Building Detection in Complex Scenes Thorough Effective Separation of Buildings from Trees

Content description: The problem of separating buildings from trees has been investigated using a combination of color, texture and dimensional cues for improved building detection performance in complex scenes.

Authors' information:

Dr. Mohammad Awrangjeb, Dr. Chunsun Zhang and Dr. Clive S. Fraser

Department of Infrastructure Engineering, The University of Melbourne

CRC for Spatial Information, 204 Lygon Street, Carlton Vic 3053 Australia

mawr@unimelb.edu.au, chunsunz@unimelb.edu.au, c.fraser@unimelb.edu.au

Abstract: Effective separation of buildings from trees is a major challenge in image-based automatic building detection. This paper presents a three-step method for effective separation of buildings from trees using aerial imagery and LIDAR data. Firstly, it uses cues such as height to remove objects of low height such as bushes, and width to exclude trees with small horizontal coverage. The height threshold is also used to generate a ground mask where buildings are found to be more separable than in so-called normalized DSM. Secondly, image entropy and color information are jointly applied to remove easily distinguishable trees. Finally, an innovative rule-based procedure is employed using the edge orientation histogram from the imagery to eliminate false positive candidates. The improved building detection algorithm has been tested on different test areas and it is shown that the algorithm offers high building detection rate in complex scenes which are hilly and densely vegetated.

1. Introduction

Building detection from remotely sensed data has a number of practical applications including urban planning, homeland security and disaster (flood or bushfire) management. Consequently, a number of automated building detection techniques have been reported over the last few decades. These can be divided into three major groups (Lee et al., 2008). Firstly, there are many algorithms which use 2D or 3D information from photogrammetric imagery (Huang and Zhang, 2011). Secondly, there have been several attempts to detect building regions from LIDAR (LIght Detection And Ranging) data. Finally, since LIDAR and imagery each have particular advantages and disadvantages in horizontal and vertical positioning resolution and accuracy, several authors have promoted an integration of LIDAR data and imagery as a means of advancing building detection. More specifically, intensity and height information from LIDAR can be used with texture and region boundary information from imagery to improve detection accuracy (Habib et al., 2010).

However, the success of automatic building detection is still largely impeded by scene complexity, incomplete cue extraction and sensor dependency of data (Sohn and Dowman, 2007). Vegetation, and especially trees, can be the prime cause of scene complexity and incomplete cue extraction. The situation becomes complex in hilly and densely vegetated areas where only a few buildings are present, these being surrounded by trees. Important building cues can be completely or partially missed due to occlusions and shadowing from trees. Trees also change color in different seasons and may be deciduous. Moreover, image quality may vary for the same scene even if images are captured by the same sensor, but at different dates and times. So, when the same detection algorithm is applied to two different images of the same scene, the outcomes may well be different. Therefore, many existing building detection techniques (Lee et al., 2003; Shorter and Kasparis, 2009) that depend largely on color information for separation of buildings from trees exhibit poor detection performance.

Application of a recently developed building detection algorithm (Awrangjeb et al., 2010b) has shown it to be capable of detecting buildings in cases where cues are only partially extracted. For example, if a section of the side of a roof (at least 3m long) is correctly detected the algorithm can also detect all or part of the entire building. However, this detector does not necessarily work well in complex scenes and hilly areas. Buildings may be surrounded by dense vegetation, they may have the same color as trees or trees may have colors other than green. Awrangjeb et al. (2010b) applies NDVI only and therefore cannot distinguish between a green building and a green tree. As a result, it fails to detect green buildings but detect many trees that are non-green in color. Moreover, in hilly areas, where there is a large variation in height within a small area, a common height threshold does not work at all. Awrangjeb et al. (2010b) divides the area into tiles of equal size, compute height threshold for each tile and then apply the same height threshold to all the LIDAR points within the tile. Consequently, in hilly areas some ground area with high heights may be detected as buildings and vice versa.

This paper presents an improved building detection algorithm that achieves effective separation of buildings from trees. It uses a combination of color, texture and dimensional cues to eliminate trees. Firstly, objects below a given threshold above the ground, such as bushes, cars and carports, are removed from the raw LIDAR data. This cue is also used to generate a ground mask where trees are found to be more separable than in the well known normalized DSM (nDSM). Secondly, image entropy and color information are employed together to remove trees that are easily distinguishable. Finally, an innovative rule-based procedure based on the edge orientation histogram from the image is used to eliminate false positive detection of trees. In addition, the local DEM height is used to avoid the problem associated with high height variation in hilly areas. The improved detector has been tested on a number of scenes covering different test areas. The data were collected with different sensors, at different dates and times of the day.

The main contributions of this paper are as follows.

▪ We have reviewed the cues that have been used for classification of buildings and trees. A building detection algorithm can apply a combination of two or more of these cues (Section 2).

▪ We have proposed an effective method of separating buildings from trees (Section 3). While the original algorithm (Awrangjeb et al., 2010b) was unable to detect green colored buildings using the NDVI only and detected a large number of false buildings in a densely vegetated area, the proposed improved algorithm preserves green colored buildings as well as removes non-green trees through a joint application of NDVI and entropy. It also removes false buildings by employing the newly developed rule-based procedure based on the edge orientation histogram.

▪ The standard values of different parameters of the proposed method have been empirically set using three sub-images from three data sets (Section 4.4). The application of the standard parameter values on a different data set, which is not used while setting standard values, shows that the proposed improved algorithm could be used on any future data sets.

▪ Through experiments (Section 4.5) we have shown that while the application of color information alone may misclassify some of the green buildings and non-green trees, the joint application of color and entropy information moderately improves the classification. The application of the newly proposed rule-based procedure eliminates a large number of false building detections resulting in a significant performance rise in complex scenes, while compared with the original detector in Awrangjeb et al. (2010b).

▪ When compared with an existing detector (Rottensteiner et al., 2005) that was tested on a same test data set, the proposed improved detector was found to offer significantly better performance (Section 4.5).

2. Cues for Tree Removal

Existing building detection algorithms make use of different cues with a view to separating buildings from trees. While cues related to color are useful only with multispectral images, cues related to width, height, area and texture can be used with both LIDAR and images.

- **Height:** A height threshold (2.5m above ground level) is often used to remove low vegetation and other objects of limited height, such as cars and street furniture (Rottensteiner et al., 2007; Cheng et al., 2011). Trees taller than the building roof cannot be removed via this height thresholding. Dash et al. (2004) used the height variation along the periphery of objects present in the data to distinguish trees from buildings. Rottensteiner et al. (2007) and Khoshelham et al. (2008) used height difference values between first and last pulse LIDAR data for the same purpose, since it can be anticipated that the differences will be large for trees but negligible for buildings. However, there are two limitations in using this cue. Firstly, the height difference between the pulses must be at least 2m (Maas, 2001). Secondly, a first pulse is not always reflected from the upper branches of a tree and a last pulse may sometimes be a reflection from a tree trunk or branches. Huang et al. (2011) used LIDAR height difference (between maximum and minimum height values), height variance and maximum-minimum (Max-Min) height values within a local area to discriminate trees with roofs and grass. They also used the average height, resulting from an object-based image classification, in a post-processing step, and experimentally showed that the Max-Min feature significantly increased classification accuracy in all classes.

- **Width, area and shape:** If the width or area of a detected object is smaller than a threshold, then it is removed as a tree. Awrangjeb et al. (2010b) used a reasonable threshold of 3m for width; however, Lee et al. (2003) have employed a high area threshold 50m^2 which can also remove small buildings. Chen et al. (2006) employed an area threshold of 30m^2 and missed small buildings. A number of shape attributes can be found in Matikainen et al. (2007).

- **Surface:** Many authors (Khoshelham et al., 2005; Zhang et al., 2006; Cheng et al., 2011) applied plane-fitting techniques on the non-ground LIDAR points to separate buildings and trees. A plane that fits a roof plane well is expected. These techniques largely depend on LIDAR density, since for a small but complex building roof the algorithm may not converge to the correct solution if the density of LIDAR points is low. Rottensteiner et al. (2007) applied a polymorphic feature extraction algorithm to the first derivatives of the DSM in order to measure the strength and directedness of surface roughness for pixels displaying a high roughness value. Their classification failed to detect buildings smaller than 30m^2 in size, since the number of (homogenous) LIDAR points was low for small buildings. Moreover, due to restrictions of surface geometry, the number of object types that can be discriminated within a DSM is limited (Haala and Brenner, 1999). Chen et al. (2006) employed slope variance using LIDAR data to measure surface roughness. A high estimation of the roughness index indicates a vegetation area. Sampath and Shan (2010) applied Eigen analysis at each LIDAR point by fitting a plane within its Voronoi neighborhood and separated planar points (roofs) from non-planar points (trees etc).

- **Colors:** A high NDVI (normalized difference vegetation index estimated using multispectral images) value for a pixel indicates vegetation, whereas a low NDVI value generally indicates a non-vegetation pixel. This cue, which is frequently employed (Sohn and Dowman, 2007; Demir et al., 2009; Huang and Zhang, 2011; Huang and Zhang, 2012), has been found to be unreliable even in scenes where trees and buildings have distinct colors (Rottensteiner et al., 2007). The situation becomes more difficult when trees change color or lose leaves in different seasons. Vu et al. (2009) applied K-means clustering on multispectral images to obtain spectral indices for clusters such as trees, water and buildings. Shorter and Kasparis (2009) used color invariants. If the majority of pixels in a segmented region represent candidate pixels for vegetation, then the segment is marked as vegetation. This technique did not work when non-vegetation pixels shared similar spectral attributes with vegetation, and the experimentation showed that it was unable to classify buildings smaller than 70m^2 (Shorter and Kasparis, 2009). Lee et al. (2003) used training pixels of different colors from roofs, roads, water, grass, trees and soil for classification. A large number of false positives, as high as 300% and caused by trees and occlusions in the vicinity of buildings, was reported as the most critical problem (Shan and Lee, 2005). This classification technique failed if an object in the test scene had a color other than its designated colors in the training set. More recently, Lee et al. (2008) used green pixel values directly to identify trees. A number of other cues generated from color image and height data can be found in Matikainen et al. (2007) and Salah et al. (2009).

- **Texture:** For the case of objects having similar spectral responses, Chen et al. (2006) used the grey level co-occurrence matrix (GLCM) of the image to quantify co-occurrence probability. However, this method does not indicate how to cope with erroneous lines (Sohn and Dowman, 2007) and it cannot detect small buildings. Some GLCM indices, eg mean, standard deviation, entropy and homogeneity, are applied to both height and image data in order to classify buildings and trees (Matikainen et al., 2007; Salah et al., 2009; Huang et al., 2011). Both Matikainen et al. (2007) and Salah et al. (2009) used complex and time consuming classification techniques on large numbers of attributes. The latter showed good detection performance, though it failed to detect small buildings. Huang et al., (2011) used multiclass support vector machines and showed about 95% classification accuracy.

- **Others:** Segmentation of LIDAR intensity data can also be used to distinguish between buildings and trees (Maas, 2001). Sampath and Shan (2007) used 1D bi-directional filter to classify ground and non-ground LIDAR points. Demir et al. (2009) made use of the density of the raw DSM and DTM. The DSM point cloud included all LIDAR points with 4 echoes per pulse and a much higher point density was observed for trees than for open terrain and buildings. In contrast, the DTM included only points on the ground, so it displayed holes at building positions, and the point density at trees was found to be lower than that in open terrain.

3. Improved Building Detection

The proposed detector, which is an improved version of that described in Awrangjeb et al. (2010b), employs a combination of height, width, color and texture information with the aim of more comprehensively separating buildings from trees. Although cues other than texture were used in the earlier version of the detector, the improved formulation makes use of additional texture cues such as entropy and the edge orientation histogram at three stages of the process, as shown in Figure 1. Awrangjeb et al. (2010b) presented different steps of the detection algorithm in detail. In this paper, we focus on how texture, dimensional (length and height) and color information can be applied jointly in order to better distinguish between buildings from trees.

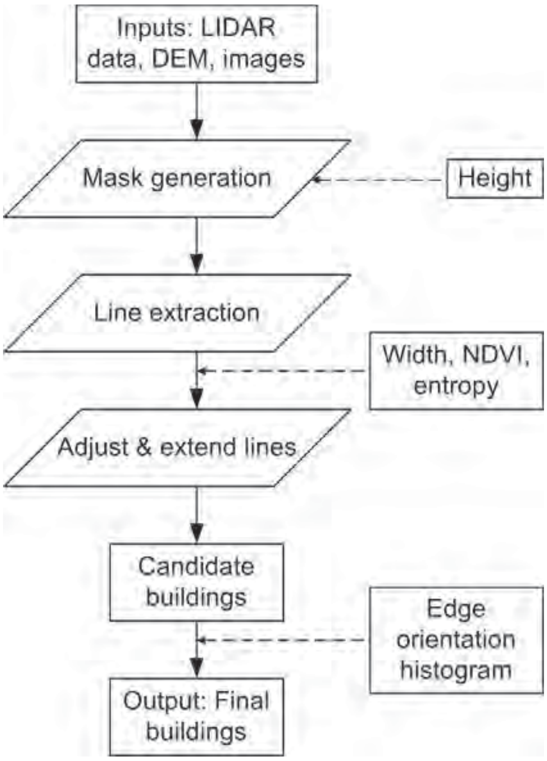

Figure 1. Flow diagram of the improved building detection technique.

Table 1 shows a list of the parameters used by the proposed algorithm. The standard values of these parameters were either chosen from the existing literature or set by an empirical study presented in Section 4.4.

Table 1: Parameters used in different sections of the proposed algorithm (T_h = height threshold, H_g = ground height, n = neighborhood size for a LIDAR point in ground mask with respect to image, d_l = LIDAR resolution, d_i = image resolution, T_{ndvi} = NDVI threshold, T_{emask} = entropy mask threshold, T_{ent} = entropy threshold, D_{bin} = histogram bin distance, N_{Tp} = number of texture pixels within a candidate building, R_{Mm1}, R_{Mm2} = ratios of highest bin-to-mean height and R_{aTp} = ratio of the detected area to N_{Tp}).

Sections	Parameters	Standard Values	Reference for values
3.1	T_h	H_g + 2.5m	Rottensteiner et al., (2005)
	n	$d_l/d_i + 2$	Awrangjeb et al. (2010b)
3.2	T_{ndvi}	10	Awrangjeb et al. (2010b)
	T_{emask}	0.8	Gonzalez et al., (2003)
	T_{ent}	0.3	Section 4.4
3.3	D_{bin}	5°	Section 4.4
	N_{Tp}	90m	Section 4.4
	R_{Mm1}	2	Section 4.4
	R_{Mm2}	4	Section 4.4
	R_{aTp}	45m	Section 4.4

3.1. Application of Height Threshold

A height threshold $T_h = H_g + 2.5$m, where H_g represents the ground height (DEM value), is applied to the raw LIDAR data. This threshold removes objects of low height (shubbery, road furniture, cars, etc.) and preserves non-ground points (trees and buildings).

The corresponding DEM height for a given LIDAR point (x, y, z) is assigned as a value of H_g. If there is no corresponding real DEM height recorded for (x, y, z), the average DEM height in its neighborhood is used. We first check within a 3×3 neighborhood around (x, y) and if there is at least one real DEM height the average of all real DEM heights within the neighborhood is considered as H_g for (x, y, z). Otherwise, the neighborhood is enlarged to 5×5.

The height threshold T_h is also used to generate a ground mask M_g, which has the same resolution as the image. All pixels in M_g are initially assigned 0 (false). If the height z of a LIDAR point (x, y, z) is less than T_h, the corresponding pixel in M_g is assigned 1. In addition, since the horizontal point distance of the LIDAR data d_l is generally higher than that of the orthoimage d_i, all the pixels in a n×n neighborhood of (x, y) are also assigned 1, where $n = 2 + d_l/d_i$. Consequently, the black areas in M_g indicate 'void areas' where there are no laser returns below T_h (ground areas covered by buildings and trees).

The generated mask is technically different from the well known nDSM where non-zero heights indicate the elevated objects above T_h. Therefore, as shown in Figure 2, buildings and

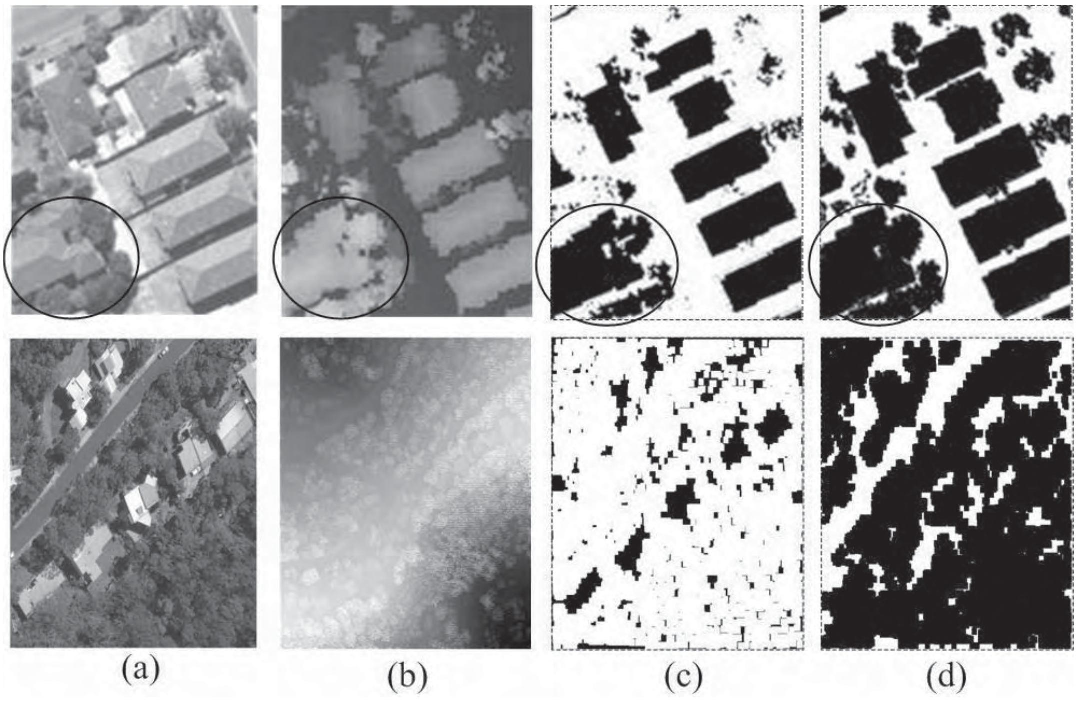

Figure 2. (a) Images of two test scenes, (b) corresponding LIDAR data (shown in gray-scale), (c) ground masks and (d) normalized DSMs. The nDSM is shown as a binary image where black pixels indicate non-zero heights greater than the height threshold.

trees are found to appear thinner in M_g than in the nDSM. In Figure 2, the nDSM is shown as a binary image where black pixels indicate non-zero heights (greater than T_h). The first row in the figure shows an example of a simple scene with a flat ground, while the second row shows the same for a complex scene with a hilly ground where buildings are mostly surrounded by dense vegetation. In both of the scenes, buildings and trees appear thinner in M_g than in the nDSM. For the first scene in Figure 2, most of the trees around the building at the lower left of the scene (shown within a circle) will be clearly separable in M_g, while in the nDSM they are almost connected to the building. For the second scene in Figure 2, each of the buildings is strongly connected to the neighboring vegetation in the nDSM, while they are clearly separable in M_g. Consequently, unlike the existing building detectors that use the nDSM for building detection, the use of M_g to obtain building candidates by the proposed detector helps to better separate trees from buildings.

3.2 Use of Width, NDVI and Entropy

The black areas in M_g are either buildings, trees or other elevated objects. Line segments around these black shapes in M_g are formed according to the process described in Awrangjeb et al. (2010b), and in order to avoid detected tree-edges, extracted lines shorter than the minimum building width $L_{\min} = 3$ m are removed. Trees having small horizontal area are thus removed. The remaining line segments are shown for a test scene in Figure 3a.

In Awrangjeb et al. (2010b), a rectangle of width 1.5m was then considered on each side of a line segment (as shown in Figure 3c for a line segment in Figure 3a). If the mean NDVI value was above a threshold T_{ndvi} at any side of the line, then the line was removed as a tree edge. The line segments that survive after application of the NDVI threshold are shown in Figure 3b.

However, the NDVI has been found to be an unreliable cue even in normal scenes where trees and buildings have distinct colors (Rottensteiner et al., 2007). In addition, it cannot differentiate between trees and green buildings when both exhibit high NDVI values. Figure 3b

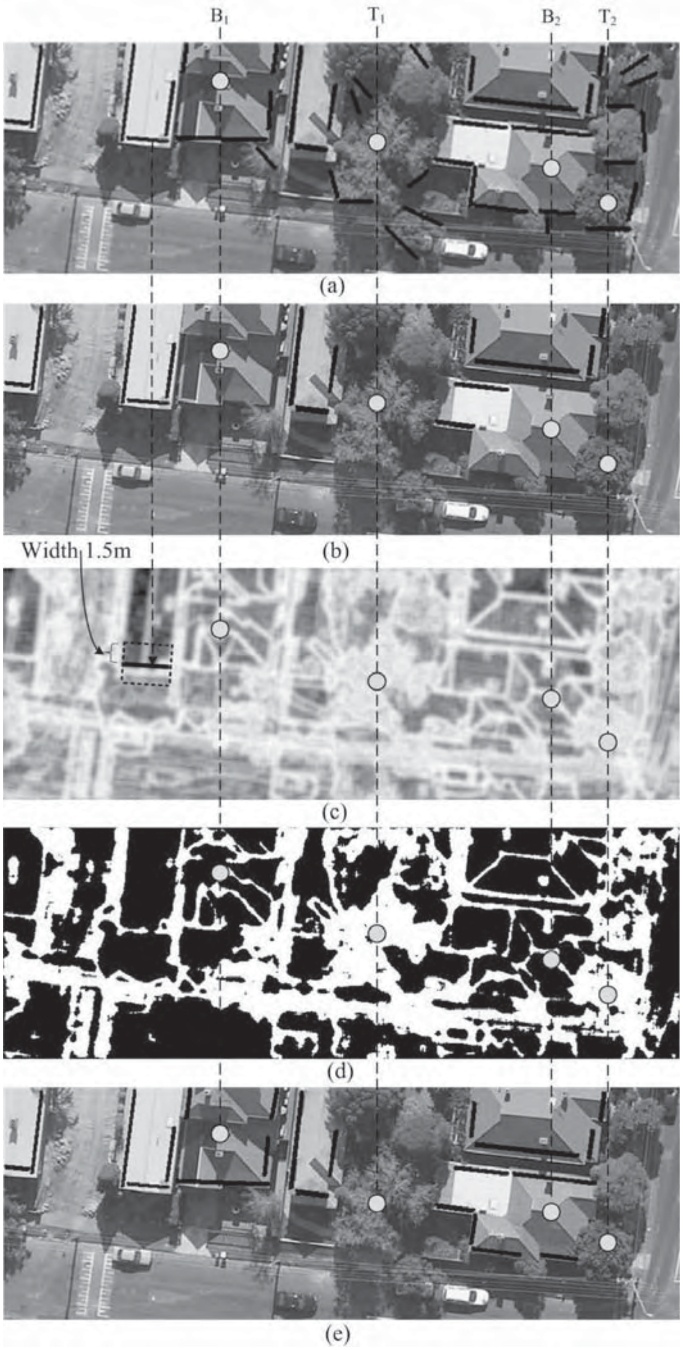

Figure 3. Detection of line segments: (a) before applying NDVI, (b) after applying NDVI, green buildings B_1 and B_2 are missed, (c) entropy image for the scene in (a), (d) entropy mask from (c), and (e) combined NDVI and entropy information is able to detect the two green buildings. In (a), (b) and (e), black solid lines represent detected buildings. In (c) and (d), while black pixels represent low entropy areas, white pixels represent high entropy areas. In (c), dotted rectangles show the two neighborhoods, used in Section 3.2, around a line segment.

shows an example where a green building B_1 cannot be detected at all, since all lines around it are rejected. However, green building B_2 can be partially detected because it has a white colored roof section. In some areas there may be non-green buildings having the same color as trees, especially when leaves change color in different seasons. In such cases, the removal of trees based on the NDVI will result in many buildings also being removed. Detection of these same buildings will likely also lead to detection of trees.

In the improved algorithm, texture information, namely entropy, and NDVI are jointly employed to remove large trees. Entropy is a statistical measure of randomness that can be used to characterize the texture of the input image (Gonzalez et al., 2003). Its adoption is based on the assumption that trees are rich in texture as compared to building roofs. While a high entropy value at an image pixel indicates a texture (tree) pixel, a low entropy value indicates a ‘flat’ (building roof) pixel.

In order to calculate entropy e at a point P of a grey-scale image, a 9×9 sub image I is considered (Gonzalez et al., 2003), where P is the centre point. A normalized histogram H for I, involving 256 bins and values between 0 and 1, is formed and entropy is calculated using non-zero frequencies as

$$e = -\sum H_i \log_2 H_i \text{, where } 1 \leq i \leq 256 \text{ and } 0 < H_i \leq 1. \qquad (1)$$

Figure 3c shows the entropy image for the scene in Figure 3a. A threshold T_{emask} 0.8 is applied to obtain the entropy mask M_e shown in Figure 3d. This threshold is selected because it is roughly the intensity value of pixels along the boundary between the textures (Gonzalez et al., 2003). We see that while we have low entropy values (black areas in M_e) on buildings B_1 and B_2, we have high entropy values (white areas in M_e) on trees T_1 and T_2.

In the improved detector, if the mean NDVI is above T_{ndvi} at any side of a line segment (using the same neighborhood shown in Figure 3c, also in Figure 5a), a further test is performed before removing this line segment as a tree-edge. This test checks whether the number of pixels having high entropy values exceeds the entropy threshold T_{ent} = 30%. In other words, whether the number of white pixels in M_e is more than T_{ent}. If the test holds, the line segment is removed as a tree edge; otherwise it is selected as a building edge. Figure 3e shows that the green buildings B_1 and B_2 can be fully detected using this approach. Note that the use of entropy alone is insufficient because trees with self-occlusions and shadows may not contain enough texture information and in such cases NDVI helps to remove them.

The remaining lines are then adjusted and extended as described in Awrangjeb et al. (2010b). Each line is adjusted based on the nearest parallel or perpendicular image line which is at least $1.5L_{min}$m long. Finally, rectangular shapes are obtained using the extended lines. In an iterative procedure, an initial candidate building position is initially detected using the first longest line segment, then another is found using the second longest line segment and so on. The final candidate building positions are obtained from their initial positions by extending each of the four sides. Image color and texture information are considered during the extension. Figure 4 shows the candidate buildings for two scenes shown in Figure 2.

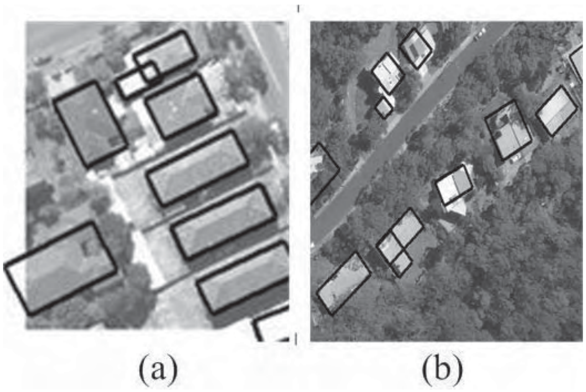

Figure 4. Candidate buildings for two scenes in Figure 2.

3.3 Application of Edge Orientation Histogram

So far we have used the local neighborhood information around the extracted lines in order to distinguish between trees and buildings. In fact, during the joint application of NDVI and entropy in the previous section we have considered a rectangle of width 1.5m (since L_{min} = 3m) on each side of an extracted line (see Figure 5a). However, due to sunlight, shadows and self-

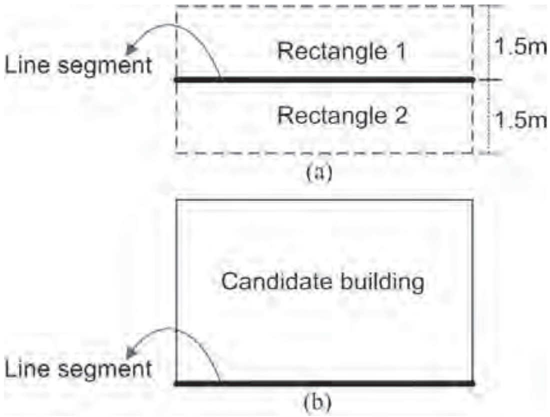

Figure 5. Neighborhood when (a) NDVI and entropy are jointly applied and (b) edge orientation histogram is applied.

occlusions among trees or when an area is mostly filled with dense vegetation, or when trees change color or lose leaves in different seasons, color and texture from such a small neighborhood may not provide distinct information and therefore many of the tree edges may still remain. Consequently, a large number of false candidates can be obtained. Such a situation is clearly evident in Figure 6b, where trees might not be pure green in color and the buildings are closely surrounded by dense vegetation. In such situations, texture information within a

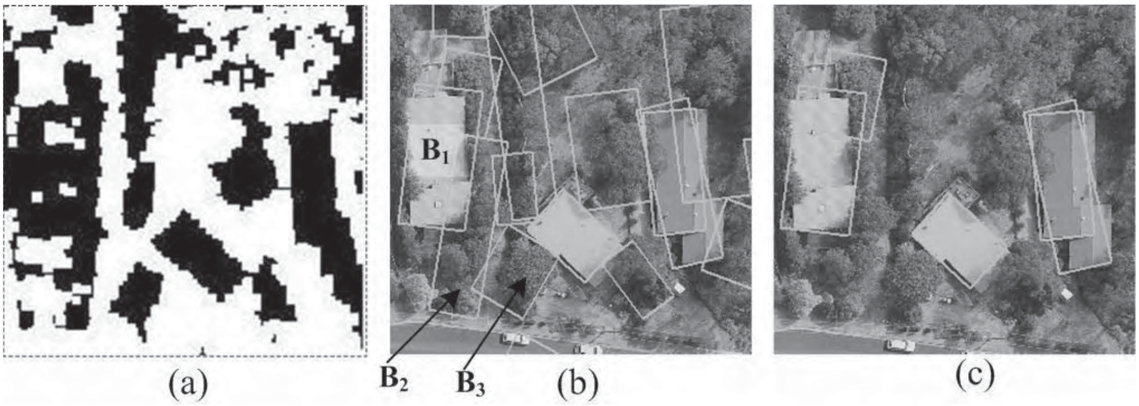

Figure 6. A complex scene: (a) ground mask, (b) detected candidate buildings with a large number of false detections and (c) final detected buildings after removing false positives.

larger neighborhood (within the candidate building rectangle, as shown in Figure 5b) can be exploited to identify the false candidates. In this section, we propose an innovative rule-based procedure based on the edge orientation histogram to remove the false building candidates.

After detecting candidate buildings, a gradient histogram is formed using the edge points within each candidate building rectangle. Edges are first extracted from the orthophoto using an edge detector and short edges (less than 2m in length) are removed. Each edge $\Gamma(t) = (x(t), y(t))$ of length n, where t is an arbitrary parameter and $1 \leq t \leq n$, is smoothed by using a Gaussian function g_σ with scale $\sigma = 3$:

$$x_\sigma(t) = x(t) * g_\sigma \quad \text{and} \quad y_\sigma(t) = y(t) * g_\sigma \tag{2}$$

where * denotes the convolution. The first order derivatives (x and y differences) are then calculated on the smoothed curve $\Gamma_\sigma(t) = (x_\sigma(t), y_\sigma(t))$ as:

$$x_\sigma'(t) = \frac{x_\sigma(t+1) - x_\sigma(t-1)}{2} \quad \text{and} \quad y_\sigma'(t) = \frac{y_\sigma(t+1) - y_\sigma(t-1)}{2}. \tag{3}$$

Finally, the gradient (tangent angle) function for a given curve $\Gamma(t)$ is calculated as:

$$\Delta_\Gamma(t) = \tan^{-1} \frac{y_\sigma'(t)}{x_\sigma'(t)} \tag{4}$$

The gradient obtained at each point using (4) will lie within the range of [– 90°, + 90°]. A histogram with a successive bin distance of $D_{bin} = 5°$ is then formed using the gradient values of all edge points lying inside the candidate rectangle.

Rectangles containing either the whole or a major part of a building should have one or more significant peaks in Δ_Γ, since edges detected on building roofs are formed from straight line segments. All points on an apparent straight line segment will have a similar gradient value and hence will be assigned to the same histogram bin, resulting in a significant peak. A significant peak means the corresponding bin height is well above the mean bin height of the histogram. Since edge points whose gradient falls into the first (at –90° to –85°) and last (at 85° to 90°) bins have almost the same orientation, located peaks in these two bins are added to form a single peak.

Figure 7 illustrates three gradient histogram functions and mean heights for candidate buildings B_1, B_2 and B_3 in Figure 6b. Figure 7a shows that B_1 has two significant peaks: 80 pixels at 0° and 117 (55+62) pixels at ±90°, these being well above the mean height of 28.6 pixels. The two significant peaks separated by 90° strongly suggest that this is a building. From Figure 7b it can be seen that B_2 has one significant peak at ±90° but a number of insignificant peaks. This points to B_2 being partly building but mostly vegetation, which is also supported by the high mean height value. With the absence of any significant peak, but a number of insignificant peaks close to the mean height, Figure 7c indicates that B_3 is comprised of vegetation. Although there may be some significant peaks in heavily vegetated areas, a high average height of bins between two significant peaks can be expected. Note that the orthophoto resolution in this case was 10cm, so a bin height of 80 pixels indicates a total length of 8m from the contributing edges.

The observations above support the theoretical inferences. In practice, however, detected vegetation clusters can show the edge characteristics of a building, and a small building having a flat roof may not have enough edges to show the required peak properties. As a result, some true buildings can be missed, while some false buildings may be detected. A number of precautions can be formulated in order to minimize the occurrence of false detections.

Two histograms are formed using edges within each detected rectangle. The first histogram, termed a collective histogram H_{col}, considers all the edges collectively (see Figure 7). The second histogram, termed an individual histogram H_{ind}, is formed for the longest individual edge within the rectangle. Figure 8 shows the three individual histogram functions for candidates B_1, B_2 and B_3 in Figure 6b. The highest peaks in individual histograms for candidates B_1, B_2 and B_3 are 9, 10 and 2 pixels respectively. It is evident that while candidates B_1 and B_2 have long straight edges, candidate B_3 does not. In other words, the individual histograms indicate that B_1 and B_2 may be on buildings, but B_3 is likely to be completely on vegetation.

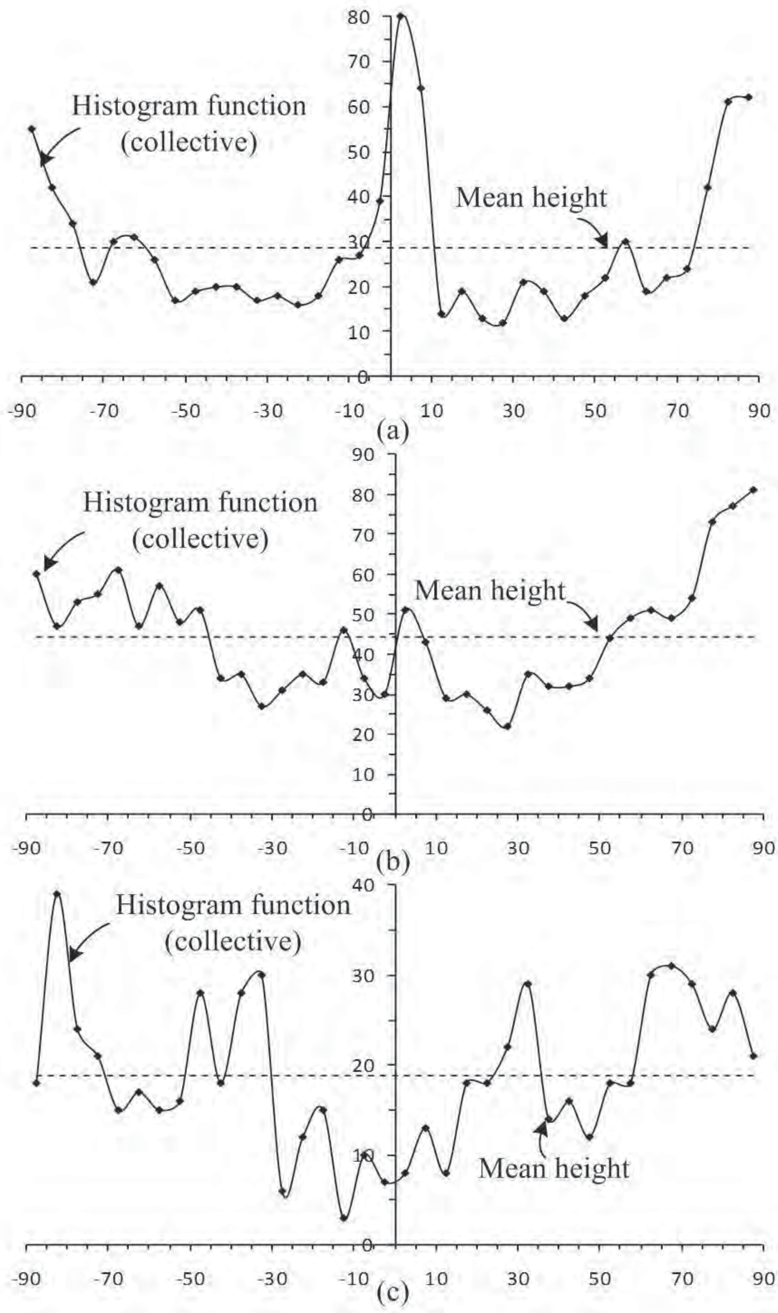

Figure 7. Gradient histogram functions (collective) and means for candidates (a) B_1, (b) B_2 and (c) B_3 in Figure 6b: *x*-axis is in degrees and *y*-axis is in pixels (bin heights). Two bins at ±90° basically form one bin, because lines in these two bins are perpendicular to the *x*-axis and reside above and below the *x*-axis. Therefore, when we have peaks at either of these bins, we accumulate their heights to form a single peak.

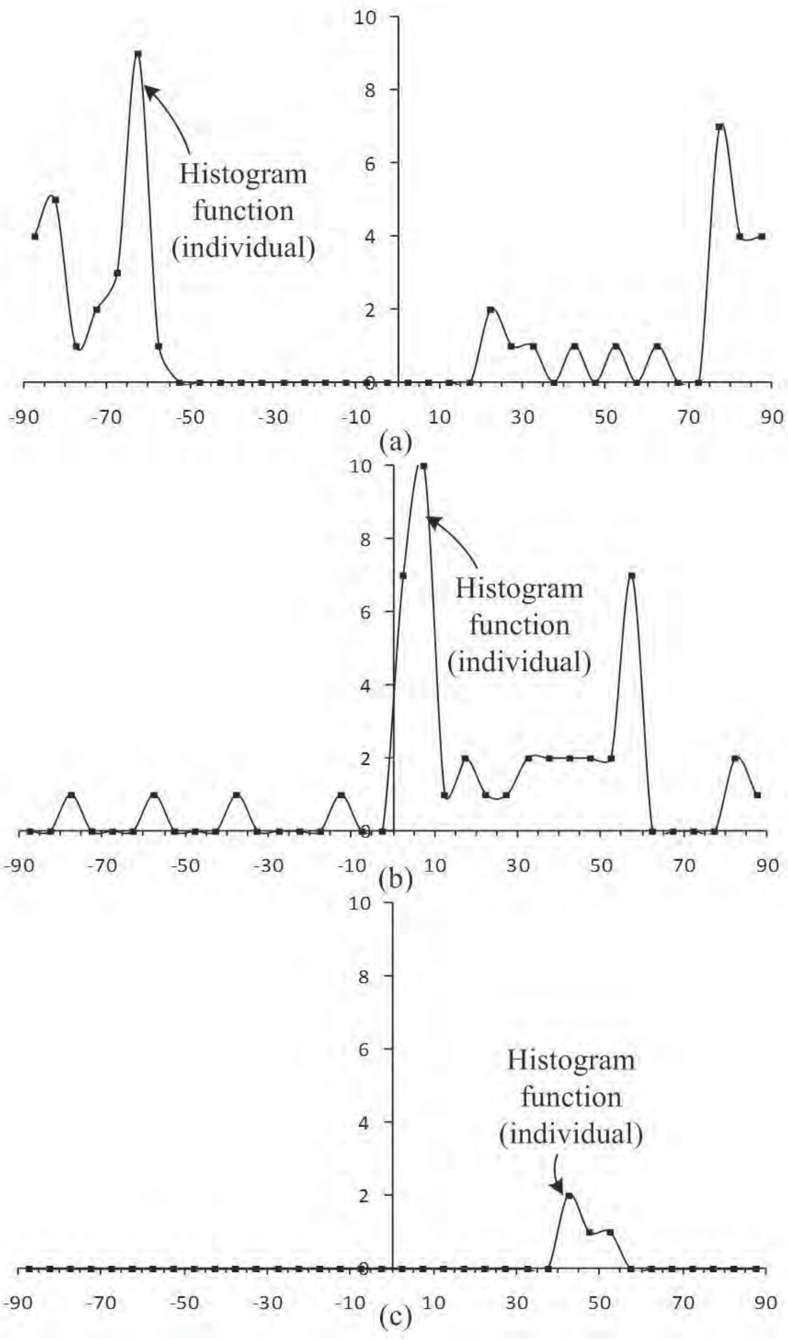

Figure 8. Gradient histogram functions (individual) for candidates (a) B_1, (b) B_2 and (c) B_3 in Figure 6b: *x*-axis is in degrees and *y*-axis is in pixels (bin heights). Two bins at ±90° basically form one bin, because lines in these two bins are perpendicular to the *x*-axis and reside above and below the *x*-axis. Therefore, when we have peaks at either of these bins, we accumulate their heights to form a single peak.

Now we carry out the following tests on H_{col} and H_{ind} to identify true buildings and to remove trees. If a detected rectangle passes at least one of the following tests it is selected as a building, otherwise it is removed as vegetation.

- **Test 1:** H_{col} has at least two peaks with heights of at least $3L_{min}$ and the average height of bins between those peaks is less than $2L_{min}$. This test ensures the selection of a large building, where at least two of its long perpendicular sides are detected. It also removes vegetation where the average height of bins between peaks is high.
- **Test 2:** The highest bin in H_{col} is at least $3L_{min}$ in height and the aggregated height of all bins in H_{col} is at most 90m. This test ensures the selection of a large building where at least one of its long sides is detected. It also removes vegetation where the aggregated height of all bins is high.
- **Test 3:** H_{col} has at least two peaks with heights of at least $2L_{min}$, and the highest bin-to-mean height ratio R_{Mm1} is at least 3. This test ensures the selection of a medium size building, where at least two of its perpendicular sides are detected. It also removes vegetation where the highest bin-to-mean height ratio is low.
- **Test 4:** The highest bin in H_{col} has a height of at least L_{min} and the highest bin-to-mean height ratio R_{Mm2} is at least 4. This test ensures the selection of a small or medium size building where at least one of its sides is at least partially detected. It also removes small to moderate sized vegetation areas where the highest bin-to-mean height ratio is low.
- **Test 5:** The highest bin in H_{ind} has a height of at least L_{min} and the aggregated height of all bins in H_{col} is at most 90m. This test ensures the selection of buildings which are occluded on at most three sides.
- **Test 6:** The ratio R_{aTp} of the detected rectangular area to the number of texture pixels (N_{Tp}, the aggregated height of all bins in H_{col}) is at least 45. This test ensures the selection of all buildings which are at least partially detected but the roof sides are missed.

The application of these tests on the complex scene in Figure 6b produces the result shown in Figure 6c.

4. Performance Evaluation

The threshold-free evaluation system (Awrangjeb et al., 2010b) involved in the performance study conducted makes one-to-one correspondences using nearest centre distances between detected and reference buildings. The words 'threshold-free' mean the evaluation system does not involve any thresholds based on human choice. In contrast, traditional approaches (Rottensteiner et al., 2005; Rutzinger et al., 2009; Lee et al., 2008) typically use one or more overlapping thresholds in determining correspondences between detected and reference building sets. The problem with threshold-based systems is that they are subjective and so there is no unique way to select the thresholds (Shufelt, 1999). Note that a building detector may use a number of thresholds whose standard values are operator selected. However, since a threshold-based evaluation system can be used to evaluate different detectors and there is no common consensus as to standard values of the thresholds (Shufelt, 1999), the evaluation results can be misleading.

We used four data sets from Australia: Fairfield, Moonee Ponds, Knox and Hobart. The experimentation was carried out in two phases. Firstly, a sensitivity analysis of six important parameters was carried out to test how the detection algorithm would perform when parameter values were changed. The standard parameter values, shown in Table 1, were chosen using three sub-images from the Moonee Ponds, Fairfield, and Knox data sets. Secondly, all parameters were set at their chosen 'standard' values and detection performance was evaluated for the full area of each data set using 15 indices in three categories: object-based, pixel-based and geometric evaluations. Since the Hobart data set was not used during the selection of the standard parameter values, the direct application of the chosen standard values to the Hobart data set for performance evaluation indicated that the standard values could be used on any other future data sets.

In order to show the efficiency of the proposed improved detector its running time is compared with the original detector. Since the test data sets differ in size, one sub-image of 2000×2000 pixels is randomly selected from each of the four data sets. For the improved detector the total running time is computed in terms of texture (entropy and edge orientation) computation and building detection, and for the original detector only the building detection time is computed. The algorithms were run on a Windows 7 Professional machine with Intel Core 2 Quad CPU @ 2.83GHz and 4GB RAM.

4.1 Threshold-Free Evaluation System

Two sets of data were used to evaluate the proposed detection process. In each, a building is represented either as a rectangular entity, for 'I' shape buildings, or as a set of rectangular entities, for 'L', 'U' and 'C' shapes. The first set $B_d = \{b_{d,i}\}$, where $0 \leq i \leq m$ and m is the number of detected rectangular entities, is known as the 'detected set'. It has been obtained from the proposed detection technique. Each entity $b_{d,i}$ is an array of four vertices and the centre of a rectangular detected entity. The second set $B_r = \{b_{r,j}\}$, where $0 \leq j \leq n$ and n is the number of reference entities, is termed 'reference set'. It is obtained from manual building measurement within the orthoimagery. Each entity $b_{r,j}$ is an array of four vertices and the centre of a rectangular reference entity.

To find the reference set B_r, manual image measurement is used. Any building-like objects above the height threshold T_h are included in B_r. As a result, some garages (car-ports) whose heights are above T_h are also included, but some building parts (verandas) whose heights are below T_h are excluded. Different building parts are referred to separate rectangular entities. Consequently, there is one entity for 'I' shape, two entities for 'L' shape, three entities for 'U' shape, four entities for 'C' shape and so on.

It is natural that different rectangular entities of the same building overlap each other. In B_r, two overlapping entities must always belong to the same building and represent two connected building parts (Figure 9a). Such an overlap is defined as a 'natural overlap' and for identification purposes a 'building identification number' b_{id} (stored in $b_{r,j}$) is assigned to each reference entity, in addition to the four vertices. Entities of the same building are assigned the same b_{id}, but those of the different buildings are assigned different b_{id} values.

For B_d, the situation is different. Here two overlapping entities may belong to the same building and represent two connected building parts. In such a case, this overlap is a 'natural overlap' (Figure 9a) and it is not counted as an error in the proposed evaluation. In all other cases, the overlap is counted as an error in the evaluation system. For example, the overlapping entities may represent the same building (multiple detection, Figure 9b) or constitute combinations of true and false detections (Figures 9c-e).

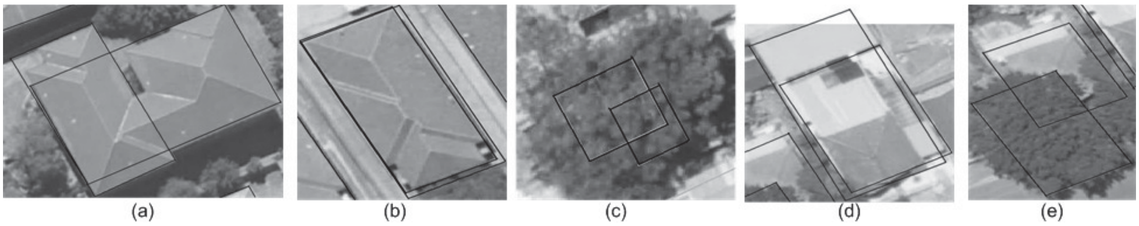

Figure 9. Different types of detection overlaps: (a) natural, (b) multiple, (c) false-false, (d) true-true and (e) true-false.

In an approach similar to that of Song and Haithcoat (2005), a detected entity is counted as correct if any of its parts overlap a reference entity. However, unlike existing evaluation systems (Rottensteiner et al., 2005; Rutzinger et al., 2009), a pseudo one-to-one correspondence is established between the detected and reference sets without using any thresholds. Pseudo one-to-one correspondence means that each entity in one set has at most one correspondence in the other set. If a detected entity overlaps only one reference entity which is not overlapped by any other detected entity, then a true correspondence is established between them. If a detected entity overlaps more than one reference entity, then the nearest reference entity (based on the distance between centers) is considered as a true correspondence for the detected entity. The same rule is applied when a reference entity is overlapped by more than one detected entity. As a consequence, there will be no correspondence for ‘false positive’ and ‘false negative’ entities.

4.2 Evaluation Indices

In the assessment of the improved algorithm, both object- and pixel-based evaluation were employed. Whereas pixel-based evaluation is based on the number of pixels within the buildings, object-based evaluation is based on the number of buildings. Seven indices were used for object-based evaluation. *Completeness* C_m, also known as ‘detection rate’ (Song and Haithcoat, 2005) or ‘producer's Accuracy’ (Foody, 2002), *correctness* C_r, also known as ‘user's accuracy’ (Foody, 2002), and *quality* Q_l have been adopted from Rutzinger et al. (2009). *Multiple detection rate* M_d, *detection overlap rate* D_o, *detection cross-lap rate* C_{rd} and r*eference cross-lap rate* C_{rr} are from Awrangjeb et al. (2010b).

A total of 7 pixel-based evaluation indices are also used, these being: *completeness* C_{mp}, also known as ‘matched overlay’ (Song and Haithcoat, 2005) and ‘detection rate’ (Lee et al., 2003), *correctness* C_{rp}, and *quality* Q_{lp} from Rutzinger et al. (2009); *area omission error* A_{oe} and *area commission error* A_{ce} from Song and Haithcoat (2005), and *branching factor* B_f and *miss factor* M_f from Lee et al. (2003).

The root-mean-square-error (RMSE) of positional discrepancy is employed to quantify geometric positional accuracy. The RMSE is measured as the average distance between a pair of detected and reference entities and it is measured for true positive entities only.

4.3 Data Sets

The test data sets employed cover four suburban areas in Australia, Fairfield, NSW; Hobart, Tasmania; Moonee Ponds, Victoria; and Knox, Victoria (see Table 2).

Table 2. Data sets (R: residential buildings, I: industrial buildings, F: flat area, H: hilly area, L: low vegetation, M: moderate vegetation, D: dense vegetation).

Scenes	**Size**	**Image (scales)**	**LIDAR**	**Buildings**	**Properties**
Fairfield	588m×417m	RGB (0.15m)	Last (0.5m)	370	R, I, F, L
Moonee Ponds	447m×447m	RGBI (0.10m)	First (1m)	250	R, I, F, M
Knox	400m×400m	RGB (0.10m)	First (1m)	130	R, H, D
Hobart	600m×600m	RGBI (0.10m)	First (1m)	200	R, H, D

The Fairfield data set covers an area of 588m×417m and contains 370 buildings, Hobart covers 600m×600m and contains 200 buildings, Moonee Ponds covers 447m×447m and has 250 buildings and Knox covers 400m×400m and contains 130 buildings. Fairfield and Moonee Ponds contain many large industrial buildings and in Moonee Ponds there were some green buildings. Knox can be characterized as outer suburban with lower housing density and extensive tree coverage that partially covers buildings. Hobart has residential buildings with moderate to dense coverage of trees that partially cover the buildings as well. In terms of topography, Fairfield and Moonee Ponds are relatively flat while Knox and Hobart are quite hilly.

LIDAR coverage comprised last-pulse returns with a point spacing of 0.5m for Fairfield, and first-pulse returns with a point spacing of 1m for Hobart, Moonee Ponds and Knox. For Fairfield and Knox, RGB color orthoimagery was available, with resolutions of 0.15m and 0.1m, respectively. Hobart and Moonee Ponds image data comprised RGBI colour orthoimagery with a resolution of 0.1m. Bare-earth DEMs of 1m horizontal resolution covered all four areas. For the data sets having only RGB color orthoimagery we have estimated the pseudo-NDVI image instead of the NDVI image following the process in Rottensteiner et al. (2005).

The orthoimagery had been created using bare-earth DEMs, so that the roofs and the tree-tops were displaced with respect to the LIDAR data. Thus, data alignment was not perfect. Apart from this registration problem, there were also problems with shadows in the orthophotography, so the NDVI and pseudo-NDVI images did not provide as much information as expected.

Reference data sets were created by monoscopic image measurement using the Barista software (Barista, 2011). All rectangular structures, recognizable as buildings and above the height threshold T_h, were digitized. The reference data included garden sheds, garages, etc. These were sometimes as small as 10m^2 in area.

4.4 Sensitivity Analysis

For the sensitivity analysis, five different values for each of the six parameters were used and object- and pixel-based qualities were estimated. These parameters comprised entropy threshold T_{ent} in percentage, orientation histogram bin distance D_{bin} in degrees, number of total texture pixels or the aggregated bin height N_{Tp} in meters, ratios of highest bin-to-mean height R_{Mm1} and R_{Mm2}, and ratio R_{aTp} of the detected area to N_{Tp} in meters. The reason for choosing *quality* as a measure for sensitivity analysis is that it provides a balance between completeness and correctness (Heipke et al., 1997). The following values were used for the six parameters:

- $\boldsymbol{T_{ent}}$: 0.2, 0.3, 0.4, 0.5 and 0.6;
- $\boldsymbol{D_{bin}}$: 3, 4, 5, 6 and 9°;
- $\boldsymbol{N_{Tp}}$: 70, 80, 90, 100 and 110m;
- $\boldsymbol{R_{Mm1}}$: 1, 2, 3, 4 and 5;
- $\boldsymbol{R_{Mm2}}$: 2, 3, 4, 5 and 6 and
- $\boldsymbol{R_{aTp}}$: 35, 40, 45, 50 and 55m.

Figure 10, in which the numbers 1 to 5 along the abscissa indicate the five values for each parameter, graphically illustrates the object- and pixel-based qualities as a percentage. When one of the parameters was changed, the others were set at their standard values. The pixel-based quality was given more weight than the object-based quality in the choice of the standard value for each parameter. This was because the object-based quality in all data sets was superior to the pixel-based quality and there was a desire to emphasize the accuracy of the detected buildings, which is better reflected by the pixel-based quality.

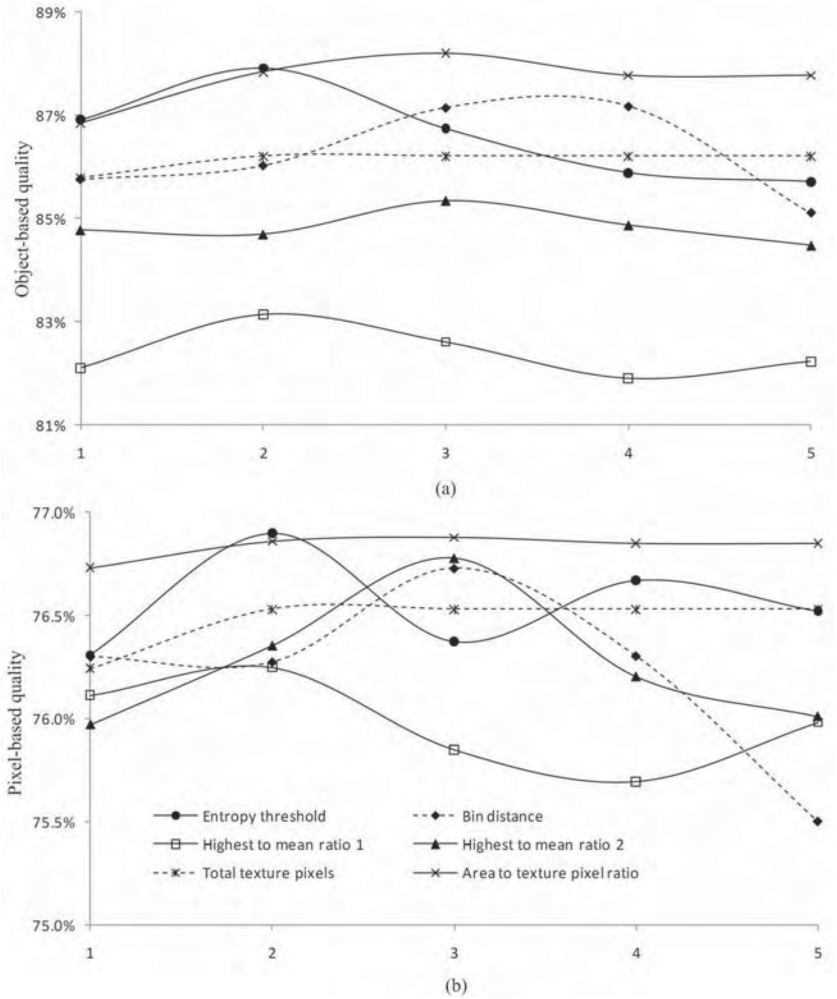

Figure 10. Sensitivity of different parameters under: (a) object-based quality and (b) pixel-based quality. The numbers 1 to 5 along the abscissa indicate the five values for each parameter mentioned in Section 4.4.

As shown in Figure 10, both object-based and pixel-based qualities were highest at entropy threshold $T_{ent} = 0.3$, ratios $R_{Mm1} = 2$, $R_{Mm2} = 4$ and $R_{aTp} = 45$. At bin distance $D_{bin} = 5°$, where the highest pixel-based quality was achieved, the object-based quality was slightly lower than the maximum.

Overall, all parameters were found to have very low sensitivity at the chosen values. In object-based evaluation, the least sensitive parameter was N_{Tp} in which the largest swing between the maximum and minimum quality values was 0.4% and the most sensitive parameter was D_{bin} for which the largest swing was 2.1%. In pixel-based evaluation, the least sensitive parameter was R_{aTp} in which the largest swing between the maximum and minimum quality values was 0.2% and the most sensitive parameter was D_{bin} for which the largest swing was 1.2%. This observation indicates that D_{bin} was the most sensitive parameter with only 2.1% object-based swing and 1.2% pixel-based swing. The chosen parameter values are recorded in Table 1.

4.5 Results and Discussion

4.5.1 Evaluation using Standard Parameter Values

Tables 3 to 5 show object-based results, pixel-based results and geometric accuracy, respectively, obtained for the four data sets with the improved building detection algorithm. Visual illustrations of selected building detection results are shown in Figures 11 to 16.

In object-based evaluation, 91% quality, on average, was observed with 95% completeness and 96% correctness (Table 3). The best performance was found in the Hobart data set: 95% quality with low error rates (M_d = 4%, D_o = 6%, C_{rd} = 3% and C_{rr} = 6%). In Knox, quality was at 87% with relatively high error rates (M_d = 5%, D_o = 13%, C_{rd} = 11% and C_{rr} = 46 %). While the performance in Fairfiled was similar to that of Hobart, the error rates in Moonee Ponds were similar to those in Knox. This was because Fairfield contained mostly residential buildings and a low vegetation density (Figure 11) and Hobart had dense vegetation with pure green leaves (Figures 13 and 14). Therefore, the NDVI information alone removed most of the vegetation in both of these scenes and a small number of false buildings were eliminated by using the edge orientation histogram. In contrast, both the moderate density vegetation in Moonee Ponds (Figure 12) and the dense vegetation in Knox were not purely green (Figures 15 and 16). As a result, most of the trees could not be removed using the NDVI. The majority of them were removed using entropy and the edge orientation histogram. Nevertheless, a low number of small trees could not be removed since small trees did not provide enough texture information in the orientation histogram. Moreover, in Moonee Ponds and Knox, some buildings were detected multiple times.

Table 3. Object-based evaluation results in percentages (C_m = completeness, C_r = correctness, Q_l = quality, M_d = multiple detection rate, D_o = Detection overlap rate, C_{rd} = detection cross-lap rate and C_{rr} = reference cross-lap rate).

Scenes	C_m	C_r	Q_l	M_d	D_o	C_{rd}	C_{rr}
Fairfield	95.1	95.4	92.2	2.7	8.6	3.5	9.7
M Ponds	94.5	95.3	89.2	6.2	13.1	7.3	17.5
Knox	94.3	91.7	86.9	5.3	13.2	10.5	45.7
Hobart	94.9	99.8	94.7	4.1	5.9	2.9	6.4
Average	**94.7**	**95.6**	**90.8**	**4.6**	**10.2**	**6.1**	**19.8**

It can be observed that the alignment of some of the detected buildings, for example, building B_1 in Figure 14, B_1 and B_2 in Figure 15 and B_2 in Figure 16, is not perfect. This is because the initial building positions were detected using the LIDAR data whose resolution was much lower than the image data. Moreover, when buildings are partially occluded by the surrounding trees, their sides may not be accurately extracted. As a result, some of the extracted line segments were not found exactly parallel to the corresponding building sides and the resulting detected buildings have a negative impact on the pixel-based accuracy measurements (Table 4). It should be emphasized that accurate alignment of the detected buildings is not conducted as yet as part of the reported process; the rectangular box shown for each building simply indicates that it has been detected.

Table 4. Pixel-based evaluation results in percentages (C_{mp} = completeness, Crp = correctness, Q_{lp} = quality, A_{oe} = area omission error, A_{ce} = area commission error, B_f = branching factor and M_f = miss factor).

Scenes	C_{mp}	Crp	Q_{lp}	A_{oe}	A_{ce}	B_f	M_f
Fairfield	83.2	84.5	72.4	15.3	12.5	13.5	20.3
M Ponds	87.2	85.4	75.3	12.7	13.2	16.7	17.3
Knox	74.1	77.3	60.9	26.4	21.0	29.4	34.9
Hobart	80.8	80.2	67.4	20.3	19.0	25.0	23.7
Average	**81.3**	**81.9**	**69.0**	**18.7**	**16.4**	**21.2**	**24.1**

Table 5. Geometric accuracy expressed via RMSE values, along with running time in terms of texture (entropy and edge orientation) computation and building detection.

Scenes	Geometric accuracy		Running time (seconds)		
	Meters	**Pixels**	**Texture**	**Detection**	**Total**
Fairfield	2.4	16.0	155.8	181.9	337.7
M Ponds	1.6	16.0	177.7	95.5	273.2
Knox	2.3	23.0	287.4	384.9	672.3
Hobart	1.9	19.0	190.4	75.7	266.1
Average	**2.1**	**18.5**	**202.8**	**184.5**	**387.3**

In pixel-based evaluation, on average 69% quality was observed with 81% completeness and 82% correctness (Table 4). With respect to different data sets, the pixel-based performance followed the same trend as the object-based performance discussed above.

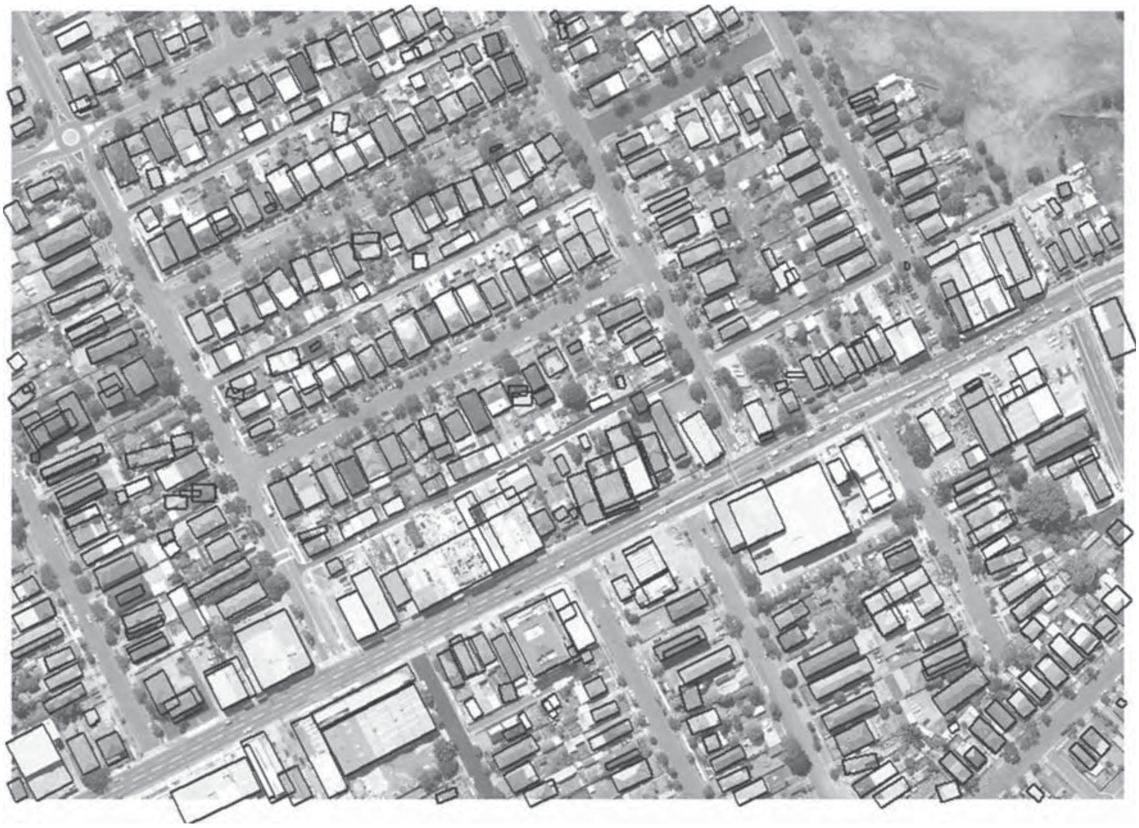

Figure 11. Building detection by the improved algorithm on a scene from Fairfield.

When compared to the object-based performance, the pixel-based performance was poor, for the following main reasons. Firstly, due to the use of bare-earth DEMs for orthoimage generation, large irregular and random registration errors were observed between the LIDAR and imagery data for all four data sets. Secondly, due to occlusions on buildings some reference buildings could not be fully digitized. Finally, the detected rectangles on the occluded buildings also included parts of the occluding trees. The final reason was also evident from the geometric accuracy shown in Table 5. There were 19 and 23 pixels RMS error in the Hobart and Knox data sets, while in Fairfield and Moonee Ponds the corresponding value was 16 pixels. The occluding trees in both Knox and Hobart caused the low geometric accuracy.

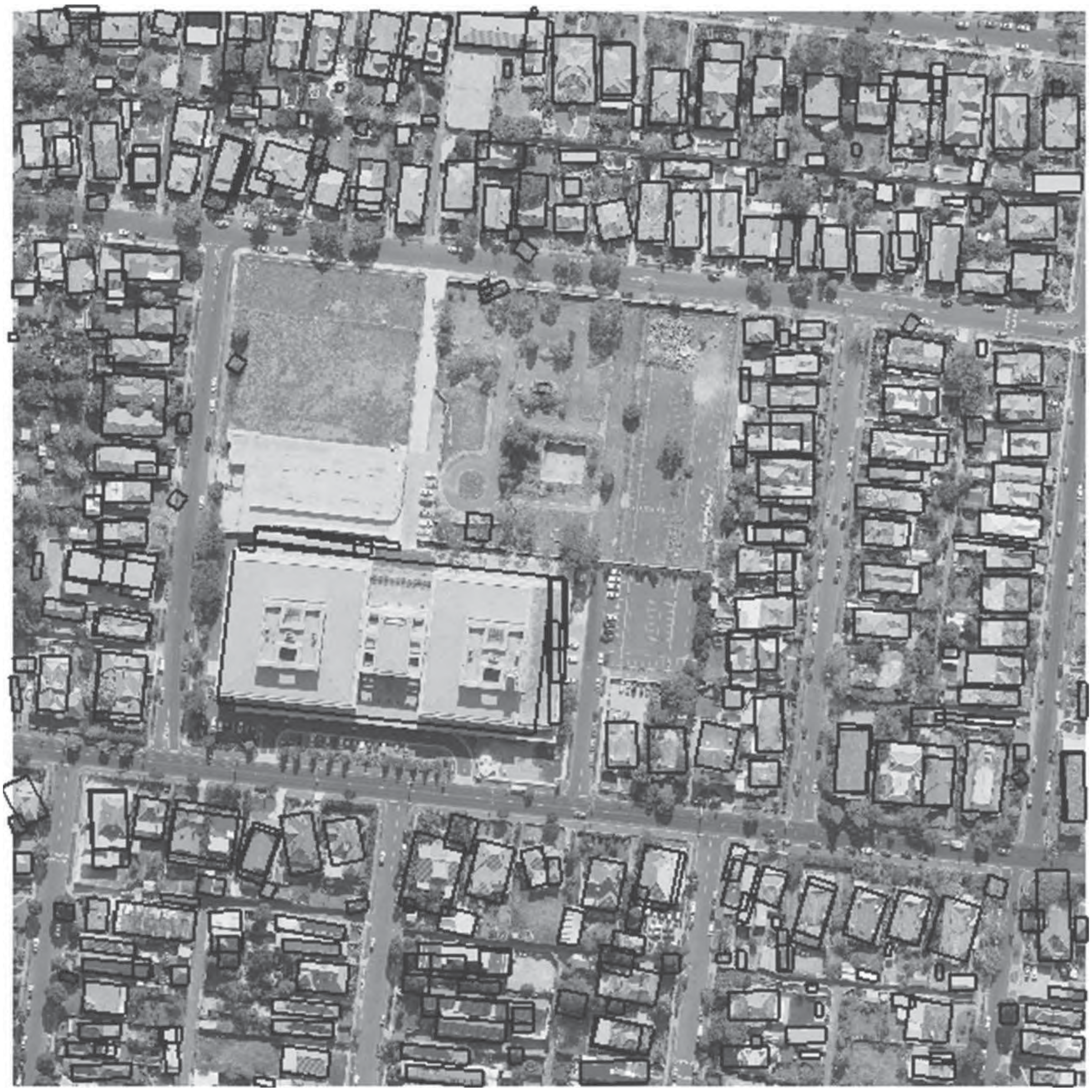

Figure 12. Building detection by the improved algorithm on a scene from Moonee Ponds.

In general, the improved detector was able to detect buildings in the case of hilly areas with dense vegetation and when the trees had colors other than pure green. It was also effective when buildings were seriously occluded by dense vegetation. For example, buildings B_1 in Figure 13; B_1 to B_7 in Figure 14; B_1 to B_3 in Figure 15; and B_2 to B_4 in Figure 16 were highly occluded by trees, and building B_6 in Figure 14c was also partially shadowed. The improved detector detected all these buildings successfully. However, in some cases, when a building was almost occluded or had low height the improved detector failed. In the first instance (see building B_8 in

Figure 14c), the improved detector was unable to detect a building edge with minimum length (at least 3m). In the second (building B_1 in Figure 16b), the height threshold estimated using the local DEM height assigned the low-height building to be ground (white region) in the ground mask.

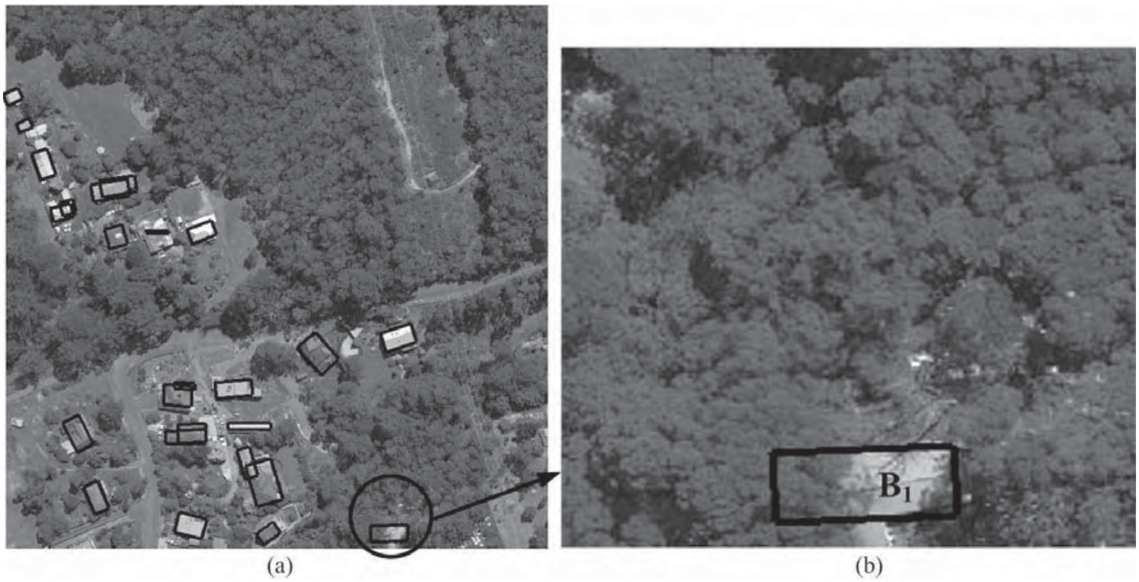

Figure 13. Building detection by the improved detector on a scene from Hobart.

Table 5 also shows the running time in seconds for the proposed improved detector. In terms of texture computation time, Fairfield was the fastest since it was low in vegetation and Knox was the slowest since it was dense in vegetation. Moonee Ponds and Hobart were moderate in vegetation and therefore they were faster than Knox but slower than Fairfield in texture computation. In terms of building detection time, Moonee Ponds and Hobart were faster, which indicates that the majority of the line segments around trees were removed by using the NDVI and entropy. Fairfield was slower than Moonee Ponds and Hobart, since it had a large number of buildings. Knox was the slowest since a large number of line segments around trees (not pure green in color) could not be removed, which resulted in a large number of false building candidates. These were removed by using the edge orientation histogram. On average, texture computation takes almost the same time as the detection procedure.

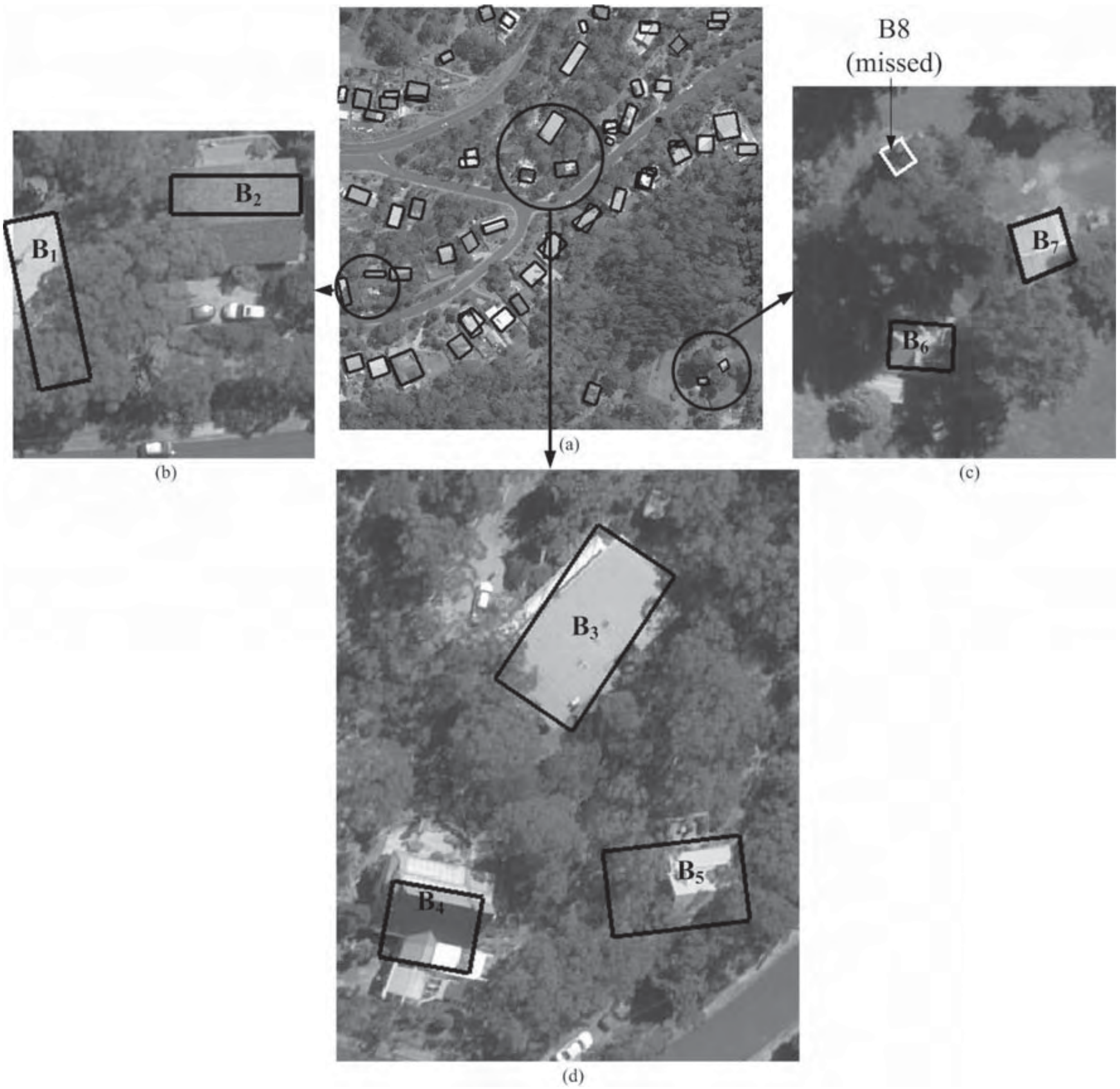

Figure 14. Building detection by the improved detector on a second scene from Hobart.

4.5. 2 Comparison with Other Detectors

Since different published detection techniques follow different evaluation systems on different data sets, they are difficult to compare. The original detector in (Awrangjeb et al., 2010b) was evaluated using the same threshold-free evaluation system on Fairfiled and Moonee Ponds data sets (Awrangjeb et al., 2010a). We have evaluated the original detector on Knox and Hobart data sets too. The Fairfield data set was also previously employed by Rottensteiner et al. (2005), Rottensteiner et al. (2007) and Rutzinger et al. (2009) to investigate automated building extraction techniques, but they used threshold-based evaluation systems. The evaluation results in Sohn and Dowman (2007) and Cheng et al. (2008) were on different data sets and also used threshold-based evaluation systems. They can therefore not be compared.

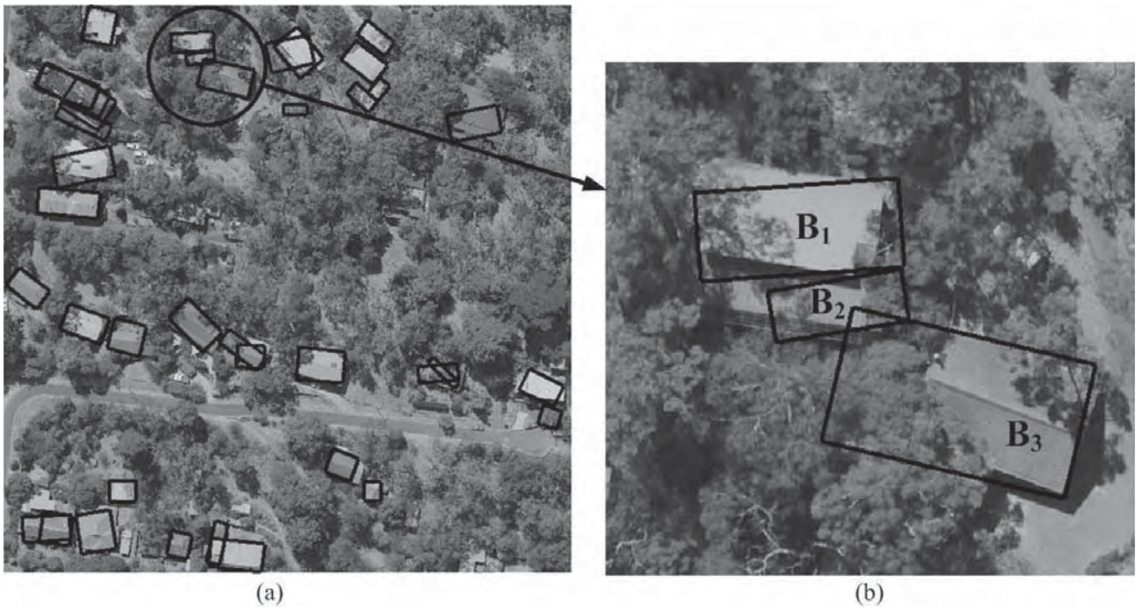

Figure 15. Building detection by the improved detector on a scene from Knox.

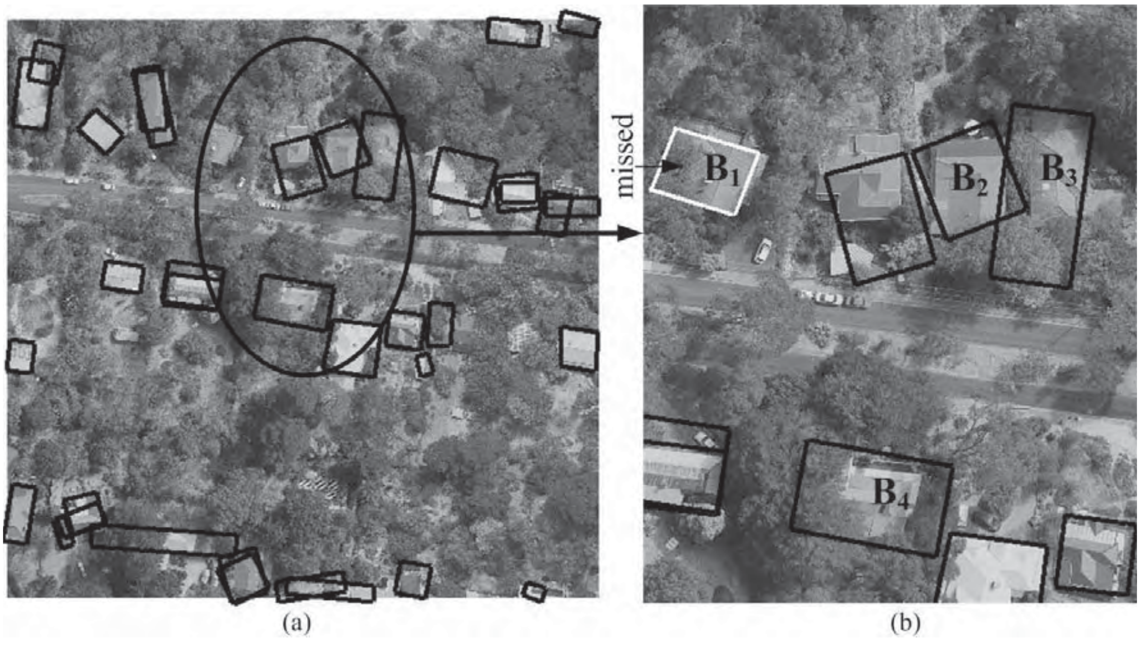

Figure 16. Building detection by the improved detector on a second scene from Knox.

When compared to the evaluation results in Awrangjeb et al. (2010a,b), the improved algorithm produced moderately better performance than the original algorithm in object-based, pixel-based and geometric accuracy for both the Fairfield and Moonee Ponds data sets. The better performance was mainly due to proper detection of large industrial buildings in both of the scenes, detection of the available green buildings in Moonee Ponds, and elimination of trees in both of the scenes.

However, the improved algorithm has been designed for enhanced performance in hilly areas with extensive tree coverage, as best exemplified by the Hobart and Knox data sets in Figures 13 to 16. In both Knox and Hobart, the improved algorithm exhibited significantly better performance over the original, due to two main reasons. Firstly, the improved algorithm better accommodated the dense tree cover that characterized the Hobart and Knox data sets. Fairfield and Moonee Ponds on the other hand exhibit low vegetation coverage. Secondly, the improved algorithm showed its merits in better handling varying topography. Hobart (maximum height H_M = 226m and minimum height H_m = 140m) and Knox H_M = 270m and H_m = 110m) are hilly areas, whereas Fairfield (H_M = 23m and H_m = 1m) and Moonee Ponds (H_M = 43m and H_m = 23m) are moderately flat.

Therefore, the original algorithm detected a large number of false buildings and missed many of the true buildings in both Hobart and Knox. For example, Figure 17 shows the building detection results, obtained with the original detector, on the same Knox scene shown in Figure 15a. It performed much worse than the proposed improved detector in both of the scenes. In the object-based evaluation in Knox, the original detector offered 56% quality with 77% completeness and 67% correctness. In contrast, as shown in Table 3, the proposed improvement provided 87% quality with 94% completeness and 92% correctness in Knox. In pixel-based evaluation, the original detector gave 27% quality with 44% completeness and 42% correctness. In contrast, Table 4 shows that the improved detector offered 61% quality with 74% completeness and 74% correctness. The geometric accuracy for Knox obtained with the original detector was 10 pixels worse than that from the improved detector.

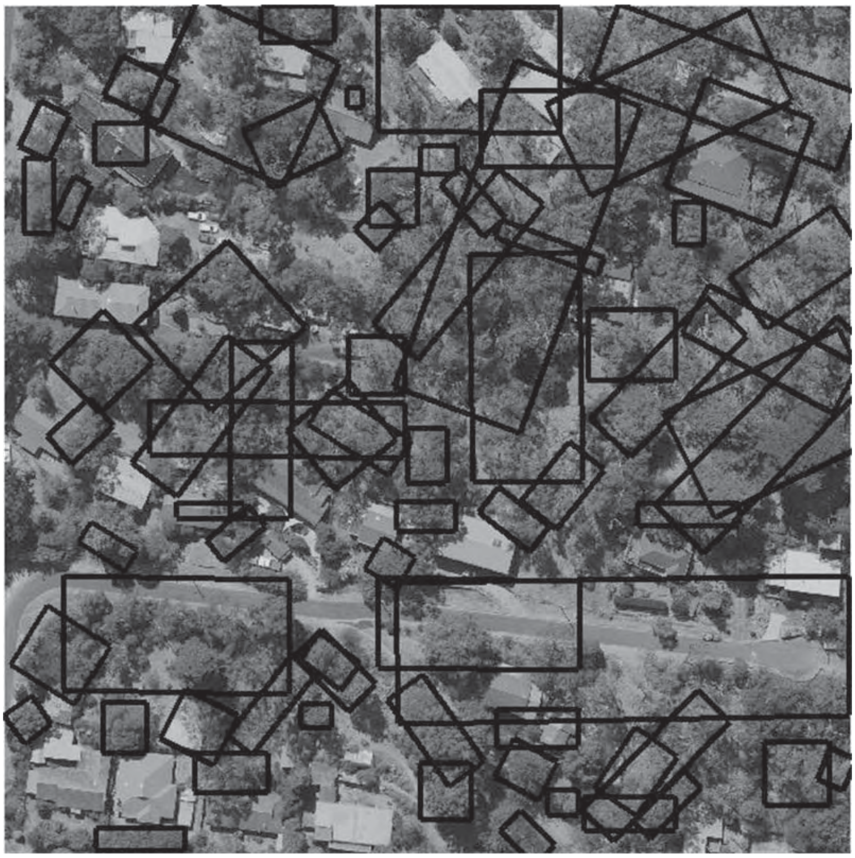

Figure 17. Building detection by the original detector on the Knox scene shown in Figure 15.

In terms of running time, the original detector (Awrangjeb et al., 2010b) does not require the texture computation step. Since this step is independent of the building detection step, texture information for the improved algorithm can be computed separately (offline or through hardware implementation). The detection time for the original detector for Fairfield, Moonee Ponds, Hobart and Knox was 150, 62, 44 and 256 seconds, respectively, with an average of 128 seconds. By comparing this with the improved detector performance in Table 5, it can be observed that the improved detector takes 44% more time for building detection. This moderate increase in computation time is well compensated by a significant increase in building detection performance in terms of object-based, pixel-based and geometric accuracy.

Rottensteiner et al. (2005), Rottensteiner et al. (2007) and Rutzinger et al. (2009) evaluated different detectors in terms of *completeness*, *correctness* and *quality* using two different threshold-based evaluation systems. Rutzinger et al. (2009) has presented results of pixel-based evaluation of the Dempster-Shafer (DS) detector showing that it can offer higher *completeness* (92.1%) and *quality* (81.8%) than the proposed detector. However, in object-based evaluation the DS detector offered much lower *completeness* (44.2%) and *quality* (43.1%) than the proposed detector. The superior performance of the DS detector in pixel-based evaluation was largely due to the adopted evaluation systems, Rottensteiner et al. (2005) and Rutzinger et al. (2009), which excluded false positive and false negative buildings from evaluation and established many-to-many relationships between the detected and reference sets. Moreover, unlike the proposed detector the DS detector was excessively sensitive to small buildings (performance deteriorated with the decrease of building size) and buildings smaller than 30m^2 could not be detected Rottensteiner et al. (2007).

5. Conclusion

This paper has presented an improved automatic building detection technique which exhibits better separation of buildings from trees. In addition to employing height and width thresholds and color information, it uses texture information from both LIDAR and color orthoimagery. The height threshold is used to generate the ground mask where, unlike the so-called normalized DSM, buildings are found to be well separated from the surrounding vegetation. The width threshold helps in removing trees of small horizontal coverage. The joint application of entropy and NDVI helps in the removal of moderate to large vegetation by making trees more easily distinguishable. Finally, a rule-based procedure based on the edge orientation histogram from the image edges assists in eliminating false positive building candidates. It is specifically helpful in removing trees which are not pure green, bearing in mind that trees may change color and lose leaves in different seasons.

Experimental evaluation of the improved algorithm was carried out in two phases. In the first, a sensitivity analysis was performed in which the selection of standard parameter values was carried out using three representative samples from three test sites, Fairfield, Moonee Ponds and Knox, each having distinct characteristics in terms of terrain slope, building type, roof structure and color, density of vegetation, scene complexity, resolution of LIDAR and orthoimages, and date and time of aerial photography. The parameters involved in the approach were found to be sufficiently insensitive to different settings. In the second phase, the standard parameter values were employed to evaluate object-based, pixel-based and accuracy performance using 15 indices. The experimental results showed that whereas the improved algorithm produced moderately enhanced performance in Fairfield and Moonee Ponds, it yielded a very significant improvement in building detection in Knox and Hobart, across all three categories of evaluation indices. It also offered better performance than Rottensteiner et al. (2005).

However, the improved detector is moderately slow in building detection as compared to the original detector in Awrangjeb et al. (2010b). It is also acknowledged that there will be some unusual situations in which the improved algorithm will fail. For example, this can occur when the entropy information employed for distinguishing trees and green buildings results in the removal of buildings with green, textured roofs. It is also likely when vegetation with shadows and self-occlusions displays very low entropy and hence no edge information. Consequently, Test 6 in Section 3.3, which employs the edge orientation histogram, may detect small vegetation clusters with self-occlusions or shadows as buildings. Finally, due to registration error between the LIDAR data and orthoimagery, some trees and especially those beside roads can still be detected as buildings. Future research will focus upon resolving these issues as well as upon the 3D reconstruction of complex building roofs.

6. Acknowledgement

The authors would like to thank the Department of Sustainability and Environment (www.dse.vic.gov.au) and Photomapping Services (www.photomapping.com.au) for providing the LIDAR data and orthoimagery for the test sites.

7. References

Awrangjeb, M., M. Ravanbakhsh, and C.S. Fraser, 2010a. Automatic building detection using lidar data and multispectral imagery, *Proceedings of the Digital Image Computing: Techniques and Applications*, Sydney, Australia, pp. 45–51.

Awrangjeb, M., M. Ravanbakhsh, and C.S. Fraser, 2010b. Automatic detection of residential buildings using lidar data and multispectral imagery, *ISPRS Journal of Photogrammetry and Remote Sensing*, 65 (5): 457–467.

Awrangjeb, M., M. Ravanbakhsh, and C.S. Fraser, 2010c. Building detection from multispectral imagery and lidar data employing a threshold-free evaluation system, *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 38 (part 3A): 49–55.

Barista, 2011. The barista software. URL www.baristasoftware.com.au.

Chen, L., T. Teo, C. Hsieh, and J. Rau, 2006. Reconstruction of building models with curvilinear boundaries from laser scanner and aerial imagery, *Lecture Notes in Computer Science*, 4319: 24–33.

Cheng, L., J. Gong, X. Chen, and P. Han, 2008. Building boundary extraction from high resolution imagery and lidar data, *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 37 (part B3): 693–698.

Cheng, L., J. Gong, M. Li, and Y. Liu, 2011. 3D building model reconstruction from multi-view aerial imagery and lidar data, *Photogrammetric Engineering and Remote Sensing*, 77 (2): 125–139.

Dash, J., E. Steinle, R.P. Singh, and H.P. Bahr, 2004. Automatic building extraction from laser scanning data: an input tool for disaster management, *Advances in Space Research*, 33 (3): 317–322.

Demir, N., D. Poli, and E. Baltsavias, 2009. Extraction of buildings using images & lidar data and a combination of various methods, *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 38 (part 3/W4): 71–76.

Foody, G., 2002. Status of land cover classification accuracy assessment, *Remote Sensing of Environment*, 80 (1): 185–201.

Gonzalez, R. C., R.E. Woods, and S.L. Eddins, 2003. *Digital Image Processing Using MATLAB*, Prentice Hall, New Jersey.

Haala, N., and C. Brenner, 1999. Extraction of buildings and trees in urban environments, *ISPRS Journal of Photogrammetry and Remote Sensing*, 54 (2-3): 130–137.

Habib, A., R. Zhai, and C. Kim, 2010. Generation of complex polyhedral building models by integrating stereo-aerial imagery and lidar data, *Photogrammetric Engineering and Remote Sensing*, 76 (5): 1–15.

Heipke, C., H. Mayer, C. Wiedemann, and O. Jamet, 1997. Evaluation of automatic road extraction, *International Archives of Photogrammetry and Remote Sensing*, 32 (part 3-2W3): 47–56.

Huang, X., and L. Zhang, 2011. A multidirectional and multiscale morphological index for automatic building extraction from multispectral GeoEye-1 imagery, *Photogrammetric Engineering and Remote Sensing*, 77 (7): 721–732.

Huang, X., L. Zhang, and W. Gong, 2011. Information fusion of aerial images and LIDAR data in urban areas: vector stacking, re-classification, and post-processing approaches, *International Journal of Remote Sensing*, 32 (1): 69–84.

Huang, X., and L. Zhang, 2012. Morphological building/shadow index for building extraction from high resolution imagery over urban areas. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, DOI: 10.1109/JSTARS.2011.2168195.

Khoshelham, K., Z. Li, and B. King, 2005. A split-and-merge technique for automated reconstruction of roof planes, *Photogrammetric Engineering and Remote Sensing*, 71 (7): 855–862.

Khoshelham, K., S. Nedkov, and C. Nardinocchi, 2008. A comparison of Bayesian and evidence-based fusion methods for automated building detection in aerial data, *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 37 (part B7): 1183–1188.

Lee, D., K. Lee, and S. Lee, 2008. Fusion of lidar and imagery for reliable building extraction, *Photogrammetric Engineering and Remote Sensing*, 74 (2): 215–226.

Lee, D., J. Shan, and J. Bethel, 2003. Class-guided building extraction from ikonos imagery, *Photogrammetric Engineering and Remote Sensing*, 69 (2): 143–150.

Maas, H. G., 2001. The suitability of airborne laser scanner data for automatic 3D object reconstruction, *Proceedings of the 3rd International Workshop on Automatic Extraction of Man-Made Objects from Aerial and Space Images*, Ascona, Switzerland, pp. 345–356.

Matikainen, L., H. Kaartinen, and J. Hyyppa, 2007. Classification tree based building detection from laser scanner and aerial image data, *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 36 (part 3/W52): 280–287.

Rottensteiner, F., J. Trinder, S. Clode, and K. Kubik, 2005. Using the Dempster-Shafer method for the fusion of lidar data and multi-spectral images for building detection, *Information Fusion* 6 (4): 283–300.

Rottensteiner, F., J. Trinder, S. Clode, and K. Kubik, 2007. Building detection by fusion of airborne laser scanner data and multi-spectral images: Performance evaluation and sensitivity analysis, *ISPRS Journal of Photogrammetry and Remote Sensing*, 62 (2): 135–149.

Rutzinger, M., F. Rottensteiner, and N. Pfeifer, 2009. A comparison of evaluation techniques for building extraction from airborne laser scanning, *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2 (1): 11–20.

Salah, M., J. Trinder, and A. Shaker, 2009. Evaluation of the self-organizing map classifier for building detection from lidar data and multispectral aerial images, *Journal of Spatial Science*, 54 (2): 1–20.

Sampath, A., and J. Shan, 2007. Building boundary tracing and regularization from airborne lidar point clouds, *Photogrammetric Engineering and Remote Sensing*, 73 (7): 805–812.

Sampath, A., and J. Shan, 2010. Segmentation and reconstruction of polyhedral building roofs from aerial lidar point clouds, *IEEE Transactions on Geoscience and Remote Sensing*, 48 (3): 1554–1567.

Shan, J., and S. Lee, 2005. Quality of building extraction from ikonos imagery, *ASCE Journal of Surveying Engineering*, 131 (1): 27–32.

Shorter, N., and T. Kasparis, 2009. Automatic vegetation identification and building detection from a single nadir aerial image, *Remote Sensing*, 1 (4): 731–757.

Shufelt, J., 1999. Performance evaluation and analysis of monocular building extraction from aerial imagery, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 21 (4): 311–326.

Sohn, G., and I. Dowman, 2007. Data fusion of high-resolution satellite imagery and lidar data for automatic building extraction, *ISPRS Journal of Photogrammetry and Remote Sensing*, 62 (1): 43–63.

Song, W., and T. Haithcoat, 2005. Development of comprehensive accuracy assessment indexes for building footprint extraction, *IEEE Transactions on Geoscience and Remote Sensing*, 43 (2): 402–404.

Vu, T., F. Yamazaki, and M. Matsuoka, 2009. Multi-scale solution for building extraction from lidar and image data, *International Journal of Applied Earth Observation and Geoinformation*, 11 (4): 281–289.

Zhang, K., J. Yan, and S.C. Chen, 2006. Automatic construction of building footprints from airborne lidar data, *IEEE Trans. on Geoscience and Remote Sensing*, 44 (9): 2523–2533.