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

Mohammad Awrangjeb, Chunsun Zhang, and Clive S. Fraser

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. First, 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. Second, 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.

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). First, there are many algorithms which use 2D or 3D information from photogrammetric imagery (Huang and Zhang, 2011). Second, 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 3 m 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 detects 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. First, 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). Second, 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 combine the use of two or more of these cues.
- We have proposed an effective method of separating buildings from trees. 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. 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, 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 improvement 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.

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 (2.5 m 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. (2011). Trees taller than the building roof cannot be removed using 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. First, the height difference between the pulses must be at least 2 m (Maas, 2001). Second, 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 postprocessing 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 3 m for width; however, Lee et al. (2003) have employed a high area threshold 50 m² which can also remove small buildings. Chen et al. (2006) employed an area threshold of 30 m² 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 30 m² 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 with 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 70 m² (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 percent 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, e.g., 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 percent 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 a 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 four 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.

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.

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 to be presented.

| Application of 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 (shrubs, road furniture, cars, etc.) and preserves non-ground points (trees and buildings).

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Standard Values</th>
<th>Reference for values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application of Height Threshold</td>
<td>$T_h$</td>
<td>$H_g + 2.5$ m</td>
</tr>
<tr>
<td>Use of Width, NDVI, and Entropy</td>
<td>$T_{ndvi}$</td>
<td>10</td>
</tr>
<tr>
<td>and Entropy</td>
<td>$T_{ent}$</td>
<td>0.3</td>
</tr>
<tr>
<td>Application of Edge Orientation Histogram</td>
<td>$D_{th}$</td>
<td>5°</td>
</tr>
<tr>
<td>Orientation Histogram</td>
<td>$N_{th}$</td>
<td>90 m</td>
</tr>
<tr>
<td>$R_{min}$</td>
<td>2</td>
<td>Sensitivity Analysis</td>
</tr>
<tr>
<td>$R_{max}$</td>
<td>4</td>
<td>Sensitivity Analysis</td>
</tr>
<tr>
<td>$R_{tp}$</td>
<td>45 m</td>
<td>Sensitivity Analysis</td>
</tr>
</tbody>
</table>

Figure 1. Flow diagram of the improved building detection technique.
assigned 1. In addition, since the horizontal point distance of the lidar data $d_i$ is generally higher than that of the orthoimage $d_j$, all the pixels in a $n \times n$ neighborhood of $(x, y)$ are also assigned 1, where $n = 2 + d_i/d_j$. Consequently, the black areas in $M_g$ indicate “void areas” where there are no laser returns below $T_{g0}$ (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 $T_{g0}$. Therefore, as shown in Figure 2, buildings and 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 $T_{g0}$). 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.

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 1.5 m 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 $T_{ndvi}$ 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 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 $9 \times 9$ sub image $I$ is considered (Gonzalez et al., 2003), where $P$ is the center 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

$$e = -\sum H_i \log_2 H_i, \quad \text{where } 1 \leq i \leq 256 \text{ and } 0 < H_i \leq 1. \quad (1)$$
In the improved detector, if the mean $\text{NDVI}$ is above $T_{\text{ndvi}}$, 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 $T_{\text{ent}} = 30\%$. In other words, whether the number of white pixels in $M_s$ is more than $T_{\text{ent}}$. 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 $\text{NDVI}$ 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 1.5 $L_{\text{min}}$ 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.

**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 $\text{NDVI}$ and entropy in the previous section we have considered a rectangle of width 1.5 m (since $L_{\text{min}} = 3$ m) on each side of an extracted line (see Figure 5a). However, due to sunlight, shadows and self-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 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 2 m 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 $g_x$ with scale $\sigma = 3$:

$$x_s(t) = x(t) * g_x \quad \text{and} \quad y_s(t) = y(t) * g_x$$  \hspace{1cm} (2)

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

$$x'_{s}(t) = \frac{x_s(t+1) - x_s(t-1)}{2} \quad \text{and} \quad y'_{s}(t) = \frac{y_s(t+1) - y_s(t-1)}{2}$$  \hspace{1cm} (3)
Finally, the gradient (tangent angle) function for a given curve $f(t)$ is calculated as:

$$
\Delta_y(t) = \tan^{-1}\left(\frac{y(t)}{x(t)}\right).
$$

(4)

The gradient obtained at each point using Equation 4 will lie within the range of $[-90^\circ, +90^\circ]$. A histogram with a successive bin distance of $D_{bin} = 5^\circ$ 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 $\Delta_y$, 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 $-90^\circ$ to $-85^\circ$) and last (at $85^\circ$ to $90^\circ$) 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 $0^\circ$ and 117 (55 + 62) pixels at $90^\circ$, these being well above the mean height of 28.6 pixels. The two significant peaks separated by $90^\circ$ strongly suggest that this is a building. From Figure 7b it can be seen that $B_2$ has one significant peak at $\pm 90^\circ$ 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 10 cm, so a bin height of 80 pixels indicates a total length of 8 m 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 $H_{coll}$, 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.

Now we carry out the following tests on $H_{coll}$ 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:** $H_{coll}$ 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 \( H_{col} \) has a height of at least \( 3L_{max} \) in height, and the aggregated height of all bins in \( H_{col} \) is at most 90 m. This test ensures the selection of a large building where at least one of its 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 \( 2L_{max} \) and the highest bin-to-mean height ratio \( R_{bin} \) is at least 2. This test ensures the selection of the size of a medium 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 \( R_{bin} \) 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 \( H_{col} \) has a height of at least \( L_{min} \) and the aggregated height of all bins in \( H_{col} \) is at most 90 m. This test ensures the selection of buildings which are occluded on at most three sides.

- **Test 6:** The ratio \( R_{col} \) 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.

### 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. First, 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. Second, all parameters were set at their chosen “standard” values and detection 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 pixels \( \times 2000 \) 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.

### 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 \( B_d = \{ b_{ij} \} \), 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 \( b_{ij} \) is an array of four vertices and the center of a rectangular detected entity. The second set \( B_r = \{ b_{ij} \} \), 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 orthoimage. Each entity \( b_{ij} \) is an array of four vertices and the center 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” \( b_{ij} \) (stored in \( b_{ij} \)) is assigned to each reference entity, in addition to the four vertices. Entities of the same building are assigned the same \( b_{ij} \), but those of the different buildings are assigned different \( b_{ij} \) values.

For \( R_b \), 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, 9d, and 9e).

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 the detected and reference sets. 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.

**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 building, object-based evaluation is based on the number of buildings. Seven indices were used for object-based evaluation. Object-based evaluation is based on the number of pixels within a building (multiple detection; Figure 9b) or constitute combinations of true and false detections.

### 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).

The Fairfield data set covers an area of 588 m \( \times \) 417 m and contains 370 buildings. Hobart covers 600 m \( \times \) 600 m and contains 200 buildings. Moonee Ponds covers 447 m \( \times \) 447 m and has 250 buildings, and Knox covers 400 m \( \times \) 400 m 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 0.5 m for Fairfield, and first-pulse returns with a point spacing of 1 m for Hobart, Moonee Ponds, and Knox. For Fairfield and Knox, RGB color orthoimagery was available. For Hobart and Knox, RGB color orthoimagery with a resolution of 0.1 m and 0.1 m, respectively. Hobart and Moonee Ponds image data comprised RGB color orthoimagery with a resolution of 0.1 m. Bare-earth DEMs of 1 m 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 10 m\(^2\) in area.

### 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 \( R_{bin1} \) and \( R_{bin2} \), and ratio \( R_{tp} \) of the detected area to \( N_{tp} \) in meters. The reason for choosing \( quality \) as a measure for sensitivity analysis is that it...

---

**Table 2. Data Sets**

<table>
<thead>
<tr>
<th>Scenes</th>
<th>Size</th>
<th>Image (scales)</th>
<th>Lidar</th>
<th>Buildings</th>
<th>Properties</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fairfield</td>
<td>588m × 417m</td>
<td>RGB (0.15 m)</td>
<td>Last (0.5 m)</td>
<td>370</td>
<td>R, I, F, L</td>
</tr>
<tr>
<td>Moonee Ponds</td>
<td>447m × 447m</td>
<td>RGBH (0.10 m)</td>
<td>First (1 m)</td>
<td>250</td>
<td>R, I, F, M</td>
</tr>
<tr>
<td>Knox</td>
<td>400m × 400m</td>
<td>RGB (0.10 m)</td>
<td>First (1 m)</td>
<td>130</td>
<td>R, H, D</td>
</tr>
<tr>
<td>Hobart</td>
<td>600m × 600m</td>
<td>RGBH (0.10 m)</td>
<td>First (1 m)</td>
<td>200</td>
<td>R, H, D</td>
</tr>
</tbody>
</table>

*PHOTOGRAMMETRIC ENGINEERING & REMOTE SENSING*  
*July 2012*  
*Page 737*
provides a balance between completeness and correctness (Heipke et al., 1997). The following values were used for the six parameters:

- $T_{\text{ent}}$: 0.2, 0.3, 0.4, 0.5 and 0.6;
- $D_{\text{bin}}$: 3, 4, 5, 6, and 9°;
- $N_{\text{TP}}$: 70, 80, 90, 100, and 110m;
- $R_{\text{m1}}$: 1, 2, 3, 4, and 5;
- $R_{\text{m2}}$: 2, 3, 4, 5, and 6;
- $R_{\text{TP}}$: 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.

As shown in Figure 10, both object-based and pixel-based qualities were highest at entropy threshold $T_{\text{ent}} = 0.3$, ratios $R_{\text{m1}} = 2$, $R_{\text{m2}} = 4$, and $R_{\text{TP}} = 45$. At bin distance $D_{\text{bin}} = 5°$, 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 $N_{\text{TP}}$, in which the largest swing between the maximum and minimum quality values was 0.4 percent and the most sensitive parameter was $D_{\text{bin}}$ for which the largest swing was 2.1 percent. In pixel-based evaluation, the least sensitive parameter was $R_{\text{TP}}$, in which the largest swing between the maximum and minimum quality values was 0.2 percent and the most sensitive parameter was $D_{\text{bin}}$ for which the largest swing was 1.2 percent. This observation indicates that $D_{\text{bin}}$ was the most sensitive parameter with only 2.1 percent object-based swing and 1.2 pixel-based swing. The chosen parameter values are recorded in Table 1.
Results and Discussion

Evaluation using Standard Parameter Values

Tables 3, 4, and 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 percent quality, on average, was observed with 95 percent completeness and 96 percent correctness (Table 3). The best performance was found in the Hobart data set: 95 percent quality with low error rates ($M_d = 4\%$, $D_o = 6\%$, $C_{oll} = 3\%$ and $C_{rr} = 6\%$). In Knox, quality was at 87 percent with relatively high error rates ($M_d = 5\%$, $D_o = 13\%$, $C_{oll} = 11\%$ and $C_{rr} = 46\%$). While the performance in Fairfield 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.

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 $B_1$ 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.

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

When compared to the object-based performance, the pixel-based performance was poor, for the following main reasons. First, 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. Second, due to occlusions on

![Figure 11. Building detection by the improved algorithm on a scene from Fairfield.](image-url)
Figure 12. Building detection by the improved algorithm on a scene from Moonee Ponds.

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

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 3 m). 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.

### Table 4. Pixel-Based Evaluation Results in Percentages ($C_{com}$ = Completeness, $C_{crp}$ = Correctness, $Q_p$ = Quality, $A_{om}$ = Area Omission Error, $A_r$ = Area Commission Error, $B_r$ = Branching Factor, and $M_r$ = Miss Factor)

<table>
<thead>
<tr>
<th>Scenes</th>
<th>$C_{com}$</th>
<th>$C_{crp}$</th>
<th>$Q_p$</th>
<th>$A_{om}$</th>
<th>$A_r$</th>
<th>$B_r$</th>
<th>$M_r$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fairfield</td>
<td>83.2</td>
<td>84.5</td>
<td>72.4</td>
<td>15.3</td>
<td>12.5</td>
<td>13.5</td>
<td>20.3</td>
</tr>
<tr>
<td>M Ponds</td>
<td>87.2</td>
<td>85.4</td>
<td>75.3</td>
<td>12.7</td>
<td>13.2</td>
<td>16.7</td>
<td>17.3</td>
</tr>
<tr>
<td>Knox</td>
<td>74.1</td>
<td>77.3</td>
<td>60.9</td>
<td>26.4</td>
<td>21.0</td>
<td>29.4</td>
<td>34.9</td>
</tr>
<tr>
<td>Hobart</td>
<td>80.8</td>
<td>80.2</td>
<td>67.4</td>
<td>20.3</td>
<td>19.0</td>
<td>25.0</td>
<td>23.7</td>
</tr>
<tr>
<td>Average</td>
<td>81.3</td>
<td>81.9</td>
<td>69.0</td>
<td>18.7</td>
<td>16.4</td>
<td>21.2</td>
<td>24.1</td>
</tr>
</tbody>
</table>

### Table 5. Geometric Accuracy Expressed Using RMSE Values, Along with Running Time in Terms of Texture (Entropy and Edge Orientation) Computation and Building Detection

<table>
<thead>
<tr>
<th>Scenes</th>
<th>Meters</th>
<th>Pixels</th>
<th>Texture</th>
<th>Detection</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fairfield</td>
<td>2.4</td>
<td>16.0</td>
<td>155.8</td>
<td>181.9</td>
<td>337.7</td>
</tr>
<tr>
<td>M Ponds</td>
<td>1.6</td>
<td>16.0</td>
<td>177.7</td>
<td>95.5</td>
<td>273.2</td>
</tr>
<tr>
<td>Knox</td>
<td>2.3</td>
<td>23.0</td>
<td>287.4</td>
<td>384.9</td>
<td>672.3</td>
</tr>
<tr>
<td>Hobart</td>
<td>1.9</td>
<td>19.0</td>
<td>190.4</td>
<td>75.7</td>
<td>266.1</td>
</tr>
<tr>
<td>Average</td>
<td>2.1</td>
<td>16.5</td>
<td>202.8</td>
<td>184.5</td>
<td>387.3</td>
</tr>
</tbody>
</table>

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

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 (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 also. 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.

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. First, 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. Second, the improved algorithm showed its merits in better handling varying topography. Hobart (maximum height \(H_M = 226\) m and minimum height \(H_m = 140\) m) and Knox \(H_M = 270\) m and \(H_m = 110\) m) are hilly areas, whereas Fairfield \(H_M = 23\) m and \(H_m = 1\) m) and Moonee Ponds \(H_M = 43\) m and \(H_m = 23\) m) 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 percent quality with 77 percent completeness and 67 percent correctness. In contrast, as shown in Table 3, the proposed improvement provided 87 percent quality with 94 percent completeness and 92 percent correctness in Knox. In pixel-based evaluation, the original detector gave 27 percent quality with 44 percent completeness and 42 percent correctness. In contrast, Table 4 shows that the improved detector offered 61 percent quality with 74 percent completeness and 74 percent correctness. The geometric accuracy for Knox obtained with the original detector was ten pixels worse than that from the improved detector.

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

Figure 16. Building detection by the improved detector on a second scene from Knox.
higher evaluation the DS detector offered much lower completeness (92.1 percent) than the proposed detector. However, in object-based detection technique which exhibits better separation of 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 30 m² could not be detected 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. Consequently, Test 6 in the Improved Building Detection Section, 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.

Acknowledgments

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.

References


(Received 07 October 2011; accepted 13 December 2011; final version 22 December 2011)