Data sewing algorithm for parallel segmentation of high-resolution remotely sensed image

HU Xiaodong[1, 2], **LUO Jiancheng**[1], **SHEN Zhanfeng**[1], **WU Wei**[1], **CHEN Qiuxiao**[3]

1. *Institute of Remote Sensing Applications*, *Chinese Academy of Sciences*, *Beijing* 100101, *China*;
2. *Graduate University of Chinese Academy of Sciences*, *Beijing* 100049, *China*;
3. *Department of Regional and Urban Planning*, *Zhejiang University*, *Zhejiang Hangzhou* 310027, *China*

Abstract: In the process and analysis of high-resolution remote sensing image, segmentation is the key step of extracting information from image data to image object. For the image segmentation tasks of large amount of data, data paralleled computing model is generally used. In this process, the effect of merging segmentation results when data gathering is related to the precision and accuracy of the subsequent object-oriented analysis. In this paper, data paralleled segmentation of remote sensing image is adopted, and a new algorithm named data sewing is proposed to solve the problem of merging segmentation results. Experiments, such as comparison of final segmentation results and assessment of computing efficiency, show that the algorithm improves the efficiency of image segmentation process. Meanwhile it guarantees the correctness of the boundary thus to ensure the credibility of the final segmentation result as well.

Key words: image segmentation, parallel computing, data sewing, mean-shift, object extraction

CLC number: TP751.1 **Document code:** A

Citation format: Hu X D, Luo J C, Shen Z F, Wu W and Chen Q X. 2010. Data sewing algorithm for parallel segmentation of high-resolution remotely sensed image. *Journal of Remote Sensing*. **14**(5): 917—927

1 INTRODUCTION

In the theory of sensor information computation (Luo *et al*., 2009), multi-scale image segmentation method, as the precondition and foundation of high spatial resolution remote sensing images information extraction and target recognition (Zhou & Luo, 2009), plays a very important role in the process of information extraction from data to object. Among the various segmentation algorithms, watershed segmentation algorithm, mean-shift segmentation algorithm and the multi-resolution image segmentation algorithm of Definiens Inc. is widely used. As an popular iterative algorithm, mean-shift segmentation algorithm has been proved efficient in convergence and applicable in cluster analysis, monitoring, image segmentation, image smoothing, filtering, image edge detection and intelligence fusion, etc (Dorin & Peter, 2002). In the aspect of implementation, researchers often adopt data paralleled segmentation (Huang, 2002; Shen, 2006) methods to meet the tasks of large amount of data.

During paralleled segmentation procedure, neither even data partition (Huang & Guo, 2002) nor more reasonable uneven data partition (Shen *et al*., 2006) adopted can avoid generating the “merging lines” after segmentation. And that will make the results unreliable and have an effect on the follow-up processes of object extraction and target recognition, especially the process of target recognition in high-resolution remote sensing information computing through shape factor, texture and skeleton, etc. Data collection during the parallelization procedure in pixel-level algorithms is relatively easy, which only needs to directly merge the sub blocks (Huang & Guo, 2002), and can also be easy to reach the level of automation. However, the result of image segmentation is a combination of several pixel blocks. Consequently, the “merging line” after paralleled segmentation will inevitably split the supposed pixel blocks into unexpected pieces. Sometimes, the pixel pieces on either side do not correspond well. This brings difficulty to the process of data collection, and some inevitable problems during high resolution remote sensing image paralleled segmentation.

At present, few study concerns this kind of problem. Therefore, after analyzing the distribution of split blocks on either side of “merging line” and taking reference of image mosaic’s techniques (Zhu & Qian, 2002), this paper proposes a new algorithm named data sewing, using the mean-shift based segmentation algorithm. Not affecting the efficiency of paralleled computing, this algorithm will make the result almost the same with the common segmentation, meanwhile ensures the credibility and accuracy of the paralleled segmentation result.

Received: 2009-07-14; **Accepted**: 2009-12-19

Foundation: National 863 Program Project (No. 2009AA12Z123, No. 2009AA12Z148 and No. 2009AA12Z121) and Key Projects in the National Science & Technology Pillar Program (No. 2006BAJ02A01, No. 2006BAJ14B08).

First author biography: HU Xiaodong (1982—), male, Ph.D. candidate, majoring in remote sensing information extraction and its implementation method with adaptive computation. E-mail: mydonne@163.com

2 PARALLEL SEGMENTATION OF REMOTE SENSING IMAGE

2.1 Mean-shift segmentation algorithm

Mean-shift segmentation algorithm is a statistical iterative algorithm, which totally depends on sample points in the feature space instead of the prior knowledge to do analysis. It enables every point to shift to the local maxima of destiny function by iteration, without pre-determining the number of categories (Zhou *et al*., 2007; Li *et al*., 2005). For different data structure, it is quite adaptable and robust. Recently, mean-shift algorithm is increasingly applied to the field of remote sensing image segmentation (Mo *et al*., 2006). Because of its clustering accuracy, stability and other characteristics, this algorithm is used in the profound analysis process of remote sensing, such as high-resolution remote sensing segmentation and object-oriented extraction in depth (Huang & Zhang, 2008).

Despite simple principle and efficient iterative, the algorithm needs to optimize the size of searching area to improve the accuracy and the efficiency in the iterative process. And it needs to map the image into a high-dimensional space in the computing process, therefore single-core PC cannot afford the large amount of data processing tasks for the higher memory capacity requirements. Thus, the power of paralleled computing is demanded to deal with such cases. This paper adopts parallel segmentation based mean-shift algorithm.

2.2 Parallel segmentation method

In aspect of implementation model, there are three kinds of model for parallelism, pipeline parallelism, functional parallelism and data parallelism. They are used to conduct the massive data, strong regularity and strong correlation of images, and adapt to the characteristics of consistency, stratification, territorial and line sequential of image processing algorithm. Pipeline parallelism makes different data lines successively flow into the various functional modules in the pipeline or the more fine-grained operations in a single function; while functional parallelism makes the data flow into the various functional modules simultaneously and do the conduction at the same time. Data parallelism can divide an image into several sub-blocks and do the same operation on every block, but will generate convergence problem on the edges of image blocks (Zhou, 2003).

With the characteristics of functional and data parallelism in pipelines, different processes accomplish different functions and different functions deal with different data. If properly designed, this kind of pipeline parallelism is able to obtain a higher efficiency, however on the premise of getting support from hardware. It is unsuitable for the current mainstream paralleled processing structure. And because of the correlative relationships between various steps in the algorithm, it is also difficult and less practical to make use of functional parallelism to process images. Considering the analysis above and especially the consistency and regionality of image, data parallelism is more appropriate to meet the tasks of paralleled segmentation and also more suitable for the current mainstream paralleled computer system, such as MPP (Massively Parallel Processing) and cluster system. Consequently, this paper adopts data parallelism to achieve the goal of segmentation and concerns about the edge convergence of segmentation blocks to carry out the in-depth research.

2.3 Implementation of parallel segmentation

This paper adopts data parallelism pattern to implement paralleled computing process of mean-shift segmentation. The steps are as follows (see Fig.1): (1) Set uniform segmentation parameters in each computing nodes in order to get the uniform segmentation results. (2) The image data is separated into pieces by the main computing node, which are assigned to different computing nodes afterward (such as different threads, cluster nodes, etc.) to do mean-shift segmentation separately. (3) The main computing node merges the blocks into areas, marks the areas as objects and then extracts the objects.

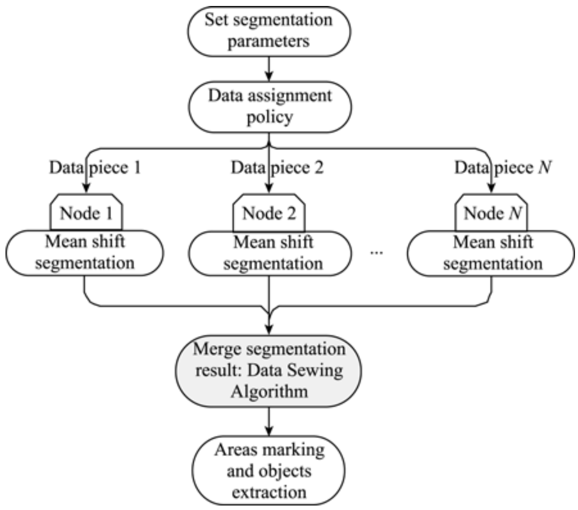

Fig. 1 Workflow of mean-shift based parallel segmentation

In this process, merging the edge of image blocks will generate the merging problem inevitably. The independence of each segmentation process leads to different segmentation results without sharing information, and different images have different global statistical features. So the outline of two blocks will be mismatched, generating an obvious "merging line" after finishing the merging and objectification of different results. The final segmentation results then become unbelievable, although the method has improved the executing efficiency and conducting capacity of segmentation algorithm. Accordingly, this paper proposes a new data sewing algorithm to solve this problem.

2.4 Data sewing

Unlike the mosaic method (Zhu & Qian, 2002) in data processing or the edge merging method (Michael *et al*., 2009) in GIS, data sewing algorithm deals with the segmented image, rather than pixel level image or vector data. Although the segmentation image is also stored as grid format, its basic unit is the block of adjacent pixels rather than pixel itself. The data sewing algorithm in this paper aims to merge and redraw such aggregate blocks on either side of the "merging line" in the segmentation image in order to generate the seamless and credible paralleled segmentation results. Therefore, data sewing is an important issue to transform remote sensing data into spatial information in the field of high performance remote sensing information computation.

3 DATA SEWING ALGORITHM

3.1 Problem description and analysis

In order to explain the problem of edge merging after separated segmentation, Fig.2 shows the "merging line" (the horizontal line pointed by an arrow) generated during the period of merging pixel blocks from paralleled segmentation results of high resolution remote sensing image. And having got careful observation, it will be found that there are two kinds of candidate segmentation areas across the "merging line". Such areas obviously different from the surrounding ones can be clustered together in two segmentation processes simultaneously, so the outlines can better correspond to each other without consideration the differences in label values. However, most of the candidate regions from various segmentation procedures are quite different in both terms of outline and label (particularly the red circle parts in Fig.2). Compared with the original surface features (see in Fig.6 (a)), roads, grass, constructions and other regions supposed to be continuous are separated by the "merging line" in Fig 2. If we use shape, area, outline and other feature factors for object-oriented analysis on this basis, the results must be unreliable. Therefore, how to eliminate the "merging line" will affect the precision and accuracy of subsequent object-oriented analysis process, and is also the key point of the data sewing algorithm.

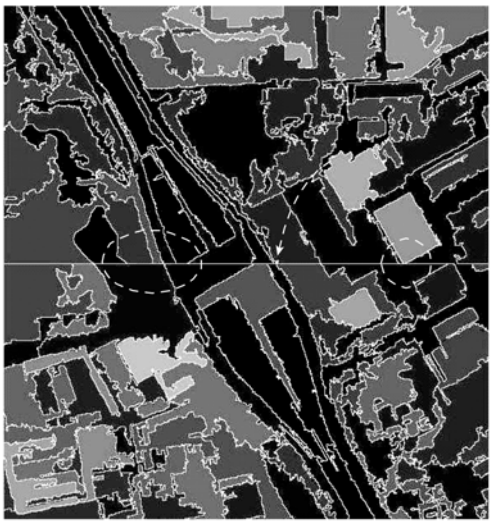

Fig. 2 The "merging line" after parallel segmentation

In this paper, the points joint to "merging line" and segmentation area outlines are classified into two categories (see in Fig.3). Among them, the endpoint of segmentation area outlines is called "class a" point of intersection ("a" point for short), and the edge-point of the whole image also belongs to this class. While the rest of the intersection points goes to "class b" point of intersection ("b" point for short). The line between two neighboring "a" points is called "class A" line segment ("A" line for short), and the rest is called "class B" line segment ("B" line for short).

The blocks along the "A" lines can be merged straightly according to the spectrum, size and shape value. But for the blocks along the "B" lines, the situation is more complicated due to the irregular arrangement of relevant segmentation blocks. However, simply merging all such kind of blocks will make segmentation inadequate eventually. In the paper, we use post segmentation to achieve the goals of eliminating merging line. This so-called "post segmentation" includes three steps, firstly finding all the groups of post-segmentation blocks along "B" line to form several small sub-image areas, secondly conducting the post segmentation of the small images, and thirdly filling the results back to the overall area at last. This does not cost too much extra computation, and still have a high sense of credibility. To find out post-segmentation blocks along "B" line, three situations should be considered. In Fig.4, (a) and (b) describe the situation where exist several "a" points and "b" points between two "a" points, not existing "A" line. While (c) gives the description of the situation that an "A" line lies between two "B" lines. For the three kinds of situation mentioned above, different search algorithms are needed to find the post-segmentation block groups and to generate the sub-image areas accordingly.

3.2 Algorithm description

According to the analysis, the core idea of data sewing algorithm in this paper is to find the "A" lines and "B" lines. Then merge the segmentation blocks along "A" lines directly, find out the post-segmentation blocks along "B" lines to form the

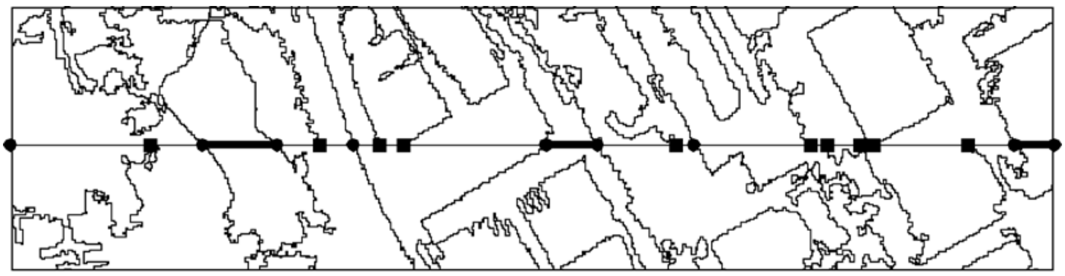

Fig. 3 Two kinds of points on the "merging line"

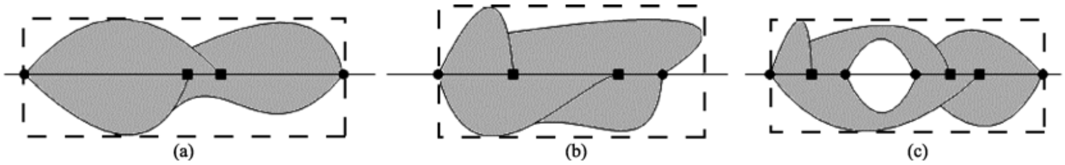

Fig. 4 Three situations of generating "post segmentation area"

sub-image areas, do post segmentation and fill back the post-segmentation results. The algorithm is as follows (see Fig.5):

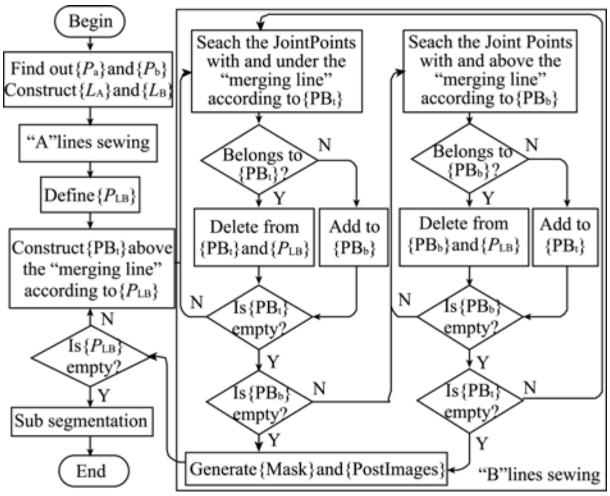

Fig. 5 Workflow of data sewing algorithm

Step(1) Traverse the "merging line" and find out the intersection sets of "a" points and "b" points, marked {P_a} and {P_b} respectively.

Step(2) Join all the adjacent "a" points to make the segment set of "A" line, marked as {L_A}. And the rest goes to the segment set of "B" line, marked as {L_B}.

Step(3) "A" lines sewing: merge all the segmentation blocks along the "A" lines respectively. The segmentation blocks are obtained by the FloodFill algorithm. Blocks on the two sides of "A" line are marked $FF_t(L_A)$ and $FF_b(L_A)$ respectively, whose "t" and "b" means "top" and "bottom" (in the paper horizontal "merging line" is taken as an example). The label value after merging is determined by Eq. (1). The prefix "T" means label value, and "AR" means area. Therefore, the label value after merging is supposed to be the weighted average of the blocks on the two sides.

$$T_{\text{merge}} = \frac{T(\text{FF}_t) \times \text{AR}(\text{FF}_t) + T(\text{FF}_b) \times \text{AR}(\text{FF}_b)}{\text{AR}(\text{FF}_t) + \text{AR}(\text{FF}_b)} \qquad (1)$$

Step(4) "B" lines sewing: assume all the points on "B" lines make up of point set {P_{LB}}. (Unlike the mathematical line segment, "B" lines here are composed of pixels. Therefore, the number of points making up the segment is limited.) Then process the following iterative algorithm.

① Take any point "P" from {P_{LB}} as seed point, and use FloodFill algorithm to find the top blocks, and put all the joint points with the "merging line" in the block into the point set {PB_t}.

② Take any point from {PB_t} as seed point to search the bottom blocks. When the point joint to the "merging line" is found, search stops and do some judgment. If it belongs to {PB_t}, then delete it from both {PB_t} and {P_{LB}}; otherwise, add it to the point set {PB_b}. Search will keep running until {PB_t} is empty. Afterward, ③ will be executed if {PB_b} is not empty, or ④ will be executed.

③ Take any point from {PB_b} as seed point to search the top blocks. When the point joint to the "merging line" is found, search stops and do some judgment. If it belongs to {PB_b}, then delete it from both {PB_b} and {P_{LB}}; otherwise, add it to the point set {PB_t}. Search will keep running until {PB_b} is empty. Afterward, ② will be executed if {PB_t} is not empty, or ④ will be executed.

④ Merge the continuous blocks generated by ①—③, add them to the image set {Mask}, and extract the corresponding pixels from the original image to make post-segmentation image, whose other parts filled with black color. Then put this image into post segmentation images set {PostImages}. Now if {P_{LB}} is not empty, ① will be executed. All the circumstances in Fig. 4 have been taken into account, and the post-segmentation images set has formed without omission.

Step(5) Segment the sub-images in the set of {PostImages}, fill the results back into the paralleled segmentation result according to {Mask} and go to the end of the algorithm.

Because the post-segmentation process for the post-segmentation blocks on both sides of the "merging line" have been done by the sewing algorithm, the final segmentation result looks more realistic and almost the same with the result of segmentation processed by single-threaded computation. The size of sub-image is small and the computation is fast by the main node, meanwhile every sub-image in the set of {PostImages} is independent, so that paralleled computation will be easily processed on this procedure. Therefore, for the overall efficiency of paralleled segmentation, the loss of computational efficiency is small.

4 EXPERIMENT AND ANALYSIS

According to the method mentioned above, multi-thread technique is adopted to realize the data-paralleled computing mode for mean-shift segmentation algorithm and the data sewing algorithm is applied to merge segmented results seamlessly.

A small block of IKONOS panchromatic image in Beijing area is used as the experimental data, whose spatial resolution is 1m and image size is 600×600. Fig.6 (a) is the original image. Fig.6 (b) shows an object-oriented result merging the paralleled segmentation data blocks directly, where we can find the merging line visible, and the dotted box brings us post-segmentation images by data sewing algorithm. Fig.6 (c) shows the object-oriented result by data sewing algorithm, where the merging line has been invisible. A high degree of credibility has been verified accordingly by the quite good results, which seems to be conducted by single-thread mean-shift segmentation.

Further, an experiment with 4-thread paralleled segmentation is taken to further assess the impact of data sewing algorithm on the overall efficiency of paralleled segmentation process. The microcomputer equipped with Intel Xeon 3.2GHz frequency 4-core CPU and 2GB memory. The experimental result on different size of data (Table 1) shows that only a few percentage of time was taken by data sewing algorithm on the overall expense. When the data becomes lager, the time percentage will gradually reduce. The experiment also shows that the final segmentation effect remains the same on different size of image. So by data sewing algorithm proposed in the paper, a good feasibility has been approved that segmentation tasks on huge data of images can be handled.

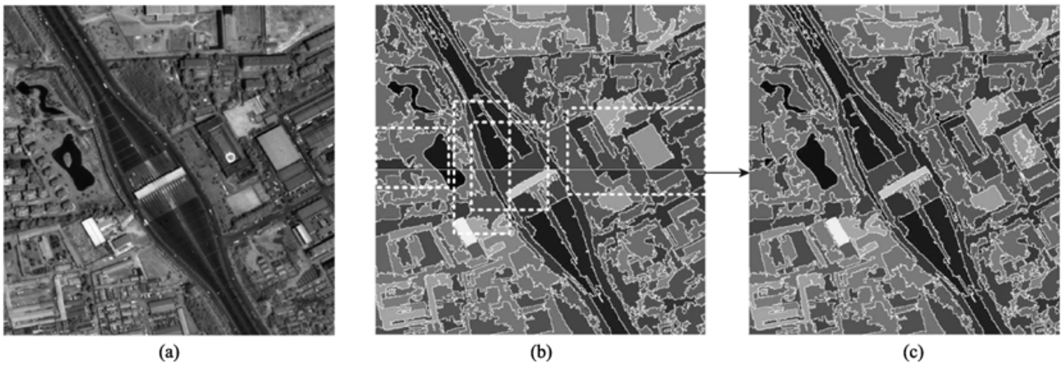

Fig. 6 Comparison of the results before and after data sewing algorithm

(a) Original image; (b) Result of parallel segmentation; (c) Result after data sewing algorithm

Table 1 Time expense of data sewing algorithm

Size of image	Time of single-thread serial segmentation/s	Block number	Time of overall parallel segmentation/s	Time of data sewing/s	Percentage of time by data sewing/%
600×600(360Kb) (Fig. 6)	6	4	6	3	50.0
2000×2000	90	4	31	11	35.5
5000×5000(25Mb)	Out of memory	6	272	47	17.3
12000×12000(144Mb)	Out of memory	35	1384	182	13.2
12000×37832(446Mb)	Out of memory	109	2479	244	9.8

5 RESULTS

As a result, a new sewing algorithm is proposed to solve the problem of "merging line" occurring in the process of merging blocks from paralleled segmentation results of high-resolution remote sensing image. Seen from the experimental results, a high sense of credibility has been verified by quite good segmentation results. On the other hand, well feasibility has also been verified by assuring the efficiency of paralleled execution. Furthermore, it can also be applied in the tasks of remote sensing segmentation for mass data, such as the serial processing of data blocks and segmentation processing of cluster machines. To a certain extent, the algorithm has avoided the bottlenecks in the process of conversion from data to information, and has a high application value.

In this paper, data sewing algorithm is proved to be effective and generally feasible. However, many problems remain to be considered and resolved. The optimization of the size of post-segmentation blocks when there are long rectangular ground covers near the merging lines is an example question. These are to be further studied, explored and practiced.

REFERENCES

Dorin C and Peter M. 2002. Mean Shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **24**(5): 603—619

Huang G M and Guo J F. 2001. Data partition for distributed-parallel processing of remote sensing imagery. *Remote Sensing Information*, (2): 9—12

Huang X and Zhang L P. 2008. An adaptive mean-shift analysis approach for object extraction and classification from urban hyperspectral imagery. *IEEE Transactions on Geoscience and Remote Sensing*, **46**(12): 4173—4185

Li X R, Wu F C and Hu Z Y. 2005. Convergence of a mean shift algorithm. *Journal of Software*, **16**(3): 365—374

Luo J C, Zhou C H and Shen Z F. 2009. A review of theory and technology for sensor information computation. *Journal of Remote Sensing*, **13**(Supplement): 1—6

Michael J S, Michael F G and Paul A L. 2009. Geospatial Analysis: A Comprehensive Guide to Principle, Techniques and Software Tools, Second Edition (Chinese Edition). Beijing: Publishing House of Electronics Industry

Mo D K, Lin H, Li J P, Sun H and Xiong Y J. 2006. VHR imagery multi-resolution segmentation based on mean shift. *Journal of Guangxi Normal University: Natural Science Edition*, **24**(4): 247—250

Shen Z F, Luo J C, Chen Q X, Huang G Y and Sheng H. 2006. Data partition policy of high-resolution remotely sensed image parallel processing. *Journal of Harbin Institute of Technology*, **11**: 1968—1971

Zhou C H and Luo J C. 2009. Geo Computing for High Resolution Remote Sensing Images. Beijing: Science Press

Zhou F F, Fan X P and Ye Z. 2007. Mean shift research and applications. *Control and Decision*, **8**: 841—847

Zhou H F. 2003. Study and Implementation of Parallel Algorithms for Remote Sensing Image Processing. Beijing: National University of Defense Technology

Zhu S L and Qian Z B. 2002. The seam-line removal under mosaicking of remotely sensed images. *Journal of Remote Sensing*, **6**(3): 183—187

高分辨率遥感影像并行分割结果缝合算法

胡晓东[1,2]，骆剑承[1]，沈占锋[1]，吴 炜[1]，陈秋晓[3]

1. 中国科学院 遥感应用研究所，北京 100101;
2. 中国科学院 研究生院，北京 100049;
3. 浙江大学 区域与城市规划系，浙江 杭州 310027

摘 要： 采用基于数据并行的遥感影像分割实现过程，提出了一种新的数据缝合算法解决分割结果合并问题。分割效果对比和运算效率分析等实验结果表明，此算法保持了分割结果合并后的边界正确性，使并行化分割在提高运算效率的同时保证了分割结果的可信度。

关键词： 影像分割，并行化，数据缝合，均值漂移，对象化提取

中图分类号： TP751.1 **文献标志码：** A

引用格式： 胡晓东，骆剑承，沈占锋，吴 炜，陈秋晓. 2010. 高分辨率遥感影像并行分割结果缝合算法. 遥感学报，**14**(5)：917—927
Hu X D, Luo J C, Shen Z F, Wu W and Chen Q X. 2010. Data sewing algorithm for parallel segmentation of high-resolution remotely sensed image. *Journal of Remote Sensing*. **14**(5): 917—927

1 引 言

在遥感信息计算体系(骆剑承等，2009)中，作为高空间分辨率遥感影像信息提取与目标识别的前提和基础(周成虎&骆剑承，2009)，多尺度影像分割方法是实现从数据到信息的对象化提取的过渡环节和关键步骤，具有十分重要的地位。在分割算法方面，分水岭分割算法、均值漂移分割算法和 Definiens 公司的多分辨率影像分割算法等被应用得较多，其中均值漂移是一种有效的统计迭代算法，并且证明具有较好的算法收敛性，广泛应用于聚类分析、跟踪、图像分割、图像平滑、滤波、图像边缘提取和信息融合等方面(Dorin & Peter，2002)；在实现方面，为了使算法效率适应大数据量任务的要求，一般采用基于数据并行的分割实现。

在并行分割算法实现过程中，不管是采用均匀数据划分方法(黄国满&郭建峰，2001)还是不均匀数据划分方法(沈占锋等，2006)，无法避免合并分割结果时产生的“缝合线”，这使并行分割结果不可信，影响到了后续的对象化提取及目标识别过程，特别是对于高分辨率遥感信息计算中根据形状因子、纹理、骨架等特征判定地物的过程，影响尤为明显。像元级算法并行化过程中的数据收集相对容易，只需将分块的影像数据进行简单地拼接(黄国满&郭建峰，2001)，也可较容易的达到自动化处理水平；影像分割后的结果是由若干像元组合而成的像元集合，并行分割结果中的“缝合线”会将原本属于同一分割块的像元集合分隔开，有时甚至线两侧的分割块并不能很好地对应，这给数据收集的过程带来了难度，也是高分辨率遥感影像并行分割中必然会遇到的问题。

目前对于该问题的研究并不多见，在分析“缝合线”两侧分割块分布情况，参考图像镶嵌的相关技术(朱述龙&钱曾波，2002)后，以基于均值漂移算法的遥感影像数据分割为例，提出了一种新的数据缝合算法，在不影响并行化运算效率的前提下，使并行分割结果与单机分割效果保持基本一致，同时确保了分割效果的可信度和准确度。

2 遥感影像并行分割

2.1 均值漂移分割算法

均值漂移算法是一种统计迭代算法，它完全依靠特征空间中的样本点进行分析，不需要任何先验

收稿日期：2009-07-14；修订日期：2009-12-19

基金项目：国家“863”计划项目(编号：2009AA12Z123，2009AA12Z148，2009AA12Z121)和国家科技支撑计划重大项目(编号：2006BAJ02A01，2006BAJ14B08)。

第一作者简介：胡晓东(1982—)，男，博士研究生，主要研究领域为遥感信息提取及其自适应计算方法。E-mail: mydonne@163.com。

知识, 对于不同结构的数据具有很好的适应性和稳健性, 不需要事先确定类别数, 能通过迭代使每一个点“漂移”到密度函数的局部极大值点(周芳芳等, 2007; 李乡儒等, 2005)。近年来, 均值漂移算法越来越多地用于遥感影像分割领域(Mo 等, 2006), 并且由于其聚类准确、稳定等特点, 被用于高分辨率遥感分割和更进一步的对象化提取等遥感深层次分析过程(Huang & Zhang, 2008)。

均值漂移算法原理简单、迭代效率高, 但迭代过程中的搜索区域大小对算法的准确性和效率有很大的影像; 运算过程中需要将图像映射到高维空间, 对内存容量要求较高, 单机无法承担大数据量任务。对于这种情况, 就需要借助并行计算的力量。为了达到较好的分割效率, 本文采用均值漂移分割算法进行并行化实现。

2.2 并行化分割

根据图像数据所具有的数据量大、规律性强、相关性强等特点以及图像处理算法所具有的一致性、分层性、领域性、行顺序性等特点, 并行的实现模式大致有 3 种: 流水线并行、功能并行、数据并行。流水线并行是将不同的数据行陆续进入流水线中的各个功能模块, 或者单个功能的更细粒度操作; 功能并行是将数据同时进入各功能模块, 同时执行; 数据并行是将一幅图像分为若干分块, 对每一块数据施加相同的操作, 但存在图像块边缘衔接的问题(周海芳, 2003)。

流水线上的各个步骤完成的功能不同, 各功能处理的数据也不同, 兼具了功能并行和数据并行的特点, 如果设计得当, 这种并行模式能获得很高的效率, 但缺点是需要硬件的支持, 不适合目前主流的并行处理结构; 单纯的功能并行对于图像处理而言难度较大, 因为同一算法内部的各个步骤之间多数是相关的, 因此这种模式的实用性较差; 对于并行分割任务, 数据并行的思想较为自然, 是因为图像具有一致性和领域性的特点, 同时这种并行模式更适合于当前主流的并行计算系统, 如 MPP (massively parallel processing, 大规模并行处理系统)、机群系统。

2.3 并行化分割实现思路

采用数据并行模式实现均值漂移分割算法的并行化过程, 首先是要设置好分割参数, 对于计算的每一个节点均采用统一的参数分割, 否则将使每一块的分割结果不一致; 然后在主节点数据划分后将不同数据块分别在不同位置(例如不同的线程、机群中的不同节点等)进行均值漂移分割; 最后由主节点进行结果合并和区域标记、对象化等后续处理, 如图 1。

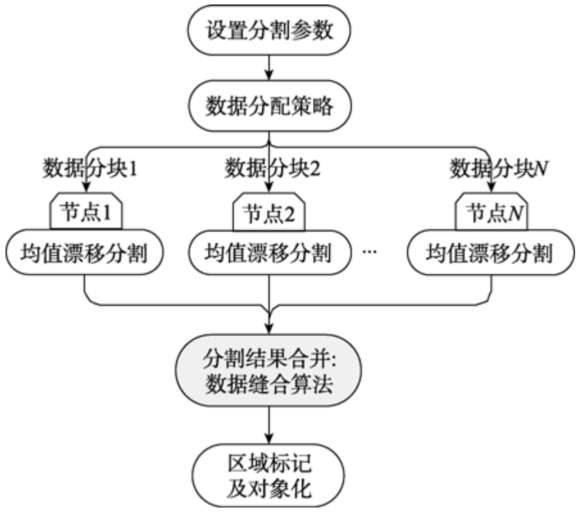

图 1 并行化均值漂移分割算法实现思路

在此过程中, 分割结果合并时存在图像分块边缘衔接的问题。由于每个分块各自的分割过程是独立的, 并没有把对方数据的特征作为滤波的参考信息, 因而彼此间并没有考虑对方数据对本块数据的影响, 不同的图像数据块具有不同的全局统计特征。因此, 在将结果进行合并和对象化后, 形成了两块数据块的分割区域轮廓不能完好对应的问题, 最终导致对象化后两个分块的边缘处有一条明显的“缝合线”。这样一来, 原本为一个对象的分割块被分成了两个或多个区域, 使分割结果不可信, 基于此的对象化分析自然也无准确性可言。因此, 这种方法虽然提高了分割算法的执行效率和处理能力, 但结果并没有达到应有的效果, 失去了并行化分割的意义。因此, 以下用一种新的数据缝合算法解决此问题。

2.4 数据缝合

与数据处理中的镶嵌(朱述龙&钱曾波, 2002)和 GIS 中的接边(Michael 等, 2009)不同, 数据缝合算法要处理的既不是像元级的影像也不是矢量数据, 而是分割图像。虽然分割图像也是以栅格形式存储, 但它并不以像元为单元, 而是以若干相邻的像元组合成的聚合块作为基本单元。本文的数据缝合算法是将分割图像中“缝合线”两侧的聚合块合并或重新划分, 形成无缝衔接并且可信的并行分割结果。由此可见, 本文的数据缝合算法是在高性能遥感信息计算中从遥感数据转化为空间信息的一个重要问题。

3 数据缝合算法

3.1 问题说明与分析

为了更好地说明分块分割结果的边缘衔接问题，图 2 示意了高分辨率遥感影像在并行分割后分块收集时产生的“缝合线”(箭头所示的水平线)。仔细观察发现，对于被“缝合线”隔开的候选分割区域，和周围差异明显的区域能同时在两个分割过程中被聚类，因而其轮廓能较好地对应，只是由于标号值不同产生了差异；然而，大部分的候选区域经过不同的分割过程后产生了无论从轮廓还是标号来看差异较大的结果(图中圈出部分尤为明显)。从图 2 看出，原本应是成片的路面、草地、建筑等区域却被“缝合线”“分”开了，若在此基础上用形状因子、面积因子、轮廓因子等特征进行对象化分析，其结果必然是不可靠的。因此，如何有效地消除这条线关系到随后对象化分析过程的精度和准确性，也是本文的数据缝合算法需要解决的问题。

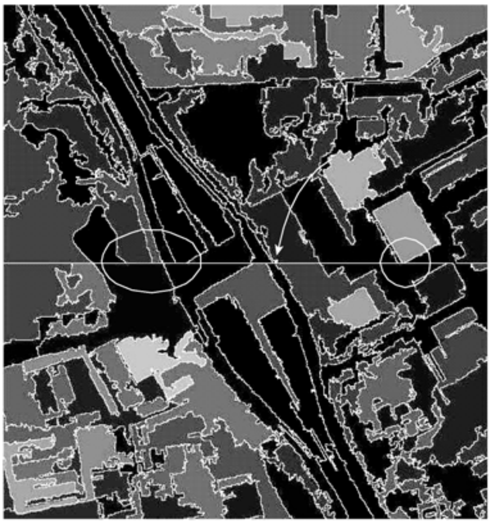

图 2 并行分割后产生的“缝合线”

将“缝合线”与两侧分割块轮廓的交点分为两类，如图 3。两侧的分割块轮廓在“缝合线”交汇的点称为 a 类交点，图像块的两个边缘处也为此类交点；单侧的轮廓与其的交点称为 b 类交点。两个相邻的 a 类交点间的线段称为 A 类线段，由图中的粗线段表示，除此之外则称为 B 类线段。

对于 A 类线段两侧的分割块根据光谱值、面积、形态等直接合并；但对于 B 类线段，情况比较复杂，由于与此相对应的分割块排列无规律可循，若简单地将所有块合并，容易造成欠分割，因此本文利用对相应区域的“次分割”达到消除“缝合线”的目的。所谓的“次分割”，也就是首先找出与 B 类线段相关的分割区域组合，形成若干小的子影像块，并对其分割，再将结果填回整体分割结果中，这样不会有太大的额外运算量，也能使缝合算法的结果可信。对于寻找与 B 类线段相关的次分割块大致分为如图 4 所示的(a) (b) (c)3 类情况，图 4(a)(b)是在两个 a 类交点之间有若干个 a 类和 b 类交点，无 A 类线段；图 4 (c)则是在一条完整的 B 类线段中包含了 A 类线段。对于这 3 种情况，需要根据不一样的搜索算法找到分割块的组合，按照其轮廓生成子影像块。

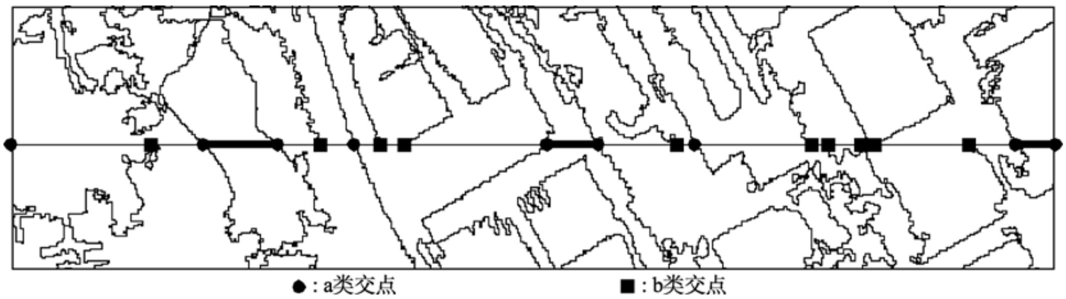

图 3 “缝合线”上的两类交点示意图

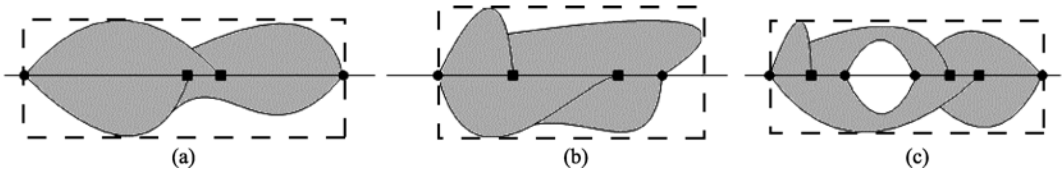

图 4 生成次分割块的三种情况

3.2　算法描述

基于以上分析，本文数据缝合算法的核心思路是要找出 A、B 两类线段，对 A 类线段两侧的分割块直接合并；对 B 类线段两侧的分割块则根据搜索算法找出子影像块进行次分割并填回数据。具体算法如下(图 5)：

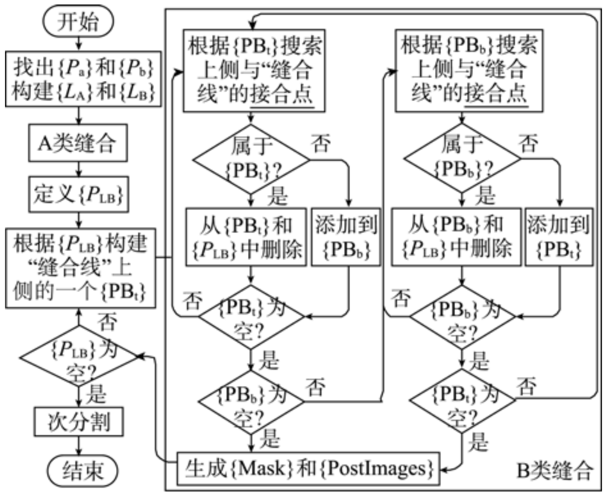

图 5　缝合算法流程图

(1) 遍历"缝合线"，找出 a 类交点集{P_a}和 b 类交点集{P_b}；

(2) 连接所有相邻的两个 P_a 作为 A 类线段集{L_A}，其余作为 B 类线段集{L_B}；

(3) A 类缝合——合并所有{L_A}两侧的分割块，分割块由 FloodFill 算法搜索得到，两侧的分割块分别简记为 $FF_t(L_A)$和 $FF_b(L_A)$, FF 的下标 t 和 b 分别表示 A 类线段上侧和下侧的分割块(这里以水平"缝合线"为例说明，因此分割块在上下两侧)，合并后的标号值由公式(1)确定，其中前缀 T 表示标号值，AR 表示面积，因此合并后的标号值应是两侧分块的加权均值；

$$T_{merge} = \frac{T(FF_t) \times AR(FF_t) + T(FF_b) \times AR(FF_b)}{AR(FF_t) + AR(FF_b)} \quad (1)$$

(4) B 类缝合——设{L_B}上的所有点组成点集{P_{LB}}(注意，不同于数学上的线段，此处的B类线段是由像素组成的，因此组成该线段的点的数量是有限的)，进行如下迭代算法：

① 取{P_{LB}}上的任一点 P 作为种子点，用 FloodFill 算法搜索上侧分割块，并将其与"缝合线"的所有接合点做上标记从而形成点集{PB_t}；

② 以{PB_t}中的任一点作为种子点搜索"缝合线"下侧分割块，搜索时遇到其与"缝合线"的接合点时，若属于{PB_t}则从中删除该点，并且删除{P_{LB}}中对应的点，否则将其添加到点集{PB_b}中，如此直至{PB_t}为空。此后若{PB_b}不为空，则执行③，否则执行④；

③ 以{PB_b}中的任一点作为种子点搜索"缝合线"上侧分割块，搜索时遇到其与"缝合线"的接合点时，若属于{PB_b}则从中删除该点，并且删除{P_{LB}}中对应的点，否则将其添加到点集{PB_t}中，如此直至{PB_b}为空。此后若{PB_t}不为空，则执行②，否则执行④；

④ 将①－③形成的连续分割区域合并，将其添加到{Mask}影像集中，并取出原始图像中与其对应的影像形成次分割影像，其余部分由黑色填充，将次分割影像添加到次分割影像集{PostImages}中。若{P_{LB}}不为空，则再跳至①。

此步骤可以把图 4 中的所有情况都顾及到，无一遗漏地形成次分割影像集。

(5) 将{PostImages}中的影像进行再一次分割，并根据{Mask}将次分割结果填充回整体分割结果中，算法完成。

此算法对"缝合线"两侧的分割块进行了合并以及重新分割，使并行分割结果更符合实际且接近于对整幅影像进行单线程运算的分割结果，使后续的信息提取等工作能更好地开展；同时，次分割影像一般比较小，在主节点中能很快运行完成，并由于{PostImages}是独立的，方便地并行化计算，对整体的并行分割不会造成太大的计算效率损失。

4　实验与效果分析

根据以上的研究和设计思路，利用多线程技术实现了基于数据并行的均值漂移分割算法的并行化，用本文的数据缝合算法处理并行分割后的影像衔接问题。

采用北京地区 1m 分辨率的 IKONOS 小幅全色影像作为实验数据，影像大小为 600×600。图 6(a)为原图，图 6(b)为并行分割后对分块数据直接拼接后的对象化结果，从中可以清楚地看到"缝合线"，虚线框为本文的数据缝合算法搜索到的次分割影像；图 6(c)为缝合算法后的对象化结果，可见，其中的"缝合线"已完全消除，其效果接近于对数据整体进行均值漂移分割后的结果，因此可信度较高。

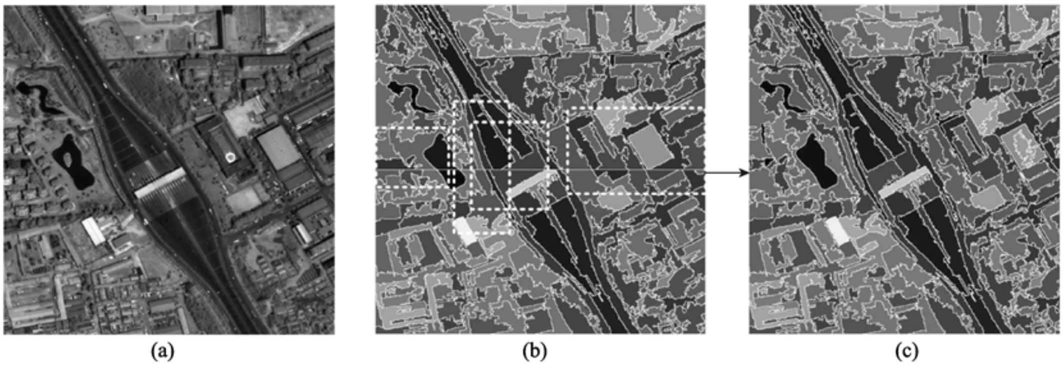

图 6 缝合算法前后对象化结果图对比
(a) 原图; (b) 并行分割结果; (c) 缝合算法处理结果

表 1 缝合算法运算时间对照表

影像大小/像元	单线程串行分割时间/s	分块数	整体并行分割时间/s	缝合算法时间/s	缝合算法占用时间百分比/%
600×600(360Kb) (图 6)	6	4	6	3	50.0
2000×2000	90	4	31	11	35.5
5000×5000(25Mb)	内存不足	6	272	47	17.3
12000×12000(144Mb)	内存不足	35	1384	182	13.2
12000×37832(446Mb)	内存不足	109	2479	244	9.8

为了进一步验证缝合算法对并行分割整体效率的影响，在微机上进行了 4 个线程并行化的实验，微机配备了Intel至强频率为3.2GHz的4核CPU，内存 2Gb。经过对不同大小的数据进行算法测试 (表 1) 后发现，缝合算法占用整体并行分割时间并不多，对其效率影响不大，随着影像的增大，所占用时间的百分比逐渐减少，同时始终保持了同样的分割效果(由于算法只针对缝合线两侧的分割块，与影像大小并无太大关系)，并使面向海量数据的分割任务成为可能，因此具有较好的可行性。

5 结 论

为了应对基于数据分块的高分辨率遥感并行分割过程中数据衔接时出现的“缝合线”问题，提出了一种新的缝合算法。从实验效果来看，该算法可使并行分割的结果接近于对影像进行整体分割的效果，使结果具有较高的可信度；并且保证并行的执行效率不受影响，具备了较好的可行性。该算法亦可用做面向大数据量遥感分割任务的数据分块串行处理，以及机群化分割过程等，具有较好的通用性。从以上可信度、可行性和通用性等的分析来看，本文的算法从一定程度上解决了从数据到信息转换过程中的瓶颈问题，具有较高的应用价值。

本文仅从算法的角度提出并验证了数据缝合算法的有效性和可行性，并没有对所有的特殊情况进行一一的考虑与解决，例如缝合线附近有长条形地物时如何确定次分割块的大小等，这些有待于进一步研究、探索与实践。

REFERENCES

Dorin C and Peter M. 2002. Mean Shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **24**(5): 603—619

Huang G M and Guo J F. 2001. Data partition for distributed-parallel processing of remote sensing imagery. *Remote Sensing Information*, (2): 9—12

Huang X and Zhang L P. 2008. An adaptive mean-shift analysis approach for object extraction and classification from urban hyperspectral imagery. *IEEE Transactions on Geoscience and Remote Sensing*, **46**(12): 4173—4185

Li X R, Wu F C and Hu Z Y. 2005. Convergence of a mean shift algorithm. *Journal of Software*, **16**(3): 365—374

Luo J C, Zhou C H and Shen Z F. 2009. A review of theory and technology for sensor information computation. *Journal of Remote Sensing*, **13**(Supplement): 1—6

Michael J S, Michael F G and Paul A L. 2009. Geospatial Analysis:

A Comprehensive Guide to Principle, Techniques and Software Tools, Second Edition (Chinese Edition). Beijing: Publishing House of Electronics Industry

Mo D K, Lin H, Li J P, Sun H and Xiong Y J. 2006. VHR imagery multi-resolution segmentation based on mean shift. *Journal of Guangxi Normal University: Natural Science Edition*, **24**(4): 247—250

Shen Z F, Luo J C, Chen Q X, Huang G Y and Sheng H. 2006. Data partition policy of high-resolution remotely sensed image parallel processing. *Journal of Harbin Institute of Technology*, **11**: 1968—1971

Zhou C H and Luo J C. 2009. Geo Computing for High Resolution Remote Sensing Images. Beijing: Science Press

Zhou F F, Fan X P and Ye Z. 2007. Mean shift research and applications. *Control and Decision*, **8**: 841—847

Zhou H F. 2003. Study and Implementation of Parallel Algorithms for Remote Sensing Image Processing. Beijing: National University of Defense Technology

Zhu S L and Qian Z B. 2002. The seam-line removal under mosaicking of remotely sensed images. *Journal of Remote Sensing*, **6**(3): 183—187

附中文参考文献

黄国满，郭建峰. 2001. 分布式并行遥感影像处理中的数据划分. 遥感信息, (2): 9—12

李乡儒，吴福朝，胡占义. 2005. 均值漂移算法的收敛性. 软件学报, **16**(3): 365—374

骆剑承，周成虎，沈占锋. 2009. 遥感信息计算研究进展. 遥感学报, **13**(增刊): 1—6

沈占锋，骆剑承，陈秋晓，黄光玉，盛昊. 2006. 高分辨率遥感影像并行处理数据分配策略研究. 哈尔滨工业大学学报，**11**: 1968—1971

周成虎，骆剑承. 2009. 高分辨率卫星遥感影像地学计算. 北京: 科学出版社

周芳芳，樊晓平，叶榛. 2007. 均值漂移算法的研究与应用. 控制与决策, **8**: 841—847

周海芳. 2003. 遥感图像并行处理算法的研究与应用. 国防科学技术大学，博士学位论文

朱述龙，钱曾波. 2002. 遥感影像镶嵌时拼接缝的消除方法. 遥感学报, **6**(3): 183—187