Fast Hierarchical Segmentation of High-Resolution Remote Sensing Image with Adaptive Edge Penalty

Xueliang Zhang, Pengfeng Xiao, and Xuezhi Feng

Abstract

A fast hierarchical segmentation method (FHS) for high-resolution remote sensing (HR) image is proposed in the paper. FHS is completely unsupervised. It is characterized by two aspects. First, the hierarchical segmentation process is accelerated by the improved linear nearest neighbor graph (LNNG) model and the segment tree model. It runs faster than other existing hierarchical segmentation methods, and can produce multi-resolution segmentations in time linear to the image size. Second, an adaptive edge penalty function is introduced to formulate the merging criterion, serving as a semantic factor. A set of QuickBird, WorldView, and aerial images is used to test the proposed method. The experiments show that the multi-resolution segmentations produced by FHS can represent objects at different scales very well. Moreover, the adaptive edge penalty function helps to remove meaningless weak edges within objects, enclosing the relation between segments and real-world objects.

Introduction

Since the within-class variability is increased in high-resolution remote sensing (HR) images, object-based image analysis (OBIA) has become the principle method to handle them (Blaschke, 2010). Image segmentation is to partition the image into a set of spatially contiguous regions. The segmented regions are viewed as image objects, which serve as the basis for OBIA.

In HR images, different objects can emerge at various scales. For example, detailed objects such as trees and houses emerge at finer scales, and main structures such as forests and urban areas are identified at coarser scales. Hence, image segmentation should be able to produce multi-scale segments and form the segment hierarchy for successive analysis (Beaulieu and Goldberg, 1989). Moreover, in the hierarchy, multi-scale objects at the same location should be nested (Benz *et al.*, 2004).

Among various segmentation methods (Pal and Pal, 1993; Cheng *et al.*, 2001; Yang and Kang, 2009; Dey *et al.*, 2010), the hierarchical method is a good choice for constructing the image object hierarchy for an HR image. It follows the stepwise optimization rule and has strong constraint for the optimization problem (Beaulieu and Goldberg, 1989). The hierarchical segmentation method can produce not just one, but a sequence of partitions, forming the segment hierarchy: in initial segmentations, detailed objects are preserved, while only main structures remain in latter segmentations. Moreover, as the coarser segments in latter segmentations are produced by merging adjacent regions in former ones, the multi-scale segments are nested. Hence, the hierarchical segmentation method is widely used for segmenting color images (Haris *et al.*, 1998; Arbeláez *et al.*, 2011), HR images (Trias-Sanz *et al.*, 2008; Gaetano *et al.*, 2009; Li *et al.*, 2010) and SAR images (Yu and Clausi, 2007; Carvalho *et al.*, 2010).

However, hierarchical segmentation is time-consuming due to the search of the most similar pair of adjacent regions within the whole image domain. In order to save the segmentation time, Kurita (1995) proposed to store dissimilarities of all the pairs of adjacent regions in a heap, rather than in a list, making the complexity decreased from P to $\log(P)$, where P was the number of adjacent region pairs. Beaulieu (1990) chose to store the best neighbor of each region in a list. Then, the length of the list was reduced to N, where N is equal to the number of regions. In the work of Haris *et al.* (1998), the nearest neighbor graph (NNG) was introduced, in which only the distance between mutual best neighbors was added in a heap. The height of heap was further reduced, but it had to scan a second-order neighborhood. A region growing engine (SEGEN) was proposed by Gofman (2006), in which the best neighbor of each segment was recorded in a priority queue. SEGEN need not scan the second-order neighborhood, but the queue height was larger than that of NNG.

In other works, different features, such as spectral homogeneity, shape (Baatz and Schäpe, 2000), texture (Ryherd and Woodcock, 1996; Hu *et al.*, 2005), and structural features (Pesaresi and Benediktsson, 2001; Akçay and Aksoy, 2008), have been used for remote sensing image segmentation. However, it is still difficult to define suitable feature with semantic meaning for HR image. The feature of edge strength was adaptively integrated in the region growing process as a semantic factor by Yu and Clausi (2008). The graduated increase edge penalty (GIEP) was proposed, and the results were appealing. But in their work, the edge penalty served as the parameter of spatial context model in the framework of Markov random field (MRF), and the incremental schedule was just determined to be experimentally satisfactory.

The objective of this paper is twofold. First, the linear nearest neighbor graph (LNNG) model is proposed to accelerate the hierarchical segmentation process. The segment tree model is adopted to represent the segment hierarchy, which

Department of Geographic Information Science and Jiangsu Provincial Key Laboratory of Geographic Information Science and Technology, Nanjing University, NO. 163, Xianlin Road, Nanjing 210023, China (xiaopf@nju.edu.cn).

Photogrammetric Engineering & Remote Sensing
Vol. 80, No. 1, January 2014, pp. 71–80.

0099-1112/14/8001–71/$3.00/0

doi: 10.14358/PERS.80.1.71

can produce multi-resolution results without repeating the time-consuming hierarchical segmentation process. Second, the feature of edge strength is adaptively integrated into hierarchical segmentation, serving as a semantic factor to enclose the relation between HR image segments and real-world objects.

Methodology

In the fast hierarchical segmentation (FHS) method, the watershed transform (Vincent and Soille, 1991) is first applied on multispectral image gradient (Xiao *et al.*, 2010) to produce over-segmented initial segmentation. Then, the region adjacency graph (RAG) (Trémeau and Colantoni, 2000; Felzenszwalb and Huttenlocher, 2004) and the linear nearest neighbor graph (LNNG) are defined on the initial segmentation. Based on the graph model, the hierarchical region merging process is performed to build the segment tree, which represents the hierarchical structure of the image. Finally, multi-resolution segmentations are produced by cutting the segment tree at different levels.

Linear Nearest Neighbor Graph

The regions in initial segmentation are labeled first (Figure 1a). Then, the RAG, $G = (V, E)$, is constructed, as shown in Figure 1b. V represents all the nodes $\{v_i\}$, and E represents the set of all the arcs $\{e_{ij}\}$. In RAG, a node v_i represents a region, and an arc e_{ij} connects two adjacent nodes v_i and v_j, indicating the adjacency between the two nodes. RAG is a weighted undirected graph. The arc weight represents the similarity between its two adjacent nodes. Setting the node number as N and the mean degree of nodes as c, then the arc number is $(c \cdot N) / 2$ in RAG. When finding the globally most similar pair of adjacent regions in hierarchical segmentation, the best arc should be searched out from all the arcs.

In order to reduce the solution space, combined with RAG, a possible NNG (Haris *et al.*, 1998) of Figure 1b can be defined as Figure 1c. NNG is a directed graph. In NNG, the out-degree of each node is equal to one. The arc starts from a node and is directed to its most similar neighbor. In addition, a cycle in NNG is defined as a sequence of connected nodes in which the starting and ending nodes coincide. However, in this paper, the node number in a cycle is limited to two, which means that a cycle is composed of two mutual best neighbors, such as the node v_1 and v_2 in Figure 1c. The globally most similar pair of adjacent nodes in RAG must belong to the cycles in NNG. Since the number of cycles is no larger than $N / 2$ (Haris *et al.*, 1998), the solution space is significantly reduced by searching the best pair of neighbors among all the cycles in NNG, rather than among all the $(c \cdot N) / 2$ arcs (Kurita, 1995) or among all the N shortest arcs of each node (Beaulieu, 1990; Gofman, 2006) in RAG. However, the NNG-based method needs to scan the second-order neighborhood of a new region.

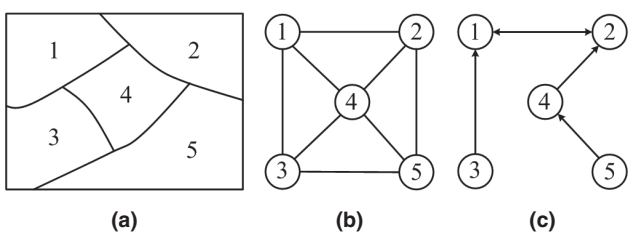


Figure 1. (a) Five-partition image, (b) corresponding RAG, and (c) a possible NNG.

Inspired by the work of Gofman (2006), the linear nearest neighbor graph (LNNG) is proposed, which performs in time linear to the neighborhood size of a new region in the region merging iteration.

The data structure of RAG and LNNG is defined as below. First, the array of arcs E is constructed, recording pointers to the adjacent nodes. Then, the array of nodes V is constructed. With each node v_i, a list L_i of pointers to incident arcs is associated. In addition, a list Q, where each member q_i points to the shortest arc connected to node v_i, is also constructed. When q_i and q_j point to the same arc e_{ij}, the corresponding nodes of v_i and v_j form a cycle, and e_{ij} is called the cycle-arc. A priority queue is constructed to record the cycle-arcs. The queue element is the pointer to cycle-arc, and the arc weight is being prioritized. The queue height B is equal to the number of cycle-arcs, which is no greater than $N / 2$.

The difference of the data structure between NNG and LNNG is the list Q recording the shortest arc for each node, which allows LNNG to avoid having to scan the second-order neighborhood.

Segment Tree from Hierarchical Merging

In hierarchical segmentation process, the globally most similar pair of adjacent regions is merged iteratively. Each merging iteration includes two stages: merging two regions, and updating the arc weight and queue.

The globally most similar pair of adjacent nodes v_i and v_j is connected by the arc e_{ij} pointed by the top of the priority queue. After merging v_i and v_j, the new node is assigned as v_i in order to reduce the computational complexity. Then, the merging stage includes: (a) remove e_{ij} in both L_i and L_j, replace node v_j to v_i in the arcs pointed by L_j, and append L_j to L_i, (b) remove the redundant arcs connecting the same adjacent nodes in L_i, and (c) recalculate the features of v_i. The process of merging two nodes is similar for different graph-based region merging algorithms.

When updating the weight of arcs connected to the new node v_i, in addition to recalculating all the arc weights, the emphasis is to re-determine the shortest arcs for the new node (q_i) and all its adjacent nodes $(\{q_k\})$. To begin with, the list L_i is scanned. For each arc e_{ki} in L_i, which connects node v_k and v_i, if its weight ω_{ki} is less than the weight of the shortest arc pointed by q_k, q_k is changed to point to e_{ki}. On the other hand, if ω_{ki} is increased, and at the same time e_{ki} is just pointed by q_k, then it needs to scan L_k to re-determine q_k by finding the new shortest arc in L_k. Altogether, if q_k is changed and at the same time the arc pointed by the old q_k is a cycle-arc, remove the old cycle-arc from the priority queue. After scanning L_i, q_i is updated by finding the shortest arc in L_i.

The first step in updating the priority queue is the removal of the arc from the top of the queue. Then, if the shortest arc connected to the new node v_i points to a cycle-arc, add it in the queue. Finally, scan $\{q_k\}$ corresponding to all the adjacent nodes $\{v_k\}$ of v_i. If any new cycle-arc emerges in $\{q_k\}$, add it in the queue.

Since the complexity of the merging stage is similar for different graph-based region merging methods, the focus is on analyzing the complexity of the updating strategy. For LNNG, the complexity of the updating stage is $O(h_S + \beta \cdot \log_2(B))$, where h_S is the total length of the lists scanned to update the arc weights in L_i, to re-determine the shortest arcs $\{q_k\}$, and to find new cycle-arcs in $\{q_k\}$. Hence, h_S is equal to the length of $2L_i + L_k$'s. β is the number of modified cycle-arcs. Compared with the NNG method, in which the complexity of updating stage is $O(h^{(2)} + \beta \cdot \log_2(B))$, the difference exists between h_S and $h^{(2)}$. The $h^{(2)}$ denotes the second-order neighborhood size of the new node, which is equal to the sum degree of the new node and all its adjacent nodes, whereas h_S in LNNG method is

linear to the neighborhood size. In terms of the SEGEN method, the complexity of updating stage is $O(h_S + p \cdot \log_2(N))$, where p is the number of modified shortest arcs in Q. Comparing LNNG with SEGEN, since the times number of updating the cycle-arcs is less than that of updating the shortest arcs, and the queue height of B is also less than N, the complexity of LNNG is significantly reduced.

At each merging iteration, the labels of two merged nodes are recorded in a list *ML*. As the merging process proceeds to that only one region remains, there are $N - 1$ records in *ML*, through which the segment tree is constructed, as shown in Figure 2. In the segment tree, a node represents a segment, and the links between nodes at different levels indicate set inclusion. The segments are getting coarser along with the increase of level. The root of the tree represents the whole image, and the leaves indicate initial segments. When cutting the tree at level k, a set of nodes, $\{S_i^k, i = 1, \ldots, N\text{-}k\text{-}1\}$, is obtained, which represents a segmentation solution. Based on *ML* and the initial segments, the key issue of cutting the segment tree is to build the set inclusion relation between the exported segments and the initial segments. The complexity of cutting the segment tree is $O(N)$. By cutting the tree at different levels, multi-resolution segmentations are produced without repeating the hierarchical merging process. The segmentation exported from higher level is at coarser scale, while that from lower level emerges at finer scale.

Merging Criterion with Adaptive Edge Penalty

The arc weight in RAG is calculated according to the merging criterion. A small arc weight indicates that two adjacent regions have larger similarity, and they are more inclined to be merged. Basically, the merging criterion integrates the features of region size and change of standard deviation (*CStd*) after a virtual merge (Baatz and Schäpe, 2000), corresponding to the segmentation scale and homogeneity, respectively. *CStd* is defined as Equation 1:

$$CStd = Std - (a_1 Std_1 + a_2 Std_2)/(a_1 + a_2), \quad (1)$$

where Std and Std_i correspond to the standard deviation of the newly created region and the original region, a and a_i correspond to the region size, respectively.

The features of *CStd* and region size do not reflect the local structure information, while the edge feature can indicate the image structure from a local sight. If the edge between two neighboring regions is weak, the neighboring regions tend to have greater possibility to be merged. Then, the local statistic, such as edge strength (*ES*), can be incorporated in the merging criterion, where a greater penalty is applied to strong edge and a lesser penalty to weak edge. Furthermore, the edge penalty term can be replaced by a monotonically increasing function of edge strength.

Since the initial watershed segmentation has one-pixel-wide region boundaries, edge strength is calculated in terms of the boundary pixels. Supposing that $R1$ and $R2$ are the two neighboring regions astride the common boundary, the edge strength of a boundary pixel (*PES*) is calculated based on the spectral difference between the $R1$ pixels and $R2$ pixels in its 8-neighborhood, as shown in Figure 3. Then, *ES* of the common boundary is calculated according to:

$$ES = \sum_{i=1}^{n} PES_i / n \quad (2)$$

where n is the pixel number of the common boundary, and PES_i is the edge strength of boundary pixel i.

As recommended by Yu and Clausi (2008), it is better to avoid choosing a single edge penalty function for a variety of scenes and applications. Moreover, since at the initial merging iterations, the segments are small and far from semantically meaningful, especially, there are some tiny structures associated with strong edges. In this case, the incorporation of strong edge penalty may produce meaningless segments, which in turn would make the segmentation even worse. Therefore, the effect of edge penalty should be weak at the initial stages and increased gradually along with the merging process. Then, the penalty function can be formulated as

$$g(ES) = \exp(-\varepsilon/ES), \quad (3)$$

where ε is the parameter adjusting the strength of edge penalty. As shown in Figure 4, supposing *ES* has been normalized to [0, 1], along with the increase of the tuning parameter ε, the

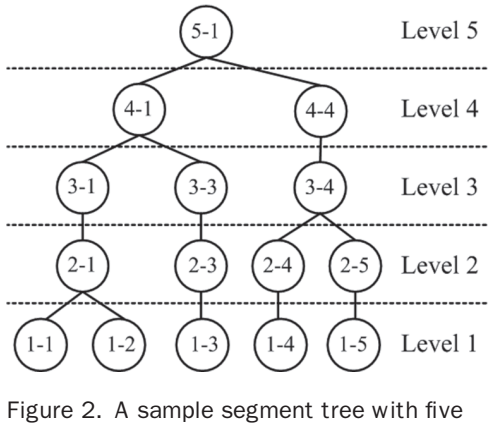


Figure 2. A sample segment tree with five initial segments.

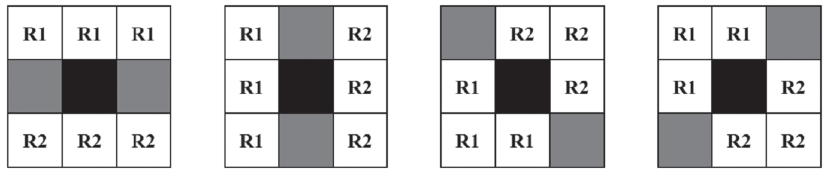


Figure 3. Samples of calculating edge strength of a pixel (*PES*). The black and gray pixels are edge pixels in the one-pixel-wide boundary. In the 8-neighborhood of the black boundary pixel, *PES* is calculated based on the spectral difference between $R1$ pixels and $R2$ pixels.

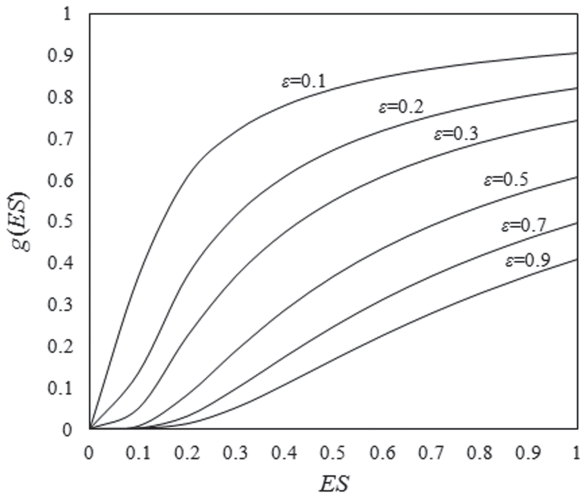


Figure 4. Edge penalty function defined by Equation 3.

edge penalty value is decreased, which means that the edge penalty makes more contribution to lessening the arc weight.

The remaining question is to determine an adaptive increment schedule for the edge penalty function parameter ε. In the hierarchical merging process, weak edges are removed first, and strong edges are preserved in the latter process. Hence, the mean edge strength of the remaining edges is increased along with the merging process (Figure 5). Moreover, the increment speed of the mean edge strength is determined by the image itself. According to extensive experiments, the mean edge strength is too significant as the tuning parameter at the initial iterations, so that the parameter ε is set as half of the mean edge strength.

Finally, the features of region size (a_1 and a_2), *CStd*, and the edge penalty function are multiplied to form the merging criterion (*MC*):

$$MC = (a_1 + a_2) \cdot CStd \cdot g(ES). \tag{4}$$

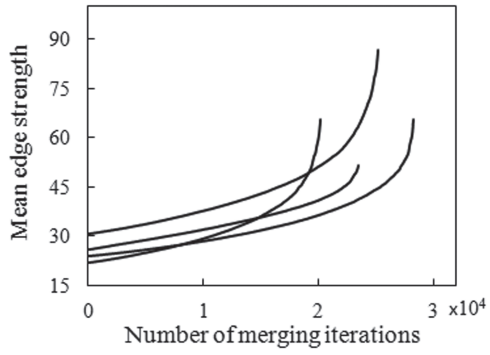


Figure 5. The increasing trend of mean edge strength along with the merging process in terms of four test images.

Experiments

Four HR images (T1, T2, T3, and T4) are used to test the FHS method. The parameters of the test images are presented in Table 1. T1 and T2 were sharpened to 0.6 m using the pan-sharpening method proposed by Zhang (2002). T1 and T2 are for an urbanized landscape, while T3 and T4 are for a rural area.

Two supervised evaluation methods are chosen to check the segmentation performance, including the method *E* (Carleer *et al.*, 2005) and the Rand index (*RI*) (Rand, 1971). Reference segmentations of T1 and T2 are produced by human interpretation, as shown in Figure 6a and 6b, respectively.

The indicator of *E* is calculated based on the number of mis-segmented pixels in the segmented images compared with the reference segmentation, indicating the segmentation precision. According to *E*, the rightly-segmented ratio (*RR*) is defined as:

$$RR = \sum_{k=1}^{N_{seg}} C_k / S. \tag{5}$$

where N_{seg} is the number of regions in the segmented result, C_k represents the number of rightly-segmented pixels in segment k, and S is the total number of pixels in the image. The rightly-segmented pixels for a segment are defined as those in the largest part when intersecting the segment with a reference region.

The indicator of *RI* is the measure of correspondence between the segmented result and the reference segmentation based on how pixel pairs are labeled in the contingency table. The *RI* indicating the agreement probability is adopted, which is defined as:

$$RI = A/(S(S-1)/2). \tag{6}$$

where the denominator represents the total number of pixel pairs in the image, and A represents the number of pairs in which the pixels are placed in the same or different labels in both the segmented result and the reference segmentation. Hence, before calculating *RI*, each region should be assigned a unique label in both the segmented result and the reference segmentation.

The multi-resolution segmentation method integrated in the commercial image analysis software eCognition® 8 is adopted to make a comparison with FHS. eCognition® uses the local mutual best region growing strategy, and the merging criterion includes the region size, spectral standard deviation, and a combined shape factor of compactness and smoothness (Baatz and Schäpe, 2000). In order to highlight the difference caused by the edge penalty function in FHS, the spectral weight of eCognition® is set as 0.9.

Table 1. Parameters of the Test Images

	Platform	Size (pixels)	Spatial resolution	Band combination	Location
T1	QuickBird	644 × 497	0.6 m	NIR, R, G	Nanjing, China
T2	QuickBird	658 × 504	0.6 m	NIR, R, G	Hangzhou, China
T3	WorldView	512 × 512	2.0 m	R, G, B	Xuzhou, China
T4	Aerial	1000 × 800	0.5 m	R, G, B	Changzhou, China

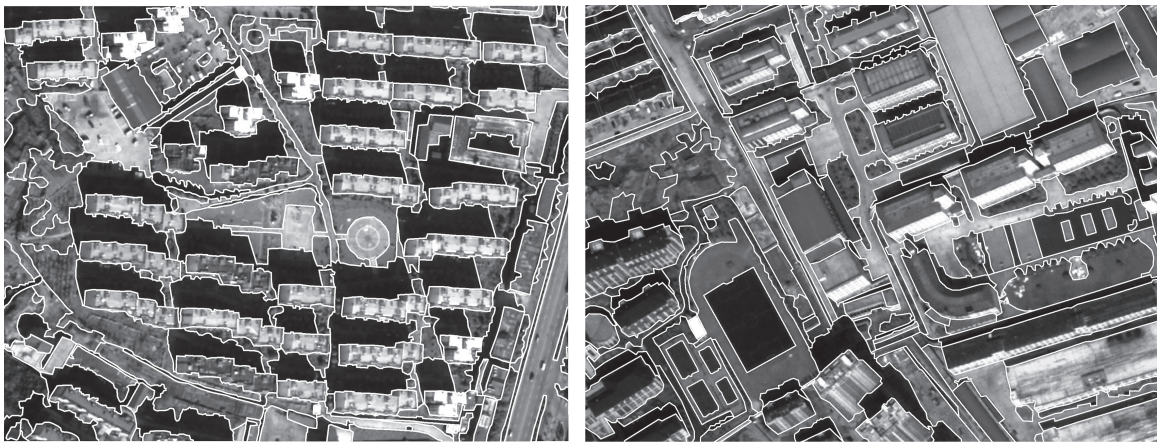


Figure 6. The reference segmentation of QuickBird test image (a) T1 and (b) T2.

Based on the reference segmentation of T1 and T2 (Figure 6), a set of multi-resolution segmentations of FHS and eCognition® are evaluated according to *RR* and *RI*. In order to show the effectiveness of the adaptive edge penalty function, the segmentation results of FHS without edge penalty are also evaluated. The evaluation results are shown in Figure 7. The region number is used to indicate the segmentation scale, where small region number indicates coarse scale and large region number corresponds to fine scale.

For the multi-resolution segmentations of T1 and T2, the *RR* differences between FHS results with edge penalty or not are weak, which shows that the edge penalty function cannot

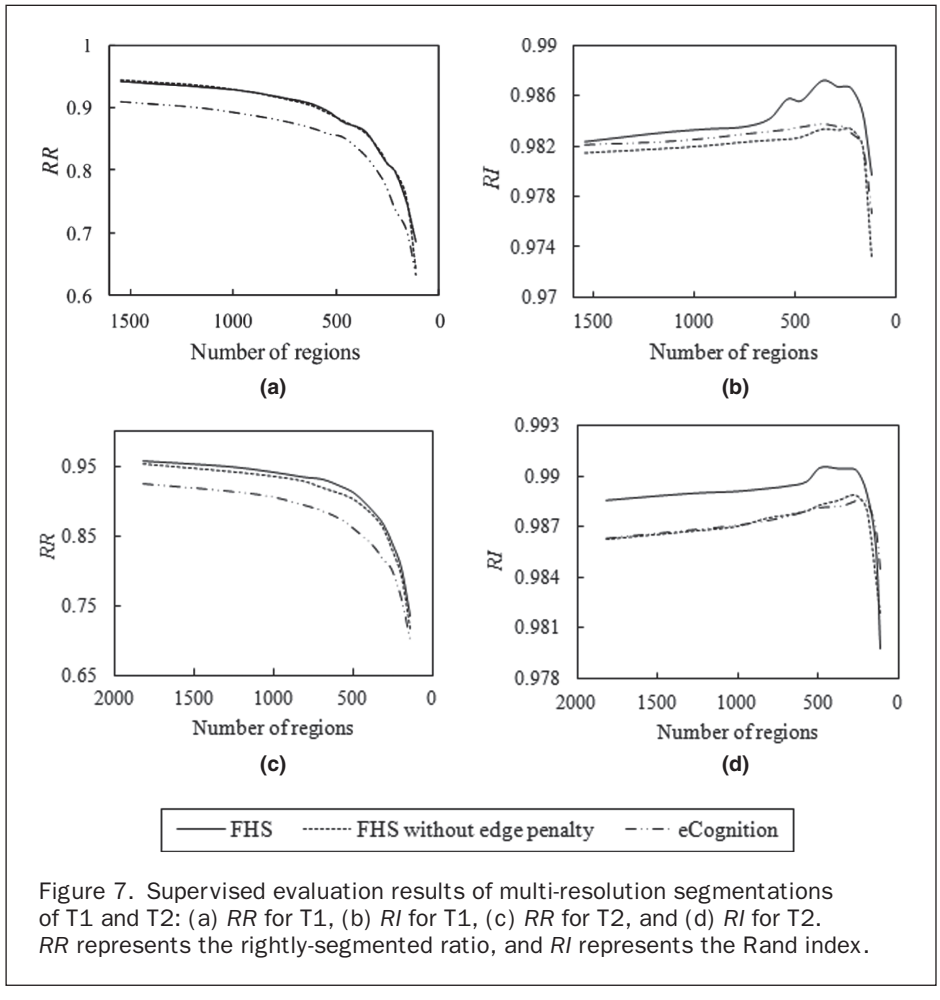


Figure 7. Supervised evaluation results of multi-resolution segmentations of T1 and T2: (a) *RR* for T1, (b) *RI* for T1, (c) *RR* for T2, and (d) *RI* for T2. *RR* represents the rightly-segmented ratio, and *RI* represents the Rand index.

improve the segmentation precision significantly. On the other hand, the *RR* values of FHS results are about 5 percent higher than those of eCognition® results. In terms of the *RI* measure, the performance of eCognition® and FHS without edge penalty is similar, but both of them are not as good as FHS, showing that the FHS results have greater correspondence to the reference segmentations. This result is owing to the semantic performance of the adaptive edge penalty function. In other words, the incorporation of adaptive edge penalty function helps to enclose the relation between the segmented result and the reference segmentation, which describes the real-world objects by human interpretation.

Typical FHS and eCognition® results of T1 are shown in Figure 8a and 8b, and those of T2 are shown in Figure 8c and 8d, respectively. The region number of FHS results are the same as those of eCognition® results. From visual assessment, both FHS and eCognition® can distinguish different objects accurately. However, the segmentation pattern of FHS seems much "cleaner" than that of eCognition®. An object would be represented by a single segment in FHS result, while it could be decomposed into several segments in the eCognition® result, especially near the boundaries. The difference mainly results from the adaptive edge penalty function in FHS, which helps to remove the "fake" edges within objects.

Figure 9 further shows the effectiveness of the adaptive edge penalty function. Figure 9b and 9d are subsets from FHS results in Figure 8a and 8c, respectively. Figure 9a and 9c are corresponding subsets of FHS results without edge penalty, respectively. The region number is set the same. Compared with the FHS results without edge penalty, some wrong segment boundaries are eliminated in FHS results. Moreover, some meaningless weak edges are removed in FHS results to generate single segments representing the real-world objects.

The multi-resolution FHS results of T3 are presented in Figure 10 to show whether they can represent the multi-scale real-world objects very well. The fine-scale result is shown in Figure 10a with 300 regions. At this scale, different objects are distinguished with each other. Even the small objects, such as boats, single houses embedded in trees, and footpaths, are described as single segments. In the medium-scale result of Figure 10b, there are 148 regions. At this scale, the small objects are merged with their adjacent similar regions. Some large objects of forest, village, road, river, and farmland are formed. When the segmentation reaches the coarse scale as in Figure 10c, there are only 70 regions remaining. The forest, bare soil, and impervious surface are segmented into single objects, and the dark-toned and light-toned farmlands are merged into large single objects, respectively.

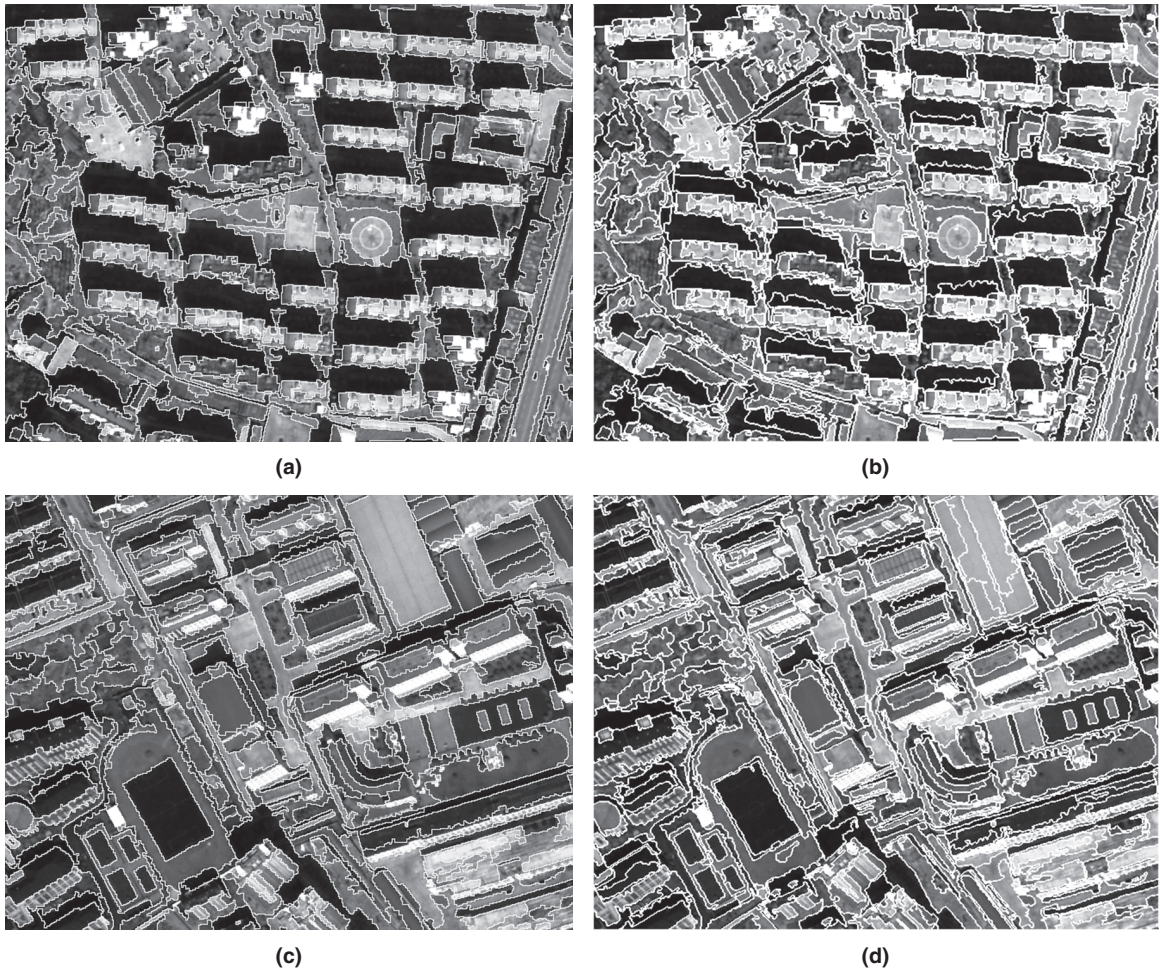


Figure 8. Visual comparison of FHS and eCognition® results: (a) FHS result of T1 with 465 regions, (b) eCognition® result of T1 with 465 regions, (c) FHS result of T2 with 429 regions, and (d) eCognition® result of T2 with 429 regions.

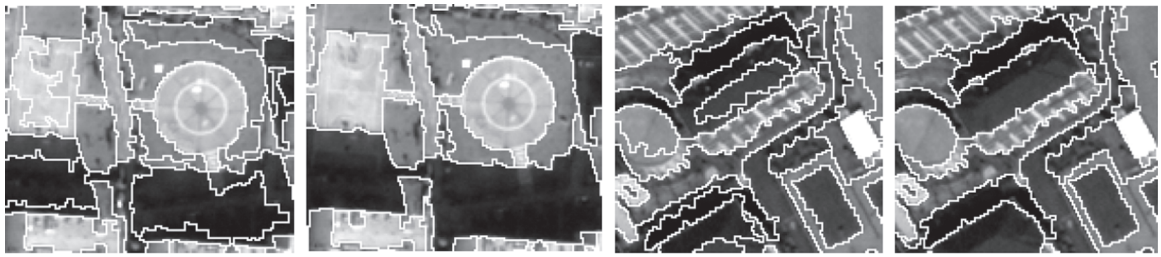
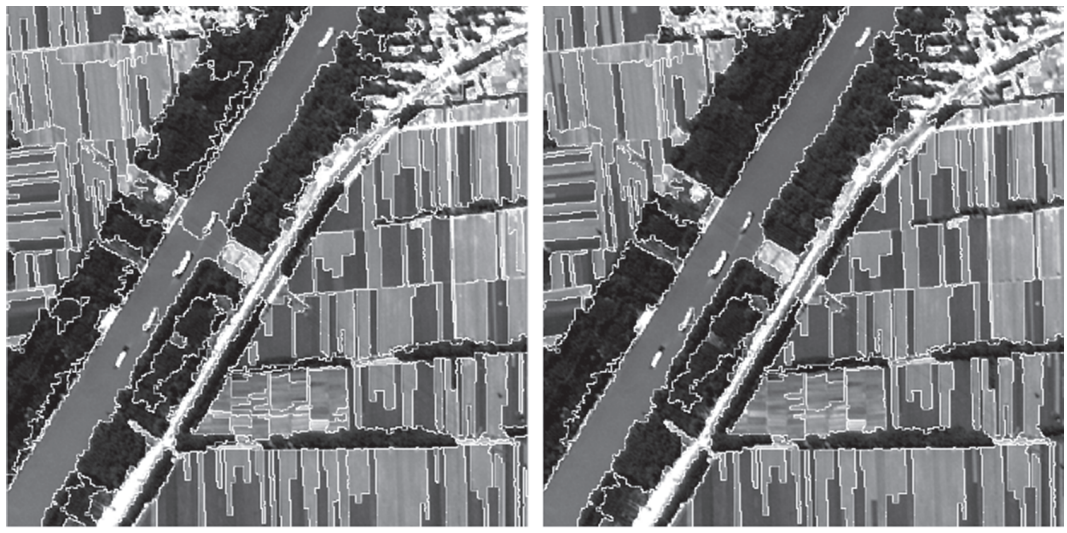


Figure 9. The effectiveness of adaptive edge penalty in FHS: (a) and (c) are FHS results without edge penalty; (b) and (d) are FHS results with edge penalty.



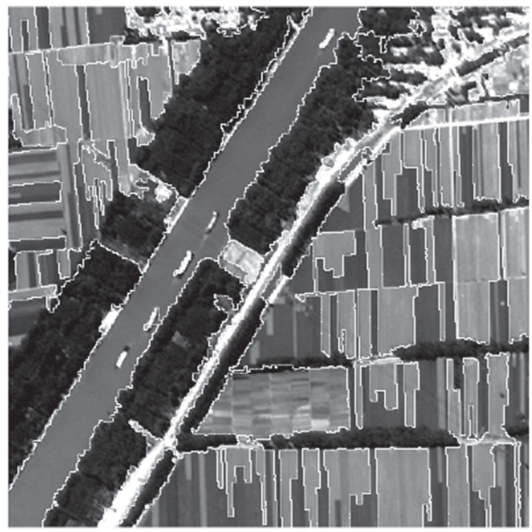


Figure 10. Multi-resolution FHS results of WorldView test image T3. The region number is 300, 148, and 70 in (a), (b), and (c), respectively.

Two FHS results of T4 are shown in Figure 11. The fine-scale segmentation result in Figure 11a has 80 regions. This result is suitable for the detailed analysis or extraction of small objects such as country lanes, embankment, small ponds and small canals because they are described as single objects. In the coarse-scale result of Figure 11b, there are only 30 regions; the large objects of farmland, forest, bare soil, river, and large ponds are segmented as single regions. At this scale, small objects are merged, while only large objects remain, which is suitable for the coarse-scale analysis.

The segmentation time performed on a laptop computer with CPU of 2.6 GHZ is shown in Table 2. Four larger test images of T'1, T'2, T'3, T'4 are used, which are from the same scenes as T1 through T4, respectively. The hierarchical region merging process based on NNG (Haris *et al.*, 1998) and SEGEN (Gofman, 2006) is applied on the same watershed initial segmentation as FHS, using the same merging criterion. The difference among NNG, SEGEN, and FHS is the complexity of the stage of updating arc weights and queue. The merging time does not take the watershed transform into account. The results show that SEGEN takes much more time than FHS. Comparing FHS with NNG, since there is no need to scan the second-order neighborhood with the benefit of LNNG, FHS reduces the segmentation time by 5 percent to 10 percent. FHS runs slower than eCognition® because the complexity of hierarchical merging strategy is higher than that of the local-oriented merging strategy. However, the segmentation time of FHS is still endurable. Moreover, it is extremely fast to cut the segment tree to produce multi-resolution segmentations.

Conclusions

The fast hierarchical segmentation method (FHS) was proposed to segment high-resolution remote sensing (HR) images in this paper. FHS is totally unsupervised, and the only user-defined parameter is the region number in segmentation result, which determines the segmentation scale directly. An adaptive edge penalty function is incorporated in FHS, using the gradually increased edge penalty function parameter of mean edge strength, which is self-adaptive to various images. Both the supervised evaluation and visual analysis of segmentation results of different test images show that the adaptive edge penalty function performs as an effective semantic factor. It cannot improve the segmentation precision significantly, but it helps to remove meaningless weak edges within objects and produce semantic segments represent the real-world objects very well.

In order to accelerate the graph-based hierarchical segmentation process, the nearest neighbor graph (NNG) model was improved to the linear nearest neighbor graph (LNNG)

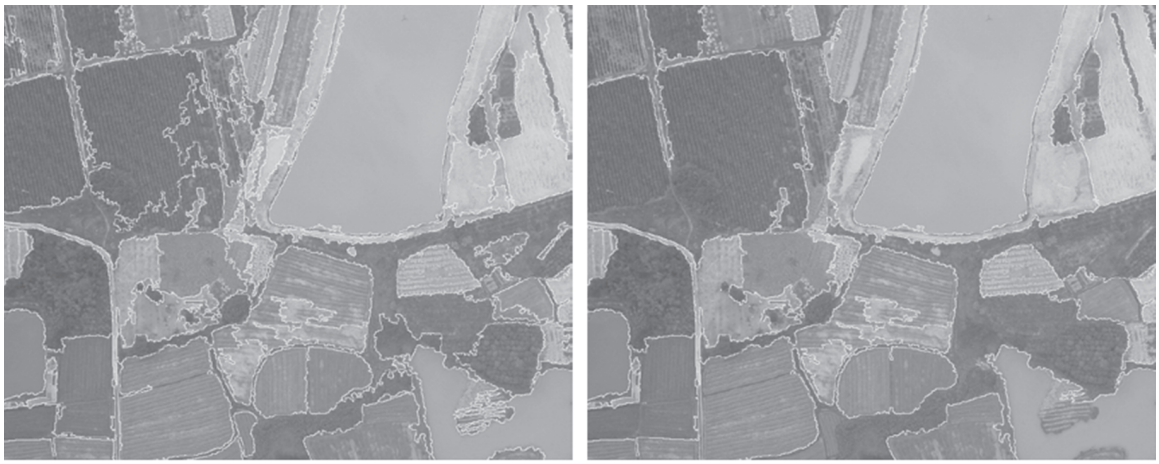


Figure 11. Multi-resolution FHS results of the Aerial test image T4. There are 80 and 30 regions in (a) and (b), respectively.

Table 2. Mean Segmentation Time Averaged Over Ten Runs.

Image	T'1	T'2	T'3	T'4
Size (pixel)	2453×2375	2781×2492	3179×2968	2000×2000
Watershed transform (s)	27	32	41	17.9
NNG (s)	112	170	284	68
SEGEN (s)	475	1296	1811	350
FHS (s)	**100**	**163**	**262**	**65**
Tree cutting (s)	**1.8**	**2.3**	**3.1**	**1.4**
eCognition® (s)	45	58	79	29

model. LNNG has the lowest priority queue height, and it does not need to search the second-order neighborhood during the merging iteration. The experiment shows that LNNG helps to reduce the segmentation time by 5 percent to 10 percent than NNG.

The hierarchical segmentation process acts as a tool to transform the graph model to a segment tree. Then, multi-resolution segmentations can be generated by cutting the segment tree at different levels without repeating the hierarchical merging process. The segment tree represents the nested multi-scale objects and can export multi-resolution results in time linear to the image size, which makes it quite useful because the problem of optimal scale for various objects is still unsolved. The fine-scale FHS results describe small objects very well, while large objects are unnecessarily decomposed into several segments at such scales. On the other hand, in the coarse-scale FHS results, the main structures are described as single objects, whereas some small objects are eliminated at these scales. Hence, when applying FHS results to successive analysis task, it needs to generate multi-resolution segmentation results in order to provide suitable scales for various objects.

Future research would focus on expanding the segment tree to the hierarchical representation of image content. The expansion task includes the integration of more effective features and the adaptive way of combining different features.

Acknowledgments

This work was supported by National Basic Research Program of China (Grant No. 2011CB952001), and a Project Funded by the Priority Academic Program Development of Jiangsu Higher Education Institutions. The authors wish to thank the anonymous reviewers for their constructive suggestions improving the paper.

References

Akçay, H.G., and S. Aksoy, 2008. Automatic detection of geospatial objects using multiple hierarchical segmentations, *IEEE Transactions on Geoscience and Remote Sensing*, 46(7):2097–2111.

Arbeláez, P., M. Maire, C. Fowlkes, and J. Malik, 2011. Contour detection and hierarchical image segmentation, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(5): 898–916.

Baatz, M., and A. Schäpe, 2000. Multiresolution segmentation - An optimization approach for high quality multi-scale image segmentation, 2000. Angewandte Geographische Informations-Verarbeitung XII (J. Strobl, T. Blaschke, and G. Griesebner, editors), Wichmann Verlag, Karlsruhe, 12–23.

Beaulieu, J.M., 1990. Versatile and efficient hierarchical clustering for picture segmentation, *Proceedings of the IEEE International Geoscience and Remote Sensing Symposium*, 20-24 May, Washington, D.C., Vol. 1663.

Beaulieu, J.M., M. and Goldberg, 1989. Hierarchy in picture segmentation: A stepwise optimization approach, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(2):150–163.

Benz, U.C., P. Hofmann, G. Willhauck, I. Lingenfelder, and M. Heynen, 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.

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

Bombrun, L., G. Vasile, M. Gay, and F. Totir, 2011. Hierarchical segmentation of polarimetric SAR images using heterogeneous clutter models, *IEEE Transactions on Geoscience and Remote Sensing*, 49(2):726–737.

Carleer, P., O. Debeir, and E. Wolff, 2005. Assessment of very high spatial resolution satellite image segmentations, *Photogrammetric Engineering & Remote Sensing*, 71(11):1285–1294.

Carvalho, E.A., D.M. Ushizima, F.N.S. Medeiros, C.I.O. Martins, R.C.P. Marques, and I.N.S. Oliveira, 2010. SAR imagery segmentation by statistical region growing and hierarchical merging, *Digital Signal Processing*, 20 (5):1365–1378.

Cheng, H.D., X.H. Jiang, Y. Sun, and J.L. Wang, 2001. Color image segmentation: Advances and prospects, *Pattern Recognition*, 34(12):2259–2281.

Dey, V., Y. Zhang, and M. Zhong, 2010. A review on image segmentation techniques with remote sensing perspective, *Proceedings of the ISPRS TC VII Symposium - 100 Years ISPRS*, Vienna, Austria, *International Archives of Photogrammetry and Remote Sensing* (W. Wagner and B. Székely, editors), XXXVIII (7A): 31–42

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

Gaetano, R., G. Scarpa, and G. Poggi, 2009. Hierarchical texture-based segmentation of multiresolution remote-sensing images, *IEEE Transactions on Geoscience and Remote Sensing*, 47(7):2129–2141.

Gofman, E., 2006. Developing an efficient region growing engine for image segmentation, *Proceedings of the IEEE International Conference on Image Processing*, 08-11 October, Atlanta, Georgia, 2413–2416.

Haris, K., S. Efstradiadis, N. Maglaveras, and A. Katsaggelos, 1998. Hybrid image segmentation using watershed and fast region merging, *IEEE Transactions on Image Processing*, 7(12):1684–1699.

Hu, X., C.V. Tao, and B. Prenzel, 2005. Automatic segmentation of high-resolution satellite imagery by integrating texture, intensity and color features, *Photogrammetric Engineering & Remote Sensing*, 71(12):1399–1406.

Kurita, T., 1995. An efficient clustering algorithm for region merging, *IEICE Transactions on Information and Systems*, E78-D (12):1546–1551.

Li, N., H. Huo, and T. Fang, 2010. A novel texture-preceded segmentation algorithm for high-resolution imagery, *IEEE Transactions on Geoscience and Remote Sensing*, 48(7): 2818–2829.

Pal, N.R., and S.K. Pal, 1993. A review on image segmentation techniques, *Pattern Recognition*, 26(9):1277–1294.

Pesaresi, M., and J.A. Benediktsson, 2001. A new approach for the morphological segmentation of high-resolution satellite imagery, *IEEE Transactions on Geoscience and Remote Sensing*, 39(2):309–320.

Rand, W.M., 1971. Objective criteria for the evaluation of clustering method, *Journal of American Statistical Association*, 66(336):846–850.

Ryherd, S., and C. Woodcock, 1996. Combining spectral and texture data in the segmentation of remotely sensed images, *Photogrammetric Engineering & Remote Sensing*, 62(2):181–194.

Trémeau, A., and P. Colantoni, 2000. Region adjacency graph applied to color image segmentation, *IEEE Transactions on Image Processing*, 9(4):735–744.

Trias-Sanz, R., G. Stamon, and J. Louchet, 2008. Using colour, texture, and hierarchical segmentation for high-resolution remote sensing, *ISPRS Journal of Photogrammetry and Remote Sensing*, 63(2):156–168.

Vincent, L., and P. Soille, 1991. Watershed in digital spaces: An efficient algorithm based on immersion simulations, *IEEE Transactions on Patten Analysis and Machine Intelligence*, 13(6):583–598.

Xiao, P., X. Feng, R. An, and S. Zhao, 2010. Segmentation of multispectral high-resolution satellite imagery using log Gabor filters, *International Journal of Remote Sensing*, 31(6):1427–1439.

Yang, Q.Q., and W.X. Kang, 2009. General research on image segmentation algorithms, *International Journal of Image, Graphics and Signal Processing*, 1(1):1–8.

Yu, Q., and D.A. Clausi, 2007. SAR sea-ice image analysis based on iterative region growing using semantics, *IEEE Transactions on Geoscience and Remote Sensing*, 45(12):3919–3941.

Yu, Q., and D.A. Clausi, 2008. IRGS: Image segmentation using edge penalties and region growing, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(12):2126–2139.

Zhang, Y., 2002. Problems in the fusion of commercial high-resolution satellite images as well as Landsat 7 images and initial solutions, *International Archives of Photogrammetry and Remote Sensing*, Vol. 34, Part 4, 236–242.

(Received 05 November 2012; accepted 31 January 2013; final version 30 July 2013)