Improvements of the divide and segment method for parallel image segmentation

Anderson Reis Soares[1], Thales Sehn Körting [1], Leila M. Garcia Fonseca [1]

[1]Instituto Nacional de Pesquisa Espaciais (INPE)
Caixa Postal 515 – 12.245-970– São José dos Campos – SP – Brazil

{anderson.soares, thales.korting, leila.fonseca}@inpe.br

Abstract. *Remote Sensing is an important source of information about the dynamics of Earth's land and oceans, but retrieve information from this technique, is a challenge. Segmentation is a traditional method in remote sensing, which have a high computational cost. An alternative to suppress this problem is use parallel approaches, which split the image into tiles, and segment each one individually. However, the divisions among tiles are not natural, which create inconsistent objects. In this work, we extended our previous work, which used non-crisp borders computed based on graph-theory. By applying this non-crisp line cut, we avoid the post-processing of neighboring regions, and therefore speed up the segmentation.*

1. Introduction

Remote Sensing is an important source of data, spatial programs, such as Landsat, that collected more than 4 million images of the Earth's surface over 40 years, are important sources to understand the dynamic of land (Bolch et al., 2010). However, the development of methods to process and analyze this data, even with current computational power, is a challenge.

Image segmentation is a traditional method in remote sensing, which demands a lot of computational power and is widely used, especially more recently with the emergence of the Geographic Object-Based Image Analysis (GEOBIA). According to Körting et al. (2013) GEOBIA firstly identify regions in the image using segmentation, then extract neighborhood, spectral and spatial descriptive features and afterwards combine regions and features for object classification. However, these elements also turn GEOBIA a complex method because the difficulties related to image segmentation (Pinho et al., 2008) and the many different methods needed to model patterns (Hay and Castilla, 2008).

Segmentation is a fundamental problem in all image-processing applications. Soille (1999) defined segmentation as a process to split an image grouping the pixels by a similar attribute, such as the gray level, so the line which splits the areas, ideally, must be an edge. Gonzalez and Woods (2008) defined an edge, as a region where the intensity of pixels varies abruptly.

The results of segmentation must create uniform areas, which allow a simpler interpretation by the users and simpler representation for classification algorithms. Because of this, algorithms must consider the context, scale, neighborhood, meaning, and computational resources, and for that, this technique demands certain computational power (Körting et al., 2011). Because of this, parallel approaches became an alternative to suppress this computational cost.

Parallel architectures have become quite popular for image analysis applications (Seinstra and Koelma, 2004). The parallel implementation of segmentation divides the process into different threads. For that, the image is split into tiles, usually using crisp lines. However, this may generate inconsistent objects since the divisions among tiles are not natural.

In our previous work (Körting et al., 2013), we proposed to create non-crisp borders between the image, using an algorithm based on the graph theory to find the best line cut over an edge image, obtained using the magnitude of gradient image. In this article, we extended this approach, to find optimal line cuts in both horizontal and vertical directions, using directional high-pass directional filters and low-pass filter. With this combination, blurred borders are created thus minimizing the occurrence of inconsistent objects.

2. Related Work

Usually, the images are split with crisp lines, however, according to Wassenberg et al. (2009), this is not acceptable because border objects are not correctly handled, and for that, inconsistent segments are created. So, it is needed to merge the segments after segmentation, to combine the tiles and recreate the full image. However, other problems are created; one of them is merge the neighboring blocks without prejudicing the homogeneity in bordering regions. The second problem is the reproducibility of the results (Happ et al., 2010; Körting et al., 2013).

Happ et al. (2010) employed the traditional parallel segmentation, using multithreading parallel implementation of a region-growing algorithm proposed originally by Baatz and Schape (2000). The use of crisp lines imply a post-processing step to treat the boundary segments.

Different approaches have been proposed for solving the problem of splitting an image without using crisp borders. According to Basavaprasad and Hegadi (2012), based on the graph theory the image elements are better structured, and because of this, solve the image problems became more simple and the computation more efficient. However, this approach increases the amount of data to be handled, but it has several attractive properties as highlighted by Felzenszwalb and Huttenlocher (2004). One of them is the possibility to use minimum cost algorithms to find the best path between two nodes. In this case, it is possible to speed-up the process of finding the best cutting line.

Brejl and Sonka (1999) proposed a method for image segmentation, in which the borders are detected automatically based on learning. In this graph-based optimal border detection method, the features were selected from a predefined global set using radial-basis neural networks.

Shi and Malik (2000) proposed an approach that treats the segmentation process as a graph-partitioning problem, through an adjacency matrix connecting all pixels of the image, and use it to perform the partitioning of the graph using normalized cuts.

Körting et al. (2013) also proposed an approach based on the graph theory. The authors proposed to divide the image into adaptive tiles, were the borders of tiles were built along the line of the maximum magnitude of the gradient image. The strategy to find the adaptive tiles used the Dijkstra's algorithm, which is a graph-based approach to the design of image processing operators based on connectivity. This method considers one image as a directed graph whose nodes are the image pixels and whose arcs are the neighboring pixel pairs. This way, the division lines should follow the natural border of the segments, in most of the cases.

Lassalle et al. (2015) proposed a different approach. The graph is not used to find the best cut line, but to perform a scalable tile-based framework for region-merging algorithms. With such techniques, the authors expected to obtain identical results, with respect to processing the whole image at once.

3. Method

As an alternative for the traditional parallel method, we extended our previous method (Körting et al. 2013) which create adaptive tiles, based on the magnitude of the gradient image, hereby called *previous approach.* In this work, trough directional high-pass filters combined with a mean filter, hereby called *our approach*, our algorithm improved in finding the cutting line. Figure 1 shows the workflow.

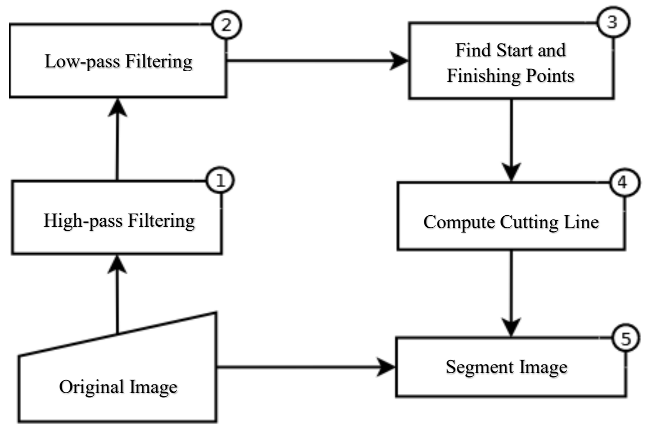


Figure 1. Scheme for segmenting images using tiles.

3.1. High-pass filtering

The first edge image was obtained using the *previous approach*. According to Gonzalez and Woods (2009), the gradient of pixels, is computed as the two-dimensional column vector, which indicates, for each pixel, the intensities of the border in horizontal and vertical directions (Equation 1). The magnitude of this vector points out the border's strength, Equation 2.

$$\nabla f = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} \quad (1)$$

$$mag(\nabla f) = \left[\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial x}\right)^2\right]^{\frac{1}{2}} \quad (2)$$

Another way to obtain the edges in an image is using directional filters. This type of filter enhance the edges in specific directions. Numerous filters has been proposed, such as Roberts, Sobel and Prewitt (Prewitt, 1970). In this work, we applied the Prewitt high-pass filters, since it produces less noise them others, as highlighted by Schowengerdt (2007). When applying a south filter (shown in Figure 2) or a north filter, the edges are enhanced on south or north directions, respectively, so it is easier for the algorithm to find the best path on horizontal direction.

$$\begin{bmatrix} -1 & -1 & -1 \\ 1 & -2 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Figure 2. Prewitt South Filter applied to images using a 3x3 neighborhood.

3.2 Low-pass Filtering

As can be seen on the example image, Figure 3, the high-pass filter normally highlights local maximum in the image. To follow a path in some edge image, we should consider not only the local maximum, but also neighboring pixels to avoid holes in the path, as occurred on our *previous approach*. In *our approach*, we fix those points using a low-pass filter, such as the mean filter, in this edge image obtained from the high-pass filter, Figure 3 (c). As can be seen on Figure 3 (d), the local maximum pixels are now blurred by the low-pass filter, because of this, the algorithm have new possibilities, and the chances of fall in a path with holes are reduced**.**

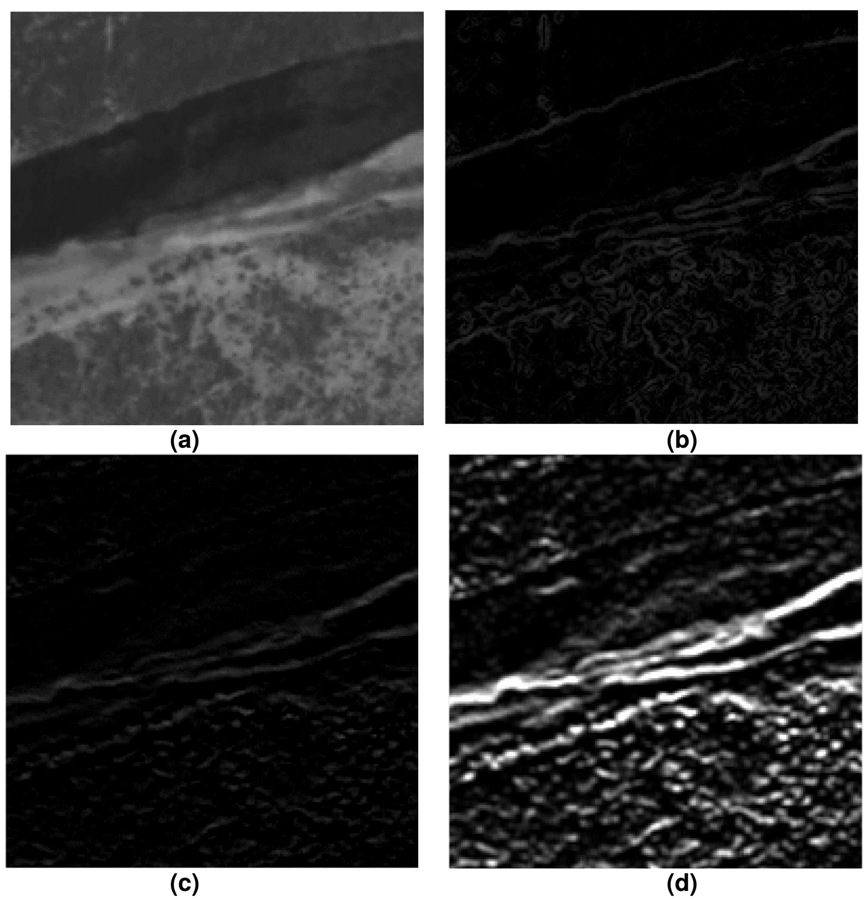


Figure 3. a) Original image, b) gradient image, c) directional high-pass image, and d) the previous result (c) with a low-pass filter.

In Figure 4 we show the histogram of the edge image obtained from both approaches. The variations on the image using *our approach* are more evident. Therefore, it is easier to find the best path. With this edge image, blurred by the low-pass filter, it is possible to create an adjacency matrix to find the optimal cutting line.

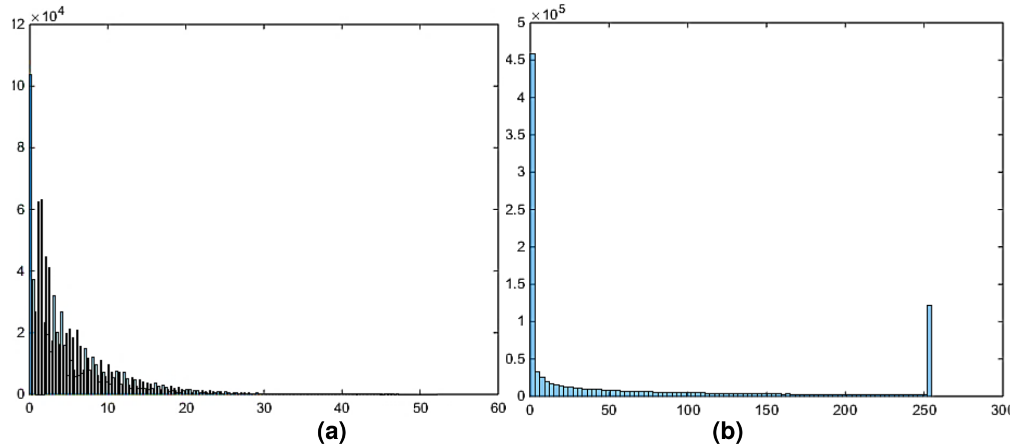


Figure 4. a) Histogram from edge image of *previous approach*, b) Histogram from edge image of *our approach* (b).

3.3. Find Starting and Finishing Points

The choice of the border is based on the pixel value. The starting point is the pixel with the highest value on the first column (or line, in case the cutting line is vertical). In case there are more than one pixel, the first on the row (or column) is assumed as the starting point. Same process is used to define the finishing point.

3.4 Compute Cutting Line

After define the starting and finishing points, the adjacency matrix is created. According to Körting et al. (2013) the adjacency matrix is a graph, whose nodes are the image pixels and whose arcs are an adjacency relation between pixels. The adjacency between the pixels is defined by five connections, including top, top-right, right, bottom-right and bottom pixels, as shown in Figure 5.

The arcs between nodes are weighted according to the pixel value. If one pixel is not a border, the distance, or the value of arc, is assigned as infinite, this way the algorithm should not go through this arc. When the adjacency pixel is a border, the distance assigned to the arc is minimum, therefore, the algorithm should go through them. With this graph is possible to apply an algorithm to compute the minimum cost between the starting and finishing points. As previously mentioned, we have employed the Dijkstra's algorithm (Dijkstras, 1959).

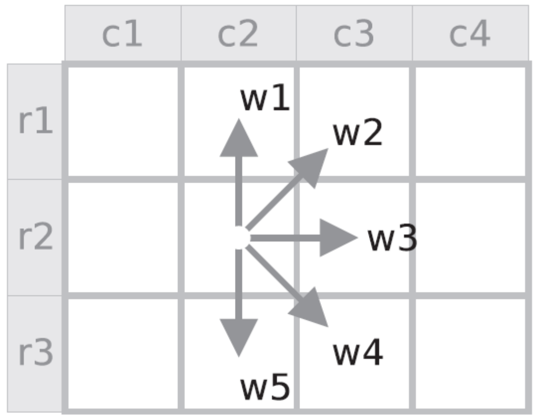


Figure 5. The five weights associated to each pixel to build the adjacency matrix. (Körting et al., 2013).

As the graph demands a lot of computational power, compute the cutting line over all image became a bottleneck, so is necessary create blocks to limit the size of the graph. Using this strategy, the candidate line will be created based on shortest path between starting and finishing edges over each blocks. To avoid tiles with very different sizes, must be defined a region of interest, a buffer zone, limited by a maximum displacement between the candidate line. This line define the non-crisp border between the tiles, and is used to split the original image.

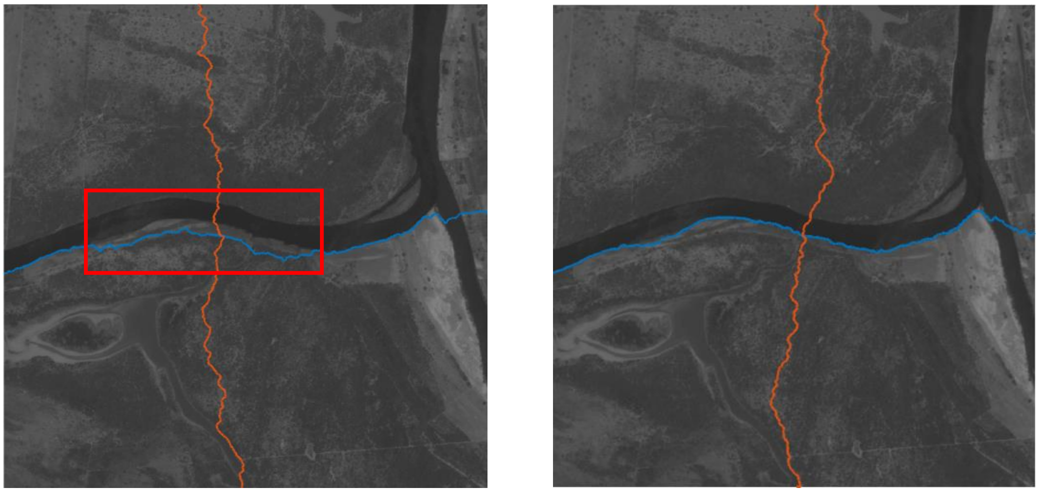
3.5. Segment Image

After computing the cutting lines, they are used to create the tiles. Each thread of a parallel segmentation scheme receive a tile, which will be segmented individually through the multiresolution segmentation algorithm (Baatz and Schäpe, 2000). Finally, after the segments are created, they are merged, creating the result, which does not need a post-processing step.

4. Results

For evaluation of the algorithm, we used three images with different contexts. The first image is a crop (1000 × 1000 pixels) of a region in the state of Bahia, Brazil, obtained by sensor HRC of satellite CBERS-2B. The second image is a crop of a Quickbird scene from São Paulo, Brazil, with 1000 × 1175 pixels. The pixels of this image have a spatial resolution of 0.6m. The third image is a crop of a WorldView-2 scene (2400 × 3200 pixels), of a region in the city of São José dos Campos, Brazil. This image has a spatial resolution of 0.5m.

All tests were performed using equivalent parameters, on all edge images. On the first experiment, Figure 6, the cutting line created using the *previous approach*, and *our approach*, followed different paths, for both directions. As can be observed on Figure 6 (a), the cutting line using *previous approach* do not followed the edge of the river, which we consider the ideal path for this image, this mistake was caused by a maximum local pixel, which was fixed using *our approach* as can be seen of Figure 6 (b). On the vertical direction, the path followed was different, but on this direction, there are not major feature to easily split the image.



(a) (b)

Figure 6. Cutting lines created using a) *previous approach* and b) *our approach*.

On the second experiment, Figure 7, the cutting line produced by both approaches were different. Using the *previous approach*, the line cut gone through a building area, this result divided some buildings in two parts as highlighted in Figure 7 (a). Using *our approach* was obtained a better cutting line, in most parts of image. At the building area, there were not major cuts on buildings, despite in some points the line did not gone through a path we consider ideal, following the street on the right side of the image.

(a) (b)

Figure 7. Cutting lines created using a) *previous approach* and b) *our approach*.

However, problems occurred over trees, which must be caused by the shadows over the canopy, this variation created the edges. Another issue in *our approach* is caused the intersection of cutting lines, which may create inconsistent objects, depending on the segmentation.

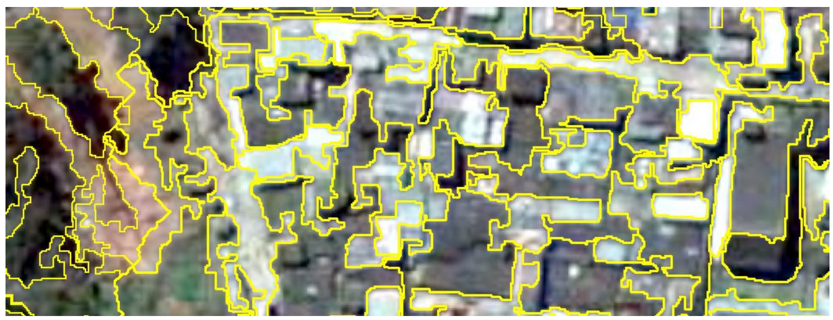
On the third experiment, Figure 8, the cutting line using *our approach* split the objects in image without major problems. Even passing near trees, the cutting lines gone to almost an optimal path, only one building, on horizontal direction, were split, which was caused by shadows. On the vertical direction, again, the cutting line split one building (as highlighted in the image), which was caused by the difference on illumination on the roof.

Figure 8. Cutting line created using *our approach*.

To analyze how the non-crisp lines effects on segmentation, we compared the results of a segmentation on tiles using crisp lines, which demands a post-processing step to merge neighboring regions obtained by different tiles, and *our approach*. Ideally, the segments created through segmentation should create homogeneous areas, however, with crisp lines the segments generated sometimes do not satisfy this condition, as exhibit on Figures 9 (a) and 10 (a). On Figure 9 the post-processing in those step did not merged some segments (note the roofs on center left of image and the soil area on the left side of image).

On Figure 10, the piece roof in the center of image was too small to create an individual region, therefore, it was merged in the region containing trees. Due to this problem, the bottom region with the rest of the roof was not merged, because the spectral difference between these two regions is too high.

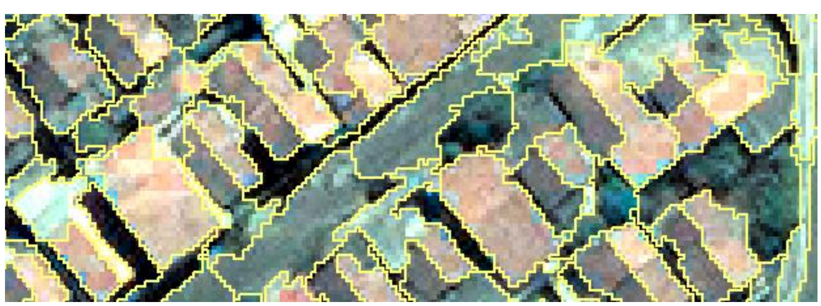
(a)



(b)

Figure 9. The intersection of tiles. Results using a) *crisp lines* and b) *our approach*.

(a)



(b)

Figure 10. The intersection of tiles. Results using a) *crisp lines* and b) *our approach*.

The presence of those crisp lines on the segments may generate some problems to classify the images, since the metrics used to characterize the objects and perform the classification are based on characteristics of the segments.

5. Conclusion

In this article we presented improvements to the divide and segment approach (Körting et al. 2013) to parallel image segmentation. The *previous approach* used the gradient image to compute the cutting line, but local maximum points influenced the results. We solved this problem in *our approach* using Prewitt directional high-pass filter combined with a low-pass filtering, with this strategy the edges are blurred and the creating new possibilities to the algorithm to find the best path.

However, in some cases the intersection of cutting lines may create some inconsistent objects, caused by the shape of lines. Dealing with these regions remains an open problem, currently unsolved by our approach. Future works include solving the problem of inconsistent objects caused by the intersection of cutting lines, and implement this approach on TerraLib library (Câmara et al. 2008).

References

Baatz, M. and Schape, A. (2000). Multiresolution Segmentation: an optimization approach for high quality multi-scale image segmentation. In: Angewandte Geographische Informationsverarbeitung, 12, 2000, Wichmann-Verlag. Proceedings... p. 12-23, WichmannVerlag: Heidelberg, 2000.

Basavaprasad, B. and Hegadi, R. S. (2012). Graph theoretical approaches for image segmentation. Aviskar–Solapur Univ Res J, 2, pp. 7-13.

Bolch, T., Menounos, B. and Wheate, R. 2010. Landsat-based inventory of glaciers in western Canada, 1985–2005. *Remote Sensing of Environment*, Vol. 114, No. 1, pp. 127–137. doi: 10.1016/j.rse.2009.08.015.

Brejl, M. and Sonka, M. (1999). Medical Image Segmentation: Automated Design of Border Detection Criteria from Examples. *Journal of Electronic Imaging*, v. 8, pp. 54-64.

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 largescale environmental and socio-economic applications. *Open Source*, p. 247-270.

Dijkstra, E. (1959). A note on two problems in connexion with graphs. *Numerische mathematik*, p. 269–271.

Felzenszwalb, P. F., and Huttenlocher, D. P. (2004). Efficient graph-based image segmentation. In: *International Journal of Computer Vision*, v. 59, n. 2, p. 167-181.

Gonzalez, R. C. and Woods, R. E. (2008). Digital Image Processing. 3rd Edition, Prentice Hall. ISBN 0-13-168728-8.

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*.

Hay, G., Castilla, G. (2008). Geographic Object-Based Image Analysis (GEOBIA): A new name for a new discipline. In: Object Based Image Analysis: Spatial Concepts for Knowledge-Driven Remote Sensing Applications, Edited by Blaschke, T., Lang, S. and Hay, G. Springer, Heidelberg.

Körting, T. S., Castejon, E. F. and Fonseca, L M. G. (2011). Divide and Segment - An alternative for parallel segmentation. In**:** *Proceedings of XII GEOINFO*, pp. 97–104. INPE, Campos do Jordão.

Körting, T. S., Fonseca, L. M. G., and Câmara, G. (2013). GeoDMA—Geographic data mining analyst. *Computers & Geosciences*, 57, 133-145.

Körting, T. S., Castejon, E. F. and Fonseca, L. M. G. (2013). The Divide and Segment Method for Parallel Image Segmentation. In: Advanced Concepts for Intelligent Vision Systems. Springer International Publishing, pp. 504-515.

Lassalle, P., Inglada, J., Michel, J., Grizonnet, M. and Malik, J. (2015). A Scalable Tile-Based Framework for Region-Merging Segmentation. In: IEEE Transactions On Geoscience And Remote Sensing, Vol. 53, n. 10. pp. 5473 - 5485.

Pinho, C., Silva, F., Fonseca, L. and Monteiro, A. (2008). Intra-urban land cover classification from high-resolution images using the C4.5 algorithm. In: ISPRS Congress Beijing 7.

Prewitt, J. MS. (1970). Object enhancement and extraction. *Picture processing and Psychopictorics*, v. 10, n. 1, p. 15-19.

Seinstra, F.J., Koelma, D. (2004). User transparency: a fully sequential programming model for efficient data parallel image processing, In: *Concurrency and Computation: Practice and Experience*, 16, pp. 611–644.

Shi, J. and Malik, J. (2000). Normalized cuts and image segmentation. In: IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(8), pp. 888–905.

Soille, P. (2013). Morphological image analysis: principles and applications. Springer Science & Business Media.

Schowengerdt, R. A. (2007). Remote Sensing Models and Methods for Image Processing. 3rd Edition, Elsevier.

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