A MULTIRESOLUTION REMOTELY SENSED IMAGE SEGMENTATION METHOD COMBINING RAINFALLING WATERSHED ALGORITHM AND FAST REGION MERGING

Min Wang[a]

[a]Key Laboratory of Virtual Geographic Environment (Nanjing Normal University), Ministry of Education, Nanjing, Jiangsu, 210046, China - sysj0918@126.com

KEY WORDS: Multiresolution, Image, Segmentation, Method, Algorithms

ABSTRACT:

Nowadays object oriented image analysis becomes a hot issue in the field of image processing and interpretation because of its more robust noise removing ability, more abundant image features and expertise knowledge involved in analysis. The first and most important step of object oriented image analysis is image segmentation, which segments an image into many visual homogenous parcels. Based on these parcels, which are 'objects' not 'pixels', more features can be involved which facilitates the succeeding image interpretation. In this work, a multi-resolution image segmentation method combining spectral and shape features is designed and implemented with reference to the basic ideas of eCognition, a famous object oriented image analyzing software package. The algorithm includes the following steps. 1) The initial segmentation parcels, so called the 'sub feature units' are obtained with rainfalling watershed algorithm for its fast speed and pretty good initial segmentation effects. 2) A fast region merging technique is designed to merge these sub feature units in a hierarchy way. A scale parameter is used to control the merging process, which stops a merge when the minimal parcel merging cost exceeds its power. A multi-resolution segmentation can be implemented with different scale parameters, for smaller scales means less cost while merging which create smaller parcels, and vice versa. Several experiments on high spatial resolution remotely sensed imagery are carried out to validate our method.

1. INTRODUCTION

Nowadays object oriented image analysis becomes a hot issue in the field of image process and interpretation. The basic idea of this kind of method is to segment an image into parcels, extract features from the parcels, and then complete the whole image interpretation with classifying the features. The main advantage of object oriented image analysis lies in that it deals with parcels, which are 'objects', not pixels, which causes more abundant features and spatial knowledge involved in analysis. Besides, with more robust pepper noise removing ability, it also brings more comprehensible interpretation results (Aplin *et al.*, 1999). eCognition (Definiens, 2007) is a world famous object oriented image analysis software, in which the multiresolution image segmentation method (Baatz *et al.*, 2007) is a key and patented technology, whose technological details hasn't been opened to the public yet. In order to implement our object-oriented image analysis software package for information extraction from high spatial resolution remotely sensed imagery, we design and implement a multiresolution image segmentation method combining spectral and shape features, with reference to the basic ideas of eCognition. Our method is validated with several successful experiments on high spatial resolution remotely sensed imagery.

2. METHOD PRINCIPLE AND STEPS

When grouping pixels into very small sub feature units at the beginning stage of our algorithm, it's of little use to import shape feature. In our method, an initial segmentation is firstly carried out only with spectral features to obtain the sub feature units. Shape can then be introduced into the algorithm to control the further merging of these feature units with suitable size. We use rainfalling watershed algorithm to create these sub feature units for its fair segmentation precision and very fast algorithm speed, which is important for processing remotely sensed imagery commonly with large data volumes. But mainly due to image noise, most watershed algorithms including rainfalling watershed have a serious over-segmentation shortcoming. Sometimes it causes that there exist a large number of very small parcels scattered in the output segmentation. A pre or post image processing should be carried out to remove this adverse influence for further analysis. In our work, we take the latter one, which deals with these very small units in a unified region merging way.

2.1 SUB FEATURE UNIT EXTRACTION

Watershed algorithm is a pretty good image segmentation method based on image grey values. A classical implementation of watershed is based on immersion simulation [Vincent *et al.*, 1991]. Watershed segmentation can also be implemented in a so called rainfalling manner. Its principle is to find a steepest routine of every pixel on the simulated image topographic surface, and a watershed base is defined as the pixel set whose downriver routine ends at a same altitude local minimum. The algorithm includes two main steps: 1) flooding stage: flood the image with some altitude threshold to create partial 'billabongs' to reduce the high frequency signal parts caused by noises so to suppress the over-segmentation of common watershed algorithms; 2) rainfalling stage: in order to classify a pixel which hasn't fallen into certain billabong, a rolling down route of a raindrop on that pixel is simulated, and all the pixels under this route will be grouped into one class (belong to a same watershed). After all these pixels are labelled, the segmentation will be terminated. A critical issue of rainfalling watershed segmentation implementation lies in correctly dealing with the local levels embedded in the slopes [Stoev, 2000].

According to Smet's work [Smet *et al.*, 2000], the efficiency of rainfalling watershed segmentation is superior remarkably to watershed algorithms based on immersion simulation. For this reason, the rainfalling watershed algorithm is chosen to initially segment an image into a set of so called sub feature units, which are the initial parcels with smaller sizes.

With these sub feature units, the spectrums, shapes (area, perimeter, etc.) and neighbourhood topology should be recorded to serve the following merging processes. These are fulfilled with Region adjacency graph (RAG) and nearest neighbour graph (NNG).

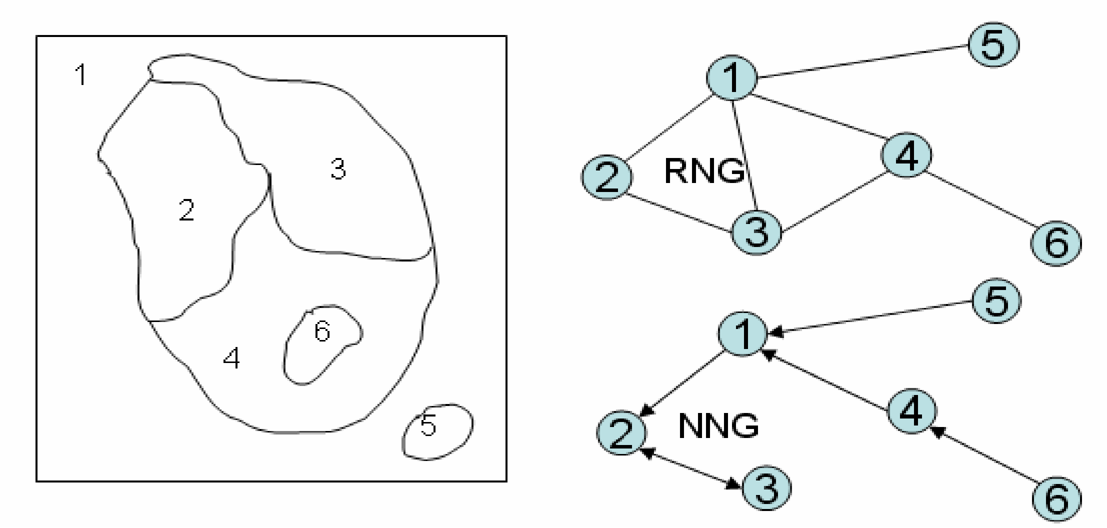

Figure 1. RNG & NNG of a parcel graph

As illustrated in Figure 1, RAG is an undirected map which can be expressed as $G = (V, E)$, in which $V=\{1,2,\ldots,k\}$ is the set of nodes, E is the set of links, and $E \subset V * V$. Every parcel is a node of the map, and a link exists if two nodes are neighbours.

Given a specified RAG and its merging cost function S, its corresponding NNG can be expressed as $G_m = (V_m, E_m)$, where V_m is just similar to V in RAG, and E_m only records the minimal merging cost of every node, which indicates NNG is a directed map. In particular, if the begin and end nodes of a link are superposed, there exists a cycle. NNG improves the merging efficiency than RAG because it obviously decreases the storage and calculation of links.

2.2 MERGING CRITERION

Based on the sub feature units, a merge cost function integrating spectral heterogeneity and shape heterogeneity is designed to guide the merging of parcels. The use of shape is to make the merged parcels more regular in shapes. With experiments, the merging cost function is similar to [Baatz *et al.*, 2007]:

$$f = w \times h_{color} + (1-w) \times h_{shape} \tag{1}$$

In which w is the weight for spectral heterogeneity falling in the interval [0, 1]. A generally suitable weight for colour is 0.9, and 0.1 for shape. Too large shape weight will bring unreasonable segmentation.

The spectral heterogeneity is the variance of the parent parcel minus the sum of the variances of the two child parcels, weighted with their respective areas:

$$h_{color} = \sum_c w_c \left(n_{Merge}\sigma_c^{Merge} - (n_1\sigma_c^1 + n_2\sigma_c^2)\right) \tag{2}$$

where, c is the band count, w_c is user specified weights for every band (1.0 by default).

The shape heterogeneity is the combination of compactness and smoothness heterogeneity:

$$h_{shape} = w_{cmpct} \times h_{cmpct} + (1 - w_{cmpct}) \times h_{smooth} \tag{3}$$

in which compactness heterogeneity is calculated as:

$$h_{cmpct} = n_{Merge} \cdot \frac{l_{Merge}}{\sqrt{n_{Merge}}} - (n_1 \cdot \frac{l_1}{\sqrt{n_1}} + n_2 \cdot \frac{l_2}{\sqrt{n_2}}) \tag{4}$$

and smoothness heterogeneity is calculated as:

$$h_{smooth} = n_{Merge} \cdot \frac{l_{Merge}}{b_{Merge}} - (n_1 \cdot \frac{l_1}{b_1} + n_2 \cdot \frac{l_2}{b_2}) \tag{5}$$

where l is the perimeter of a parcel, n is the pixels, b is the perimeter of its bounding box. A commonly suitable setting of w_{cmpct} is 0.5.

The merged parcel variance can be got with Formula 6 to avoid redundant calculation:

$$\sigma_{Merge} = \sqrt{\begin{array}{l}((n_1-1)\sigma_1^2 + (n_2-1)\sigma_2^2)/(n_{merge}-1) \\ + n_1 n_2 (m_1 - m_2)^2 / (n_{Merge})(n_{Merge}-1)\end{array}} \tag{6}$$

where m_1, m_2 are the means of the two child parcels.

2.3 FAST PARCEL MERGING

With the initial sub feature units and merging costing function, the commonly used merging schemes which can be adopted include the following two kinds:

The main steps of scheme one:

Step1. Input the RAG (NNG) of the initial segmentation;

Step2. Loop until some merging terminating condition is satisfied:

- Select the link of the minimal merging cost in the RAG (NNG);
- Merge the corresponding region pair into a new node, and delete the old pair;
- Update RAG (NNG);

Step3. Output the merged RAG (NNG) and terminate the whole merging.

The main steps of scheme two:

Step1. Input the RAG (NNG) of the initial segmenting;

Step2. Loop until all nodes are merged:

Starting from a node, search its neighbourhood nodes. If the merging cost is less than some threshold, merge them into a new node until the merging cost of the new node exceeds some specified merging threshold. Exclude these merged nodes from the following merging;

Step3. Output the merged RAG (NNG) and terminate the whole merging.

The advantage of the first merging scheme lies in that it can guarantee the current merging pair is the minimal cost one of the un-merged pairs, which thus indicates that it is a globe minimal merging cost strategy. But it has severe shortcoming of very low efficiency because the links of a merging node with all its neighbours should be rebuilt to find the minimal cost link to update the NNG, which is a very time-consuming business. Our experiments indicate that it isn't suitable for segmenting remotely sensed imagery commonly with large data volume and with the merge cost function in section 2.2.

The second merging scheme needs to visit the parcels only once, and then is of faster speed; but the problems rely in the selection of merging criterions. If area is selected as the control factor, it may cause many small parcels with distinct spectral difference with the neighbours to be merged compulsively. If the merging criterion in section 2.2 is used, it often brings a defect that sometimes a merge will be too greedy: it often merge too many unsuitable parcels because more merges will occasionally cause smaller merging costs, which makes the merging unstoppable.

With a lot of experiments, a quick merging strategy is designed to merge these sub feature units in a repetitive way. It includes the following four steps.

Step1. Input the NNG of the initial segmentation of section 2.1;

Step2. Loop until all nodes are merged:

Start from node *A*, find its pointing node *B* (its minimal merging cost node). Merge *A*, *B* if the merging cost of this node pair is under some specified threshold, and then create a new node in the output NNG. If exceeding the threshold, copy node *A* into the new NNG directly. In the above merging, if *B* has been merged before, (for example, into *C* in the new NNG), then *A* will be directly merged into *C* and no new nodes will be created.

Step3. Re-build topology for the new NNG;

Step4. Redo the above steps on the new NNG if the terminating condition hasn't been reached; otherwise output the merged RAG (NNG) and complete the segmentation.

The characteristic of the method is that we don't remove a parcel from the merging list after it is merged. That's to say that a parcel can be merged many times, until all the nodes are visited once. This merging strategy avoids the high consuming performances including topology rebuilding, merged node searching and deleting, etc. We only rebuild the topology once after the entire merging of an image has been accomplished. It's proved that this merging strategy doesn't decline the visual feeling of the segmentation, but greatly improves the algorithm efficiency.

Just similar to eCognition, a scale parameter is used to control the merging processing. If all parcel merging costs exceed the power of the scale parameter, the whole merge cycle breaks and the segmentation is over. Through experiments, we find that the minimal merging cost doesn't increase steadily with the merging times. It fluctuates, which means namely the latter merging cost sometimes maybe be lower than the former. But generally it will increase post after certain times of merging. Experiments indicate that totally after 7 to 8 iterations, the whole merging will be terminated. The scale parameter controls the iterating times, which indirectly controls the average size of the parcels. With changing the scales, a multiresolution segmentation can be realized.

3. EXPERIMENTAL ANALYSIS

Our algorithm is implemented with visual C++ 2003, and is tested on the platform of windows XP， with Pentium 4 2.93GHz CPU, 1G memory. Because image segmentation is only the first step for image information extraction, over-segmentation to some degree will not bring serious influences to succeeding analyses. Keeping this in mind, the evaluation of method precision is based on whether a method well prevents different ground objects from falling into same segmenting parcels. Several comparative experiments on different types of images such as SPOT-5, IKONOS with eCognition 5.0 segmentation module are carried out. To facilitate the comparisons, the inputs of our method and eCognition are unified to the default setting of the latter: w=0.7，w_c=1.0，w_{cmpct}=0.5.

Figure 1, 2, 3 illustrate the experimental results, in which the left graphs are the results of eCogntion, the right are of our methods. Table 1 presents the comparisons of the two methods on segmentation precision and efficiency.

With these proofs, it can be found that the two methods give similar and good segmentation in vision, and both have their respectively local visual worse-or-better segmenting parcels. Although eCognition generally produces more regular shape parcels, it often brings some fragmentized parcels distributing around the boundary of many even-tone, large-size parcels (see the pond in the upper-left corner of Figure 1, the river in Figure 2 and the playground in Figure 3). Our method doesn't have this kind of defects yet. In efficiency comparisons, our method trails eCognition. Maybe there exist two reasons that cause the lag: 1) with same scales, our method perhaps merges more times than eCognition (see the parcel number in Table 1), which causes more time consuming; 2) a lot of superfluous time is wasted on our merging steps (for example, the re-calculation of parcel topology), which may be improved with introducing spatial indexes to automatically maintain the topology between parent and child parcels.

Image	Parcel number		Time consuming		Evaluation of segmentation precision
	Our method	eCognition	Our method	eCognition	
SPOT-5 multi-spectral (687×569)	229	366	About 28s	About 8s	Both methods segment the urban, rivers, ponds, mountains etc. correctly, but our method better keeps the boundary of the ponds.
IKONOS panchromatic （1142×787）	283	515	About 65s	About 15s	The segmentations of both methods are totally similar. eCognition is more regular in shapes, but with fragmentized parcels distributing around the boundary of many even-tone, large-size parcels.
Google Earth screen capture （1208×796）	811	1778	About 58s	About 17s	Similar in segmentations. Both with some errors. eCogntion is more regular in shapes, but with the above mentioned fragmentized parcels.

Table 1. Comparison of our method and the multi-resolution segmentation module of eCognition

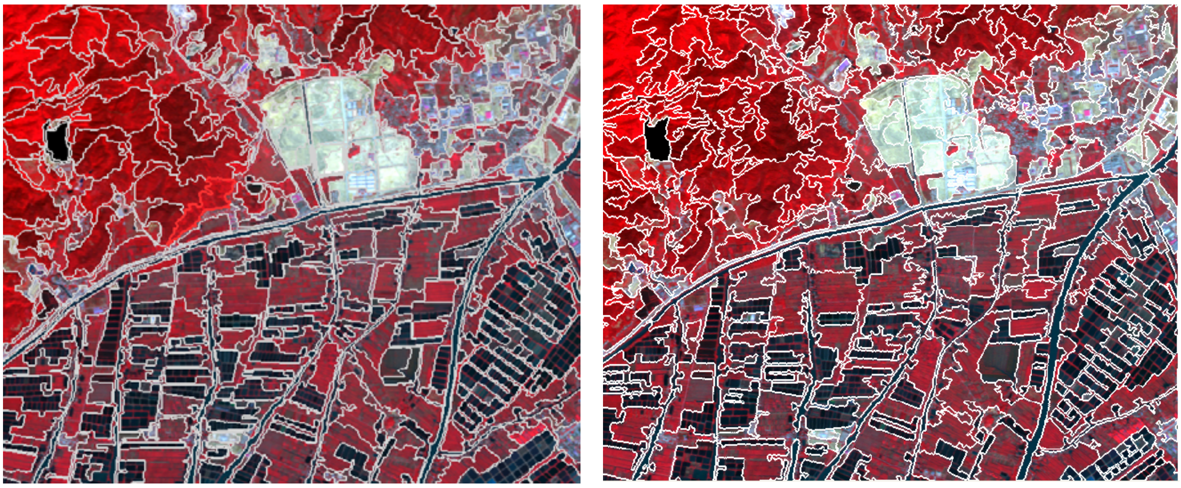

Figure 1. Segmentation of a SPOT-5 image with scale 40

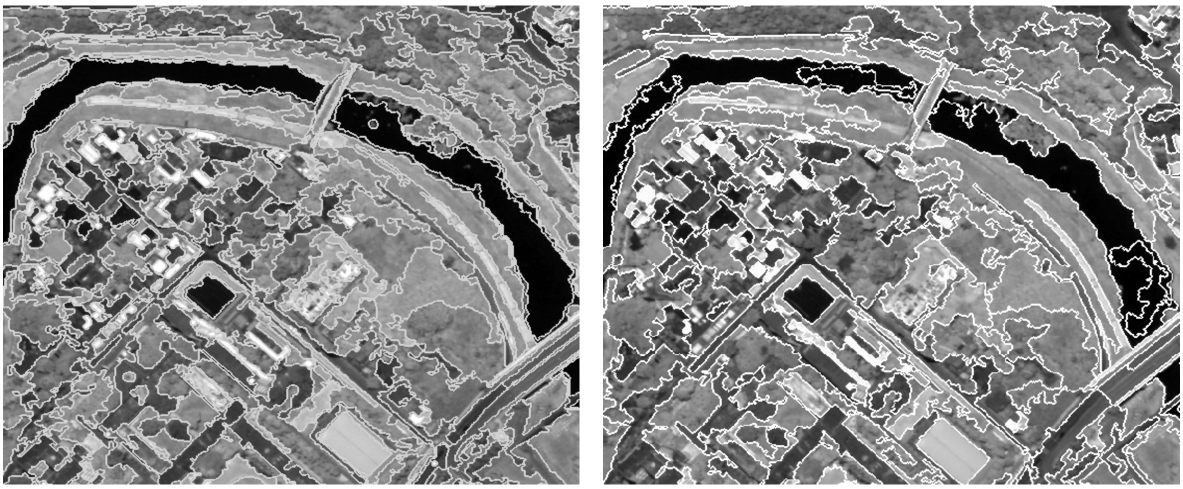

Figure 2. Segmentation of an Ikonos image with scale 80

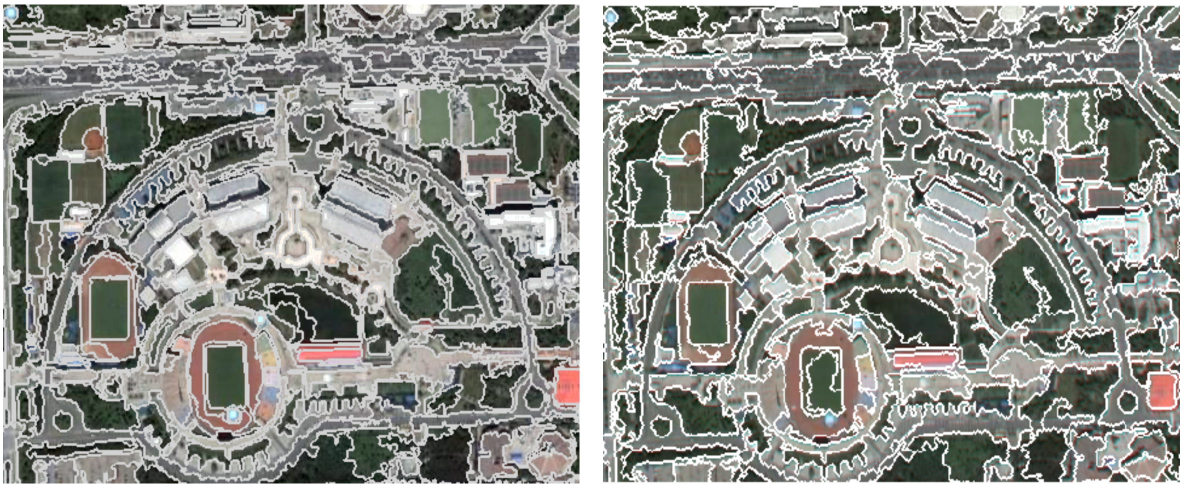

Figure 3. Segmentation of a Google Earth screen capture image with scale 50

4. CONCLUSIONS

In this work, a multi-resolution image segmentation method combining spectral and shape features is proposed, with reference to the basic idea of eCogntion. In most cases, our segmentation method produces highly visually homogeneous parcels in arbitrary resolution on different types of images. We declare that our method reaches the level of eCognition in segmenting precision, and satisfies the practical need of segmenting remotely sensed imagery with fair algorithm efficiency. As a first step for further analyses, multiresolution segmentation can be used to produce image object primitives. Starting from this, we can carry out a lot of higher level image interpretation including image classification, information extraction, and object recognition, etc.

REFERENCES

Aplin P., Atkinson P., Curran P., 1999. Per-field Classification of Land Use Using the Forthcoming Very Fine Resolution Satellite Sensors: Problems and Potential Solutions. In: Advances in Remote Sensing and GIS Analysis. Chichester: Wiley and Sons Inc., 219-239

Definiens website, 2007. http://www.definiens.com (access 2007-12-30)

Baatz M. and Schape A., 2007. Multiresolution Segmentation-an Optimization Approach for High Quality Multi-scale Image Segmentation. http://www.definiens-imaging.com (access 2007-12-30)

Stoev S. L., 2000. RaFSi – A Fast Watershed Algorithm Based on Rainfalling Simulation. In Proceedings of 8-th International Conference on Computer Graphics, Visualization, and Interactive Digital Media, Plzen City, Czech Republic

Vincent L. and Soille P., 1991. Watershed in digital spaces: an efficient algorithm based on immersion simulations. IEEE Trans. Patt. Anal. and Mach. Int., 13(6): 583-598.

Patrick De Smet and Rui Luís V.P.M. Pires, 2000. Implementation and analysis of an optimized rainfalling watershed algorithm. Proc.SPIE, 3974:759-766

ACKNOWLEDGEMENTS

This work is supported by Chinese National Programs for High Technology Research and Development (No.2007AA12Z224, 2006AA12Z146), and startup fund of Nanjing Normal University (No. 2006105XGQ0035)