a mutual best match in order to merge the two objects (Tzotsos and Argialas 2006). The Nth pass module, is called iteratively until the algorithm converges. The algorithm is considered finished, when during the nth pass no more merges occur and the algorithm converges (Figure 1). Then, the objects are exported and marked as final primitives.

2.2 MSEG algorithm – Advanced Profile Overview

The simple profile of the MSEG algorithm included the pass modules, as basic elements of a region merging segmentation procedure. The extension of the Simple Profile was used to include extra functionality algorithms and innovative techniques for improving the results. The Advanced Profile, as implemented at present, included the following modules:

- the Multi-scale Algorithm (MA), and
- the Global Heterogeneity Heuristics (GHH)
- the Advanced Texture Heuristics

The Multi-scale Algorithm module was designed to give to the MSEG algorithm the ability to create multiple instances of segmentations for an image, with different scale parameters. Thus, the produced primitive objects could vary in size and therefore, the multi-scale representation could model large image entities, as well as small ones. In order to include multiple instances of segmentation, inside an object-oriented image analysis system, those instances must be properly constrained to be integrated and used together (Tzotsos and Argialas 2006).

The problem when dealing with multiple segmentations, is the compatibility between scales, in order to combine information and objects. One simple way to deal with this problem is to create a multi-level representation, and incorporate the multiple segmentations within this representation, hierarchically.

But a single-level hierarchy is sometimes not flexible, when dealing with remote sensing classification problems (Argialas and Tzotsos 2004). A multi-level-hierarchy approach or a branch-based hierarchy model can represent more complex spatial relations. Thus, in the present Multi-scale segmentation Algorithm, every new level depends only from the nearest (scale-wise) super-level or the nearest sub-level, or both (Tzotsos and Argialas 2006).

The Simple Profile passes were based on the merging criterion of the mutual best match between neighboring objects. This criterion, sometimes is sub-optimal, due to computation cost (many virtual merges occur and few of them are producing real merges). This heterogeneity heuristic was found optimal at minimizing the scene heterogeneity after region merging procedures (Baatz and Schäpe 2000). An accuracy-to-speed ratio module has been implemented, including the global heterogeneity heuristics. The accuracy refers to the global heterogeneity cost that is added to the image with each merge that occurs during segmentation (Tzotsos and Argialas 2006).

2.3 Advanced Texture Heuristics

The basic objective of the Advanced Texture Heuristic module was to build upon MSEG's simple profile modules, in order to improve segmentation results. Since texture is a key photo-interpretation element, it was decided to use rather more complex texture features, than first order texture features (e.g. standard deviation, variance).

Since second order texture features are widely used in pixel classification (Haralick et al 1973, Materka and Strzelecki 1998), there was a need to test them for segmentation purposes and specifically as an add-on to the MSEG algorithm (Tzotsos and Argialas 2006).

Given that MSEG is a region merging algorithm, it should be taken under consideration that not all state of the art methods for modeling texture are compatible with a hybrid segmentation algorithm. The recent literature has shown that Markov Random Fields, Wavelets and Gabor filters, have great potential for texture analysis (Materka and Strzelecki 1998). Their disadvantage is that they are very complex and time consuming to be used with a procedure, involving thousands of virtual merges. At the same time, wavelets and Gabor filters are incapable to be used locally, within the boundaries of a single – and sometimes very small - primitive object. Markov Random Fields are easier to adopt for region-based texture segmentation, but they were found incompatible with the current merging search method, since they are based on Bayesian reasoning.

A traditional method for modeling texture, which has been proved very good for practical purposes in supervised classification (Haralick et al 1973, Schroder and Dimai 1998), is based on the Grey Level Co-occurrence Matrix (GLCM). GLCM is a two dimensional histogram of grey levels for a pair of pixels separated by a fixed spatial distance. The Grey Level Co-occurrence Matrix approximates the joint probability distribution of this pair of pixels. This is an insufficient approximation for small windows and a large number of grey levels. Therefore the image data have to be pre-scaled to reduce the number of grey levels in the image. Directional invariance can be obtained by summing over pairs of pixels with different orientations (Schroder and Dimai 1998).

From the GLCM several texture measures can be obtained, such as homogeneity, entropy, angular second moment, variance, contrast etc (Haralick et al 1973). To compute the GLCM, several optimization methods have been introduced (Argenti et al 1990, Miyamoto and Merryman 2006). Most applications of GLCM for remote sensing images, at pixel-level, included computation of the co-occurrence matrix less often for the whole image, and more often for a predefined image sliding window of fixed size.

In order to employ the second order texture features into MSEG, it was obvious that a GLCM should be computed for each primitive image object, during the merging procedure (Figure 2).

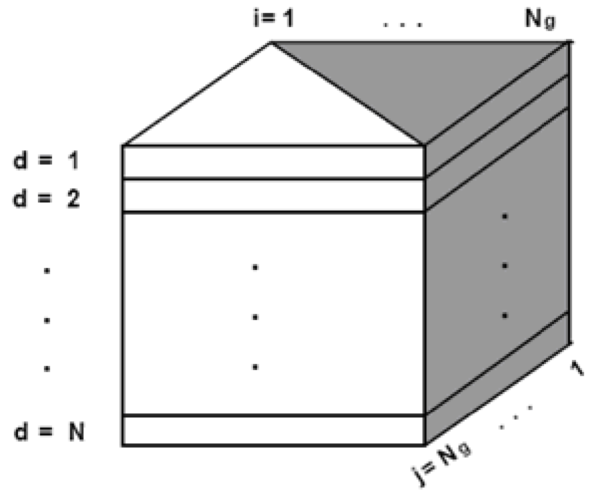

Figure 2: A 3-dimensional representation of the Co-occurrence matrices that have to be computed for a given orientation. Ng are the possible grey levels and N is the total number of primitive image objects (source: Miyamoto and Merryman 2006).

The Advanced Texture Heuristic module, performed the GLCM computation for each initial image object before any merge could occur at any given segmentation pass. Then, for each primitive object texture features were computed. The basic idea of this module was the implementation of a similarity measure, in order to decide whether two image objects are compatible in texture to be merged. A good similarity measure would be a homogeneity criterion based on the second order texture features. Haralick has indicated as good texture similarity measures, the Homogeneity (Equation 1) and the Angular

Second Moment (ASM) (Equation 2) (Haralick 1979). These were implemented into the Advanced Texture Heuristic module.

$$Homogeneity = \sum_{i,j=0}^{N-1} \frac{P_{i,j}}{1+(i-j)^2} \quad (1)$$

$$ASM = \sum_{i,j=0}^{N-1} P_{i,j}{}^2 \quad (2)$$

where P_{ij} is the GLMC value.

When an object was retrieved from the MSEG priority list (Tzotsos and Argialas 2006), the texture homogeneity features were computed from the GLCM. The mutual best match search procedure compared neighbour objects to the selected one, and computed the homogeneity texture features for those as well. Before, the color and shape heterogeneity criteria were computed and involved to the scale parameter comparison, texture heterogeneity was computed, as the difference of the values of the homogeneity texture features. These values, one for each direction and GLCM, were then compared with a threshold called texture parameter which can be defined by the user. If the two objects are found to be compatible by the texture parameter, then the computation of the spectral and shape heterogeneity takes place, in order to fulfil the mutual best match criterion, and the merge to occur.

The described heuristic, takes advantage of the texture parameter, to reduce the heterogeneity computations. This means that, when activated, the Advanced Texture Heuristic module, has greater priority than the scale parameter, but cannot perform any merging, without color and shape compatibility of image objects. If one wishes to perform segmentation using only texture features, the scale parameter can be set to a very large value, so that not to constrain the merging by the color and shape criteria.

In the following section, the optimization procedure for the GLCM computation is described.

2.4 Implementation

Having to compute thousands of co-occurrence matrices, during a region merging segmentation procedure can be computationally painful. If for each primitive object, a grey level reduction and a co-occurrence computation are performed, then the segmentation algorithm would slow down.

In order to tackle this problem, for the object-oriented algorithm, the GLCM computation should be optimized to be used with objects, rather than pixels. A modification to the traditional methods was performed, so that to make the procedure faster but not less accurate.

At first, there was the problem of image band selection. If the computation of the GLCM was to be performed for each band separately, the whole segmentation process would not be performed optimally. Given that the Starting Point Estimation, worked very well with one selected band, the idea to use the same selection was tested. So, instead of using all bands, the Advanced Texture Heuristic module can use the intensity band of the HSI colorspace, or the Y band of the YCbCr colorspace (used as default), or a principal component band of the image, or finally a single image band.

After the band selection, a grey level reduction was performed at the selected band. The final number of grey levels can be selected by the user, with a quantization parameter. The default value, as used by many other GLCM implementations was set to 32 grey levels (Miyamoto and Merryman 2006). The grey level reduction took place using histogram equalization technique and then a look-up table was computed to hold the new grey level information.

The optimal way to compute the GLCMs was designed to perform some kind of global pre-computation, so that to speed up the inter-object GLCM creation function. For each of the image pixel, a direction search was performed, to evaluate the grey level pair co-occurrences. For the 4 different directions, a vector was designed to hold the overall co-occurrence information. This way, no direction search was to be performed twice during the pass stages. After the completion of these vectors for all pixels, the segmentation procedure was initiated. Each time an object co-occurrence matrix was to be used, the object triggered a pointer structure to call all pixel co-occurrence vectors and performed very quick texture feature computation within the object boundaries.

This procedure was not tested for algorithmic complexity, but was compared to a simple GLCM implementation and was found more stable and faster.

The implementation of the Advanced Texture Heuristic module was performed in C++, as the MSEG algorithm. The modified version of the algorithm was called Texture-based MSEG.

3. RESULTS AND DISCUSSION

The implemented version of the MSEG algorithm was tested on a variety of image data, in order to assess the quality of the results, its ability to be generic and its speed. Evaluating the results of a segmentation algorithm does not depend on the delivery of semantic objects, but rather on the generation of good object primitives useful to further classification steps.

The algorithm was designed to provide over-segmentation so that merging of segments, towards the final image semantics, to be achieved by a follow up classification procedure. Boundary distinction and full-scene segmentation were of great significance. Since the eCognition software (eCognition User Guide 2005) is greatly used for object oriented image analysis purposes the evaluation of results was mainly based on comparison with outputs from eCognition. Also, a comparison was made to the Simple Profile results of MSEG, to show how the texture features performed along with region merging segmentation.

For the evaluation of the algorithms a Landsat TM image was used. The eCognition software was used to provide segmentations with scale parameters 10 and 20. The color criterion was used with a weight of 0.7 and the shape criterion with weight 0.3. The results are shown in Figures 3 and 4. Then, the simple profile of MSEG was used to provide segmentations with scale parameters of 400 and 700 respectively, to simulate the mean object size of eCognition's results. It should be noted that scale parameters are not compatible between the two algorithms, but are implementation dependent. MSEG results are shown in Figures 5 and 6.

In Figures 7, 8 and 9, the results from the texture-based MSEG are shown. Similar scale parameters with MSEG's simple profile results have been used, and also the same weights for color and shape criteria. In Figure 7, a scale parameter 400 and the texture parameter of 2.0 were used. In Figure 8, scale parameter was the same, but texture parameter was 1.0. Finally in Figure 9, scale parameter was set to 2500 and texture parameter was set to 3.0.

Figure 3: Segmentation result as provided by eCognition for scale parameter 10.

Figure 4: Segmentation result as provided by eCognition for scale parameter 20.

Figure 5: Segmentation result as provided by MSEG for scale parameter 400.

Comparing results between Figures 3, 5 and 7, shows that similar sized objects can be obtained by all 3 segmentation algorithms. For the scale parameter of 400, the MSEG seems to be less sensitive to spectral heterogeneity, but still more sensitive than eCognition's result with scale parameter 10. Both algorithms keep good alignment of object boundaries with the image edges and both provide usable over-segmentations of the initial image. The texture-based MSEG also provided good segmentation of the image, improving the simple profile result, but at the same time, working against the shape criterion for objects, providing less compact or smooth boundaries.

In both systems, MSEG (with or without texture) and eCognition, the thematic category boundaries are well respected by the segmentations.

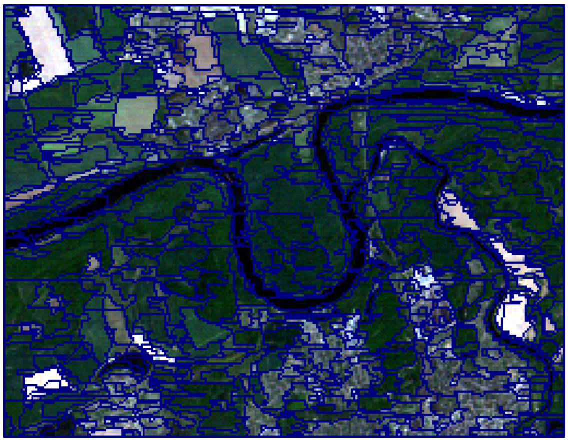

Figure 6: Segmentation result as provided by MSEG for scale parameter 700.

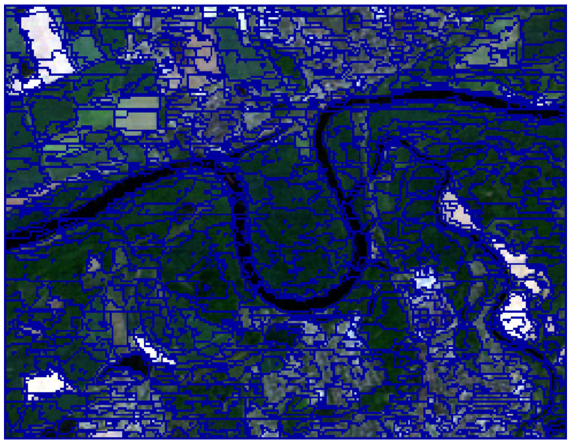

Figure 7: Segmentation result as provided by the texture-based MSEG for scale parameter 400 and texture parameter 2.0

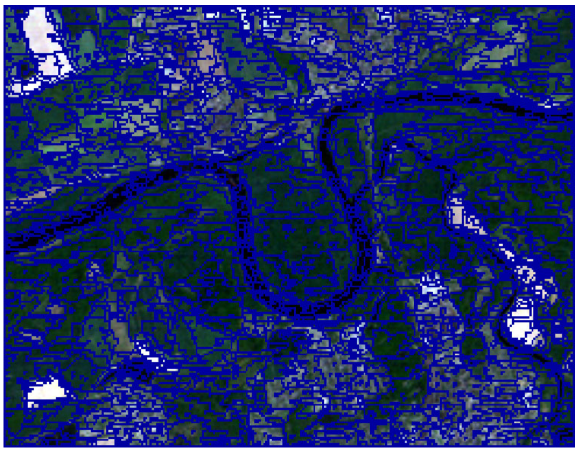

Figure 8: Segmentation result as provided by the texture-based MSEG for scale parameter 400 and texture parameter 1.0

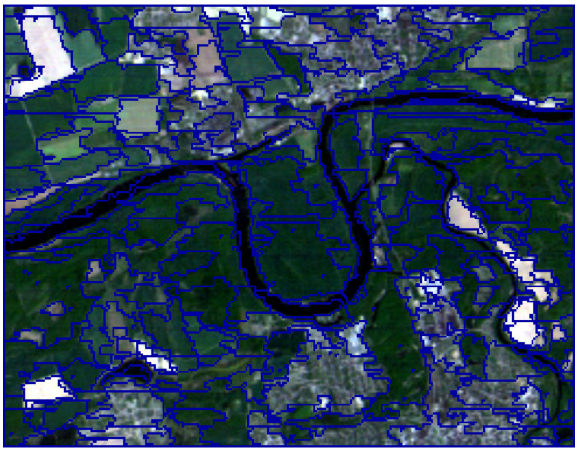

Figure 9: Segmentation result as provided by texture-based MSEG for scale parameter 2500 and texture parameter 3.0

In the last step of evaluation, segmentations were produced from the algorithms using larger scale parameters: eCognition's result for the scale value of 20 is presented in Figure 4 while MSEG's result for scale parameter of 700 is presented in Figure 6. A comparable result was produced by the texture-based MSEG when a very large (2500) scale parameter was used (so that the scale parameter would not significantly interfere with the final mean object size) and the texture parameter was set to 3.0 (Figure 9). The texture-based MSEG result was very good especially inside the urban areas, with complex texture patterns, where the object primitives were merged in such a way, so to provide larger homogenous objects than eCognition or the simple profile MSEG.

Finally in Figures 7 and 8, it can be observed that the segmentation result can be different when the texture parameter is changed. Smaller texture parameter provides smaller primitive objects. It should be noticed that texture free objects, like the very bright white areas in the image, don't get affected by the texture parameter change, as expected.

4. CONCLUSIONS AND FUTURE WORK

Overall, the proposed image segmentation algorithm, gave very promising segmentation results for remote sensing imagery. With the addition of the Advanced Texture Heuristic module, it was shown to be a good and generic segmentation solution for remote sensing imagery. The extracted boundaries of the primitive objects in each case were compatible with the semantic object edges. Thus, for object oriented image analysis, the texture-based MSEG is qualified as a successful low level processing algorithm.

MSEG has however some disadvantages that have to be further investigated. Its shape heterogeneity criteria are not quite effective and must be further tested and optimized to provide better results.

An integration of the MSEG algorithm with higher level artificial intelligence and pattern recognition methods, will be developed for full-scale object oriented image analysis.

5. ACKNOWLEDGEMENTS

The project is co-funded by the European Social Fund (75%) and Greek National Resources (25%) - Operational Program for Educational and Vocational Training II (EPEAEK II) and particularly the Program PYTHAGORAS II.

6. REFERENCES

Argenti F., L. Alparone and G. Benelli, 1990. Fast Algorithms for Texture Analysis Using Co-Occurrence Matrices. IEEE Proceedings, 137, F, 6, 1990, 443-448.

Baatz, M. & Schäpe, A. 2000. Multiresolution Segmentation – an optimization approach for high quality multi-scale image segmentation. *In: Strobl, J. et al. (eds.): Angewandte Geographische Infor-mationsverarbeitung XII. Wichmann, Heidelberg, pp. 12-23*.

Benz U., Hoffman P., Willhauck G., Lingenfelder I. and M. Heynen, 2004. Multi-resolution, object-oriented fuzzy analysis of remote sensing data for GIS-ready information. *ISPRS Journal of Photogrammetry and Remote Sensing 58* pp. 239-258.

Chen, J., Pappas, T.N., Mojsilovic, A., Rogowitz, B. 2002. Adaptive image segmentation based on color and texture. *Proceedings International Conference on Image Processing. Evanston, IL, USA, 777- 780 vol.3*

eCognition User Guide, 2005, Definiens, Munchen. http://www.definiens.com (accessed 15/06/2006)

Miyamoto E., T. Merryman. Fast Calculation of Haralick Texture Features. www.ece.cmu.edu/~pueschel/teaching/18-799B-CMU-spring05/material/eizan-tad.pdf (accessed 15/06/2006)

Fauzi, M. FA., Lewis, P. H. 2003. A Fully Unsupervised Texture Segmentation Algorithm. Harvey, R. and Bangham, J. A., Eds. *Proceedings British Machine Vision Conference 2003*, pages 519-528.

Haralick R.M., Shanmugan K. and I. Dinstein 1973. Textural features for image classification. IEEE Trans. On Systems, Man and Cybernetics, 3(6):610-621, Nov. 1973.

Haralick R.M.1979. Statistical and Structural Approaches to Texture. Proceedings of the IEEE, Vol. 67, No. 5, May 1979, pp. 786-804.

Havlicek, J. P., Tay, P.C. 2001. Determination of the number of texture segments using wavelets. *Electronic Journal of Differential Equations*, Conf. pp 61–70.

Liapis, S., Alvertos, N., Tziritas, G. 1998. Unsupervised Texture Segmentation using Discrete Wavelet Frames. *IX European Signal Processing Conference*, Sept. 1998, pp. 2529-2532

Materka A., M. Strzelecki 1998. Texture Analysis Methods – A Review, Technical University of Lodz, Institute of Electronics, COST B11 report, Brussels 1998

Pal, N. R., Pal, S. K. 1993. A review on image segmentation techniques. *Pattern Recognition*, vol. 26, pp. 1277-1294.

Sonka, M., Hlavac, V. Boyle, R., 1998. *Image Processing, Analysis, and Machine Vision* - 2nd Edition, PWS, Pacific Grove, CA, 800 p., ISBN 0-534-95393-X.

Schroder M., A. Dimai 1998. Texture Information in Remote Sensing Images: A Case Study. Workshop on Texture Analysis, WTA 1998, Freiburg, Germany.

Sukissian L., S. Kollias and Y. Boutalis 1994. Adaptive Classification of Textured Images Using Linear Prediction and Neural Networks. Signal Processing, 36, 1994, 209-232.

Tzotsos A. and D. Argialas, 2006. MSEG: A generic region-based multi-scale image segmentation algorithm for remote sensing imagery. Proceedings of ASPRS 2006 Annual Conference, Reno, Nevada; May 1-5, 2006.