

Research Article

Individual Building Rooftop and Tree Crown Segmentation from High-Resolution Urban Aerial Optical Images

Jichao Jiao and Zhongliang Deng

School of Electronic Engineering, Beijing University of Posts and Telecommunications, Beijing 100876, China

Correspondence should be addressed to Jichao Jiao; jiaojichao@bupt.edu.cn

Received 2 September 2015; Accepted 8 December 2015

Academic Editor: Chimam Kwan

Copyright © 2016 J. Jiao and Z. Deng. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

We segment buildings and trees from aerial photographs by using superpixels, and we estimate the tree's parameters by using a cost function proposed in this paper. A method based on image complexity is proposed to refine superpixels boundaries. In order to classify buildings from ground and classify trees from grass, the salient feature vectors that include colors, Features from Accelerated Segment Test (FAST) corners, and Gabor edges are extracted from refined superpixels. The vectors are used to train the classifier based on Naive Bayes classifier. The trained classifier is used to classify refined superpixels as object or nonobject. The properties of a tree, including its locations and radius, are estimated by minimizing the cost function. The shadow is used to calculate the tree height using sun angle and the time when the image was taken. Our segmentation algorithm is compared with other two state-of-the-art segmentation algorithms, and the tree parameters obtained in this paper are compared to the ground truth data. Experiments show that the proposed method can segment trees and buildings appropriately, yielding higher precision and better recall rates, and the tree parameters are in good agreement with the ground truth data.

1. Introduction

With the fast pace of industrialization and urbanization, 3D models are more and more necessary for urban planning, flight simulator, and military training. It is important to identify buildings and trees in high-resolution aerial photographs for displacement maps in real-time, which is a key procedure for building 3D models, because buildings and trees not only are significant features for city modeling, but also often occlude other elements in 3D urban models. Therefore, the first step of displacement maps is to detect buildings and trees from aerial photographs. Automatic detection of trees and buildings is a challenging work because, first of all, the input data sets of aerial photographs are huge; second, the features of buildings and trees are various, and it is hard to find salient ones for training purposes; third, it is difficult to classify building from ground and classify tree from grass in desert climate regions like Arizona State in USA.

Recently, many methods have been proposed for aerial images objects detection. In this section, we will provide a description of prior art of building and tree detection from

three aspects: (1) buildings and trees segmentation; (2) superpixel refinement; (3) salient region feature.

(A) *Building and Tree Segmentation.* Building and tree segmentation from aerial image is used in different applications and analysis, for example, mapping, video games, 3D modeling, environmental management and monitoring, and disaster management. Huertas and Nevatia [1] used a generic model of shapes of structures which were rectangular or were combined by rectangular components to detect the buildings. However, they assumed the visible building surfaces consisted of smooth regions and that building sides consisted mostly of vertical structures, which makes it difficult to detect complex buildings. Liow and Pavlidis [2] proposed two methods integrating region-growing and edge detection to extract buildings. However, some small buildings were not detected. Then, Rottensteiner and Briese [3] used a hierarchical application of robust interpolation using a skew error distribution function to extract the building points from the LiDAR data. However, this method needed much time to calculate the digital terrain model (DTM). Rottensteiner

et al. [4] proposed a hierarchic approach to extract the buildings from the LiDAR data and multispectral images by using a classification. This method extracted large and small buildings. However, they put more emphasis on detecting all buildings in the test data set than on reducing false alarm rate. Cheng and his colleagues [5] proposed an algorithm of integration of LiDAR data and optical images for building roofs segmentation. However, point normal estimation needs to be used for initial segmentation, which made the algorithm complicated. In order to achieve roof segmentation from aerial images, El Merabet et al. [6] proposed a line and region-based watershed method based on the cooperation of edge- and region-based segmentation methods. However, this method is not robust to the aerial image resolution, contrast, noise, and so forth.

There are many methods for tree segmentation from aerial urban images. Iovan et al. [7] proposed a supervised classification system based on Support Vector Machines (SVM) to detect trees in high density urban areas using high-resolution color infrared images. Unlike our proposed method, this algorithm only uses textures to segment trees from buildings, which reduces the segmentation accuracy. Another pixel-level classifier based on Adaboost was implemented to segment trees from the aerial urban images [8]. However, this method cannot segment the neighbor trees. Chen and Zakhor [9] proposed a random forest classifier for the tree segmentation without using color information. In this method, the edges of buildings and trees were not discriminated because of lacking some other features. In order to accurately segment the trees, Brook and Ben-Dor [10] proposed an approach based on a sequence merge including different classifiers. However, this classifier was a complex classifier that was difficult to be implemented. Risojević et al. [11] presented an evaluated classifier based on SVM using only Gabor features. However, because of lacking colorful information, the objects with smooth structures cannot be correctly segmented. Swetnam and Falk [12] proposed a method for tree canopy segmentation and estimating tree's parameters by using LiDAR data. In order to calculate the forest structural parameters, Huang et al. [13] proposed a method for calculating the tree height and crown width by using high-resolution aerial imagery with a low density LiDAR system.

(B) FAST Superpixel Refinement. In order to overcome the undersegmentation, the number and boundaries of superpixels should be appropriately determined. A superpixel is a patch whose boundary matches the edge of an object. Superpixels represent a restricted form of region segmentation, balancing the conflicting goals of reducing image complexity through pixel grouping while avoiding undersegmentation. They have been adopted primarily by those attempting to segment, classify, or label images from labeled training data. Superpixels are segments in an image which can serve as basic units in the further image processing. Their purpose is to reduce the redundancy in the image and increase efficiency from the point of view of the next processing task. Many methods for the computation of superpixels were already presented.

Levinshtein et al. [14] presented a method to produce smooth segment boundaries. Turbopixels algorithm was a geometric-flow based algorithm for computing a dense over-segmentation of an image, often referred to as superpixels. It produced segments that on one hand respect local image boundaries, while on the other hand limiting undersegmentation through a compactness constraint. It was very fast, with complexity that was approximately linear in image size, and can be applied to megapixel sized images with high superpixel densities in a matter of minutes. However, it produced a fixed number of superpixels. Engel et al. [15] used the medial features to detect the undersegmentation region, and Sobel operator was used to detect the edges of undersegmentation regions. However, the computational burden of this method was high because of implementing iterations, which increased the run-time. Kluckner and Bischof [16] proposed an automatic building extraction method based on superpixels, which is similar to our method. However, they applied a conditional Markov random field (CRF) to refinement on superpixels twice, which was a computationally expensive process. Therefore, we proposed a fast and simple method based on image complexity to determine the number of superpixels and refine boundaries of superpixels.

Therefore, we can find that a drawback of most of these methods is their high computational complexity and hence high computational time consumption. Second, a sophisticated method for marker image calculation which respects both the remaining natural edges in the image and the regularity of marker placement in still regions of the image still has to be developed.

(C) Salient Region Feature. Salient region, which was introduced by Itti et al. [17], attracts much more attention by human brain and visual system than other regions in the image. It is inspired by the Human Vision System (HVS), and therefore the features extracted from salient regions are invariant to viewpoint change, insensitivity to image perturbations, and repeatability under intraclass variation [18]. These features are key properties for image classification. Therefore, salient region is introduced for the image classification. Hahn et al. [19] presented a representation for 3D salient region features, and these features were used to be implemented for medical image registration. The authors used a region-growing based approach for extracting 3D salient region features. In order to extract cartographic features from aerial and spatial images, Kicherer et al. [20] proposed an interactive method based on image texture and gray-level information for image region segmentation. Bi et al. [21] proposed a method for saliency tiles detection from the entire image using low-level features. However, because of selecting tiles whose edges were regular, some edges of an object were missed and much computation was needed. Therefore, we extract the salient features from superpixels, and these features are used for the aerial image classification.

(D) Research Objective. To automatically generate a realistic 3D urban model just depending on a single 2D large high-resolution urban image, we should segment and detect all the objects in the 2D image, such as trees, buildings, and streets.

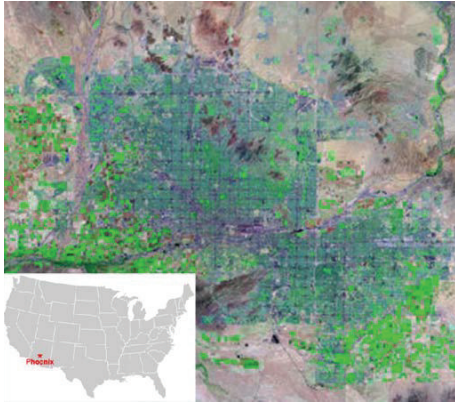


FIGURE 1: The overview area of Phoenix, USA.

However, in this work we limit our focus to buildings and trees. Our aim is to produce a label for each pixel in an aerial photograph indicating whether building or tree is present at its location. We are given a high-resolution aerial or satellite photograph from a nadir viewpoint that includes visible color (*RGB* or *HSV*) which is an important feature for building and tree segmentation. We do not consider additional raster data that may be available, such as LiDAR data, infrared, hyperspectral data, stereo data, or digital surface models, because they can be more difficult to obtain. Therefore, we claim the following specific contributions:

- (1) Proposing a method to automatically estimate the superpixels number for an aerial photo based on measurement of image complexity and detect boundaries of the undersegmentation region. The new method makes our approach robust against missing or erroneous metadata about image resolution.
- (2) Proposing a salient feature vector for training the classifier, which is simple and efficient and easy to be implemented.
- (3) Proposing an approach for estimating the location and radius of a tree crown and the tree height without additional information, which is simple and efficient and easy to be implemented.

2. Materials

2.1. Study Area. The study area is located at Phoenix, Arizona, United States, and its location is ($33^{\circ}27'N$, $112^{\circ}4'W$), which is shown in Figure 1. The area of the study region is about 291.7264 km^2 . Arizona has an abundance of mountains and plateaus in addition to its desert climate. Despite the state's aridity, 27% of Arizona is forest. The largest stand of Ponderosa pine trees in the world is contained in Arizona. Therefore, the geography of Arizona makes it more difficult to segment the buildings and trees from images than from other areas.

2.2. Aerial Imagery. The test images are from urban or sub-urban regions of Arizona, USA. These aerial images were

TABLE 1: Main EXIF information of experimental images.

Tag	Value
Manufacturer	Leica
Model	ADS40
DateandTime	2008:12:11 3:45:32
ExposureTime	1/659 seconds
ComponentsConfiguration	4.01
MaxApertureValue	2.0
FocalLength	20.1 mm
ColorSpace	sRGB
PixelXDimension	6250
PixelYDimension	6250

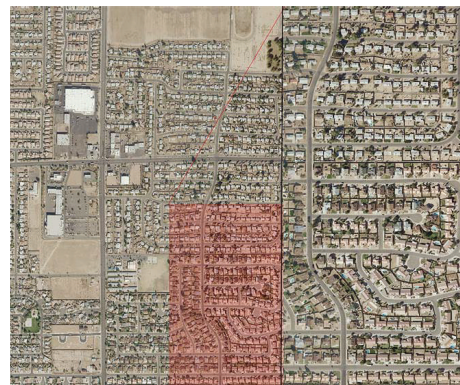


FIGURE 2: The original image with its zoom-in area.

collected at August, 2008, using the ADS40 airborne digital sensor. The height taken in the images is 600 meters. This sensor incorporates a line-array of charge-coupled devices (CCDs) and is capable of acquiring visible to near-infrared stereo data at ground resolutions of 0.5 m/pixel . The detail of this sensor can be found in [22]. The resolution of each aerial image shown in Figure 2 is 8540×8540 in this paper. The main EXIF information of experimental images is shown in Table 1. Moreover, in order to train the classifier, 50 images which are labeled trees, building, and others are used and the examples of the original labeled images are shown in Figure 3 whose resolution is 2048×1642 pixels. Furthermore, 500 images are used to evaluate the proposed algorithm.

3. Proposed Algorithm

In this paper, we propose a building and tree segmentation method using superpixels from a single large high-resolution urban image. The procedure of this algorithm is shown in Figure 4.

3.1. Presegmentation Based on Refined Simple Superpixel. In this paper, an improved method is proposed to presegment the large scale image with a high accuracy. This method is based on Turbopixels whose number of superpixels was given by users. However, different from the Turbopixels, our

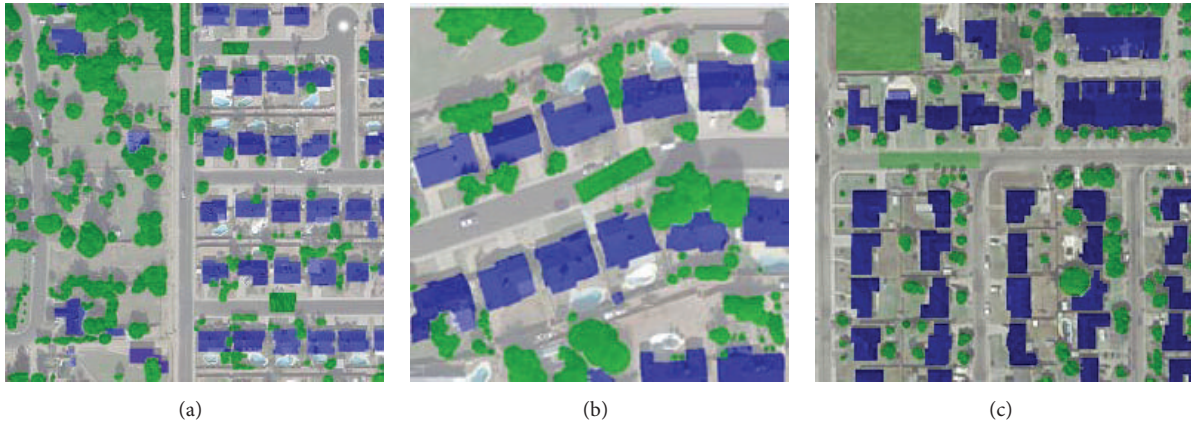


FIGURE 3: Three examples of the original labeled images for classifier training.

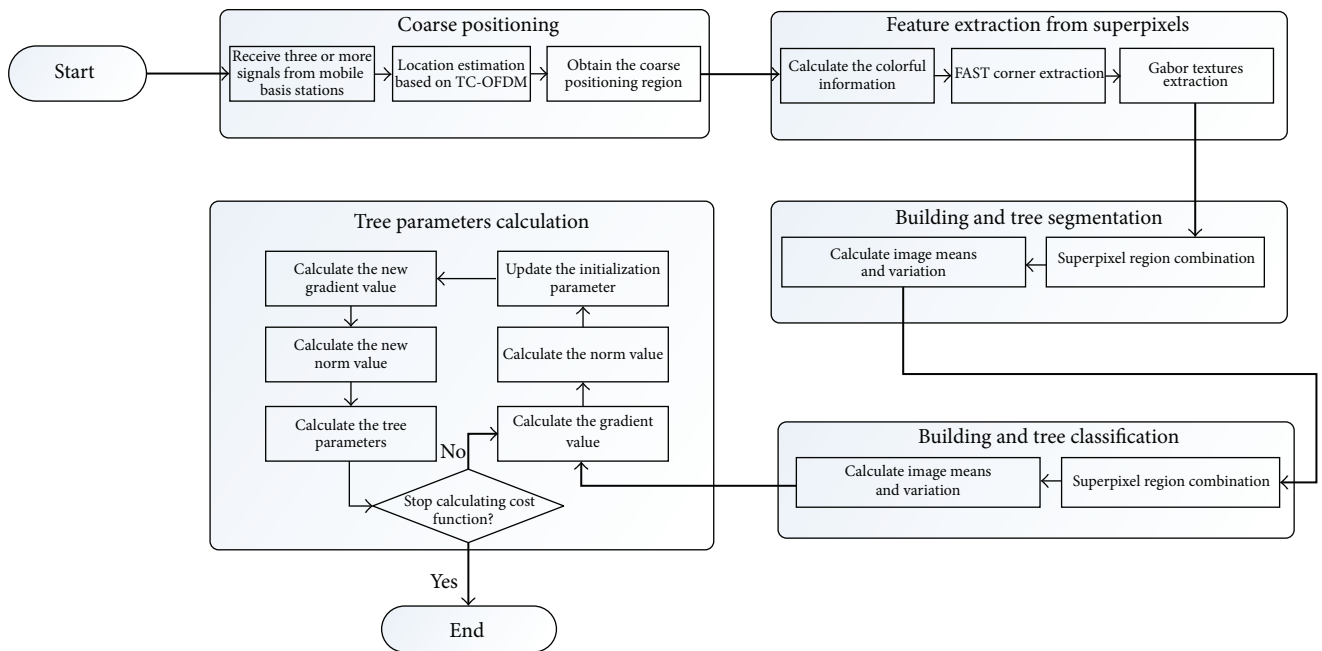


FIGURE 4: The flow chart of our algorithm.

method used two formulas to overcome the undersegmentation caused by Turbopixels.

In order to use superpixels [14, 23] to presegment images to reduce computation while preserving exact pixel boundaries, different types of images should need different numbers of superpixels. Therefore, it is required to automatically calculate the number of superpixels according to the processed images. Moreover, it is necessary to overcome the undersegmentation by refining results given by Turbopixels. Therefore, we use $N_{\text{superpixel}}$ that is defined in Section 3.1.1 to calculate the number of superpixels automatically according to the image complexity and R_{edge} is used to refine the presegmentation. This superpixel is called Refined Simple Superpixel.

3.1.1. Simple Superpixel Segmentation. The aim of superpixels is to reduce the problem by replacing pixels with regularly spaced, similarly sized image patches whose boundaries lie on edges between objects in the image, so a pixel-exact object segmentation can be accomplished by classifying superpixel patches rather than individual pixels. However, the boundaries of the superpixels will not match with the edges of the objects as well if the number of superpixels generated is too small, and the computation will be expensive when the number is too large.

In this paper, we use the image complexity to calculate the number of superpixels. Image complexity is defined as a measure of the inherent difficulty of finding a true target in a given image. The image complexity metric is determined by

the gray-level feature or the edge-level feature [24]. According to Rigau et al. [25], image complexity depended on the entropy of the image intensity histogram. Therefore, we use the entropy to measure the image complexity. The following formula is used to calculate ϵ :

$$\epsilon = - \sum p_i \log_2 p_i, \quad (1)$$

where p_i is the probability value of i th gray. This metric was originally developed for automatic target recognition. However, according to lots of experiments, we suggest another novel use of this metric to choose the number of superpixels using the following formula:

$$N_{\text{superpixel}} = \frac{\epsilon \times M}{\phi \times 100}, \quad (2)$$

where M is the image size and ϕ is a weighted value. According to (2), we can find that the number of superpixels is dependent on the image complexity and the image size. The bigger ϵ is, the more useful information is included in the aerial image. Moreover, the bigger M is, the more objects are contained in the aerial image. Therefore, $N_{\text{superpixel}}$ must be bigger, which means that more superpixels whose boundaries lie on edges between objects in the image are needed to segment the images. However, in order to obtain a balance between the segmentation accuracy and computation, a penalty weight ϕ is introduced in (2).

Testing different types of images which were collected by our lab, we suggest that the internal of ϕ is 6~13 and in our experiment ϕ is 8.1. In order to balance under- and oversegmentation, several parameters were introduced in (2). According to our research, oversegmentation or undersegmentation is caused by inappropriate numbers of superpixels. Furthermore, more details are included in the complex image. Therefore, the entropy ϵ in (2) was introduced to calculate the image complexity. Besides, a larger image includes more details than smaller one, which means that more superpixels are required for image segmentation. Therefore, the image size M was introduced in (2). Finally, the parameter ϕ was introduced to suppress the oversegmentation. As shown in Figure 2, we can find that the trees, buildings, and other objects are covered by different superpixel patches; even the trees are adjacent, which makes it easy to segment buildings and trees. From Figure 2, we can also find that most of the superpixels cover the pixels which belonged to the same objects. However, some boundaries of the superpixels are missed. Therefore, R_{edge} is implemented to detect these missing boundaries in Section 3.1.2.

3.1.2. Refined Superpixel. In this paper, we use a method based on Canny detector to overcome the undersegmentation. Canny [26] suggested that, in an image, edges were usually defined as sets of points with a strong gradient magnitude. An edge can be of almost arbitrary shape and may include junctions. Therefore, if the variance of pixels in the same superpixel is bigger than the threshold, the Canny detector should be used to refine this superpixel. Then the new boundary of the superpixel could be found; otherwise, this superpixel does not need to be refined.

In this paper, we propose a function R_{edge} to determine which superpixels need to be refined. The function is shown as follows:

$$\begin{aligned} \sigma_i > \text{thr} & \quad \text{Detection,} \\ \sigma_i \leq \text{thr} & \quad \text{No Detection,} \end{aligned} \quad (3)$$

where σ_i is the standard deviation of the i th superpixel and thr is the threshold which is obtained by the experiment. In this paper, we suggest that thr is the threshold whose interval is [9, 19] that is calculated based on [27].

Simply defining thr to be the three-dimensional feature in color space will cause inconsistencies in clustering behavior for different superpixel sizes. This produces compact superpixels that adhere well to image boundaries. For smaller superpixels, the converse is true.

3.2. Salient Features Extraction from Refined Superpixels. A superpixel-based classifier is proposed for segmenting buildings and trees from a presegmentation image. In order to train the classifier, we assign {building, tree, others} label to each superpixel to create the training data, and then a salient feature vector is derived from an entire superpixel. In this paper, we extract the color feature, corners, and texture feature for training the classifier.

3.2.1. Color Features. In our method, *RGB* and *HSV* (hue, saturation, and value) are extracted to be as color features. The input *RGB* indices fill a color-cube which the *HSV* hexcone was designed to perfectly fit [28]. *RGB* is a basic color space, and it can be transformed to any color space including *HSV*, *YUV*. At the same time, *HSV* is a more reasonable color space and it has a relationship with the Human Vision System (HVS). Therefore, they were both important for training a classifier. The following features calculated based on *RGB* and *HSV* are extracted from each superpixel:

$R_{\text{mean}}, G_{\text{mean}}, B_{\text{mean}}, H_{\text{mean}}, S_{\text{mean}},$ and V_{mean} : the means of each channel of *RGB* and *HSV*.

$R_{\text{std}}, G_{\text{std}}, B_{\text{std}}, H_{\text{std}}, S_{\text{std}},$ and V_{std} : the standard deviation of each channel of *RGB* and *HSV*.

$R_{\text{max}}, G_{\text{max}}, B_{\text{max}}, H_{\text{max}}, S_{\text{max}},$ and V_{max} : the value of the pixel with the greatest intensity in the superpixel.

$R_{\text{min}}, G_{\text{min}}, B_{\text{min}}, H_{\text{min}}, S_{\text{min}},$ and V_{min} : the value of a pixel with the lowest intensity in the superpixel.

3.2.2. Robust FAST Corners Extraction. Corner refers to a small point of interest with variation in two dimensions. There are many corner detectors which can be classified into two categories. (1) Corner detector based on edges: an edge in an image corresponds to the boundary between two areas and this boundary changes direction at corners. Many early algorithms detected corners based on the intensity changes. (2) Corner detector based on chained edge: lots of methods have been developed which involved detecting and chaining edges with a view to analyzing the properties of the edge. Rosten et al. proposed a high-speed corner detection method based on FAST (Feature from Accelerated Segment Test) and

machine learning [29], which is shown by (4). However, FAST corner is not robust because the threshold in (4) is a constant. Therefore, a novel function which is used to calculate the threshold is proposed in this paper, which is shown by (5):

$$S_l = \begin{cases} d, & I_l \leq I_{cp} - T_{\text{FAST-corner}} \text{ (darker)} \\ s, & I_{cp} - T_{\text{FAST-corner}} \leq I_l \leq I_{cp} + T_{\text{FAST-corner}} \text{ (similar)} \\ l, & I_l \geq I_{cp} + T_{\text{FAST-corner}} \text{ (lighter)}, \end{cases} \quad (4)$$

where I_l is the gray values of candidate corners in a superpixel, I_{cp} is the gray values of neighbors around I_l , and $T_{\text{FAST-corner}}$ is the FAST corner number which is calculated by the following equation:

$$T_{\text{FAST-corner}} = \frac{f^2(x_i, y_i)}{(1/(M \times N)) \sum_{i=0}^{M \times N} f(x_i, y_i)} + \sigma, \quad (5)$$

where σ is the variance of an aerial image $f(x, y)$ and $f(x_i, y_i)$ is the value of the i th pixel in the image $f(x, y)$. According to (4), we can find that the bigger σ is, the more FAST corners are extracted from the superpixels, because more salient corners are in an image whose variance is bigger.

3.2.3. Gabor Textures. In our method, we use the Gabor filter to extract structure features of buildings and trees. Structure is an important feature to segment buildings from street and ground, and it is also useful to segment trees from grass. Furthermore, frequency and orientation representations of Gabor filters are similar to those of the Human Visual System (HVS), and they have been found to be particularly appropriate for texture representation and discrimination. Among various approaches for extracting texture features, Gabor filter has emerged as one of the most popular ones [30]. A Gabor descriptor of an image is computed by passing the image through a filter bank of Gabor filter. In the spatial domain, a 2D Gabor filter is a Gaussian kernel function modulated by a sinusoidal plane wave with a complex sinusoid [31]. The function is shown as follows:

$$g(x, y) = \left(\frac{1}{2\pi\sigma_x\sigma_y} \right) \exp \left[-\frac{1}{2} \left(\frac{x^2}{\sigma_x^2} + \frac{y^2}{\sigma_y^2} \right) + 2\pi j\Omega x \right], \quad (6)$$

where Ω is the frequency of the Gabor function and σ_x and σ_y determine its bandwidth.

From the zoom-in region of Figure 2, we can find that (1) buildings have similar color with the ground and streets and (2) trees and grass have similar color. These two characteristics make it difficult to segment buildings and trees from aerial images. However, the textures of buildings and trees are much more obvious than ground and grass. Besides, in order to balance the conflicting goals of reducing computation while obtaining a satisfying accuracy, the number of frequencies ω ($\omega_1 = \sqrt{2}\pi/2$, $\omega_2 = \sqrt{2}\pi/3$, $\omega_3 = \sqrt{2}\pi/4$, $\omega_4 = \sqrt{2}\pi/5$, and $\omega_5 = \sqrt{2}\pi/6$) is 5 and number of orientations θ ($\theta_1 = 0$, $\theta_2 = \pi/8$, $\theta_3 = \pi/4$, $\theta_4 = 3\pi/8$, $\theta_5 = \pi/2$, $\theta_6 = 5\pi/8$, $\theta_7 = 3\pi/4$, and $\theta_8 = 7\pi/8$) is 8.

Through concatenating the above features based on SVM, a salient feature vector is generated for each superpixel. Then we use this feature vector to train a classifier for building and tree segmentation. Finally, Naive Bayes classifier is chosen to be used for tree and building classification in our algorithm because it is efficient and easy to be implemented [32].

3.3. Proposing a Cost Function for Tree Parameters Estimation. After segmentation, a tree location (x_T, y_T) , a tree radius r_T , and a tree height h_T are estimated by a cost function integrating shadows.

3.3.1. The Proposed Cost Function. The tree model matching is applied to 3 channels: Y , U , and V of the image $I(x, y)$. The main advantage of the YUV model is that the luminance and the color information are independent. Therefore, the luminance component can be processed without affecting the color contents [33]. In order to match trees in $I(x, y)$ with the tree model $M(x, y)$, we propose a cost function which is shown as follows:

$$\text{cost} = \arg \min (w_0 + w_Y F_Y + w_U F_U + w_V F_V + w_S F_S), \quad (7)$$

where w_0 , w_Y , w_U , w_V , and w_S are the weights to control the convergence speed of the cost function, F_Y , F_U , and F_V are used to calculate the root of the square error (RSE) of Y , U , and V channel between $I(x, y)$ and $M(x, y)$ within the tree's canopy, and F_S is used to calculate the RSE of shadows in $I(x, y)$ and $M(x, y)$ which is a circle. These functions are shown as follows:

$$\begin{aligned} F_Y(I, M) &= \gamma \sqrt{\sum_{x, y \in C(M)} [I_Y(x, y) - M_Y(x, y)]^2}, \\ F_U(I, M) &= \gamma \sqrt{\sum_{x, y \in C(M)} [I_U(x, y) - M_U(x, y)]^2}, \\ F_V(I, M) &= \gamma \sqrt{\sum_{x, y \in C(M)} [I_V(x, y) - M_V(x, y)]^2}, \\ F_S(I, M) &= \omega \sqrt{\sum_{x, y \in C_S(M)} [I_S(x, y) - M_S(x, y)]^2}, \end{aligned} \quad (8)$$

where $\gamma = 1/P$, P is a weight, and $P \in [0, 1]$,

$$C(M) = \{x, y \mid (x - x_0)^2 + (y - y_0)^2 \leq r_0^2\}. \quad (9)$$

$C(M)$ is supposed to be circular in our algorithm. (x_0, y_0) is the initializing center of the model. r_0 is the initializing radius and $\omega = 1/Q$, Q is a weight, and $Q \in [0, 1]$:

$$C_S(M) = \{x, y \mid (x - x_S)^2 + (y - y_S)^2 \leq r_0^2\}. \quad (10)$$

$C_S(M)$ is also supposed to be a circle and its radius is the same with the tree model. (x_S, y_S) is dependent on the centroid of the tree which is detected. In Section 1, we showed that there was a relationship between tree and its shadow,

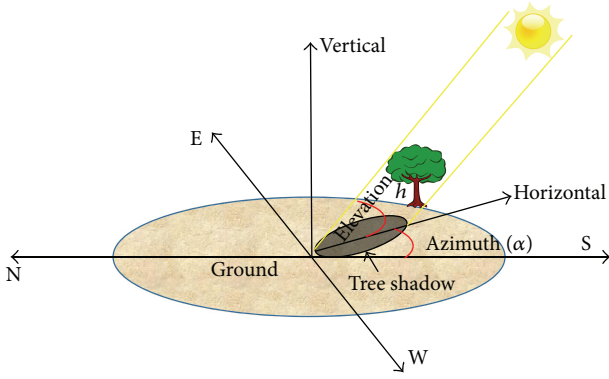


FIGURE 5: The relationship between tree and its shadow.

which is shown in Figure 5, and this relationship can be calculated by using the following functions:

$$\begin{aligned} x_S &= x_0 - \cos^{-1} \alpha \cot^{-1} \beta h_0, \\ y_S &= y_0 - \sin^{-1} \alpha \cot^{-1} \beta h_0, \end{aligned} \quad (11)$$

where α is the sun azimuth angle, β is the sun elevation angle, and they could be calculated using the time when the image was taken, h_0 is the tree height, (x_0, y_0) is the center of the tree crown, and (x_S, y_S) is the center of the shadow. Here, we estimate the tree height h using (11) and the cost function. It is noted that all these parameters are initialization parameters, and the unit of the tree parameters is pixel, because we calculate them without additional information.

3.3.2. The Outline of Calculating Tree's Parameters. In order to calculate the tree's parameter, the gradient descent method is employed to achieve iterations. The convergence speed is robust. This procedure is shown in the following steps.

Step 1. Calculate the gradient value g_0 by the initialization parameter set $\text{int}_0 = \{(x_0, y_0), r_0, h_0\}$, where (x_0, y_0) is the centroid of a superpixel, $r_0 = 2 \times x_0$, and $h_0 = 0$.

Step 2. Calculate the norm value of g_0 which can be expressed as gn_0 , where $g_0 = 5$.

Step 3. Update the int_0 using the formula $\text{int}_1 = \text{int}_0 - \text{oldspeed} \times g_0$.

Step 4. Calculate the new gradient value g_1 using int_1 .

Step 5. Calculate the norm value of g_1 which can be expressed as gn_1 .

Step 6. Calculate the ratio of g_0 and g_1 ; rate_g can be obtained.

Step 7. If $\text{rate}_g > T$, $\text{newspeed} = (\text{rate}_g/L) \times \text{oldspeed}$; else $\text{newspeed} = \text{oldspeed}$, where T is a threshold, L is a constant, and $L > 1$ and in our experiment $L = 100$. In our method, the initialization centroid (x_0, y_0) is the centroid of a superpixel that belonged to a tree. The number of iterations is 600.

This procedure is finished until getting the minimum value of tree parameters (x^*, y^*, r^*, h^*) .

Our method was implemented using Microsoft Visual Studio 2010 and Matlab 2010a, and this method was coded by integrating C# and Matlab. The proposed algorithm was tested on many large high-resolution urban images taken from Phoenix, Arizona, USA.

4. Results

In our experiment, the method is implemented by two steps: (1) the first step is presegmentation, and the result is compared with the result which is got by Turbopixel and Entropy Rate Superpixel (ERS); (2) the second is to segment buildings and trees, and the segmentation result is compared to the result got by using Turbopixel and ERS [34]; (3) the tree's parameters estimation result is evaluated by the root mean square error (RMSE).

4.1. Presegmentation Result. Figure 2 is an original image which includes trees, grass, buildings, and other objects. The zoom-in area of Figure 2 shows that the tree color is similar to grass color; the textures of trees are coarse and salient, while the textures of grass are smooth. Moreover, the shapes of buildings and trees are different. Figure 6 shows the presegmentation result. By comparing Figures 6(a), 6(b), and 6(c), we can find that the superpixel in Figure 6(a) was refined by our method. It is noted that the number of superpixels is 8200 before the aerial images are presegmented by using Turbopixels.

Figure 6(a) shows the result which is obtained by (3). The boundaries are marked by red, and the boundaries which are got by (3) are marked by green.

4.2. Segmentation Result. Figure 7 shows four segmentation results; the first one is based on our method; the second one is based on the Turbopixels; the third one is obtained by using ERS; and the last one is ground truth segmentation result. By subjectively comparing these three results, we found that the segmentation result got by our method is the best. Because of training the features from the superpixels, computation of our method is less than that of the method used by ERS. For the superpixel-level segmentation, the computational complexity is $O(\epsilon \times M/2^{10})$. However, the computational complexity of pixel-level is $O(M)$. ϵ is the image complexity, and M is the pixel number. Therefore, the running time of our method is considerable, which is very important in large scale urban aerial image processing. The average running time of our method is 45% less than the average running time of ERS.

Furthermore, two standard metrics which were undersegmentation error [35] and boundary recall [36] were used for evaluating the quality of superpixels. Undersegmentation error (UE) measures fraction of pixel leak across ground truth boundaries, which is shown by (12). The comparison results of UE were shown in Figure 8(a):

$$\text{UE}_T(S) = \frac{\sum_i \sum_{k: S_k \cap T_i \neq \emptyset} |S_k - T_i|}{\sum_i |G_i|}, \quad (12)$$

where $\mathbf{T} = \{G_1, G_2, \dots, G_n\}$ represent ground truth segmentation and G_i denotes the segment size and S_k is the k th

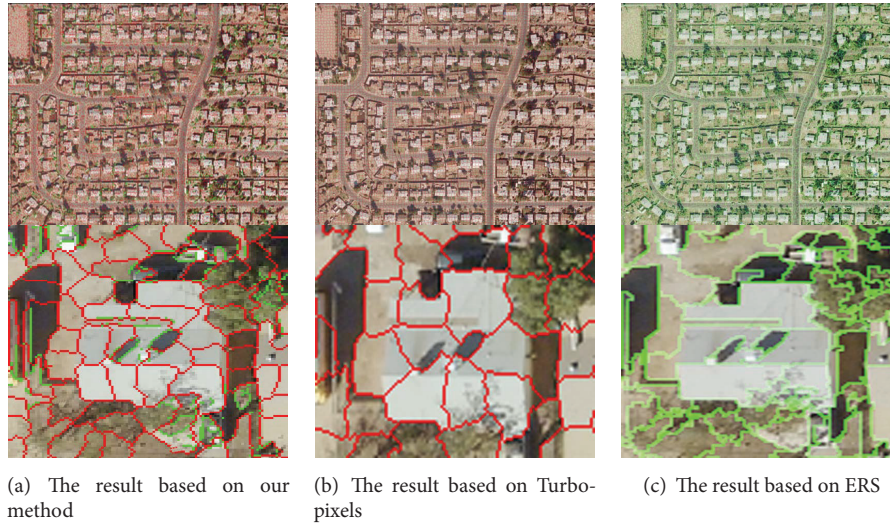


FIGURE 6: The comparative presegmentation result.

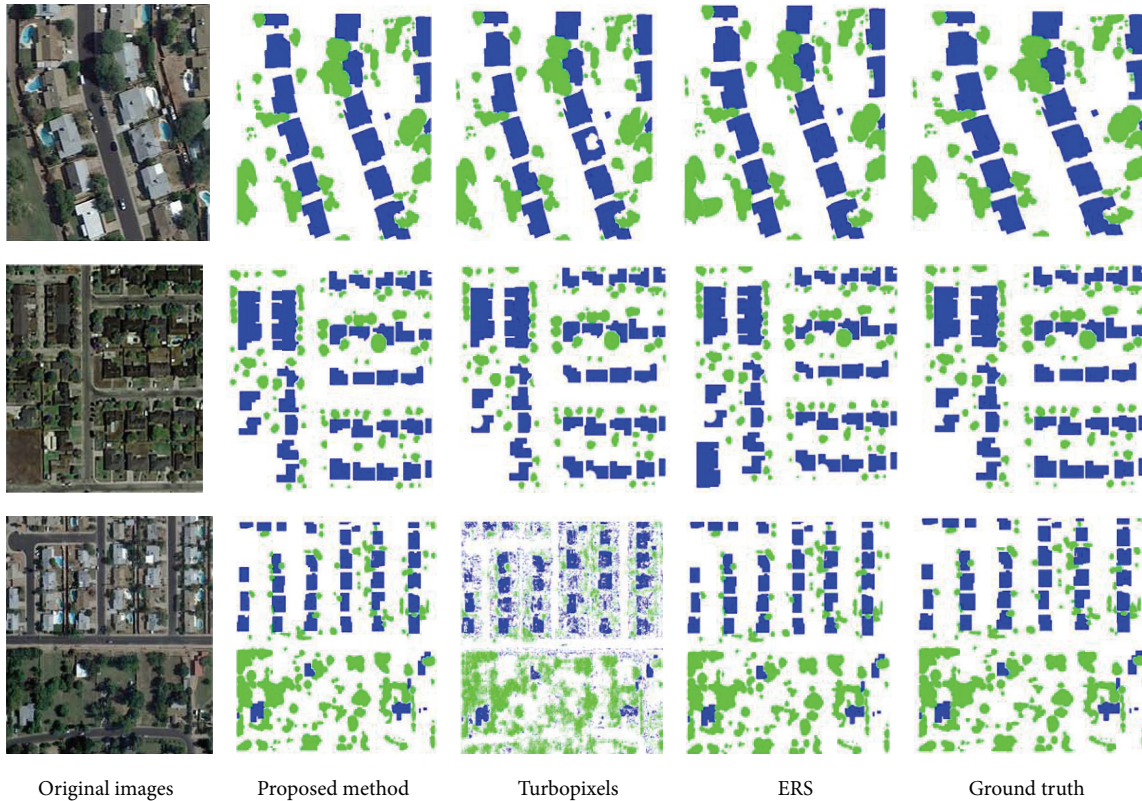


FIGURE 7: The comparative segmentation result.

superpixel. Boundary recall (BR) measures the percentage of the natural boundaries recovered by the superpixel boundaries, which is represented by (13). The comparison results of BR were shown in Figure 8(b):

$$BR_T(S) = \frac{\sum_{p \in \delta T} \nabla(\min_{q \in \delta S} \|p - q\| < \epsilon)}{|\delta T|}, \quad (13)$$

where δT and δS denote the union sets of ground truth boundaries and superpixel boundaries, respectively. The indicator function ∇ checks if the nearest pixel is within distance. In our experiments we set $\nabla = 2$. These performance metrics are plotted against the number of superpixels in an image.

According to Figure 8, we can find that the proposed algorithm in this paper performs significantly better than

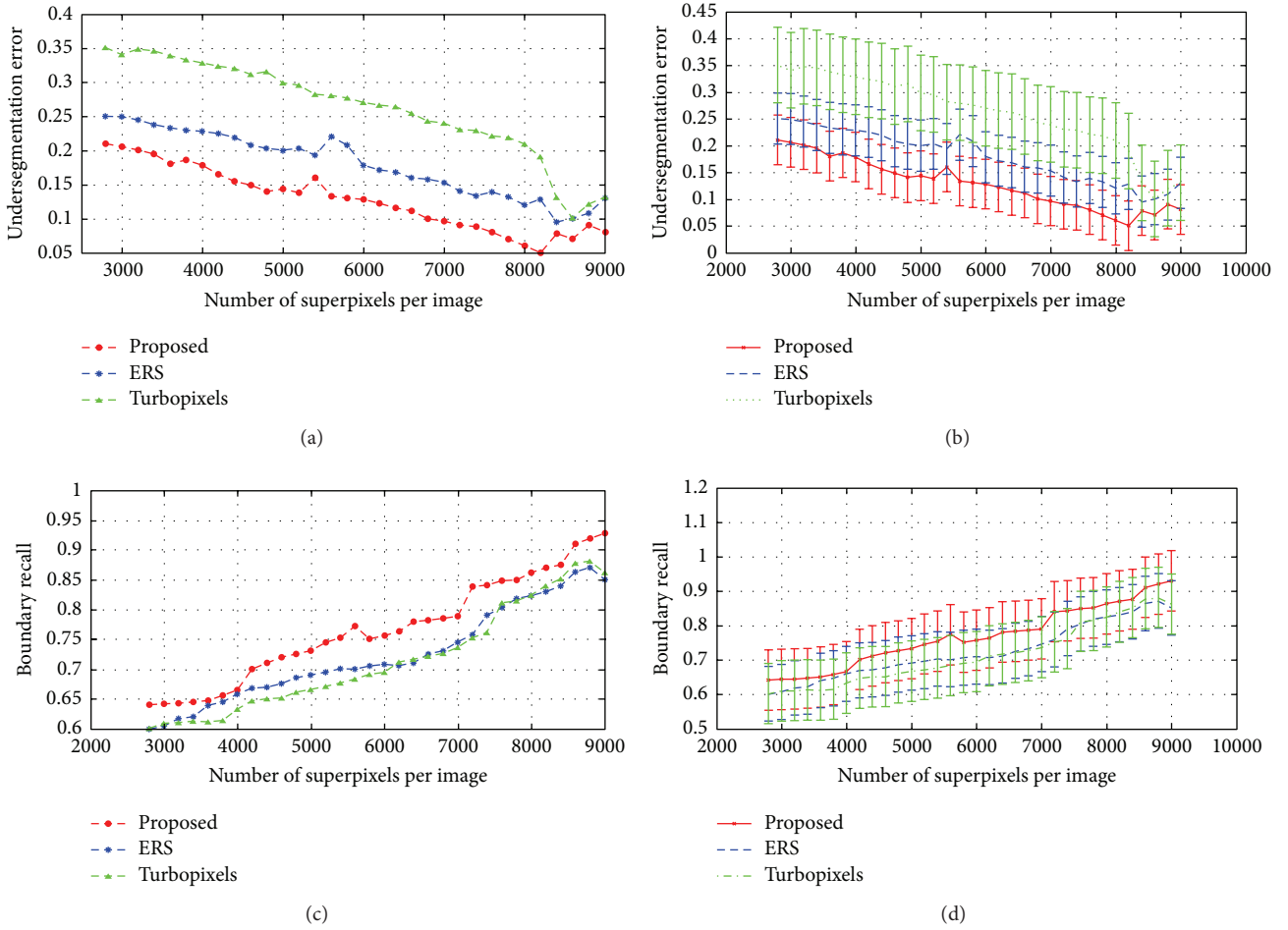


FIGURE 8: The comparison results of UE and BR. (a) Undersegmentation error curves; (b) undersegmentation error bar; (c) boundary recall curves; (d) boundary recall error bar.

the state of the art in all the metrics when the number of superpixels is calculated by the proposed function.

4.3. Building and Tree Classification Result. To justify the use of Naive Bayes classifiers which is trained by the proposed algorithm in our approach, we tried the popular intersection kernel and Chi-square kernel on our feature vector including color, FAST corner, and Gabor feature for comparison. Figure 9 shows the building and tree classification results with zoom-in regions.

4.4. Tree's Parameters Estimation Result. In our method, we calculated the tree parameters one by one. Each tree with shadow was extracted from every segmentation image, such as Figure 7, and then this tree is matched with tree model that is presented in this paper. The estimation result is shown in the panel of the GUI when the tree is marked by using red circular, which is shown in Figure 10. After calculating the parameters, all of the trees are covered by a round model.

RMSE is introduced to evaluate the estimation results and the function is shown as follows:

$$\text{RMSE} = \sqrt{\frac{1}{L} \sum_{i=1}^L \left(\frac{|p_i - p_{\text{true}}|}{p_{\text{true}}} \right)^2}, \quad (14)$$

where L is the number of trees, p_i is the estimated parameter, and p_{true} is the ground truth of the parameters which are manually measured. It is noted that p_i and p_{true} are used to describe the same tree in the image. The RMSE of x_T , y_T , r_T , and h_T is 1.142 pixels, 1.381 pixels, 1.993 pixels, and 5.992 pixels, respectively. At the same time, we manually measured the parameters for 270 trees which are chosen from Figure 9(a). The height of the tree cannot be measured directly because the image is 2D. However, we can measure the parameters of the tree's shadow by human, so the height can be calculated by (11). Therefore, the tree's height calculated by (11) could be considered as ground truth. Then these ground truth

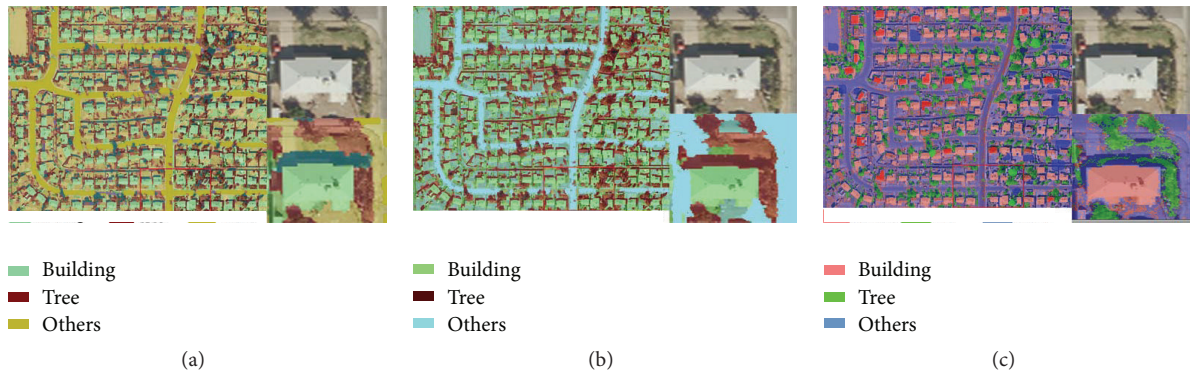


FIGURE 9: The visual comparison of building and tree classification result. (a) Classification result by our proposed method; (b) the classification based on intersection kernel; (c) the classification result based on Chi-square kernel.

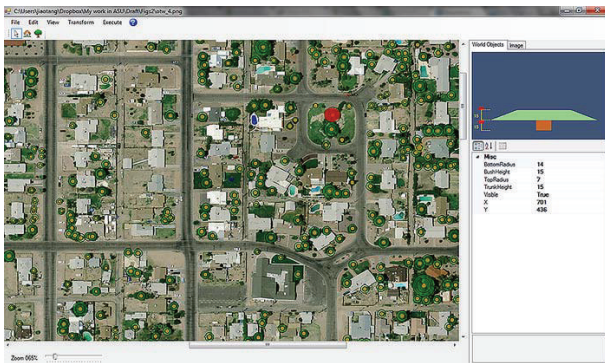


FIGURE 10: The segmentation result which is shown by GUI.

parameters are compared with the parameters obtained by using the proposed cost function (7). The result is shown in Figure 11.

5. Discussion

In this paper, we proposed a novel method for building and tree segmentation from large scale urban aerial images, and we also proposed a new approach for estimating tree parameters. In Section 4.2, we compared our segmentation method with the method proposed by Levinshtein et al. [14]. From Figure 6, it is shown that our approach of superpixel can be segmented along with the edge of objects, while the method of Levinshtein and his colleague segmented several objects into one superpixel. We also compare our segmentation results with the pixel-level segmentation approach using ERS. In Figures 9(a) and 9(c), the results indicate that our method can not only detect smaller building objects but also has less false alarm detection of building objects on the street. In our paper, the superpixel code is programmed by Matlab code. Therefore, the time complexity is a little high. However, we are doing the work of installing this algorithm by C#, which is used to reduce the run-time and easy to be used in commercial areas.

In Section 4.4, we presented a novel approach of calculating tree parameters, such as locations, radiuses, and heights, depending on a single aerial image without additional information. Our method needs less data than existing methods [9, 37, 38] that used additional information from LiDAR data and another database. Based on our novel segmentation method and the high-resolution aerial photographs, our tree parameters can be estimated with high accuracy since (a) the edges of neighbor trees can be detected more precisely because our segmentation method can separate adjacent trees using different superpixels; (b) the edges of shadow of trees are clearer and easier for segmentation in the high-resolution aerial images; (c) the time information for calculating heights by shadows of trees can be obtained from EXIF info in aerial photographs. However, our approach is implemented with two restrictions. The first restriction is that the color of building roof cannot be the same as the color of ground. The second restriction is that the distance between trees cannot be too close or the nearby tree cannot be separated and will be considered as the same one.

In Section 1(A), we showed many existing segmentation algorithms in aerial urban image processing using the pixel-level as the underlying representation. However, the pixel-level representation is not a natural representation of visual scenes. Each pixel cannot represent the geometric structure property of an object, while the superpixel is perceptually meaningful and each superpixel is a perceptually consistent unit [23]. The comparison of Figures 9(a), 9(b), and 9(c) shows that the pixel-level segmentation misses some edges of the buildings, while the superpixel-level segmentation keeps the edges. Furthermore, superpixel image has low computation cost and reduces the complexity of high-resolution aerial images. Therefore, many researchers introduced the superpixels for the image segmentation. For example, Saxena et al. [39] used the superpixel to segment objects during modeling 3D scene structure from a single stationary image. Kluckner et al. [40] used the superpixels to segment the large scale aerial images and the conditional random field (CRF) was used to refine the result. The presented approach of segmenting tree stems goes one step further than existing

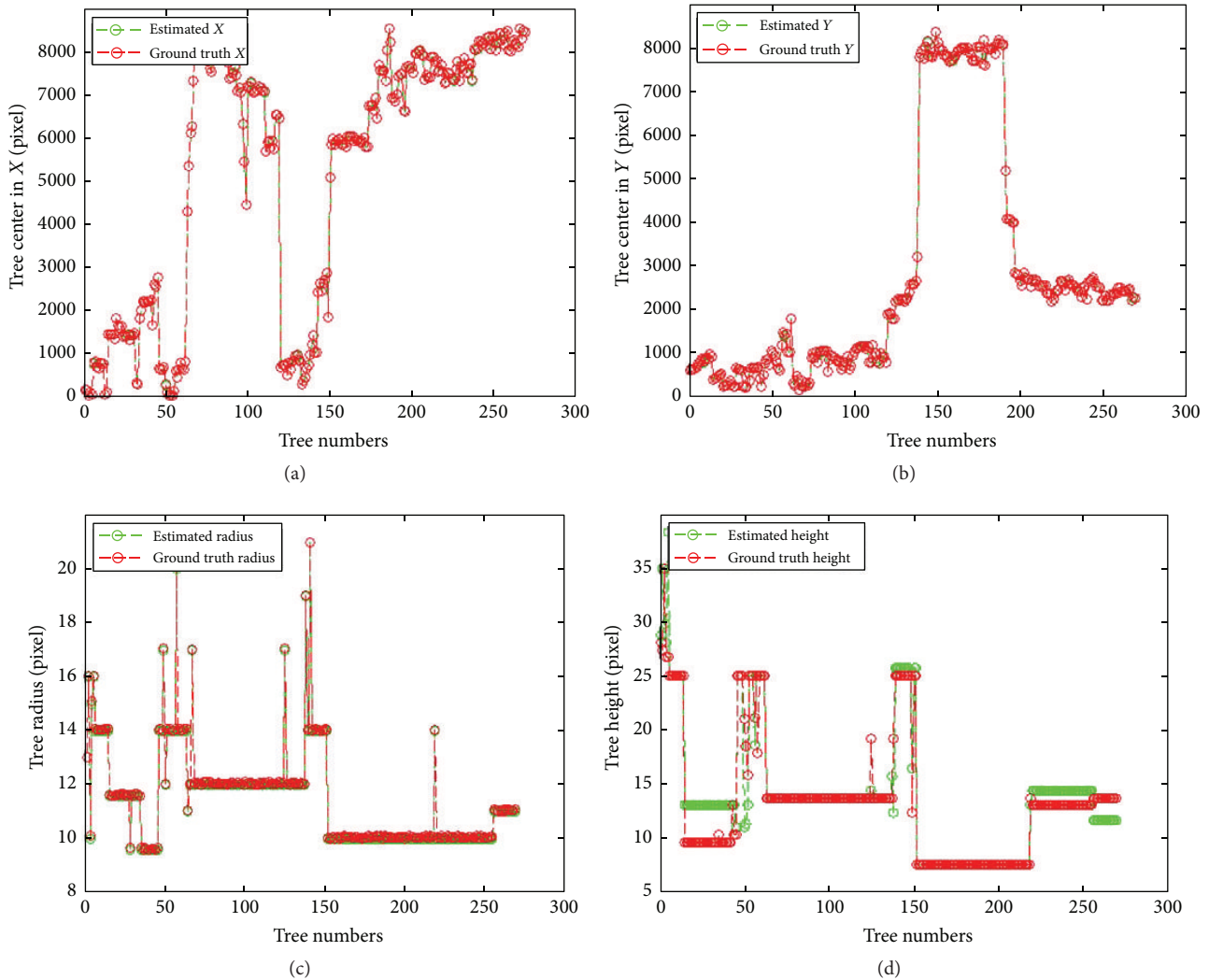


FIGURE 11: The comparing result. (a) The centers of trees in horizontal direction; (b) the centers of trees in vertical direction; (c) the radiuses; (d) the height.

methods by using additional information between the superpixels. Most of missing edges of superpixels are detected.

Tree model is widely used to detect trees. Morsdorf et al. [41] used the local maxima of canopy height model (CHM) to calculate the tree height using airborne laser scanning raw data. The method from Vastaranta et al. [42] used a raster CHM which was created from normalized data to detect the individual tree. Apparently, LiDAR data information is needed for detecting trees. However, our proposed method just detects the single tree from the aerial image without any additional complex information. The experiments show that the parameters of the trees can be calculated from a single high-resolution aerial image. In order to calculate the parameters of the trees, we use shadows, tree crowns, and the time of taking aerial image as input data and estimate the height of a tree based on (11).

Finally, the experiments show that the locations and radiuses of the trees estimated by our approach are approximate

to the manual-picking methods. However, there are some heights of trees higher or lower than the ground truth ones shown at Figure 11(d). It might be because the shapes of the tree are irregular. This problem could be solved if we consider a new tree model which is more complex than the model proposed in this paper.

6. Conclusion

In this paper, we proposed a building and tree detection algorithm by using improved superpixels from large high-resolution urban images, and we also proposed a method to calculate the tree parameters depending upon a cost function and shadows. A function was proposed to automatically calculate the number of superpixels, and a function is used to refine the boundaries of superpixels. We provided a new tree model with a cost function that can be minimized using gradient decent in order to identify the optimal properties

of individual tree. We evaluated our method by using many aerial images and compared our method with other two state-of-the-art methods. Experiments showed that our method is fast and robust, while still being simple and efficient, and they also indicate that the shadow is a good feature to estimate the tree height. The results of our method can be implemented for generating 3D urban models. The main purpose of this paper is to design fast, accurate segmentation and classification, characteristics not focused on the learning process. However, we will examine the different supervised learning algorithms in the future research work, improving the effectiveness of the proposed algorithm. Future work will focus on comparing our method with other supervised algorithms including methods of integrating these data sources into our solution.

Parameters used in the rule sets are sample values specifically for the study area we chose and may vary for other locations of interest, but the similar principles and procedures can be applied to other areas. Using additional ancillary data, such as 1-meter resolution LiDAR (Light Detection and Ranging), may further help generate more accurate land-cover maps, which is among our planned future work.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

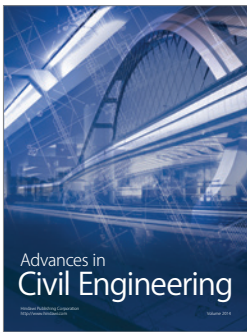
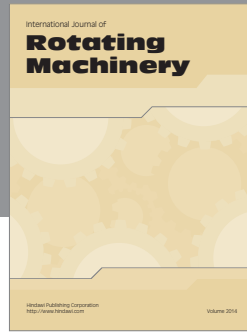
Acknowledgments

This project is sponsored by the The National High Technology Research and Development Program of China (no. 2014AA123103 and no. 2015AA124103) and National Natural Science Foundation of China (no. 61401040).

References

- [1] A. Huertas and R. Nevatia, "Detecting buildings in aerial images," *Computer Vision, Graphics, and Image Processing*, vol. 41, no. 2, pp. 131–152, 1988.
- [2] Y.-T. Liow and T. Pavlidis, "Use of shadows for extracting buildings in aerial images," *Computer Vision, Graphics, and Image Processing*, vol. 49, no. 2, pp. 242–277, 1990.
- [3] F. Rottensteiner and C. Briese, "A new method for building extraction in urban areas from high-resolution lidar data," *International Archives of Photogrammetry Remote Sensing and Spatial Information Sciences*, vol. 34, pp. 295–301, 2002.
- [4] F. Rottensteiner, J. Trinder, S. Clode, and K. Kubik, "Building detection using lidar data and multispectral images," in *Proceedings of the 7th Digital Image Computing: Techniques and Applications*, C. Sun, H. Talbot, S. Ourselin, and T. Adriaansen, Eds., vol. 2, pp. 673–682, Sydney, Australia, December 2011.
- [5] L. Cheng, L. Tong, Y. Chen et al., "Integration of LiDAR data and optical multi-view images for 3D reconstruction of building roofs," *Optics and Lasers in Engineering*, vol. 51, no. 4, pp. 493–502, 2013.
- [6] Y. El Merabet, C. Meurie, Y. Ruichek, A. Sbihi, and R. Touahni, "Building roof segmentation from aerial images using a lineand region-based watershed segmentation technique," *Sensors*, vol. 15, no. 2, pp. 3172–3203, 2015.
- [7] C. Iovan, D. Boldo, and M. Cord, "Detection, characterization, and modeling vegetation in urban areas from high-resolution aerial imagery," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 1, no. 3, pp. 206–213, 2008.
- [8] L. Yang and X. Wu, "Tree detection from aerial imagery," in *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (GIS '09)*, pp. 131–137, Seattle, Wash, USA, November 2009.
- [9] G. Chen and A. Zakhor, "2D tree detection in large urban landscapes using aerial lidar data," in *Proceedings of the IEEE International Conference on Image Processing (ICIP '09)*, pp. 1693–1696, Cairo, Egypt, November 2009.
- [10] A. Brook and E. Ben-Dor, "Quantitative detection of sediment dust analog over green canopy using airborne hyperspectral imagery," in *Proceedings of the 2nd Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS '10)*, pp. 1–5, IEEE, Reykjavik, Iceland, June 2010.
- [11] V. Risojević, S. Momić, and Z. Babić, "Gabor descriptors for aerial image classification," in *Adaptive and Natural Computing Algorithms: 10th International Conference, ICANNGA 2011, Ljubljana, Slovenia, April 14–16, 2011, Proceedings, Part II*, vol. 6594 of *Lecture Notes in Computer Science*, pp. 51–60, Springer, Berlin, Germany, 2011.
- [12] T. L. Swetnam and D. A. Falk, "Application of metabolic scaling theory to reduce error in local maxima tree segmentation from aerial LiDAR," *Forest Ecology and Management*, vol. 323, pp. 158–167, 2014.
- [13] H. Huang, P. Gong, X. Cheng, N. Clinton, and Z. Li, "Improving measurement of forest structural parameters by coregistering of high resolution aerial imagery and low density lidar data," *Sensors*, vol. 9, no. 3, pp. 1541–1558, 2009.
- [14] A. Levinshtein, A. Stere, K. N. Kutulakos, D. J. Fleet, S. J. Dickinson, and K. Siddiqi, "TurboPixels: fast superpixels using geometric flows," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 12, pp. 2290–2297, 2009.
- [15] D. Engel, L. Spinello, R. Triebel, R. Siegwart, H. H. Bühlhoff, and C. Curio, "Medial features for superpixel segmentation," in *Proceedings of the 11th IAPR Conference on Machine Vision Applications (MVA '09)*, pp. 248–252, Yokohama, Japan, May 2009.
- [16] S. Kluckner and H. Bischof, "Image-based building classification and 3d modeling with super-pixels," in *Proceedings of the ISPRS Technical Commission 3rd Symposium on Photogrammetry Computer Vision and Image Analysis*, pp. 233–238, Paris, France, September 2010.
- [17] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 11, pp. 1254–1259, 1998.
- [18] T. Kadir, A. Zisserman, and M. Brady, "An affine invariant salient region detector," in *Computer Vision—ECCV 2004*, vol. 3021 of *Lecture Notes in Computer Science*, pp. 228–241, Springer, Berlin, Germany, 2004.
- [19] D. A. Hahn, G. Wolz, Y. Sun et al., "A practical salient region feature based 3D multi-modality registration method for medical images," in *Proceedings of the Medical Imaging: Image Processing*, vol. 6144 of *Proceedings of SPIE*, p. 61442P, San Diego, Calif, USA, March 2006.
- [20] S. Kicherer, J. A. Malpica, and M. C. Alonso, "An interactive technique for cartographic feature extraction from aerial and satellite image sensors," *Sensors*, vol. 8, no. 8, pp. 4786–4799, 2008.

- [21] F. Bi, F. Liu, and L. Gao, "A hierarchical salient-region based algorithm for ship detection in remote sensing images," in *Advances in Neural Network Research and Applications*, vol. 67 of *Lecture Notes in Electrical Engineering*, pp. 729–738, Springer, Berlin, Germany, 2010.
- [22] V. Casellaa, M. Franzinia, G. Banchinib, and G. Gentilib, "Initial evaluation of the secondgeneration leica ADS40 camera," in *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, pp. 527–532, Postgraduate Medicine, Beijing, China, 2008.
- [23] X. Ren and J. Malik, "The ecological statistics of good continuation: multi-scale markov models for contours," *Journal of Vision*, vol. 2, no. 7, article 708, 2002.
- [24] R. A. Peters and R. N. Strickland, "Image complexity metrics for automatic target recognizers," in *Proceedings of the Automatic Target Recognizer System and Technology Conference*, pp. 1–17, 1990.
- [25] J. Rigau, M. Feixas, and M. Sbert, "An information-theoretic framework for image complexity," in *Proceedings of the 1st Eurographics Conference on Computational Aesthetics in Graphics, Visualization and Imaging Computational Aesthetics*, L. Neumann, M. Sbert, B. Gooch, and W. Purgathofer, Eds., pp. 177–184, Eurographics Association Aire-la-Ville, 2005.
- [26] J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, no. 6, pp. 679–698, 1986.
- [27] V. M. Nazareth, K. Amulya, and K. Manikantan, "Optimal multilevel thresholding for image segmentation using contrast-limited adaptive histogram equalization and enhanced convergence particle swarm optimization," in *Proceedings of the 3rd National Conference on Computer Vision, Pattern Recognition, Image Processing and Graphics (NCPVPRIG '11)*, pp. 207–210, Hubli, India, December 2011.
- [28] J. C. Femiani and A. Razdan, "Interval HSV: extracting ink annotations," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '09)*, pp. 2520–2527, Miami, Fla, USA, June 2009.
- [29] E. Rosten, R. Porter, and T. Drummond, "Faster and better: a machine learning approach to corner detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 1, pp. 105–119, 2010.
- [30] W. Li, K. Z. Mao, H. Zhang, and T. Chai, "Selection of gabor filters for improved texture feature extraction," in *Proceedings of the 17th IEEE International Conference on Image Processing (ICIP '10)*, pp. 361–364, IEEE, Hong Kong, September 2010.
- [31] V. Risojević, S. Momić, and Z. Babić, "Gabor descriptors for aerial image classification," in *Adaptive and Natural Computing Algorithms*, vol. 6594 of *Lecture Notes in Computer Science*, pp. 51–60, Springer, Berlin, Germany, 2011.
- [32] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The WEKA data mining software: an update," *ACM SIGKDD Explorations Newsletter*, vol. 11, no. 1, pp. 10–18, 2009.
- [33] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, Prentice-Hall, 2002.
- [34] M.-Y. Liu, O. Tuzel, S. Ramalingam, and R. Chellappa, "Entropy rate superpixel segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '11)*, pp. 2097–2104, IEEE, Providence, RI, USA, June 2011.
- [35] O. Veksler, Y. Boykov, and P. Mehrani, "Superpixels and super-voxels in an energy optimization framework," in *Computer Vision—ECCV 2010*, vol. 6315 of *Lecture Notes in Computer Science*, pp. 211–224, Springer, Berlin, Germany, 2010.
- [36] X. Ren and J. Malik, "Learning a classification model for segmentation," in *Proceedings of the 9th IEEE International Conference on Computer Vision*, vol. 1, pp. 10–17, IEEE, Nice, France, October 2003.
- [37] J. Reitberger, C. I. Schnörr, P. Krzystek, and U. Stilla, "3D Segmentation of single trees exploiting full waveform LIDAR data," *International Society for Photogrammetry and Remote Sensing*, vol. 64, no. 6, pp. 561–574, 2009.
- [38] J. Liu, J. Shen, R. Zhao, and S. Xu, "Extraction of individual tree crowns from airborne LiDAR data in human settlements," *Mathematical and Computer Modelling*, vol. 58, no. 3–4, pp. 524–535, 2013.
- [39] A. Saxena, M. Sun, and A. Y. Ng, "Make3D: learning 3D scene structure from a single still image," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 5, pp. 824–840, 2009.
- [40] S. Kluckner, M. Donoser, and H. Bischof, "Super-pixel class segmentation in large-scale aerial imagery," in *Proceedings of the 34th Annual Workshop of the Austrian Association for Pattern Recognition (OeAGM '10)*, pp. 131–138, Zwettl, Austria, May 2010.
- [41] F. Morsdorf, E. Meier, B. Allgower, and D. Nuesch, "Clustering in airborne laser scanning raw data for segmentation of single trees," *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 34, part 3, p. W13, 2003.
- [42] M. Vastaranta, V. Kankare, M. Holopainen, X. Yu, J. Hyyppä, and H. Hyyppä, "Combination of individual tree detection and area-based approach in imputation of forest variables using airborne laser data," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 67, pp. 73–79, 2012.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

