Automatically Extracting Manmade Objects from Pan-Sharpened High-Resolution Satellite Imagery Using a Fuzzy Segmentation Method

Yu Li, Jonathan Li and Michael A. Chapman

Geomatics Engineering Program, Department of Civil Engineering, Ryerson University, 350 Victoria Street, Toronto, Ontario, Canada M5B 2K3T.
Email: {y6li, junli, mchapman}@ryerson.ca

Abstract

The paper describes a new method for extracting objects from high resolution color remote sensing images. This method is based on the fuzzy segmentation algorithm which has been developed in our previous works. The proposed object extraction method is following three steps. (1) Segmenting color images, (2) Detecting objects from segmented images, and (3) Post-processing of extracted object. The paper also gives experimental results from using the proposed method to extract centerlines of road networks and roofs of building from QuickBird and Ikonos Images.

1 Introduction

Very high spatial resolution satellite imagery such as 1 m Ikonos and 0.6 m QuickBird, particularly its high temporal resolution (e.g., 1-3 days), has implied that this kind of image data acquired from spaceborne sensors can provide a viable alternative to aerial photography for emergency response planning. Unfortunately, such satellite imagery has not been readily adopted by metropolitan mapping agencies and emergency response personnel for quick detection of emergency change information for planning, monitoring and damage assessment. In the applications of high resolution satellite imagery, object extraction is the most basic and important task. Though many object extraction algorithms for satellite or aerial images have been proposed in the past years, most of them processed grayscale imagery only. The objective of this paper is to present an effective approach to man-made object extraction from the pan-sharpened high-resolution satellite imagery. This method is based on the fuzzy segmentation algorithm which has been developed in our previous works. The proposed object extraction method consists of three steps: segmenting color imagery, detecting objects from the segmented imagery, and post-processing of the extracted objects. Several high-resolution satellite images with different scenes in urban residential areas have been examined and the results presented illustrate the potential of the proposed approach.

The paper is organized as follows. Section 2 introduces notations and algorithms for color segmentation from our previous work. The proposed new method for object extraction is described in Section 3. The results on extracted objects from pan-sharpened QuickBird and Ikonos color images by using the proposed method are illustrated in Section 4. Finally, conclusions are drawn in Section 5.

2 Fuzzy-Based Segmentation

2.1 New Fuzzy C-partition Method

Fuzzy c-partition algorithm has been wildly used method to solve the clustering problems in pattern recognition (Tou and Gonzalez, 1974; Zeng and Starzyk, 2001), image segmentation (Liew and Yan, 2001), unsupervised learning (Langan et al., 1998), and data compression (Zhong et al., 2000).

Consider a vector set $\boldsymbol{V}$ formed by n vectors in L-dimensional real number space R^m, i.e., $\boldsymbol{V} = \{\boldsymbol{V}_1, \boldsymbol{V}_2, \ldots, \boldsymbol{V}_n\}$, $\boldsymbol{V}_j = [\boldsymbol{V}_{j1}, \boldsymbol{V}_{j2}, \ldots, \boldsymbol{V}_{jL}]$ and $j = 1, 2, \ldots, n$, a fuzzy c-partition on $\boldsymbol{V}$ is represented by

$$\boldsymbol{P} = [p_{ij}],\ i = 1, 2, \ldots, c \text{ and } j = 1, 2, \ldots, n \tag{1}$$

where $\boldsymbol{P}$ is a fuzzy partition matrix and satisfies

$$\sum_{i=1}^{c} p_{ij} = 1,\ \ for\ \ j = 1,\ 2, \cdots, n \tag{2}$$

$$0 < \sum_{j=1}^{n} p_{ij} < \mathrm{n},\ \ \mathrm{for}\ \ \mathrm{i} = 1, 2, \cdots, \mathrm{c} \tag{3}$$

where c is the positive integer to indicate the number of the clusters in the partition, and $p_{ij} \in [0,1]$ is the fuzzy membership value of $\boldsymbol{V}_j$ belonging to ith cluster (George and Bo, 1995).

Before using fuzzy c-partition to design a clustering algorithm, the following two issues should be solved. First one is how to determinate the number of clusters for a clustering. Another issue is how to calculate the fuzzy c-partition matrix.

Unfortunately, in most of situations, the number of clusters is unknown a prior and sometimes it is difficult to specify any desired number of clusters. For example, the situations often happen in the segmentations of remote sensing images, because the ground truth is always not available for these images. Based on our previous work, a histogram-based procedure is used to obtain the number of the clusters (Li, 2004). The second issue is solved by following procedure.

Given a vector set $\boldsymbol{V} = \{\boldsymbol{V}_1, \boldsymbol{V}_2, \dots, \boldsymbol{V}_n\}$, and the number of the clusters c. Based on this number, a central vectors set can be selected, that is, $\boldsymbol{V}_{CV} = \{\boldsymbol{V}_{CV1}, \boldsymbol{V}_{CV2}, \dots, \boldsymbol{V}_{CVc}\}$ and $\boldsymbol{V}_{CV} \subset \boldsymbol{V}$ The fuzzy c-partition matrix can be calculated as follows.

$$p_{ij} = \frac{\mu(\boldsymbol{V}_{CVi}, \boldsymbol{V}_j)^{\frac{1}{m-1}}}{\sum_{k=1}^{c} \mu(\boldsymbol{V}_{CVk}, \boldsymbol{V}_j)^{\frac{1}{m-1}}}, \text{ for } i = 1, 2, \dots, c \text{ and } j = 1, 2, \dots, n \qquad (4)$$

where $m \in (1, \infty)$ is the weighting exponent on each fuzzy membership. The larger m is the fuzzier the partition is, $\mu(\boldsymbol{V}_i, \boldsymbol{V}_j)$ is a similarity measure between vectors $\boldsymbol{V}_i$ and $\boldsymbol{V}_j$ and can be calculated by

$$\mu(\boldsymbol{V}_{CVi}, \boldsymbol{V}_j) = \exp(-k_1 d(\boldsymbol{V}_{CVi}, \boldsymbol{V}_j)) \cos(k_2 \theta(\boldsymbol{V}_{CVi}, \boldsymbol{V}_j)) \qquad (5)$$

where $d(\boldsymbol{V}_i, \boldsymbol{V}_j)$ and $\theta(\boldsymbol{V}_i, \boldsymbol{V}_j)$ are the distance and the angle between $\boldsymbol{V}_i$ and $\boldsymbol{V}_j$ as follows, and k_1 and k_2 are parameters.

$$d(\boldsymbol{V}_{CVi}, \boldsymbol{V}_j) = \left(\sum_{l=1}^{L} | \boldsymbol{V}_{CVil} - \boldsymbol{V}_{jl} |^2 \right)^{1/2} \qquad (6)$$

$$\theta(\boldsymbol{V}_{CVi}, \boldsymbol{V}_j) = \arccos\left(\frac{\sum_{l=1}^{L} \boldsymbol{V}_{CVil} \boldsymbol{V}_{jl}}{\sqrt{\sum_{l=1}^{L} \boldsymbol{V}_{CVil} \sum_{l=1}^{L} \boldsymbol{V}_{jl}^{2}}} \right) \qquad (7)$$

In order to obtain the best fuzzy c-partition, a set of the best central vectors is tried to find. It is modeled as an integer programming (IP) problem as follows.

Given a vector set $\boldsymbol{V} = \{\boldsymbol{V}_1, \boldsymbol{V}_2, \ldots, \boldsymbol{V}_n\}$, and the number of the clusters c,

Max

$$\sum_{i=1}^{c} \sum_{j=1}^{n} p_{ij} \qquad (8)$$

Subject to

$$\boldsymbol{V}_{CVi} \in \boldsymbol{V} \qquad (9)$$

After finding the best centre vector set $\boldsymbol{V}_{BCV} = \{\boldsymbol{V}_{BCV1}, \boldsymbol{V}_{BCV2}, \ldots, \boldsymbol{V}_{BCVc}\}$, the best fuzzy c-partition matrix is calculated with Equation (4).

2.2 Segmentation by the Proposed Fuzzy C-Partition Algorithm

Based on the above fuzzy c-partition algorithm, the color segmentation approach is developed. The approach consists of three steps: (1) Pre-clustering. This process includes determining of the number of clusters, finding an initial centre vector set V_{CV0}, and indicating the ranges in which the centre vectors are chosen in the following optimal procedure. This procedure is finished by using a histogram-based technique. (2) Searching the best fuzzy c-partition. It is realized by solving an integer programming problem to find a good fuzzy c-partition. (3) Post-processing. It means a defuzzification procedure to convert the fuzzy c-partition matrix to the crisp c-partition matrix.

Pre-clustering. For the given color image ***CI***, the color histogram H (***CI***) can be obtained (Li, 2004). It is obvious that if an image is composed of distinct objects with different colors, its color histogram usually shows different peaks. Each peak corresponds to one object and adjacent peaks are likely to be separated by a valley. The height of a peak implies the number of the pixels falling in the bin corresponding to the location of the peak.

The pre-clustering procedure is carried out by thresholding the color histogram of a color image. For a selected threshold, the peaks having higher magnitudes than the threshold can be detected. The number of all detected peaks is chosen as the number of clusters, and the bins corresponding to the detected peaks determine the ranges in which the centre vectors are investigated for the purpose of the optimization. The initial centre vectors consist of the minimum vectors of all bins. On the other hand, they can also be produced randomly, as long as they are located in the selected bins.

The threshold is determined by either a manual or an automatic way. In the manual case, the number of clusters is determined by observing the color image and the color histogram of the image. In the automatic case, the criterion to determine the threshold should be given first. For example, the mean of all peaks can be used as the criterion. It means that the peaks with the higher magnitudes than the mean are valid.

Optimizing. To solve the optimization model introduced in the previous section, there are many methods, such as a branch-and-bound approach (Winston, 1991), and genetic algorithms (Goldberg, 1989). However, they are very time consuming and not practical in the real world. Therefore, the use of a heuristic, which gives a good but sometimes not optimal or the best solution, is necessary.

Post-processing. In order to obtain the segmented image, it is necessary to transform the fuzzy c-partition matrix to the crisp partition matrix. In this study, the following defuzzification scheme is used.

Let $\boldsymbol{P} = [p_{ij}]$ $i = 1, 2, \ldots, c$ and $j = 1, 2, \ldots, n$ be the fuzzy c-partition matrix, it is well known that p_{ij} presents the membership grade for pixel j belonging to cluster i. A percent partition matrix, $\boldsymbol{P}_p$, is defined as

$$p_{pij} = \frac{p_{ij}}{\sum_{j=1}^{n} p_{ij}} \qquad (10)$$

In terms of the percent partition, the crisp partition matrix, $\boldsymbol{P}_c = [p_{cij}]$, is defined as

$$p_{cij} = \begin{cases} 1, & p_{pij} = \max_{i=1}^{c}(p_{pij}) \\ 0, & otherwise \end{cases} \qquad (11)$$

It is clear that in the crisp-partition matrix each pixel belongs to a certain cluster.

3 Our Object Extraction Method

In this section, the segmentation method based on the fuzzy c-partition algorithm is utilized to extract objects from high-resolution remote sensing images. It consists of three main steps: (1) segmenting color images based on the above segmentation method; (2) detecting objects from segmented images; (3) post-processing of extracted object, for example, delineating road centerlines from the extracted road networks or . The discussions are mainly focused on the Steps 2 and 3.

3.1 Extraction objects

Once the segmented images are obtained by the above segmentation method, the binary object image can be extracted from it by selecting the pseudo-color corresponding to the object regions. In general, the objects in the binary image are corrupted by noise objects, which have the similar colors to objects. In order to make the object regions clear, it is necessary to filter the corrupted object image. To this end, binary morphological operations are used. For example, depending on the shapes of noise objects, the appropriate combinations of binary dilation, erosion, opening, and closing should be chosen.

3.2 Post-Processing of Extracted Objects

Delineation of road centerlines. An important process for representing the structural shape of the detected road regions is to reduce it to a graph. This work can be accomplished by a thinning algorithm. The thinning algorithm developed by Zhang and Suen (1984) for thinning binary regions is utilized in this study. It is assumed that the road pixels in the binary road network images have value 1 (black), and those background (non-road) pixels have the value 0 (white). The method consists of the successive passes of two basic steps applied to the contour pixels of the given images, where a contour pixel is any pixel with value 1 and has at least one 8-neighour value 0. With reference to the 8-neighbourhood definition shown in Figure 1, the first step indicates a contour pixel p for deletion (from black to white) if the following conditions are satisfied:

- $2 \leq N(p) \leq 6$
- $S(p) = 1$
- $p_0 \cdot p_1 \cdot p_3 = 0$
- $p_3 \cdot p_5 \cdot p_7 = 0$

where $N(p)$ is the number of nonzero neighbors of p, i.e.,

$$N(p) = \sum_{i=0}^{7} p_i \qquad (12)$$

and $S(p)$ is the number of 0-1 transitions in the ordered sequence of p_0, p_1, …, p_6, p_7.

In the second step, first two conditions remain the same, but the last two conditions are changed to

- $p_0 \cdot p_1 \cdot p_7 = 0$
- $p_0 \cdot p_5 \cdot p_7 = 0$

Extraction Build Profiles. To extract the building regions according to the color features of the buildings and uses an edge extraction algorithm to detect the skeletons of the detected buildings. To this end, a boundary extractor is designed and described in this section.

Following the definition of 8-neighborhood shown in Figure 1, the boundary pixel for building is determined if it is a contour pixel and satisfies the following condition:

- $0 < N(p) < 8$

where $N(p)$ is the number of nonzero neighbors of pixel p.

p_7	p_0	p_1
p_6	p	p_2
p_5	p_4	p_3

Fig. 1. Neighborhood arrangement

4 Experiments and Results

The proposed road extraction algorithm has been tested on two types of high-resolution satellite images, including 0.6 m QuickBird and 1 m Ikonos image data (see Figure 2). All test images have a size of 150 × 150 pixels and cover a sub-scene of a typical urban residential area in Toronto, Ontario.

Fig. 2. Tested images: (a) and (c) QuickBird, (b) and (d) Ikonos images

The pseudo-color segmented images generated from the test images shown in Figure 2 are illustrated in Figure 3.

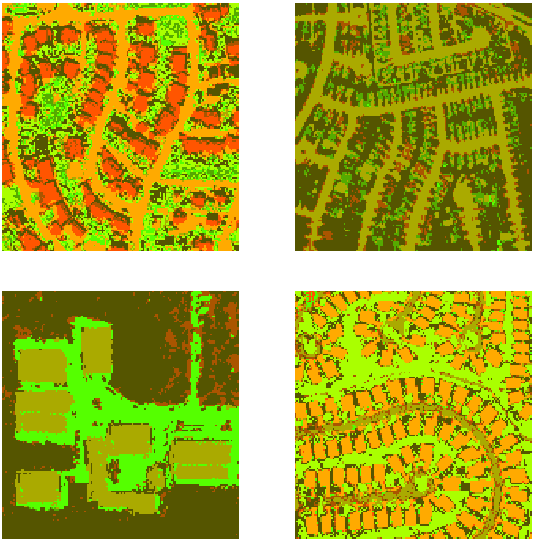

Fig. 3. Segmented images: (a) and (c) QuickBird, (b) and (d) Ikonos images

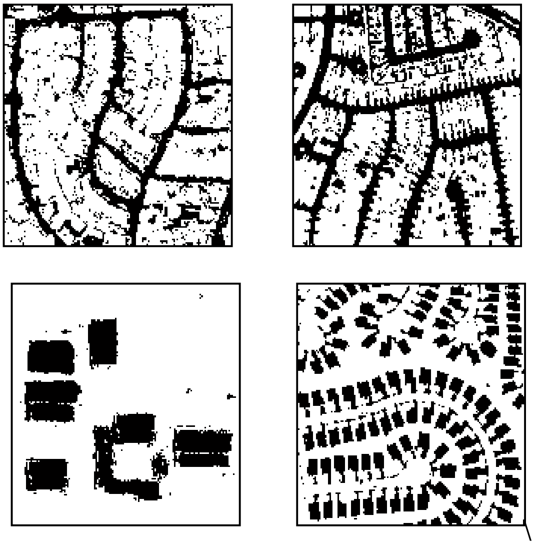

Fig. 4. Binary images of object regions: (a) and (c) QuickBird, (b) and (d) Ikonos images.

Figure 4 shows the binary images of the objects extracting from the segmentation images shown in Figure 3. It can be observed in all four images shown in Figure 4 that the segmented objects are corrupted by other objects with similar colors to objects

Figure 5 shows the object regions obtained after filtering the segmented images depicted in Figure 4 using the binary morphological operators. A visual comparison of the images clearly favors the filtered images (see Figure 5) over the segmented images (see Figure 4). Figure 5a shows the results obtained by filtering Figure 4a using binary dilating with a structuring element of 3 × 3, followed by eroding with a structuring element of 5 × 5. Figure 5b shows the results obtained by dilating Figure 4b with a structuring element of 3 × 3 and eroding with a structuring element of 5 × 5. Figure 5c shows the results obtained by closing Figure 4c with a structuring element of 4×4. Figure 5d shows the results obtained by eroding Figure 4d with a structuring element of 3×3 followed by closing with a structuring element of 4×4.

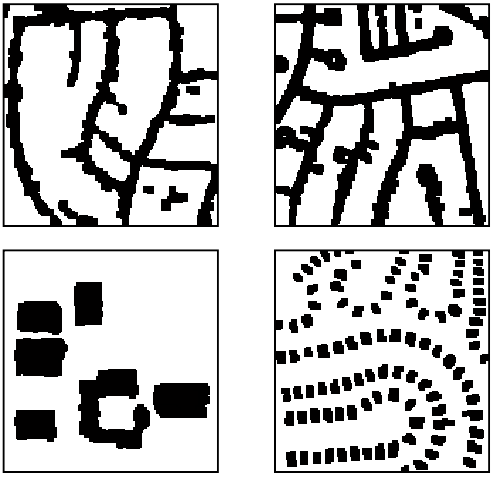

Fig. 5. Object regions after filtering: (a) and (c) QuickBird, (b) and (d) Ikonos images

The road centerlines and the edges of the extracted building roofs are delineated using the thinning algorithm and the proposed boundary extractor discussed above, and the results are shown in Figure 6.

In order to illustrate the accuracy, the extracted road centerlines and the edges of the extracted building roofs are overlaid on the original image, see Figure 7. In the overlay images the thin red lines indicate the road centerlines and the edges of the extracted building roofs. It can be obverted in

Figure 7 that most centerlines and the edges of the extracted building roofs match well the roads and buildings.

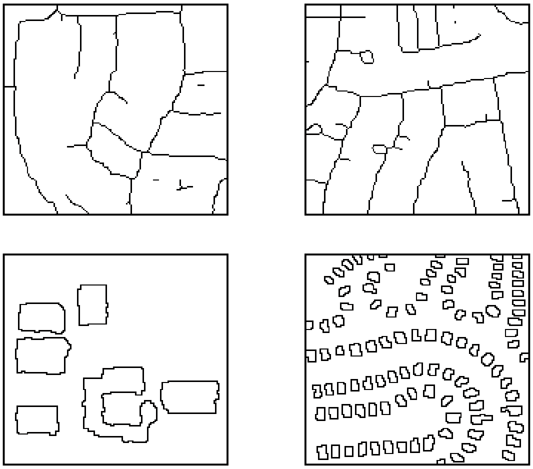

Fig. 6. Road centerlines and building roofs: (a) and (c) QuickBird, (b) and (d) Ikonos images

Fig. 7. Road centerlines and building edges (in red) overlaid on tested images: (a) and (c) QuickBird, (b) and (d) Ikonos images

5 Conclusions

A new method for extracting manmade objects from high-resolution satellite imagery such as QuichBird and Ikonos has been presented in this paper. The method employs a segmentation algorithm proposed in our previous work and works in three steps: (1) segmenting color images, (2) detecting objects from the segmented images, and (3) post-processing of the extracted objects. The proposed method has been examined by extracting road networks and buildings from pan-sharpened QuickBird and Ikonos images. The results demonstrate that the proposed method for object extraction is very effective.

Acknowledgements

This research was partially supported by a Natural Sciences and Engineering Research Council of Canada (NSERC) discovery grant.

References

George JK, Bo Y (1995) Fuzzy Sets and Fuzzy Logic: Theory and Applications, Prentice Hall PTR, Upper Saddle River, New Jersey, USA

Goldberg DE (1989) Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley¸ Reading, Mass, USA

Langan DA, Modestino JW, Zhang J (1998) Cluster validation for unsupervised stochastic model-based image segmentation. IEEE Transactions on Image Processing, 7(2), pp 180 -195

Li Y (2004) Fuzzy Similarity Measure and Its Application to High Resolution Color Remote Sensing Image Processing, Master Thesis, Ryerson University

Liew AW, Yan H (2001) Adaptive spatially constrained fuzzy clustering for image segmentation. Proceedings of 10th IEEE International Conference on Fuzzy Systems, University of Melbourne, Australia, Dec., 2001, Vol 2, pp 801-804

Tou JT, Gonzalez RC (1974) Pattern Recognition Principles, Addision-Wesley, Reading, Mass, USA

Winston WL (1991) Introduction to Mathematical Programming: Applications and Algorithms, Duxbury Press, Belmont, Ca, USA

Zhang TY, Suen CY (1984) A fats parallel algorithm for thinning digital patterns. Communications of the ACM, 27(3), pp 236-239

Zhong JM, Leung CH, Tang YY (2000). Image compression based on energy clustering and zero-quadtree representation. IEE Proceedings on Vision, Image and Signal Processing, 147(6), pp 564 –570