Divide and Segment – An alternative for parallel segmentation

Thales Sehn Korting[1], Emiliano Ferreira Castejon[1], Leila M. Garcia Fonseca[1]

[1]National Institute for Space Research (INPE) – Image Processing Division (DPI)
12201-010 São José dos Campos – SP – Brazil

{tkorting,castejon,leila}@dpi.inpe.br

Abstract. *Remote sensing images with large sizes are usual. They also include several spectral channels, increasing the volume of information. To get valuable information from data automatically, computers need higher amounts of memory and efficient processing techniques. Segmentation is a key technique to deal with remote sensing. It identifies regions in images. Therefore, it deals with large amounts of information. Even with current computational power, some image sizes exceed the memory limits, which need different solutions. An alternative to overcome such limits is to employ divide and conquer strategy, splitting the image into tiles, and segmenting each one individually. However, arises the problem of merging neighboring tiles and keeping the homogeneity in such regions. In this work, we propose an alternative to create the tiles, by defining noncrisp borders between tiles, but adaptive borders for the tiles. By applying our method, we avoid the postprocessing of neighboring regions, and therefore speed up the final segmentation.*

1. Introduction

Segmentation in remote sensing is a challenging field. Techniques for segmentation are the first step in all analysis tasks. Their results are expected to describe the objects, allowing a deeper interpretation by experts or classification algorithms. The work of [Haralick and Shapiro 1985] defined segmentation as a way to separate the image into simple regions with homogeneous behavior.

In remote sensing, segmentation techniques are not new – see [Bins et al. 1996], [Câmara et al. 1996]. However, the field of GEOBIA (GEographic Object-Based Image Analysis) emerged recently. It makes a link between objects and radiometric properties [Blaschke 2010].

To partition automatically an image into regions, algorithms must consider the context, scale, neighborhood, meaning, and computational resources. However, according to [Wassenberg et al. 2009], good quality results often come at the price of high computational cost.

For example, the collection rate for IKONOS satellite is about 890 megapixels each minute [Dial et al. 2003]; for CBERS-2B is about 120 megapixels each minute. According to [Wassenberg et al. 2009], even a tuned sequential segmentation algorithm is far slower than these rates.

Remote sensing images often present large sizes. The variety of spectral channels, that in one side contain rich information about the land targets, in other side increases the volume of information. Even with current computational power, certain sizes exceed the memory limits, claiming new solutions. Methods based on divide and conquer strategy arise as an alternative for these limits. Such methods split the image into tiles, and segment each one individually. The problem of this approach is to merge neighboring tiles, keeping the homogeneity in these regions.

In this article we tackle the problem of creating tiles for parallel segmentation. After defining tiles, any algorithm can run on them in an independent way, since the implementation is based on multiprogrammed techniques. We argue that by defining noncrisp borders between tiles, we avoid the postprocessing of neighboring regions, and therefore speed up the final segmentation.

2. Related Work

Several techniques arise from "region growing" strategy, relying on the similarity of near pixels. [Bins et al. 1996] applied this approach in remote sensing images. Their method is based on the likeness between neighboring pixels and the smallest area allowed for a region.

[Baatz and Schape 2000] is another example of region growing technique. In this approach, the algorithm minimizes the average heterogeneity of the regions. The heterogeneity balances the object's smoothness and compactness, resulting in more regular objects. It deals with the standard deviation of pixels for each band as well. We suggest the reading of [Meinel and Neubert 2004] for a comparison of these two algorithms and alternatives for remote sensing segmentation.

According to [Lenkiewicz et al. 2009], parallel architectures are becoming a standard for handling complex operations that need significant computational power. Large size images include medical data sets of magnetic resonance imaging (MRI) [Prassni et al. 2010], or remote sensing hyperspectral and multitemporal images [Valencia et al. 2008, Plaza et al. 2011]. Therefore parallel algorithms arise as an alternative to overcome these limits. Such methods usually split the image into tiles, and segment each one individually. A postprocessing step is necessary to couple bordering regions.

The tiles often have regular sizes to be assigned equally among the processors [Bader et al. 1996]. However, according to [Wassenberg et al. 2009] this is not acceptable because border objects are not correctly handled. A common solution is to adopt overlapping tiles, which is also inadequate because there is no upper bound on the size of objects of interest (e.g. rivers or roads).

The work of [Singh et al. 1999] proposed a parallel method for the seeded region growing algorithm ([Adams and Bischof 1994]), based on spreading seeds in different processes, each one growing in parallel. The authors needed to deal with simultaneous access for the same pixels. To avoid this problem, images were divided in square windows, employing a postprocessing step to join regions.

[Happ et al. 2010] employed the traditional parallel segmentation. Each tile was processed by a different thread, through a sequential algorithm [Baatz and Schape 2000]. This method falls on the same problem of treating boundary segments. The number of boundary objects, which depends on the image, can be prohibitive in certain cases. Therefore this paper deals with adaptive tiles, minimizing the problem of boundary regions.

In the area of merging and mosaicking, [Bagli and Fonseca 2006] presented a technique for image blending, based on multi-resolution decomposition. The authors defined a cut line, considering texture information from overlapping regions of mosaicking images. The method found automatically the transition zone size and the cut line on satellite and aerial images.

3. Method

Traditional parallel schemes first divide the image into crisp tiles, and then treat bordering regions. Some creates tiles with overlapping regions, but fall into the same problem of postprocessing. We propose to create adaptive tiles using non crisp borders. Figure 1 shows our approach.

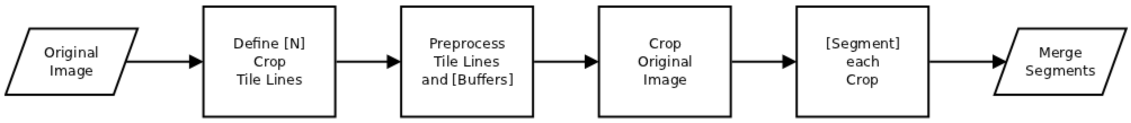

Figure 1. Main diagram from our technique. User defined parameters are in [braces].

We suggest to first define the crisp tiles, and analyze them to adapt the local features, in a strategy of presegmentation. We use two basic parameters , a maximum neighborhood for each pixel in the line ($t1$), transverse to the tile line, and a buffer around the original tile line ($t2$). Figure 2 depicts the approach, where the original tile line is defined as the middle line between the buffer lines. Let the neighborhood of each pixel be called profile. The algorithm is described as follows:

```
1. get pixel in tile line and its profile
2. find border
3. change the tile position to this border
4. assign next pixel to the same border position
5. if there are more pixels in the tile line, back to 1
6. crop image using new tile line
```

To detect the edges, our strategy calculates the first slope of each profile. The maximum absolute value of the slope for each profile points a border. However, alternative methods to detect borders in the profiles can be employed. In this paper, we describe and test only horizontal tile lines. Although the same scheme can be applied if tiles contain vertical lines as well. After adapting the tiles, individual and parallel methods segment each tile. The result is the merging from all tiles.

Noisy tile lines can be created, depending on the right choice of $t1$ and $t2$. The buffer size $t2$ can be defined based on the parameters of average region size from the segmentation algorithms. The choice of proper values depends on the image scale and the dimension of the objects of interest.

4. Results and Discussion

To apply our method, we used the algorithm of region growing [Bins et al. 1996] available at TerraLib C++ library [Câmara et al. 2008]. This library presents parallel methods for segmentation, dividing the image into crisp tiles, and merging neighbors afterwards.

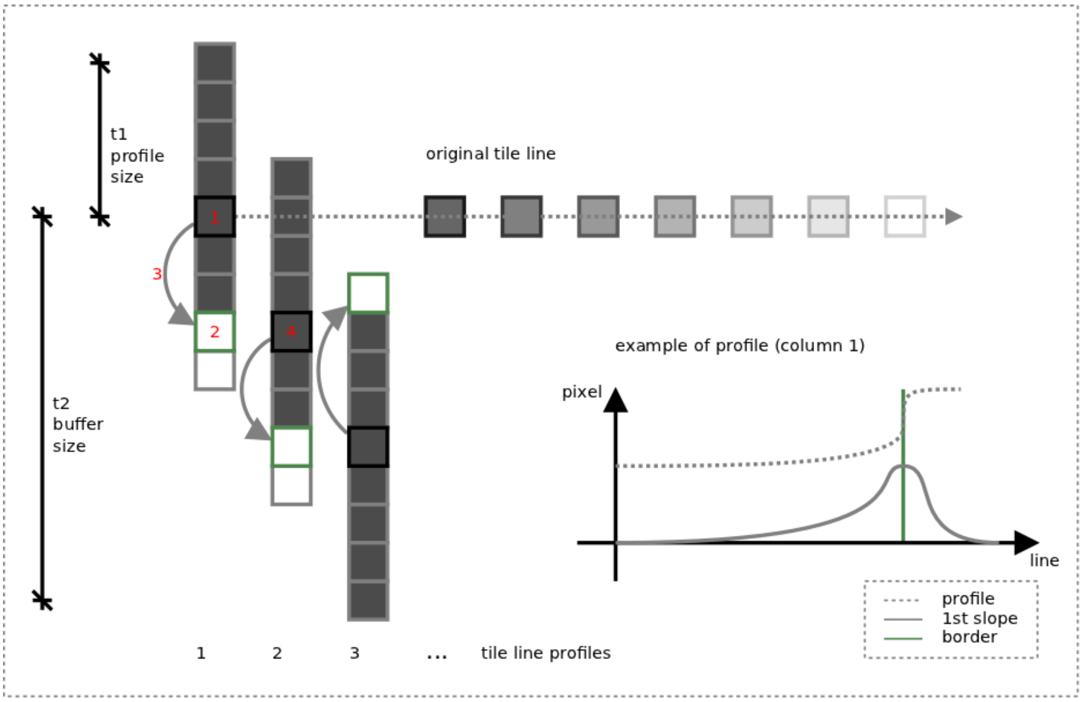

Figure 2. Scheme for adapting the tile lines. Steps 1 to 4 are highlighted (red).

The sequential algorithm (without dividing the image into tiles) is available as well. We compared our approach to the parallel algorithm available in TerraLib, using the same parameters. We used 3 images with different contexts to evaluate the method.

The first example shows a Quickbird image from São José dos Campos, Brazil, obtained in 2005. The size of the image is 1024×1024 pixels. The parameters used to adapt the tile line were $t1 = 8$ and $t2 = 20$. The adapted tile line is shown in Figure 3. The segmentation parameters (see [Bins et al. 1996]) were minimum area of 200 pixels, and Euclidean distance of 80 pixels. To compare the results of the segmentation, Figure 4 shows both approaches (adapted tile line, and crisp tile line with postprocessing). It is possible to see that our approach reduced crispy resultant objects.

The second image is a crop (1000×1000 pixels) of a region in the state of Bahia, Brazil, using satellite CBERS-2B, instrument HRC[1]. The pixels of this image have $2.7m^2$. The choice of parameters was $t1 = 5$ pixels, and a buffer of 100 pixels ($t2$). The adapted tile line is shown in Figure 5. The parameters were minimum area of 50 pixels, and Euclidean distance of 14 pixels. To compare the results of the segmentation, Figure 6 shows both approaches (adapted tile line, and crisp tile line with postprocessing). In this case the tile line has adapted to the river, allowing a better result of segmentation than to use crisp tiles.

The third image is a crop of a Quickbird scene from São Paulo, Brazil, with 1000×1175 pixels. The pixels of this image have $1m^2$. The choice of parameters was $t1 = 6$ pixels, and a buffer of 70 pixels ($t2$). The adapted tile line is shown in Figure 7. The parameters were minimum area of 300 pixels, and Euclidean distance of 50 pixels. To compare the results of the segmentation, Figure 8 shows both approaches (adapted tile

[1]Free remote sensing imagery at `http://www.dgi.inpe.br/CDSR/`.

Figure 3. The adapted tile line (yellow) for the first example. Red and Green lines shows the buffer size (parameter t2), and the Blue line shows the original tile line.

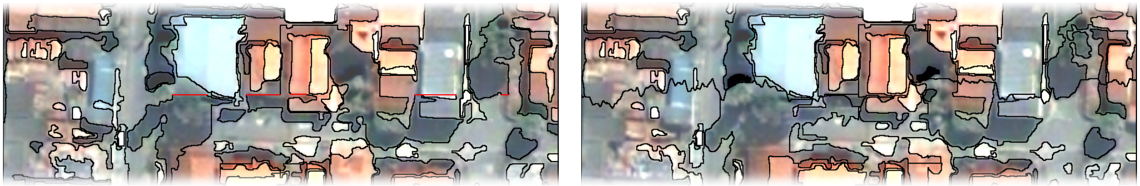

Figure 4. Comparison of approaches for Quickbird scene. Left is the segmentation using crisp tiles. Right is our approach using adapted tile lines.

line, and crisp tile line with postprocessing). One can note the presence of certain crisp polygons, which couldn't be merged due to their spectral differences. However using our alternative, the segmentation achieved a smoothest result between the tiles.

5. Conclusion

This article tackled the problem of defining tiles for parallel segmentation. Current methods create crisp tiles, needing postprocessing steps to get final regions. In certain cases, such methods create inconsistent objects, demanding computational power to deal with bordering regions. Postprocessing detects the bordering regions, and test the best combination of regions to merge. This step aims to keep the consistence of the new regions to specific segmentation parameters, as spectral homogeneity and size.

From the results it is clear to see that new tile borders remain at the end of segmentation. For most of the cases this is the expected result, and should not influence the overall segmentation. However, the shape of certain regions near the adaptive tile lines will not split image targets properly. Therefore dealing with such problem still remain as an open problem, currently unsolved by this method. Future works also include initialization issues, since our method begins in the leftmost pixel. Since this method is extendable to vertical tile lines, next steps also include testing images with horizontal and vertical tile lines. Besides, we aim at defining automatically the parameters ($t1$ and $t2$) based on the specific segmentation parameters.

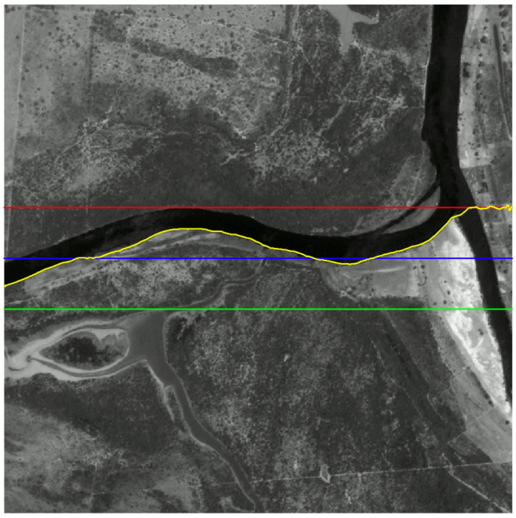

Figure 5. The adapted tile line (yellow) for the second example. Red and Green lines shows the buffer size (parameter t2), and the Blue line shows the original tile line.

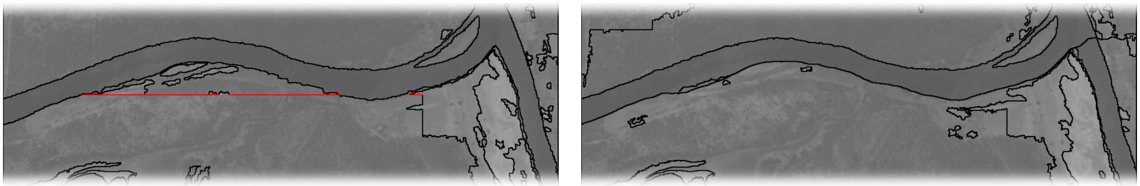

Figure 6. Comparison of approaches for CBERS-2B HRC. Left is the segmentation using crisp tiles. Right is our approach using adapted tile lines.

This work is a first step towards the definition of a method for creating adaptive tiles, developed for the main purpose of dealing with remote sensing images with large sizes. By defining noncrisp borders between tiles, our method avoided the processing of neighboring regions, creating adaptive tiles. The algorithm runs with a complexity of $O(n)$, and was developed using the TerraLib library, in C++.

References

Adams, R. and Bischof, L. (1994). Seeded region growing. *Pattern Analysis and Machine*, 16.

Baatz, M. and Schape, A. (2000). Multiresolution Segmentation: an optimization approach for high quality multi-scale image segmentation. In Wichmann-Verlag, editor, *XII Angewandte Geographische Informationsverarbeitung*, Heidelberg. Herbert Wichmann Verlag.

Bader, D., Jaja, J., Harwood, D., and Davis, L. (1996). Parallel algorithms for image enhancement and segmentation by region growing with an experimental study. *Proceedings of International Conference on Parallel Processing*, pages 414–423.

Bagli, V. and Fonseca, L. (2006). Seamless mosaicking via multiresolution analysis and cut line definition. In *Signal and Image Processing*. ACTA Press.

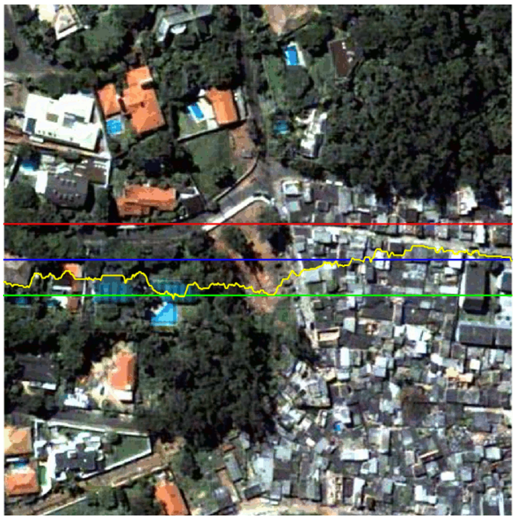

Figure 7. The adapted tile line (yellow) for the third example. Red and Green lines shows the buffer size (parameter t2), and the Blue line shows the original tile line.

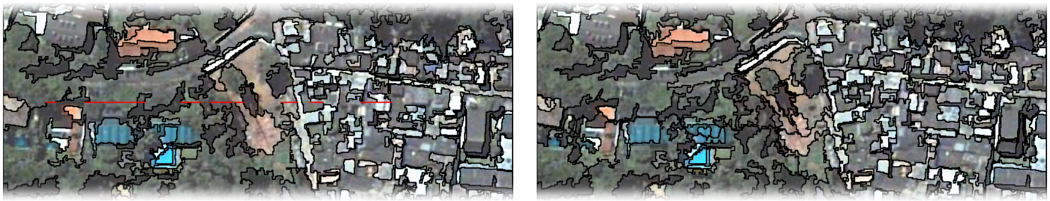

Figure 8. Comparing approaches for the third image. Left is the segmentation using crisp tiles. Right is our approach using adapted tile lines.

Bins, L., Fonseca, L., Erthal, G., and Li, F. (1996). Satellite imagery segmentation: a region growing approach. *Brazilian Remote Sensing Symposium*, 8.

Blaschke, T. (2010). Object based image analysis for remote sensing. *ISPRS Journal of Photogrammetry and Remote Sensing*, 65(1):2–16.

Câmara, G., Souza, R., Freitas, U., Garrido, J., and Li, F. (1996). Spring: Integrating remote sensing and gis by object-oriented data modelling. *Computers and Graphics*, 20(3):395–403.

Câmara, G., Vinhas, L., Ferreira, K., Queiroz, G., Souza, R., Monteiro, A., Carvalho, M., Casanova, M., and Freitas, U. (2008). TerraLib: An open source GIS library for large-scale environmental and socio-economic applications. *Open Source*, pages 247–270.

Dial, G., Bowen, H., Gerlach, F., Grodecki, J., and Oleszczuk, R. (2003). IKONOS satellite, imagery, and products. *Remote Sensing of Environment*, 88(1-2):23–36.

Happ, P., Ferreira, R., Bentes, C., Costa, G., and Feitosa, R. (2010). Multiresolution segmentation: a parallel approach for high resolution image segmentation in multicore architectures. In *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*.

Haralick, R. and Shapiro, L. (1985). Image segmentation techniques. *Applications of Artificial Intelligence II., 1985,*, 548:2–9.

Lenkiewicz, P., Pereira, M., Freire, M., and Fernandes, J. (2009). A new 3D image segmentation method for parallel architectures. *2009 IEEE International Conference on Multimedia and Expo*, pages 1813–1816.

Meinel, G. and Neubert, M. (2004). A comparison of segmentation programs for high resolution remote sensing data. *International Archives of Photogrammetry and Remote Sensing*, 35(Part B):1097–1105.

Plaza, A., Plaza, J., Paz, A., and Sanchez, S. (2011). Parallel Hyperspectral Image and Signal Processing. *IEEE Signal Processing Magazine*, 28(May):119–126.

Prassni, J., Ropinski, T., and Hinrichs, K. (2010). Uncertainty-aware guided volume segmentation. *IEEE transactions on visualization and computer graphics*, 16(6):1358–65.

Singh, D., Heras, D., and Rivera, F. (1999). Parallel Seeded Region Growing Algorithm. In *VIII Simposium Nacional de Reconocimiento de Formas y Análisis de Imágenes*, Bilbao, Spain.

Valencia, D., Lastovetsky, A., O'Flynn, M., Plaza, A., and Plaza, J. (2008). Parallel Processing of Remotely Sensed Hyperspectral Images On Heterogeneous Networks of Workstations Using HeteroMPI. *International Journal of High Performance Computing Applications*, 22(4):386–407.

Wassenberg, J., Middelmann, W., and Sanders, P. (2009). An Efficient Parallel Algorithm for Graph-Based Image Segmentation. In *Computer Analysis of Images and Patterns*, pages 1003–1010. Springer.