Large Remote Sensing Image Segmentation with Stitching Strategy Based on Dominant Color

Haizhong Zhang[1], Ligang Wang[1], Fei Tong[2*]

[1] Information Center of Natural Resources, Department of Natural Resources of Ningxia Hui Autonomous Region, Yinjiaqu North Street 25, Ningxia 750004, China.

[2] Department of Geodesy and Geomatics Engineering, University of New Brunswick, 15 Dineen Drive, P.O. Box 4400, Fredericton E3B 5A3, NB, Canada.

* Corresponding author: Email: ftong@unb.ca

Manuscript submitted June 10, 2020; accepted October 20, 2020.
doi: 10.17706/jsw.16.2.67-79

Abstract: Large remote sensing image segmentation is a crucial issue in object-based image analysis. It is common sense that a segmentation framework consists of three components: (1) dividing large remote sensing image into blocks for overcoming the constraint of computer memory; (2) executing segmentation algorithm for each block individually; (3) stitching segmentation results of all blocks into a complete result for eliminating artificial borders created by dividing blocks. However, there is a lack of mature technologies to eliminate artificial borders produced by dividing blocks. In this paper, we proposed a new stitching strategy based on the dominant color similarity measure and modified the traditional method of dominant color similarity measure to make it more suitable for measuring the similarity of two segmented regions. A multi-scale segmentation algorithm is adopted for segmenting each block. External memory is used to store intermediate segmentation results and exchange data with internal memory. We tested the algorithm with three different images and validated that the algorithm can implement the segmentation for large remote sensing images in a common computer. Experiments demonstrate that the stitching strategy based on the similarity measure of dominant color can effectively eliminate artificial borders.

Key words: Large image segmentation, stitching algorithm, dominant color, similarity measure, dividing block.

1. Introduction

With the increase in the resolution of the Earth-observing system, remote sensing images are becoming larger and larger. As the primary step of image analysis, image segmentation can provide useful information for subsequent processes such as target objects classification, land cover investigation, and other applications [1]–[5]. However, it is very difficult to obtain good segmentation results of large remote sensing images using a common computer. The memory capacity of a common computer is usually limited. It cannot meet the requirement of large remote sensing image segmentation. Therefore, we have to divide the large image into many small blocks [6], and then segment each small block. During the procedure, there exist two main problems. The first problem is the memory limitation. Although dividing blocks makes the segmentation of large image possible, the data volume of the segmentation results of all blocks is too large to be stored in the memory of the computer. The second problem is the artificial borders. In the process of dividing blocks, artificial borders will be generated. When stitching blocks together, artificial borders exist in the segmentation result and negatively affect the quality of the segmentation result.

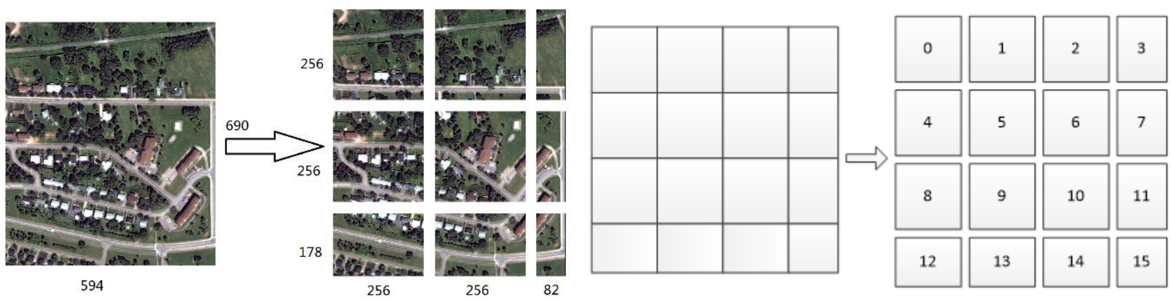
For the problem of memory limitation, the open-source software Orfeo ToolBox [7] utilizes external memory to store intermediate segmentation results so that it can segment arbitrary large remote sensing images. Once the segmentation of a block is finished, its result will be stored in external memory (hard disk). In this paper, we adopt a similar external memory strategy. However, we select the multi-scale segmentation algorithm used in eCognition to replace the Mean Shift segmentation algorithm [8], [9] used in Orfeo ToolBox. The multi-scale segmentation algorithm is based on a region growing and merging technique. It can segment images to produce multiple layers on different scales, then build an image semantics network between different layers. The algorithm has been widely applied to object-based classification because of its' high quality compared to other segmentation approaches [10], [11]. According to the literature review by Blaschke [1], 50%-55% articles out of more than 800 papers used eCognition. Although the multi-scale segmentation algorithm embedded in eCognition Developer has many advantages, it is unable to segment large images in the common computer.

For the problem of artificial borders, several methods have been proposed to remove the artificial borders created by dividing blocks. The paper [12] proposed an idea that the image contour is taken as cutting lines to create adaptive chunks. In this way, all chunks are divided with suitable borders. Therefore, no artificial borders need to be removed. However, it cannot guarantee that the detected cutting line is identical to the boundary of regions produced by the segmentation algorithm. Hence, this method may create inconsistent objects in certain cases. The paper [13] defines an overlapped area for every two adjacent tiles. When one part of a region is in a tile and the other part of the region is in the overlapped area, the method can avoid producing the artificial borders. But when a large region straddles an overlapped area, it will be partitioned into several parts and artificial borders still exist. In [8], [9], the authors proposed the concept of segmentation stability, which includes the inner property and the cover property. The concept is very good for evaluating the tile-based large image segmentation algorithm. According to the concept, the authors designed a scheme of dividing images into overlapped blocks. Two adjacent blocks have a pixel overlapped margin. Then any pair of overlapped segments from a pair of overlapped tiles will be merged after finishing the segmentation of all blocks. The method can eliminate all artificial borders and guarantee that the segmentation result will exactly match the segmentation result of the whole image at once. In [14], the authors utilize a topological criterion to remove the artificial borders on the tile edges. If the contact surface of two adjacent segments separately belonged to two adjacent tiles is large enough, the two segments will be merged. However, two segments that have a small contact surface may belong to the same ground object, and two segments that have a large contact surface may belong to different ground objects. In [15], the authors present a parallel implementation of the mean-shift segmentation algorithm. They use a novel buffer-zone-based data-partitioning strategy to avoid the inconsistency on the boundaries of adjacent data chunks. In [16], the authors apply a tile-wise processing framework similar to the methodology developed in [9] in SLIC superpixel segmentation algorithm for big remote sensing image segmentation.

In this paper, we propose a new stitching method to remove the artificial borders. Segmented regions adjacent to artificial borders will be judged whether they should be merged according to the similarity measure. To achieve the best effect, we adopt the dominant color proposed by MPEG (Moving Picture Experts Group) [17], which has been widely applied in the field of CBIR (Content-Based Image Retrieval) [18], [19], to measure the similarity between segmented regions. The dominant color describes the salient color distributions in an image or a region of interest. It provides an effective, compact, and intuitive representation of colors that is consistent with human visual perception. In addition, we improve the traditional similarity measure of dominant color to make it more effective for measuring the similarity of two segmented regions. Therefore, our proposed method can effectively and successfully remove artificial borders.

2. Methodology

To overcome the constraint of internal memory, we have to divide the original large image into many small blocks. Assuming that the size of the original image is $m \times n$ and the width and height of blocks are w and h, there will be $((m-1)/w+1) \times ((n-1)/h+1)$ blocks, as shown in Fig. 1(a). Each block is assigned to an index that will be used in the following file reading and writing operation. Fig. 1(b) shows the process of dividing an image into 16 blocks and all blocks are given a unique index. Then segmentation of each block will be executed independently. The segmentation result of each block will produce artificial borders (red lines in Fig. 2 are artificial borders), which don't exist if the whole image is segmented without dividing blocks. To eliminate the artificial borders, we have to stitch the segmentation results of all blocks into a whole segmentation result. However, the storage space occupied by the segmentation results of all blocks is far large than the memory available on computers. Therefore, we need to save the segmentation result of each block to external memory. We design an effective framework of large image segmentation according to the analysis above. The framework of large image segmentation is shown in Fig. 3. Initially, we set parameters for multi-scale algorithm and block size. Then we divide the original large image into blocks and give each block a unique index. A multi-scale segmentation algorithm is applied to segment each block and each segmentation result is written into two binary files stored in external memory. Finally, a stitching algorithm is used to eliminate artificial borders and generate the ultimate segmentation result.



(a) (b)

Fig. 1. (a): A 594×690 image is divided into 9 blocks and the assigned block size is 256×256.
(b): An image is divided into 16 blocks and each block is assigned to an index.

Fig. 2. Dividing a 512×512 image into 2×2 blocks and segmenting each block (Red lines are artificial borders).

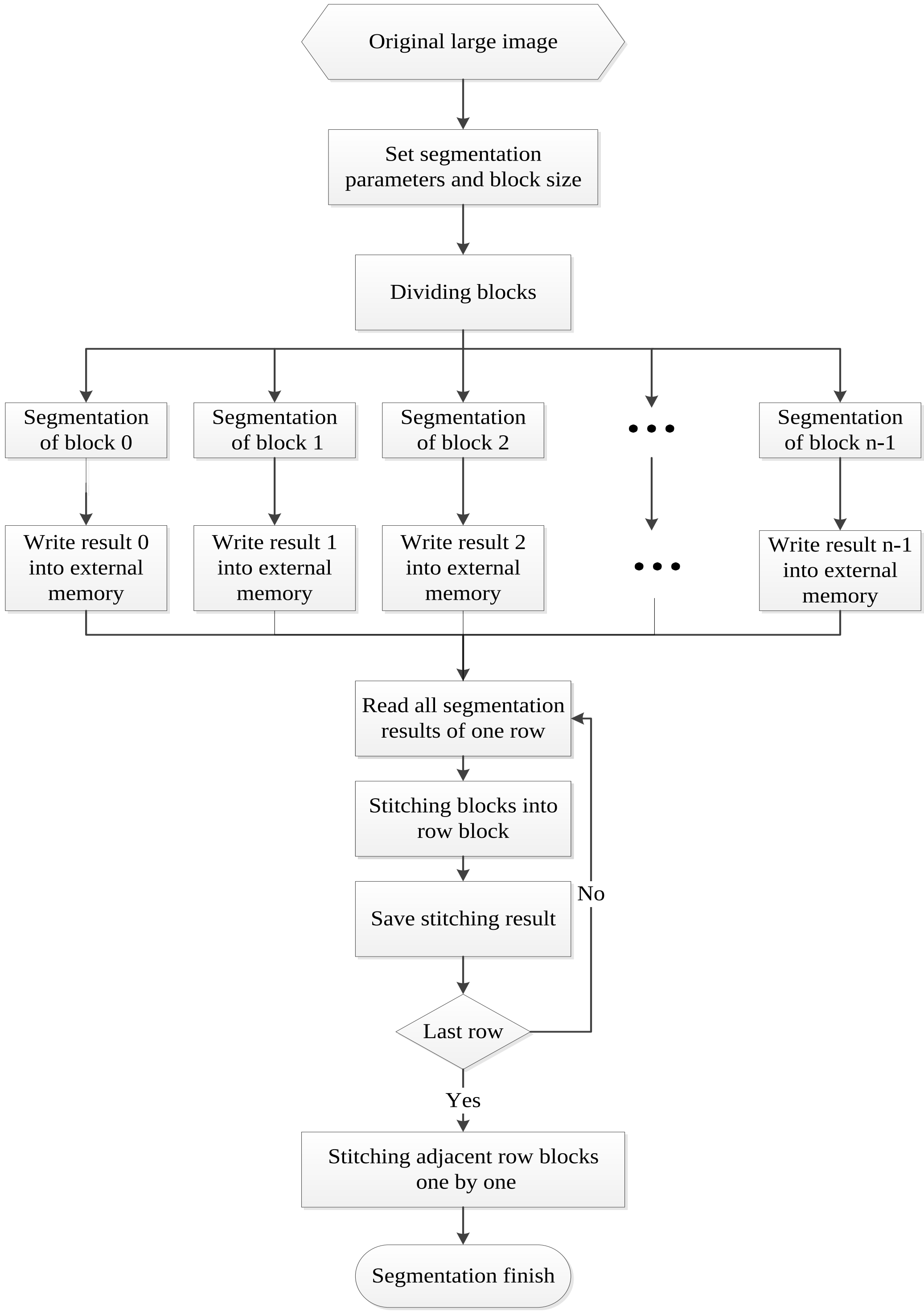


Fig. 3. The framework of large image segmentation.

2.1. Multi-scale Segmentation Algorithm

The main idea of the multi-scale segmentation algorithm in the paper is from FNEA (Fractal Net Evolution Approach) which was proposed by Baatz and Schäpe [20]. The algorithm is based on a region growing and merging technique. We adopt the merging strategy of Global Mutual Best-Fitting (GMBF) in the algorithm implementation. GMBF means that the selected pair of objects to be merged are the best among all pairs. The criterion that determines whether an object should merge with its neighbor is the heterogeneity increase [20], [21]. The heterogeneity increase is the so-called 'fusion' value (f) [21], which is given by:

$$f = w_{spectral} \bullet h_{spectral} + (1 - w_{spectral}) \bullet h_{shape} \quad (1)$$

where $w_{spectral}$ is the user assigned weight associated with spectral heterogeneity increase, $h_{spectral}$ is the spectral heterogeneity increase and h_{shape} is the shape heterogeneity increase. The merging between two adjacent objects will be considered if $f < s^2$, where is the user-specified threshold, referred to as the Scale parameter.

$h_{spectral}$ is a measure of the spectral heterogeneity increase resulting from the potential merging of two adjacent objects ($obj1$ and $obj2$). It is given by:

$$h_{spectral} = \sum_c w_c (n_{obj1+obj2} \bullet \sigma_c^{obj1+obj2} - (n_{obj1} \bullet \sigma_c^{obj1} + n_{obj2} \bullet \sigma_c^{obj2})) \quad (2)$$

where c represents the different spectral bands of a multispectral image, w_c is the user assigned weight associated with the band c, n is the number of pixels in the object, and σ_c is the standard deviation of pixel values within the band c.

h_{shape} is the weighted average of $h_{compact}$ (compactness heterogeneity increase) and h_{smooth} (smoothness heterogeneity increase). The calculation of h_{shape} is given by:

$$h_{shape} = w_{compact} \bullet h_{compact} + (1 - w_{compact}) \bullet h_{smooth} \quad (3)$$

where $w_{compact}$ is the user assigned weight associated with the compactness increase.

$h_{compact}$ and h_{smooth} are defined as:

$$h_{compact} = n_{obj1+obj2} \bullet \frac{l_{obj1+obj2}}{\sqrt{n_{obj1+obj2}}} - (n_{obj1} \bullet \frac{l_{obj1}}{\sqrt{n_{obj1}}} + n_{obj2} \bullet \frac{l_{obj2}}{\sqrt{n_{obj2}}}) \quad (4)$$

$$h_{smooth} = n_{obj1+obj2} \bullet \frac{l_{obj1+obj2}}{b_{obj1+obj2}} - (n_{obj1} \bullet \frac{l_{obj1}}{b_{obj1}} + n_{obj2} \bullet \frac{l_{obj2}}{b_{obj2}}) \quad (5)$$

where n is the number of pixels comprising an object, l is the perimeter of an object, and b is the perimeter of an object's bounding box.

The generation process of fusion value is graphically represented in Fig. 4. The fusion value is used to compare with the user-specified threshold, scale parameter (s), which determine whether two adjacent objects should be merged or not.

2.2. Stitching Algorithm Based on Dominant Color Similarity

After segmentation block by block, we have to stitch the segmentation results of all blocks into a complete result in order to remove artificial borders. The segmentation results of all blocks are very large thus they cannot be stored in the memory of a computer. Therefore, when the segmentation of a block is finished, the segmentation result will be written to the hard disk and the memory occupied by this block should be released. Only border regions in a block are used in the stitching process. In order to reduce unnecessary file read-write operations, the segmentation result is written into two binary files. One file stores border regions and the other file stores internal regions. Memory is not enough if border regions of all blocks are read into memory simultaneously for a large remote sensing image. Therefore, we just read segmentation results of a row of blocks into memory every time and stitch these blocks. Then when all large row blocks are produced, we stitch adjacent large row blocks step by step until all large row blocks are stitched into a complete segmentation result. The stitching procedure is shown in Fig. 5. During the stitching procedure, a primary operation is the stitching of two adjacent regions. The stitching of two adjacent regions is based on the similarity measure of the dominant color.

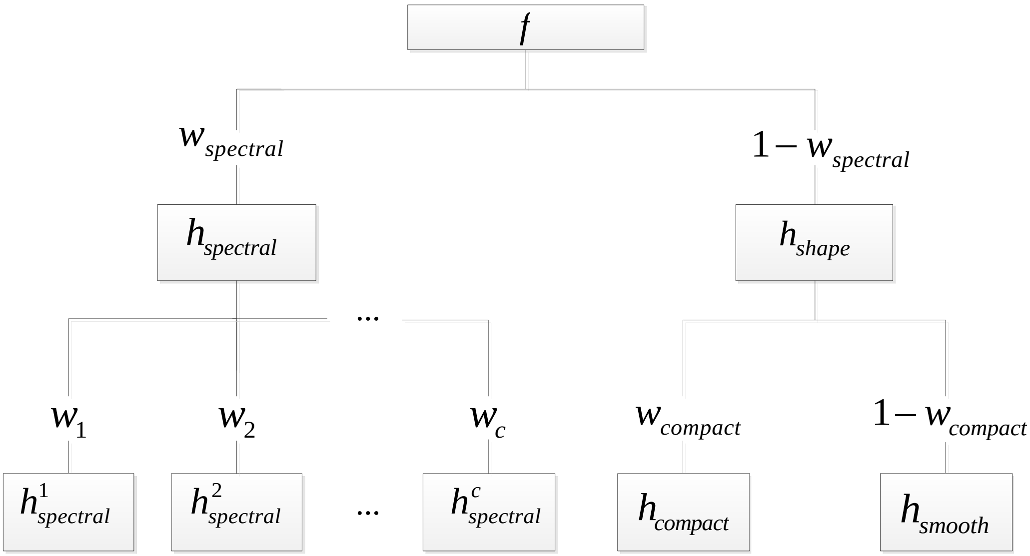


Fig. 4. The generation process of fusion value. Usually, the weights for individual spectral bands ($w_1, w_2, \ldots w_c$) are set to 1. Scale parameter (s), spectral weight ($w_{spectral}$) and shape weight ($w_{compact}$) are determined by users.

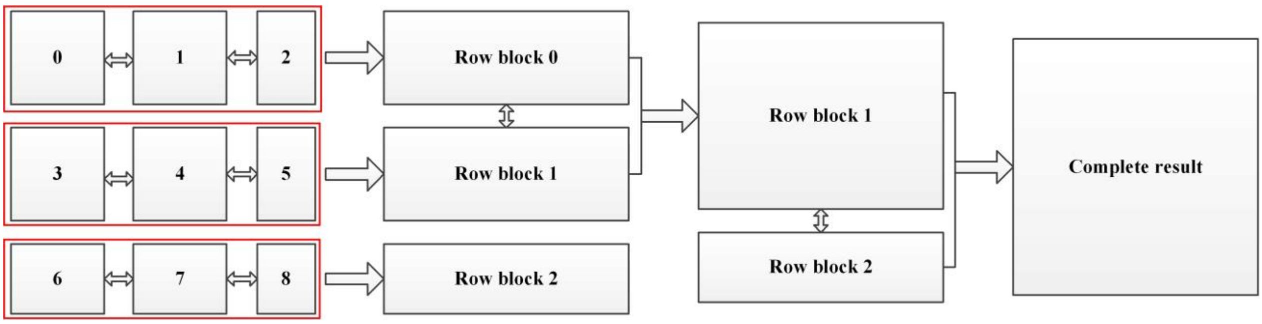


Fig. 5. The procedure of stitching 3×3 blocks' segmentation results into a complete result.

The similarity measure of dominant color is based on the dominant color descriptor. The dominant color descriptor is proposed by MPEG-7 standard [19] and now it has been used extensively in the field of image retrieval. The dominant color descriptor is a set of colors that occupy a large percentage in an image. According to visual psychology theory, when we observe an image, we are usually attracted by colors that occupy a large part of the image. Therefore, the dominant color descriptor can represent the color characteristic of an image or a region. In the field of image retrieval, two regions whose similarity of dominant color is large enough will be considered similar and can be matched. Therefore, this paper uses the similarity of dominant color to determine whether two adjacent regions should be merged or not.

The dominant color descriptor in the MPEG-7 is defined as:

$$F = \{\{c_i, p_i\}, (i = 1, 2, ..., N)\} \tag{6}$$

where *F* represents the dominant color descriptor of a region and c_i is *i*-th dominant color in the set, p_i is the percentage of *i*-th dominant color in the region, *N* is the number of dominant colors. In MPEG-7 standard, the number of dominant colors is suggested to be no more than 8.

The dominant color extraction algorithm is based on clustering. We choose GLA (Generalized Lloyd Algorithm) to extract dominant color [19], [22]. We should choose a suitable color space to represent color before extraction. In the real world, RGB color space is generally used to describe the color of the region. The three components of RGB color space are relative. However, the relativity of three components in the LUV color space is weak. Furthermore, LUV color space is more suitable than RGB color space for the human visual system. Therefore, the LUV color model is selected to extract the dominant color. In LUV color space, L represents the brightness, U and V represent the chromaticity. RGB color space can be transferred to LUV color space and vice versa [23]. The details about how to extract dominant color are explained in [24].

The similarity measure of dominant color descriptors can influence the accuracy of regions merging. The traditional similarity measure of dominant color is described in chapter three of [24]. Let $F_1 = \{\{c_i, p_i\}, (i = 1, 2, ..., N_1)\}$ and $F_2 = \{\{b_j, q_j\}, (j = 1, 2, ..., N_2)\}$ be two dominant color descriptors for two different regions. The dissimilarity of the two regions is denoted by the distance between F_1 and F_2. It is defined as:

$$D^2(F_1, F_2) = \sum_{i=1}^{N_1} p_i^2 + \sum_{j=1}^{N_2} q_j^2 - \sum_{i=1}^{N_1}\sum_{j=1}^{N_2} 2a_{i,j} p_i q_j \tag{7}$$

where $a_{i,j}$ is the similarity coefficient between two dominant colors c_i and b_j.

The similarity coefficient is given by:

$$a_{i,j} = \begin{cases} 1 - d_{i,j}/d_{\max} & , \ d_{i,j} \le T_d \\ 0 & , \ d_{i,j} > T_d \end{cases} \tag{8}$$

where $d_{i,j}$ is the Euclidean distance between two dominant colors c_i and b_j, the threshold T_d is the maximum distance for two dominant colors. If the distance of the two dominant colors is greater than T_d, two dominant colors are considered to be dissimilar. $d_{\max} = \alpha T_d$ and α is a parameter that is equal to 1 or greater than 1. A normal value for T_d is between 10 and 20 in the LUV color space and for α is 1.0 ~1.5. It is easy to find that the larger T_d and α are, the smaller distance $D^2(F_1, F_2)$ is.

The similarity measure can reflect the similarity of regions well in the field CBIR, because retrieved images are ranked by the similarity with a searched image without threshold. But in our algorithm, a threshold of similarity should be set to determine whether two regions should be merged or not. Actually, it is difficult to choose a suitable threshold if we adopt the similarity measure mentioned above. Two examples in Table 1 demonstrate that the similarity measure above is not suitable for segmented region merging. The first row of Table 1 shows the two parts of a red roof that should be similar according to human perception. $F_{region1}$ and $F_{region2}$ represent dominant color descriptors of region 1 and region 2. The distance between $F_{region1}$ and $F_{region2}$ is 1.05, even T_d is set to the maximum value 20 and α is set to the maximum value 1.5. The second row of Table 1 shows two different regions: one belongs to the red roof and the other belongs to the blue roof. The color characteristic of region 1 and region 2 are dissimilar according to visual perception. The distance between $F_{region1}$ and $F_{region2}$ is 0.86, no matter what T_d and α are set. It is obvious that the distance in the second is less than the distance in the first row. We will conclude that two regions in the second row are more similar than the two regions in the first row. However, the conclusion is contrary to the fact. According to the analysis above, we can conclude that the similarity measure is not suitable for determining whether two regions are similar or not in our algorithm. Therefore, we propose a new method of the dominant color similarity measure. The new method only selects one color, whose percentage is largest in the dominant color descriptor, to represent the dominant color of the region. The new dominant color similarity measure is to calculate the Euclidean distance between two selected dominant colors. If the distance of two regions is less than the threshold T_d, two regions should be merged. The reasons that we choose the new similarity measure are: (1) the color whose percentage is largest in a region will always draw our attention according to visual physiology theory; (2) the color distribution of segmented region concentrates on one color, which can denote the color characteristic of a region.

Table 1. Two Examples That Are not Consistent with Visual Perception

Regions	$F_{region1}$	$F_{region2}$	Distance
	{(45.21, 13.83, 6.36), 1.0}	{(58.23, 20.61, 12.26), 1.0}	1.05
	{(75.53, -17.11, 22.22), 0.01} {(22.47, -8.47, -31.02), 0.02} {(64.27, -22.83, -50.82), 0.10} {(43.11, -16.59, -46.82), 0.21} {(58.44, -24.46, -76.64), 0.66}	{(91.68, 14.96, 20.44), 0.02} {(85.17, 31.52, 22.35), 0.06} {(78.92, 47.05, 22.23), 0.07} {(73.52, 67.15, 23.06), 0.09} {(68.13, 88.89, 24.76), 0.22} {(64.26, 105.93, 26.91), 0.54}	0.86

3. Experimental Results

The proposed algorithm is tested on three different remote sensing images. The images were acquired by different sensors, with different band numbers and spatial resolution. The default block size is set to 1024×1024. Parameters in the multi-scale segmentation algorithm are same for the three images. To test the robustness of the algorithm, we choose a relative smaller Scale parameter (*s*=50) for the multi-scale segmentation algorithm. The spectral weight parameter ($w_{spectral}$) is set to 0.9 and the Compactness weight parameter ($w_{compact}$) is set to 0.5. The segmentation result will consist of a large number of regions. According to Tobler's First Law of Geography [25], [26], near regions are more related to each other. Therefore, T_d is set to the maximum value 20 in order to merge adjacent and similar regions as much as possible. Table 2 shows the time cost distribution of multiple steps in segmentation. All experiments were conducted on an Intel Core i7-4790 at 3.60GHZ. The memory size of the computer is 8G. The code is written in C++ and the program is run in Microsoft Visual Studio 2013.

All images are segmented well with the proposed algorithm. Almost all regions on the edge of blocks are merged accurately. Fig. 6 shows the segmentation result of a 512×512 image, which was divided into 4 blocks. We successfully segmented a pan-sharpened image, whose size is 24060×21512. The image has four spectral bands (blue, green, red, and near-infrared), 16 bits, and 0.7 m resolution for KOMPSAT-3 scene of Beijing (the size of the file is 5.26G; Overviews is 32 levels; the size of the tile is 256×256). Because the segmentation result's size of a large remote sensing image is far larger than the page size of a paper, it cannot be displayed clearly in the paper. To better show segmentation results, all segmentation results for large images were uploaded to http://39.101.162.99:8085/.

Table 2. Time Cost Distribution in Segmentation for three different Images

Sizes	Block Number	T_d	Time Cost (min)		
			Segmentation	Stitching	Vectorization
4000×5000	20	20	7	1	1
16384×16384	256	20	85	3	13
24060×21512	528	20	177	5	30

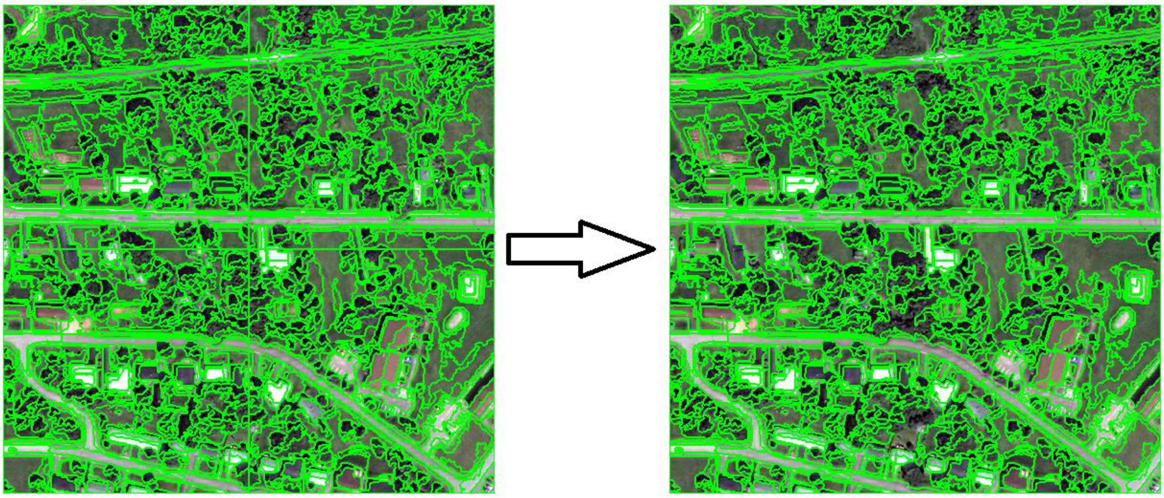


Fig. 6. Stitching 2×2 blocks into a complete one (the size of each block is 256×256).

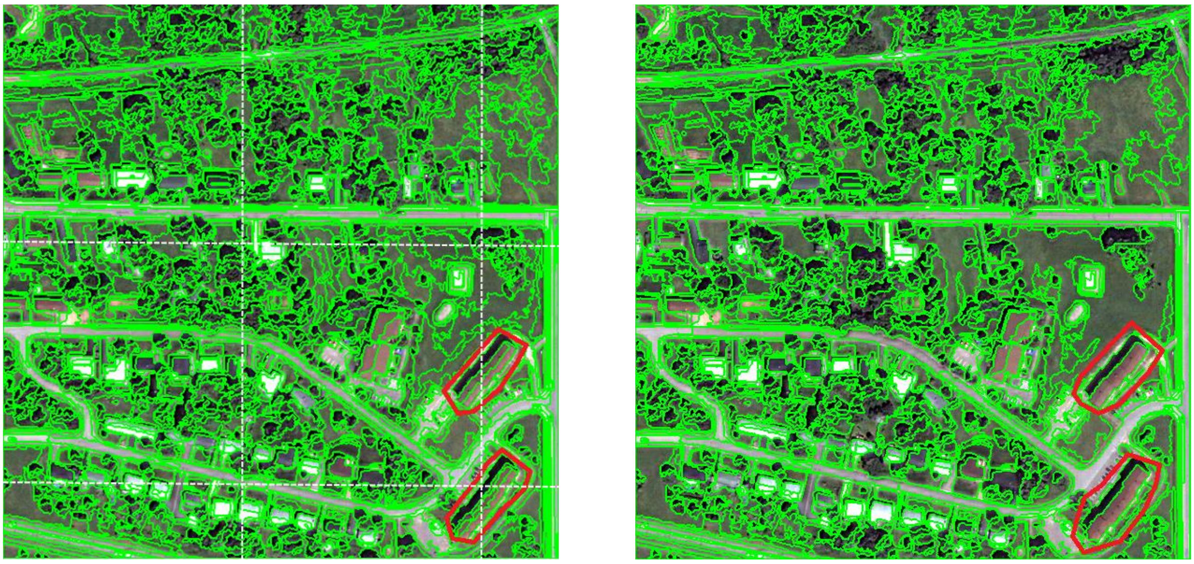
4. Discussion

We divide a large remote sensing image into many small blocks and perform the segmentation algorithm block by block. Moreover, we use external memory to save intermediate segmentation results. When we perform the stitching algorithm, we only read segmentation results of a row of blocks into memory every time and stitch these blocks. Then we stitch two adjacent large row blocks step by step to produce a complete segmentation result. The stitching procedure is shown in Fig. 5. During the whole procedure, we allocate and release memory dynamically and use external memory to save intermediate segmentation results. Therefore, we can segment arbitrary large remote sensing images in theory.

It is necessary to select a suitable block size before segmentation. The larger the block size is, the fewer artificial borders, file read-write operations, and stitching operations are. Therefore, the larger the block size is, the more effective the large image segmentation is. However, if the block size is too large, the memory cost will exceed the limitation of memory in the common computer. On the contrary, the smaller the block size is, the more artificial borders, file read-write operations, and stitching operations are. It will lead to the low efficiency of large image segmentation. A suitable block size can make full use of the available internal memory of the computer and make the segmentation more effective. In order to make the program run smoothly, the default block size is set to 1024×1024.

The threshold T_d in dominant color similarity measure can influence the accuracy of stitching. If T_d is set to a small value, the constraint condition for the merging operation of two regions is strict, which may result in that a few similar regions are not merged. Consequently, a few artificial borders are not removed. If T_d is set to a large value, artificial borders can be removed correctly, and over-segmentation regions located at the edge of blocks can be reduced. As shown in Fig. 7, the segmentation result with dividing blocks in Fig. 7(b) is different from the segmentation result without dividing blocks in Fig. 7(a). The marked red roofs in Fig. 7(a) consist of several small regions, but the red roofs are merged perfectly in Fig. 7(b). However, if adjacent regions located at the edge of blocks exist different ground objects but with similar or same spectral characteristics, they will be merged improperly.

The time cost of the proposed segmentation algorithm is expensive when dealing with large images, because the segmentation of blocks is serial and external memory is used to exchange intermediate segmentation results with memory. A parallel scheme based on CUDA (Compute Unified Device Architecture) or Hadoop platform can effectively solve the problem.



(a) (b)

Fig. 7. The size of image is 594×594 and block size is 256×256. (a) Segmentation result without dividing blocks; (b) segmentation result with dividing blocks.

5. Conclusion

In this paper, we proposed a multi-scale segmentation algorithm to segment large remote sensing images and overcome the limitation of computer memory capacity. The algorithm utilized a dominant color similarity measure and an external memory technology to improve segmentation quality and efficiency. To find an effective method for color similarity measure, we introduced the concept of dominant color from the research field of CBIR into the field of remote sensing, and modified the similarity measure of dominant color to suit to the similarity measure of two segmented regions. Experimental results demonstrated that our algorithm can segment very large remote sensing images well and remove the artificial borders caused by image blocks.

Segmentation of large remote sensing images is a key step for Object-Based Image Analysis (OBIA). The proposed method demonstrated its potential, but still has room for improvement. Further work is to improve the performance of our algorithm by using parallel computing technology based on CUDA or Hadoop platform programming.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

Haizhong Zhang and Ligang Wang conceived and designed the study, proposed the stitching strategy based on the dominant color, wrote the first version of the manuscript. Haizhong Zhang and Ligang Wang made an equal contribution to the paper. Fei Tong implemented the proposed methodology, performed the experiments. Hengjian Tong provided remote sensing image data, reviewed the first version of the manuscript, gave some good suggestions, and modified the first version to form the final version of the manuscript. All authors read and approved the manuscript.

Acknowledgment

This research was supported by the National Natural Science Foundation of China under Grant No. 41171339. The original Ikonos, QuickBird, and KOMPSAT-3 images were provided by the City of Fredericton, CFB Gagetown, and Beijing Space Eye Innovation Technology Co., Ltd., respectively.

References

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

[2] Asokan, A., & Anitha, J. (2019). Change detection techniques for remote sensing applications: A survey. *Earth Science Informatics, 12(2)*, 143–160.

[3] Chen, G., Weng, Q., Hay, G. J., & He, Y. (2018). Geographic object-based image analysis (GEOBIA): Emerging trends and future opportunities. *GIScience & Remote Sensing, 55(2)*, 159–182.

[4] Costa, H., Foody, G. M., & Boyd, D. S. (2018). Supervised methods of image segmentation accuracy assessment in land cover mapping. *Remote Sensing of Environment, 205*, 338–351.

[5] Csillik, O. (2017). Fast segmentation and classification of very high resolution remote sensing data using SLIC superpixels. *Remote Sensing, 9(3)*, 243.

[6] Xing, J., Sieber, R., & Kalacska, M. (2014). The challenges of image segmentation in big remotely sensed imagery data. *Annals of GIS, 20(4)*, 233–244.

[7] Orfeo ToolBox project homepage. Retrieved July 26, 2019, from https://www.orfeo-toolbox.org.

[8] Michel, J., Youssefi, D., & Grizonnet, M. (2014). Stable mean-shift algorithm and its application to the segmentation of arbitrarily large remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing, 53(2)*, 952–964.

[9] Lassalle, P., Inglada, J., Michel, J., Grizonnet, M., & Malik, J. (2015). A scalable tile-based framework for region-merging segmentation. *IEEE Transactions on Geoscience and Remote Sensing, 53(10)*, 5473–5485.

[10] Hay, G. J., Blaschke, T., Marceau, D. J., & Bouchard, A. (2003). A comparison of three image-object methods for the multiscale analysis of landscape structure. *ISPRS Journal of Photogrammetry and Remote sensing, 57(5–6)*, 327–345.

[11] Hossain, M. D., & Chen, D. (2019). Segmentation for object-based image analysis (OBIA): A review of algorithms and challenges from remote sensing perspective. *ISPRS Journal of Photogrammetry and Remote Sensing, 150*, 115–134.

[12] Körting, T. S., Castejon, E. F., & Fonseca, L. M. G. (2013). The divide and segment method for parallel image segmentation. *Proceedings of the International Conference on Advanced Concepts for Intelligent Vision Systems* (pp. 504–515). Springer, Cham.

[13] Crisp, D. J., Perry, P., & Redding, N. J. (2003, February). Fast segmentation of large images. *Proceedings of the 26th Australasian Computer Science Conference* (pp. 87–93).

[14] Michel, J., Grizonnet, M., Jaen, A., Harasse, S., Hermitte, L., Guinet, J., ... & Savinaud, M. (2012). Open tools and methods for large scale segmentation of very high resolution satellite images. *Proceedings of the Open Source and Geospatial Research and Education Symposium* (pp. 179–184).

[15] Wu, T., Xia, L., Luo, J., Zhou, X., Hu, X., Ma, J., & Song, X. (2018). Computationally Efficient Mean-Shift Parallel Segmentation Algorithm for High-Resolution Remote Sensing Images. *Journal of the Indian Society of Remote Sensing, 46(11)*, 1805–1814.

[16] Derksen, D., Inglada, J., & Michel, J. (2019). Scaling up SLIC superpixels using a tile-based approach. *IEEE Transactions on Geoscience and Remote Sensing, 57(5)*, 3073–3085.

[17] Zhang, Dengsheng (2019). Color Feature Extraction. *In Fundamentals of Image Data Mining* (pp. 49–80). Springer: Cham.

[18] Mansourian, L., Abdullah, M. T., Abdullah, L. N., Azman, A., & Mustaffa, M. R. (2018). An effective fusion model for image retrieval. *Multimedia Tools and Applications, 77(13)*, 16131–16154.

[19] Yang, N. C., Chang, W. H., Kuo, C. M., & Li, T. H. (2008). A fast MPEG-7 dominant color extraction with new similarity measure for image retrieval. *Journal of Visual Communication and Image Representation, 19(2)*, 92–105.

[20] Baatz, M.; Schäpe, A. (2000). Multiresolution Segmentation: An Optimization Approach for High Quality Multi-scale Image Segmentation. *Proceedings of the Angewandte Geographische Information Sverarbeitung XII* (pp. 12–23).

[21] Benz, U. C., Hofmann, P., Willhauck, G., Lingenfelder, I., & Heynen, M. (2004). Multi-resolution, object-oriented fuzzy analysis of remote sensing data for GIS-ready information. *ISPRS Journal of Photogrammetry and Remote Sensing, 58(3–4)*, 239–258.

[22] Hu, K. C., Tsai, C. W., & Chiang, M. C. (2018). A highly efficient method for improving the performance of GLA-based algorithms. *Journal of Visual Communication and Image Representation, 50*, 290–302.

[23] Color Converter. Retrieved March 5, 2018, from https://github.com/THEjoezack/ColorMine.

[24] Yamada, A.; Pickering, M.; Jeannin, S.; Cieplinski, L.; Ohm, J.-R.; Kim, M. (2000) MPEG-7 Visual part of Experimentation Model Version 8.0. France: La Baule.

[25] Leitner, M., Glasner, P., & Kounadi, O. (2018). Laws of geography. *In Oxford Research Encyclopedia of Criminology and Criminal Justice*.

[26] Miller, H. J. (2004). Tobler's first law and spatial analysis. *Annals of the Association of American Geographers, 94(2)*, 284–289.

Haizhong Zhang was born in 1966. From September 1986 to July 1988, he studied at Wuhan Technical University of Surveying and Mapping, majoring in cartography for junior college education; from February 2004 to January 2007, he studied computer science and technology at Xi'an Jiaotong University for bachelor's degree. He is currently a professorate senior engineer in Information Center of Natural Resources, Department of

Natural Resources of Ningxia Hui Autonomous Region. He is mainly engaged in the application research of remote sensing technology in natural resource management.

Ligang Wang was born in 1984. From 2002 to 2006, he studied information and computing science at Changsha University of Science and Technology and obtained a bachelor's degree in science; in 2014, he graduated from Wuhan University with a master's degree in surveying and mapping engineering from the State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing. He is currently a senior engineer in Information Center of Natural Resources, Department of Natural Resources of Ningxia Hui Autonomous Region. He is mainly engaged in the application research of remote sensing technology in natural resource management.

Fei Tong was born in 1993, he received the M.E. degree from China University of Geosciences (Wuhan) in 2018. He is currently pursuing the Ph.D. degree in University of New Brunswick. His primary research interests are remote sensing image processing and artificial intelligence.