SPT 3.1: A FREE SOFTWARE FOR AUTOMATIC TUNING OF SEGMENTATION PARAMETERS IN OPTICAL, HYPERSPECTRAL AND SAR IMAGES

P. Achanccaray[1], V.A. Ayma[1], L. I. Jimenez[2], S. Bernabe[2], P. N. Happ[1], G. A. O. P. Costa[1,3], R. Q. Feitosa[1,4], A. Plaza[2]

[1]Department of Electrical Engineering, Pontifical Catholic University of Rio de Janeiro, Brazil
[2]Department of Technology of Computers and Communications, University of Extremadura, Spain
[3]Department of Informatics and Computer Science, Rio de Janeiro State University, Brazil
[4]Department of Computer Engineering, Rio de Janeiro State University, Brazil

ABSTRACT

The Segmentation Parameter Tuner (SPT) is a tool designed for automatic tuning of segmentation parameters. In SPT, the goodness of a set of parameter values is given by the level of agreement between the segmentation result and a given reference (representing the desired outcome) quantified by a metric selected by the user (empirical discrepancy methods). This metric is used as the fitness function of an optimization algorithm that searches the parameter space for the minimum value, which is expected to correspond to the segmentation outcome most similar to the reference. SPT 3.1 offers many interesting features such as: five segmentation algorithms (for Optical, Hyperspectral and SAR images), four optimization algorithms (stochastic and direct search optimization methods) and seven discrepancy metrics (pixel and object-based). This paper describes the optimization procedure underlying SPT 3.1, the features added to this version as well as an experiment that illustrates the operation of the tool.

***Index Terms*—** Image Segmentation, Optimization, Hyperspectral Imaging, Synthetic Aperture Radar, Public domain software

1. INTRODUCTION

In the field of object based image analysis (OBIA), segmentation is a key step. Image segmentation splits an image into homogeneous regions – image segments – according to given homogeneity criterion. There are many approaches to accomplish this task and each implementation has its own parameters, which have to be properly tuned to the target application.

The selection of a set of parameter values for certain application is usually done manually through a time consuming, trial-and-error process, using some a priori knowledge. There is no guarantee that this procedure leads to the optimal set of values for two main reasons. First, the process relies on the subjective perception of a human user. Second, the relationship between input segmentation parameters and the segmentation outcome is far from being evident. Alternatively, automatic methods have been proposed to minimize human intervention. These rely on optimization algorithms to search the parameter space for the optimal set of values. These algorithms minimize a given fitness function that represents the level of disagreement between the segmentation outcome and a given set of reference segments provided by the user.

The *Segmentation Parameter Tuner* (SPT) is a free software tool that implements such strategy. SPT was firstly introduced in [1] and extensions have been added later on [2]. This paper describes SPT's main functions and focuses on the improvements brought by its newest version (SPT 3.1). Among the new features of SPT 3.1, it provides a larger number of segmentation algorithms, able to handle hyperspectral images, and a more efficient parallel implementation that assigns a thread for each specific task performed by SPT.

This paper is organized as follows. Section 2 describes how the automatic tuning of segmentation parameters works and how it is implemented in SPT 3.1. Section 3 provides an overview of the current functionalities available in SPT. Section 4 presents the enhancements embedded in this new version relative to the earlier one. Section 5 presents experimental results that illustrate SPT's outcome. Concluding remarks are made in Section 6.

2. AUTOMATIC TUNING OF SEGMENTATION PARAMETERS

Let's define $p_1, p_2, \dots, p_n$ as a set of parameters of a given segmentation algorithm. The goodness of a segmentation outcome is measured by a metric that expresses numerically how well a segmentation result fits a set of references provided by the user to represent the ideal segmentation for the target application and for the input image. This approach is often called empirical discrepancy [3]. The lower the metric's value is, the more similar is the segmentation

outcome to the reference. Thus, the problem of parameter tuning reduces to finding the set of parameters values which yields the minimum value of the selected metric; this set of values will be denoted henceforth as $\hat{p}$, formally:

$$\hat{p} = \underset{p_1, p_2, \dots, p_n}{\textbf{argmin}} \quad metric\big(Seg.Alg.(p_1, p_2, \dots, p_n)\big) \qquad (1)$$

where $metric$ is the selected metric and $Seg.Alg$ is a particular segmentation algorithm.

The methodology designed to solve this optimization problem is illustrated in Figure 1. First, the input image is segmented taking an initial set of parameter values. Later, the fitness function is calculated by comparing the segmentation result with the references provided by the user. This process is repeated iteratively, using new sets of segmentation parameters, until the minimum value or a convergence criterion (e.g., maximum number of iterations) is reached.

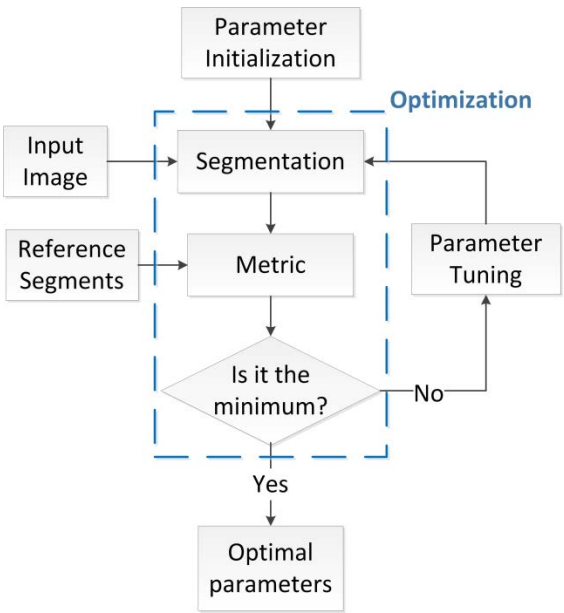

Figure 1. SPT optimization methodology.

3. SEGMENTATION PARAMETER TUNER

The SPT user interface is constituted by four different tabs with specific functionalities (see Figure 2) related to the automatic parameter tuning task. They are the *Reference Tool*, the *Optimization Tool*, the *Segmentation Tool* and the *Assessment Tool*.

The *Reference Tool* allows the user to create or load a set of references that represents the desired segmentation outcome. These references could be in raster or vector format. As a visual guide to delineate them, a three bands color composition can be selected by the user.

The *Optimization Tool* refers to the essential functionalities of SPT. In that tab an optimization algorithm, a segmentation algorithm and a discrepancy metric may be selected (see Figure 2), and the parameter tuning itself can be started. The number of iterations until convergence is achieved depends, among other factors, on the initial solution. Additionally, there is an option to save the intermediate results obtained in all iterations of the optimization procedure for further analysis.

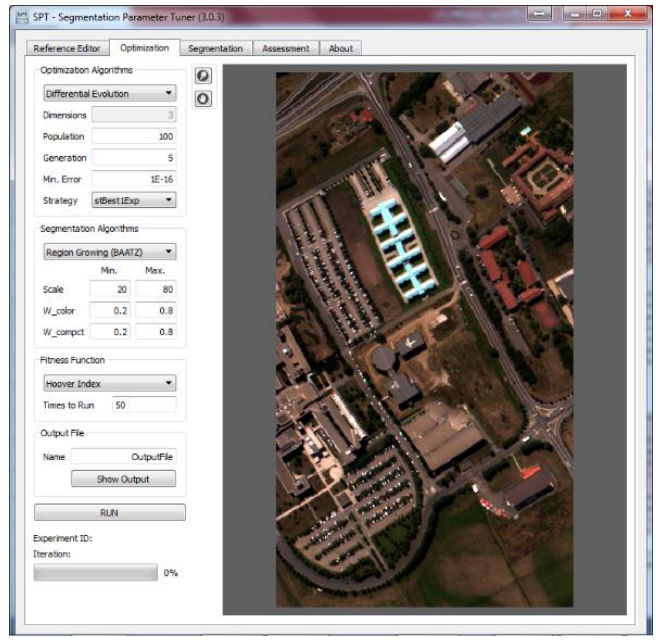

Figure 2. SPT 3.1 Graphic User Interface

SPT offers four optimization algorithms for parameter tuning: *Differential Evolution* [4], *Generalized Pattern Search* [5], *Mesh Adaptive Direct Search* [5] and *Nelder-Mead* [6]. Furthermore, it provides five segmentation algorithms. Four of them are designed for optical and hyperspectral images: *MeanShift* [7], *Graph-Based* proposed in [9] and the *Region Growing* algorithms proposed in [10] and [11]. The fifth one, called *MultiSeg* [12] was conceived primarily for SAR images, although it can also be used for optical images. One out of a set of seven metrics can be selected to express segmentation quality, namely *Hoover Index*, *Shape Index, Area-Fit-Index, Precision & Recall, Reference Bounded Segments Booster, Segmentation Covering and Rand Index* [1].

The *Segmentation Tool* provides five segmentation algorithms along with graphical support for definition of parameter settings. Moreover, the tool allows the user to load from a file a set of references, in raster or vector format, and to compute how well the segmentation outcome fits the set of references.

Finally, with the *Assessment Tool* one can compute any of the available metrics for a segmentation outcome (either obtained though SPT or through any other software) and a set of reference segments.

4. NEW FEATURES

One important improvement of SPT 3.1 relative to the previous versions refers to the number of segmentation algorithms for hyperspectral images. New versions of Mean Shift and Graph-Based algorithms able to handle hyperspectral images were added to this version of SPT.

MeanShift is a robust method for finding local extrema in the density distribution of a data set, in our case, a set of vectors each one containing the values of a single pixel through all its bands. The modified implementation of MeanShift in SPT 3.1 for hyperspectral imagery is based on EDISON's MeanShift [8].

The implementation of the Graph-Based segmentation algorithm added to this version, first developed by Felzenszwalb [9], was adapted to work with hyperspectral images.

In addition, the current version exploits parallel processing power available in multicore CPUs. Specific tasks, such as the optimization procedure, the graphic user interface (GUI) and image segmentation were encapsulated in different threads that run in parallel to facilitate the user interaction with the tool.

5. EXPERIMENTAL RESULTS

In order to illustrate the new functionalities of SPT 3.1, this section shows the results of an experiment that aims to find proper parameter values for the segmentation of a hyperspectral image. The image was produced by the ROSIS optical sensor over the urban area of the University of Pavia, Italy [13]. The image size is 610 × 340 (see Figure 3), with spatial resolution of 1.3 meters per pixel and 103 data channels (with spectral range from 0.43 μm to 0.86 μm). The experiment was carried out using the Nelder-Mead optimization algorithm, with the Multiresolution Segmentation, SPRING, MeanShift and Graph-based as segmentation procedures and Precision & Recall as the fitness function to be optimized. The references were created using the Reference Tool. They represent roofs of buildings (see Figure 3). The values of the fitness function obtained are shown in Table 1 and the segmentation outcomes corresponding to the optimal set of values are presented in Figure 4. Notice that for visualization purposes, only snips of the original image and the segmentation outcomes are shown.

The values of the fitness function obtained after the tuning procedure were close to zero for Multiresolution and Graph-based segmentation and greater than zero for SPRING and MeanShift. This behavior is related to the nature of each segmentation algorithm. Those who consider shape attributes and potential boundaries between segments led to better results (see Figure 4a and 4d) considering the references used in this experiment. Results obtained with segmentation algorithms based only on spectral attributes and/or pixel position led to poorer results (see Figure 4b and 4c).

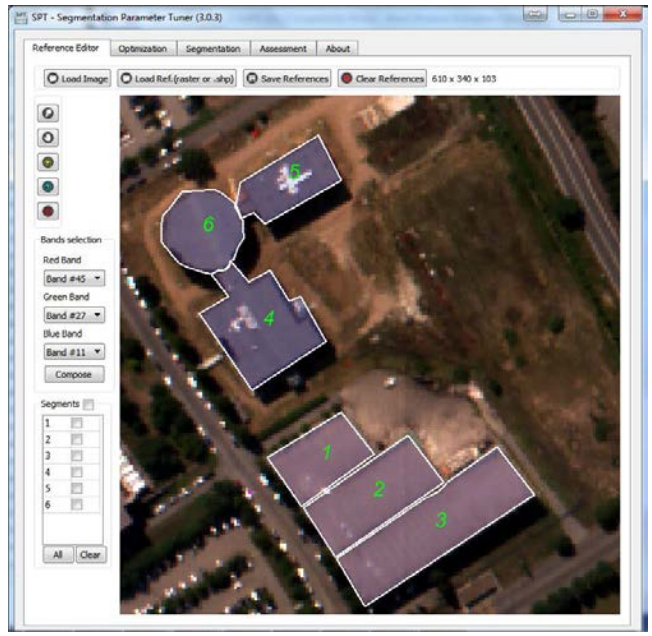

Figure 3. Set of references used in the experiments.

Table 1. Values of fitness functions obtained using SPT 3.1

Segmentation Algorithm	Fitness Function
Multiresolution	0.0258489
SPRING	0.192207
MeanShift	0.845313
Graph-based	0.0246502

It is worth mentioning that the segmentation is by itself a computational intensive task and its associated processing time grows with the size of the image being segmented. In SPT the selected segmentation algorithm is executed tens or even hundreds of times till the stop condition is achieved, what may take hours or even days depending on the image size. Thus, it is advisable to run SPT on a small portion of the image, where a sufficient number of representative references can be drawn.

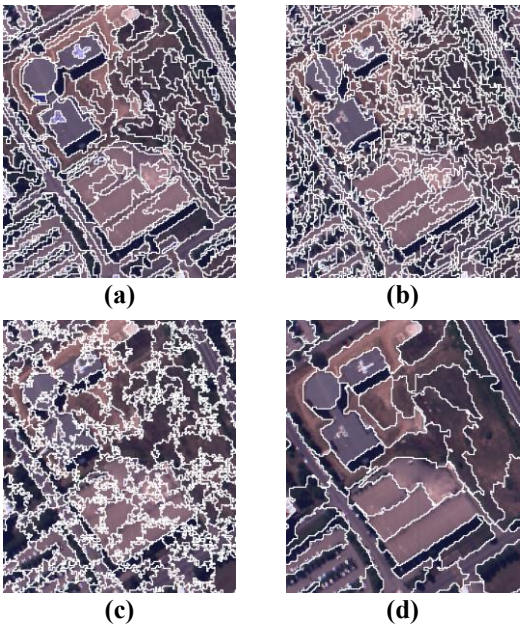

Figure 4. Segmentation outcomes corresponding to the optimal set of values found for different segmentation algorithms: (a) Multiresolution Region Growing; (b) SPRING; (c) MeanShift; and (d) Graph-based.

6. CONCLUSIONS

In this work, the Segmentation Parameter Tuner (SPT 3.1) was presented, which is a free tool designed to find optimal parameter values of segmentation algorithms according to a given reference. The software is available at: *http://www.lvc.ele.puc-rio.br/wp/?p=1403*.

This version has all the characteristics described in the paper, supporting the tuning of segmentation parameters for segmentation methods that can handle optical, hyperspectral and SAR images. Moreover, in this version the use of threads for specific tasks facilitate user interaction.

The new version of SPT is a convenient tool for future investigations, such as comparisons of distinct segmentation algorithms, quality metrics, and optimization methods.

7. ACKNOWLEDGMENTS

The authors acknowledge the support provided by CNPq (Conselho Nacional de Desenvolvimento e Pesquisa), CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior), FAPERJ (Fundação Carlos Chagas Filho de Amparo à Pesquisa do Estado do Rio de Janeiro) and FP7 (Seventh Framework Programme) in the scope of the TOLOMEO project.

8. REFERENCES

[1] Achanccaray, P.; Ayma, V. A.; Jimenez, L.; Garcia, S.; Happ, P.; Feitosa, R. Q.; Plaza, A., "A free software tool for Automatic Tuning of Segmentation Parameters," South-Eastern European Journal of Earth Observation and Geomatics, vol. 3, pp. 707-712, 2014.

[2] Achanccaray, P. M.; Ayma, V. A.; Jimenez, L. I.; Garcia, S. B.; Happ, P. N.; Feitosa, R. Q.; Plaza, A. SPT 3.0: A free software for automatic segmentation parameters tuning. In: Simposio Brasileiro De Sensoriamento Remoto, 17. (SBSR), 2015, João Pessoa, PB. Anais do 17mo Simpósio Brasileiro de Sensoriamento Remoto, João Pessoa,PB: INPE, 2015. p. 5578-5581. 1 DVD.

[3] Y. Zhang, "A survey on evaluation methods for image segmentation," Pattern Recognition, vol. 29, no. 8, pp. 1335-1346, 1996.

[4] Storn, R. & Price, K., 1997. Differential Evolution - A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. Journal of Global Optimization, Volume 11, pp. 341-359.

[5] Audet, C., 2012. A survey on direct search methods for blackbox optimization and their applications, Montreal, Canada: s.n.

[6] Nelder, J. & Mead, R., 1965. A Ssimplex Method for Function Minimization. The Computer Journal, 7(4), pp. 308-313.

[7] D. Comaniciu and P. Meer, "Mean Shift: A robust approach toward feature space analysis," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, pp. 603-619, 2002.

[8] Christoudias, C. M.; Georgescu, B. & Meer, P. Synergism in low level vision, 2002. Proc. 16th International Conference on Pattern Recognition, 2002, 4, 150-155.

[9] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," vol. 59, no. 2, pp. 167-181, 2004.

[10] P. Happ, R. Feitosa, C. Bentes and R. Farias, "A Region-Growing Segmentation Algorithm for GPUs," IEEE Geoscience and Remote Sensing Letters, vol. 10, no. 6, pp. 1612-1616, November 2013.

[11] Bin, L. S.; Fonseca, L. M. G.; Erthal, G. J.; Ii, F. M., "Satellite imagery segmentation: a region growing approach," in VIII Simpósio Brasileiro de Sensoriamento Remoto, Salvador, BA, 1996.

[12] M. A. S. Jr., "Segmentação multi-níveis e multi-modelos para imagens de radar e ópticas," Tese (Doutorado em Computação Aplicada), p. 131, 2007.

[13] Hypercomp Research Group, “Hyperspectral Repository”, 2014. [Online]. Available: www.hypercomp.es/repository/ [Accessed 2015].