6

3D building reconstruction from airborne lidar point clouds fused with aerial imagery

Jonathan Li and Haiyan Guan

Lidar mapping technology can be used to provide building data that is three-dimensional (3D), accurate, timely, and increasingly affordable in complex urban areas. However, the automated creation of reliable and accurate 3D city models is still challenging. Commercial software tools for building modeling generally require a high degree of human interaction and most automated approaches described in literature stress steps of such a workflow individually. This chapter first briefly reviews the existing techniques for building extraction using airborne lidar range data. Then, we present a comprehensive approach for automated creation of 3D building models from airborne lidar point cloud data fused with color aerial imagery. Our two-step building extraction strategy, building detection followed by building reconstruction, is then detailed and implemented using a set of lidar data covering a Toronto study area collected by Optech ALTM 3100 system. Finally, the results obtained are presented and discussed.

Urban Remote Sensing: Monitoring, Synthesis and Modeling in the Urban Environment, First Edition. Edited by Xiaojun Yang.

6.1 Lidar-drived building models: related work

This section first provides an overview of lidar data processing towards 3D builing reconstruction. The output of a lidar mapping system is a cloud of irregularly spaced 3D points which include not only the bare ground but also all kinds of objects (buildings, trees, cars, *etc.*). Therefore, the generation of reliable and accurate building models from lidar data requires a number of processes, including building detection, outline extraction, roof shape reconstruction, model generation and regularization, and finally, model quality analysis (Dorninger and Pfeifer, 2008). The majority of available literature concentrates on individual aspects only. For example, methods on building region detection in rasterized lidar data were described in the literature (e.g., Hug and Wehr, 1997; Maas, 1999; Morgan and Tempfli, 2000; Nardinocchi and Forlani, 2001; Alharthy and Bethel, 2002; Matikainen, Hyyppä and Hyyppä, 2003; Tóvari and Vögtle, 2004; Gross, Thoennessen and Hansen, 2005; Li, Li and Chapman, 2010). Techniques on roof reconstruction in lidar point clouds with known building boundaries were presented in the literature (e.g., Vosselman and Dijkman, 2001; Hofmann, Maas and Streilein, 2003). Approaches considering detection and reconstruction were presented in the literature (e.g., Rottensteiner and Briese, 2002; Lafarge *et al.*, 2008). The reconstructed models presented in these two references are, however, restricted. In both cases digital surface model (DSM) data of relatively low density is processed. This does not allow for exact positioning of building outlines and prevents the reconstruction of small roof features. Furthermore, in the latter reference the complexity of building models is restricted to a composition of predefined building parts. A general and up-to-date overview on the topic of lidar mapping technology and data processing, in particular, information extraction can be found in Shan and Toth (2008). This chapter does not aim at covering a complete bibliography, but it gives a brief summary of the existing building extraction methods developed thus far mainly in Europe and North America.

To increase the reliability of building reconstruction, additional knowledge on buildings has to be incorporated into the modeling process. Typical assumptions are to define walls as being vertical and roofs as being a composition of planar faces. This leads to an idealization of the buildings. The transition zone of two neighboring roof faces, for example, becomes a straight line defined by the intersection of two roof planes (Dorninger and Pfeifer, 2008).

Many methods have been developed for semiautomated or fully automated extraction of buildings using lidar data in the past 15 years. Recognizing that fully automation is not attainable yet, we aim to reduce the complexity of the building reconstruction task by including a user and automated processes in the system to work sequentially. For example, the user supplies the automated processes with inputs and cues. Then the automated processes produce a scene model based on these inputs. Finally, the user corrects mistakes in that scene model. In this chapter, the building extraction task is addressed by a two-step strategy: building detection followed by building reconstruction (Li, 1999; Hu, 2003).

6.1.1 Building detection

Building detection is often performed on resampled (i.e., interpolated) grid data, thus simplifying the 3D content of lidar data to 2.5D. Roughness measures, i.e. local height variations, are often used to identify vegetation. Open areas and buildings can be differentiated by first computing a digital terrain model (DTM) with so-called filtering methods (Kraus and Pfeifer, 1998; Sithole and Vosselman, 2004). Thereafter, a normalized digital surface model (nDSM) is computed by subtraction of the DTM from the DSM (Weidner and Forstner, 1995; Haala and Brenner, 1999; Gamba and Houshmand, 2002; Rottensteiner and Briese, 2002; Hu, Tao and Collins 2003), hence representing local object heights (Hug and Wehr, 1997; Maas and Vosselman, 1999; Alharthy and Bethel, 2002; Tóvari and Vogtle, 2004; Gross, Thoennessen and Hansen, 2005). High objects with low roughness correspond to building areas. Other approaches identify blobs in the DSM, based on height jumps, high curvature, etc. (Morgan and Tempfli, 2000; Nardinocchi and Forlani, 2001; Matikainen, Hyyppä and Hyyppä, 2003; Rutzinger *et al.*, 2006).

Building reconstruction may include two parts: building footprint detection and roof reconstruction. For those studies focusing on geometric reconstruction of upper roof lines instead of building footprints, the reliable reconstruction of complex building roof boundaries is a key step. Most algorithms work well only under specific assumptions, which limit roofs to simple shapes such as rectangles or low quality polygons (Weidner and Forstner, 1995; Vosselman, 1999; Wang and Schenk, 2000). Other algorithms, which do not make such assumptions, often got distorted boundaries expressed by edges detected from lidar DSMs (Baltsavias, 1995; Weidner, 1995; Yoon *et al.*, 1999; Wang and Schenk, 2000; Rottensteiner and Briese, 2002). These boundaries need to be refined using a set of geometric regularity constraints (Vestri and Devernay, 2001).

To distinguish buildings from vegetated regions, the classification is often based on shape measures assuming some geometric regularity constraints (Wang and Schenk, 2000) or the roughness of the point clouds. These measures limit the detectable buildings to a narrower spectrum, and also are not very reliable for complex scenes such as densely forested areas. The shape measures often make use of 2D properties such as area and perimeter; while complex building roofs may present close values when calculating the roughness measures. The use of the lidar multiple return data can benefit the separation of buildings and vegetation since building roofs must be solid surface (Zhan, Molenaar and Tempfli, 2002; Hu, Tao and Collins, 2003). Lidar cannot penetrate solid surfaces and will get a single return only for them. That is, the first and last returns are same in elevation at solid surfaces but are different at vegetated regions. However, lidar gets the similar effect at building boundaries as that at vegetated areas.

6.1.2 Building reconstruction

Building reconstruction recovers the geometrical parameters of the roofs and walls of a located building (Weidner and Forstner,

1995). Building boundaries, the intersection of the buildings with its surroundings (e.g., ground), need to be derived from the classified building points. Typically, the building boundary generation is initiated by detecting a coarse approximation of the outline, followed by a generalization and a regularization (Sampath and Shan, 2007; Jwa *et al.*, 2008). Two fundamentally different approaches for building reconstruction can be distinguished: model-driven and data-driven approaches.

In the model-driven methods a predefined catalog of roof forms is prescribed (e.g., flat roof, gable roof, etc.). The shape and position parameters are determined by fitting models to lidar point clouds (Weidner and Forstner, 1995; Haala, Brenner and Anders, 1998; Maas and Vosselman, 1999). These methods will lead to a reliable reconstruction if all the constraints in building models are well satisfied and produce pretty results. For instance, an algorithm may assume that there exists a main orientation of the building and all edges are either parallel or perpendicular to that orientation. This is especially appropriate for low point densities. An advantage is that the final roof shape is always topologically correct. A disadvantage is, however, that with low quality in shape (Gamba and Houshmand, 2002). In the data-driven methods the roof is "reassembled" from roof parts found by segmentation algorithms. The results of the segmentation process are sets of points, each one ideally describing exactly one roof face. Some roof elements (e.g., small dormers, chimneys, etc.) may not be represented. The challenge is to identify neighboring segments and the starting and ending point of their intersection.

Building reconstruction is based on the assumption that individual buildings can be modeled properly by a composition of a set of planar surfaces. Hence, it is based on a reliable 3D segmentation algorithm, detecting planar faces in a point cloud. This segmentation is of crucial importance for the outline detection and for the modeling approach (Dorninger and Pfeifer, 2008). Segmentation allows for a decomposition of a building in a lidar point cloud into planar surfaces and other objects. This requires the definition of a homogeneity criterion according to which similar items (e.g., points) are grouped. As homogeneity criteria, approximate height similarity or/and approximate normal vector similarity are commonly used. Determination of planar faces for roof modeling from point clouds acquired from airborne platforms is studied in the literature (Maas and Vosselman, 1999; Lee and Schenk, 2002; Rottensteiner *et al.*, 2005). To reduce the complexity of the problem and to increase the performance of the implementation, again, 2.5D grid representations are commonly used instead of the original points. This requires the definition of a reference direction (e.g., the vertical z-axis) to resample the given points to a regular grid defined as a scalar function over the horizontal xy-plane. Thus, only one distinct height value can be assigned to an arbitrary pair of xy-coordinates. Advantages of 2.5D approaches are a possible reduction of the amount of input data and the implicitly defined neighborhood by means of the grid representation. By contrast, for processing original point clouds, such a neighborhood (e.g., for the estimation of normal vectors) has to be defined explicitly (e.g., Filin and Pfeifer, 2006). Unfortunately, the grid resampling process introduces smoothing effects especially at sharp surface structures. Segmentation approaches based on range images suffer from these restrictions as well (e.g., Hug and Wehr, 1997; Maas and Vosselman, 1999).

6.2 Our building reconstruction method

[illegible] surfaces and roofs with slopes. As such the building extraction based on either single image data or single lidar data cannot reach a satisfying result. To this end, this chapter presents a combination of lidar point cloud data and color aerial image data for 3D building extraction.

6.2.1 Our strategy using fused data

The proposed building extraction strategy is based on the sequential determination of individual building models from both lidar point cloud data and aerial image data. Figure 6.1 shows the workflow of our strategy.

Both an unstructured lidar point cloud with the first and last returns and a color aerial image covering the same area are used as the input. Spatial registration of lidar point cloud data and optical image is performed as data preprocessing.

At the building detection step, filtering is first used to categorize the lidar data into two classes: on-terrain and off-terrain points. The filtered lidar points are then classified into three feature classes, named buildings, trees, and ground (or on-terrain points). The line features extracted from the aerial image, geometric information, such as height information, discrete measurement, and the height difference between the first and last

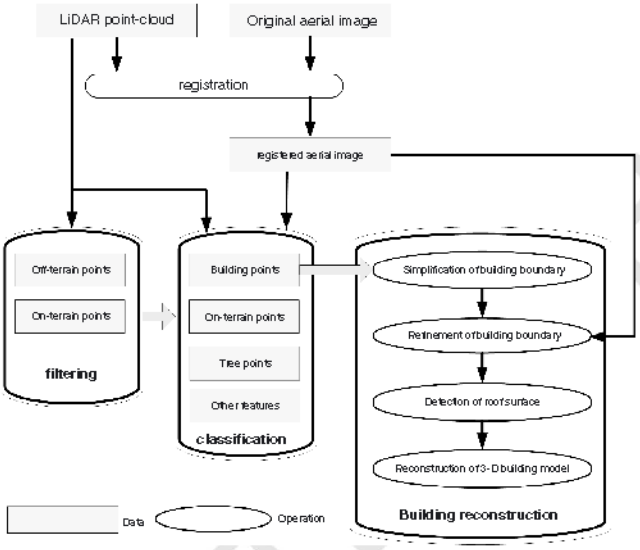


FIGURE 6.1 The workflow of our strategy from building detection to building reconstruction.

returns, provided by lidar point cloud data, are all used as the clues for classification. The homogeneous regions of the aerial image can be attained by the watersheds transform segmentation algorithm and boundary tracking which can be treated as the object of classification. In each homogeneous region, both spectral and geometric information can be comprehensively used for the classification of lidar data in accordance with the experiments to determine the conditions and rules of classification.

At the building reconstruction step, first, the coarse building roof boundary (a polygon) is determined based on the classified building regions. Boundary reconstruction is formulated as a regularization or simplification problem. Because lidar points are randomly collected, the traced boundary cannot be directly used as the final building boundary due to its irregular shape and possible artifacts introduced in the previous steps. Second, the well-known Douglas–Peucker algorithm (Douglas and Peucker, 1973; its improved algorithm by Hershberger and Snoeyink, 1992) is employed for simplification of building region polygons. Third, the least-squares template matching (Ackermann, 1983) with and the right-angle constraint are used for extraction of simple right-angle shaped buildings. Due to disadvantages of lidar data mentioned earlier, some bias exists in the building boundaries. Therefore, the simplified building boundaries are projected onto the registered aerial image for further refinement. Fourth, the random sample consensus (RANSAC) algorithm (Fischler and Bolles, 1981) is used to detect the roof surfaces and then the ridge points of a gable building can be reconstructed through building the neighbor correlation of the roof surfaces, from which the ridge lines of the roof can be attained. The most challenging process is the modeling of the individual roof faces representing the roof. We define a roof surface as the closed, polygonal boundary of a roof segment.

6.2.2 Building detection

Separation of individual buildings from lidar point cloud data is the key to an accurate reconstruction. Like buildings, trees are also one of the dominant features in urban areas. Thus, a new object-oriented supervised classification method is proposed to detect individual buildings and differentiate trees from lidar point clouds at the same time.

6.2.2.1 Region-based segmentation

Segmentation, a process of partitioning an image space into some non-overlapping meaningful homogeneous regions (polygons), is crucial to the classification result. Segmentation of color aerial image contributes to the quality of classification because they can provide more additional information than gray level images. On the other hand, abundant spectrum information of color image may lead to increase of the segmentation difficulty to some extent which can be discriminated based on the height and textural information from lidar data as additional channels. The color aerial image segmentation includes two issues: (1) the choices of color space, and (2) the choice of segmentation methods. The RGB color space is suitable for color expression, but not good for color image segmentation and analysis because of the high correlation among the red, green, and blue bands. The

CIE color space can control color and intensity information more independently and simply than the three RGB primary colors. It is especially efficient in the measurement of small color difference. The CIELAB color space, which is a color-opponent space with dimension L for lightness and a and b for the color-opponent dimensions, based on non-linearly compressed CIE XYZ color space coordinates, is adopted because it is more consistent with human perception.

In general, the segmentation techniques for monochrome images can be extended to segment color images by using R, G and B or their linear or non-linear transformations. Gonzalez and Woods (2008) grouped the existing segmentation methods into three categories: cluster-based approach, edge-based approach, and region-based approach. The cluster-based approach sometimes cannot handle the imagery with high speckle noise due to the assumption that pixels are spatially independent in image space (Li, Li and Chapman, 2010). Moreover, one of the drawbacks of feature space clustering is that the cluster analysis does not utilize any spatial information (Haralick and Shapiro, 1985). Due to image noise, the most three common problems of the edge-based approach are false detections (edge presence in a location where there is no border), missed detections (no edge presence where a real border exists), and selecting an appropriate threshold for the gradient image (this threshold separates significant from non-significant edge information) (Sonka, Hlavac and Boyle, 2002).

The region-based approach, comparing with edge-based segmentation, works generally better on noisy images, where borders are extremely difficult to detect. Watersheds transform is a powerful tool, referring to region-based algorithm, for image segmentation. The concept of watershed and catchment basin are well known in topography. Image data can be interpreted as a topographic surface where the gradient image gray-levels represent altitudes (Sonka, Hlavac and Boyle, 2002). Thus, the expression watershed can be used to denote a labeling of the image, such that all points of a given catchment basin have the same unique label, and a special label, distinct from all the labels of the catchment basins, is assigned to all points of the watersheds (Roerdink and Meijster, 2001). In this chapter, watershed segmentation is used to generate closed contours for each region in the original image since it can effectively divide individual catchment basins in a gradient image.

6.2.2.2 Lidar data preprocessing by filtering

Before the lidar data can be used further, it has to be preprocessed. Filtering is commonly used to separate the on-terrain points and the off-terrain points (Kraus and Pfeifer, 1998). Commonly used filtering algorithms are the morphological filtering, the progressive TIN densification, or the robust filtering. An extensive overview of different filtering approaches can be found in (Sithole and Vosselman, 2004). In this study, the progressive TIN densification that was first proposed by Axelsson (1999) is used through assigning different thresholds to various land cover types. The algorithm works particular well when modeling surfaces with discontinuities, which is common in urban areas.

The filtering method assumes that objects on the ground, such as trees, cars and buildings, etc. are usually higher than those of on-terrain points. In other word, the lowest point in lidar data must belong to on-terrain points. But there are points, generated from both multipath errors and system errors from the laser scanner, do not exist in the landscape. If these points, also called outliers, are misclassified the lowest points, erosion of points in the neighborhood of outliers will be occurred. It is necessary to remove outliers from the dataset before filtering. Because outliers generally are a single point or a small number of points, KD-tree (known as multidimensional binary search tree) structure provide quick query of region neighbors to remove outliers from data, because its performance far surpass the best currently known algorithms for query time and effectiveness (Bentley, 1985). If the number of points within a region is below the given threshold, these points are referred to as outliers.

First, the lowest points in lidar data, as seed points, within a user-defined grid of a size greater than the largest type of features are selected to join an on-terrain dataset. In order to minimize grid size sensitivity to the selection of seed points, a parametric plane is derived. Seed points will be removed if their perpendicular distances to the fitting plane do not satisfy with the median value estimated from the histograms. Then the rest of seed points generate a sparse TIN as initial digital elevation model (DEM). For each iteration, points in each TIN facet are added to the on-terrain data set if they meet the criteria based on the calculated parameters, distances to the TIN facets, and angles to the nodes. At the end of each iteration, the TIN and the thresholds are recomputed. Iterative process continues until no more points meet the threshold values.

6.2.2.3 Extraction of height difference from first and last returns

Difference of the first and last returns usually differentiates tree features from lidar data. As described in Fig. 6.2(a), one of the lidar system's characteristics is the ability of a laser beam to penetrate the tree's canopy through small openings. The number of echoes counts on the object within the travel path of the laser pulse. Many commercial lidar systems can measure multiple returns. Although the laser beam can penetrate the tree's canopy to the ground, it is unreliable to distinguish the tree from the lidar data with only the difference between the first- and the last-echo. This is because, first, shown in Fig. 6.2(a), the laser beam hits on the edge of building also generates two returns. Second, if the density of trees is high, the small-footprint lidar cannot penetrate the tree's canopy.

6.2.2.4 Extraction of spatial features by eigen-analysis

Discrete measurement of lidar data is define as the spatial feature of each point by calculating dispersion matrix of its neighbors. It is another indicator for distinguishing tree points from the other features. Eigen-values of a point's dispersion matrix can reflect this point spatial information.

For each point v_j under consideration, its neighborhood points can be found by the TIN. A 3×3 dispersion matrix S_j of point v_j is given by

$$S_j = \sum_{i=0}^{n} (P_i - M)^T (P_i - M) \quad \forall (j = 0, 1, 2 \ldots N-1) \quad (6.1)$$

where, S_j is the dispersion matrix of point v_j, n is the number of neighborhood points of point v_j, P_i is the coordinate (x_j, y_j, z_j)

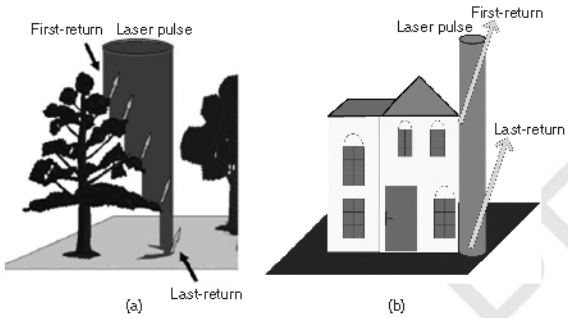


FIGURE 6.2 Height difference of first and last returns.

of point v_j, N is the number of all lidar points, M is the mean matrix of its neighborhood points.

In this 3×3 dispersion matrix, each point v_j has three eigenvalues. An eigenvalue represents the spatial information of a lidar point because it is a scalar of association with the eigenvector that reflects the spatial distribution of a lidar point. Three possibilities are considered. (1) If one of the three eigenvalues is much larger than the others, a lidar point is labelled as an "edge" point. (2) If two of the three eigenvalues are much larger than the other, the lidar point is labelled as a "plane" point. (3) If the three eigenvalues are all larger than a given threshold, the lidar point is labelled as a "discrete" point. Figure 6.3(a) shows the result of dispersion matrices of lidar points in Fig. 6.3(b), in which the white, black, and gray points represent "discrete", "edge," and "plane" points, respectively. As illustrated in Fig. 6.3, trees show the divergence property, while bare ground and most of buildings exhibit the local planarity.

6.2.2.5 Extraction of linear features from aerial imagery

In general, a building consists of regular geometric primitives (lines, corners, etc.). Given the fact that the lines (or edge) of building boundaries appear clearer that those in lidar point clouds, we apply the Canny edge operator and 2D Hough transform (Sonka, Hlavac and Boyle, 2002) to extract the line segments and use them as one of the clues in further classification.

6.2.2.6 Object-oriented supervised classification

Comparing with an unsupervised approach, a supervised approach is preferred by most researchers because it generally gives more accurate class definitions and higher classification accuracy. As a rule, a statistical supervised classification can be

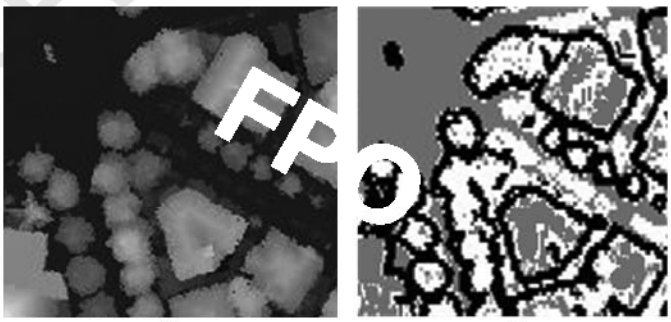


FIGURE 6.3 Discrete measurements by eigen-analysis.

carried out by the following three steps (Tso and Mather, 2001): (1) to define the number and nature of the information classes, and collect sufficient and representative training data for each class; (2) to estimate the required statistical parameters from the training data; and (3) to apply an appropriated decision rule.

Given the class $w_i (i = 1, 2, \cdots n)$, where n is the number of the classes. Features mentioned above yield a D dimension feature space F. So, the probability of a region represented by its feature vector $X (X \in F)$ belongs to class w_i, is defined by Bayes' rule:

$$P(w_i/X) = \frac{P(X/w_i) \bullet P(w_i)}{P(X)} \tag{6.2}$$

where $P(w_i)$ is the prior probability of class w_i, $P(X/w_i)$ is the conditional probability of class w_i has data X, $P(w_i/X)$ is the posterior probability of data X belonging to class w_i, $P(X)$ can be considered as constant value for class w_i. Therefore, Equation 6.2 can be reduced to

$$d_i(X) = P(X/w_i) \bullet P(w_i) \tag{6.3}$$

where, $J = 1, 2,$ and 3, defining three distinctive classes: buildings w_1, bare ground w_2, and trees w_3. The objects that do not belong to these three classes are labelled as unclassified ones w_4. The prior probabilities of buildings $P(w_1)$, bare ground $P(w_2)$, trees, $P(w_3)$ and unclassified objects $P(w_4)$ are obtained according to specific training data set that can represents the typical features of urban areas, and meet

$$P(w_1) + P(w_2) + P(w_3) + P(w_4) = 1 \tag{6.4}$$

Therefore, the next step is to quantify features before determining conditional probability of each class.

1. The spatial information of the lidar point using eigen-analysis $X_1(f)$ is the ratio of the number of points belonging to "scatter" and "edge" points to the amount points in the homogenous region. $1 - X_1(f)$ means the percentage of "plane" points in the region.
2. The filtering result $X_2(f)$ is the ratio of the number of points labelling as on-terrain points to the amount points in the homogenous region. $1 - X_2(f)$ implies the percent of off-terrain points in region.
3. The height difference of first-echo and last-echo $X_3(f)$ is the ratio of the number of points that height differences are over the given threshold, to the amount points in the homogenous region.
4. The linear features of the aerial image $X_4(f)$ are the ration of the length of line segments near to the region to the length of region boundary.

The conditional probability of the class w_i is ascertained by the choices of the weights (m_1, m_2, m_3, m_4) to features $(X_1(f), X_2(f), X_3(f), X_4(f))$ in

$$P(X/w_i) = m_1 \times X_1(f) + m_2 \times X_2(f) + m_3 \times X_3(f) + m_4 \times X_4(f) \tag{6.5}$$

For each region, we can get three class values $\{d_1(X), d_2(X), d_3(X)\}$ according to Equation 6.3. The maximum of three results $d_i(X) = \max\{d_1(X), d_2(X), d_3(X)\}$ labels the region as to which of classes it belongs. If $d_1(X) \approx d_2(X) \approx d_3(X)$, the region is temporary to be labeled as the unclassified class.

6.2.3 Building reconstruction

After having successfully detected the isolated building regions from lidar data, the next step is building reconstruction.

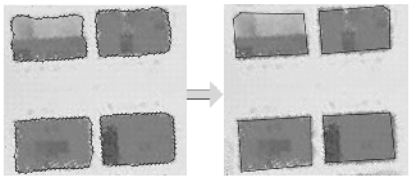
6.2.3.1 Simplification of building boundaries

The majority of buildings in real world are the vertical walls which form the boundaries of buildings. As shown in Fig. 6.4(a), the extracted building boundary at first is recorded as an order sequence of feature points which form a set of ragged small line segments. Thus, the boundaries of buildings are very noisy. The Douglas–Peucker algorithm is first employed to generalize boundary segments so that redundant points can be removed. Figure 6.4(b) illustrates the simplification of building regions by the Douglas–Peuker algorithm.

However, the Douglas–Peuker algorithm can only remove redundant points from those small line segments that are not perpendicular to each other. Therefore, least-squares template matching with right-angle constraint is further applied to accurately determine the building boundaries with a rectangular shape.

6.2.3.2 Least-squares template matching with right-angle constraint

The least-squares matching proposed by Ackermann (1983) uses plenty of information in image to construct an adjustment model, by which the matching precision can reach 0.1 pixels, even 0.01 pixels. Gruen (1985) extended the least-squares matching to the least-squares template matching. Assuming two image regions



(a) (b)

FIGURE 6.4 An example of boundaries simplified by Douglas–Peuker algorithm.

are given as discrete 2D functions $g_1(x,y)$ and $g_2(x,y)$, which might have been derived from analog images. $g_1(x,y)$ and $g_2(x,y)$ can be defined as conjugate regions of a stereo pair in the 'left' and the 'right' image respectively. $g_1(x,y)$ is interpreted in the following as the 'template', $g_2(x,y)$ as the 'picture'. Correlation is (ideally) established if

$$g_1(x,y) = g_2(x,y) \tag{6.6}$$

However, because of the random noise n, Equation 6.6 is not consistent. In other word, there is error v between the template and the picture.

$$v = g_1(x,y) - g_2(x,y) \tag{6.7}$$

Equation 6.7 can be considered as a non-linear observation equation, from which the distances between the gray levels in the template and the picture can be estimated by the least-squares adjustment.

Depending on the edge shape, the edge template can be designed in different forms. Figure 6.5 shows the designed edge template profile and the generated image patch template with the size of 7×5 pixels, respectively.

If the edge template is defined as $g_1(x,y)$ and the aerial image is defined as $g_2(x,y)$, then the image matching from Equation 6.7, can be given by

$$v(x,y) = \frac{\partial g}{\partial x}\mathrm{d}x + \frac{\partial g}{\partial y}\mathrm{d}y - \Delta g$$
$$\Delta g = g_1(x,y) - g_2(x,y)$$
$$\begin{cases} v(x,y) = \dfrac{\partial g}{\partial x}\mathrm{d}x + \dfrac{\partial g}{\partial y}\mathrm{d}y - \Delta g \\ \Delta g = g_1(x,y) - g_2(x,y) \end{cases} \tag{6.8}$$

where $\partial g/\partial x$ is the derivative in the x direction, $\partial g/\partial y$ is the derivative in the y direction. They can be approximated by the first-order difference.

As mentioned above, most of the buildings in the area have rectangular or near rectangular shape. Therefore, we could assume that the line segments as shown in Fig. 6.5 (right), $L_{(i,i+1)}$ and $L_{(i-1,i)}$ are perpendicular each other. Each line segment can be represented by its two end points.

$$L_{(i,i+1)} :: \quad (y - y_i) = \frac{y_{i+1} - y_i}{x_{i+1} - x_i}(x - x_i) \tag{6.9}$$

$$L_{(i-1,i)} :: \quad (y - y_i) = \frac{y_{i-1} - y_i}{x_{i-1} - x_i}(x - x_i) \tag{6.10}$$

where, $L_{(i,i+1)}$ is perpendicular to $L_{(i-1,i)}$, which can be represented as

$$(x_{i+1} - x_i)(x_{i-1} - x_i) + (y_{i+1} - y_i)(y_{i-1} - y_i) - l_i = 0 \tag{6.11}$$

By linearizing Equation 6.11, we can write

$$\begin{aligned} &(x_{i+1} - x_i) \cdot \mathrm{d}x_{i-1} - (x_{i+1} + x_{i-1} - 2x_i) \cdot \mathrm{d}x_i \\ &+ (x_{i-1} - x_i) \cdot \mathrm{d}x_{i+1} + (y_{i+1} - y_i) \cdot \mathrm{d}y_{i-1} \\ &+ (y_{i+1} + y_{i-1} - 2y_i) \cdot \mathrm{d}y_i + (y_{i-1} - y_i) \cdot \mathrm{d}y_i \end{aligned} \tag{6.12}$$

A combination of Equations 6.8 and 6.12 represents the indirect adjustment with geometric right-angle constraint by which the corners coordinate of building will be attained accurately in range image. This iterative matching processing will go on until the correction value of the two endpoints of a line meet the criteria or arrive the given maximal iteration times.

6.2.3.3 Roof surfaces detection

After regularization of the building boundary, the roof points of the lidar data in the boundary are available to reconstruct 3D building models. The automated detection of planes is an essential operation because it can eventually determine the roof shapes. Currently, there are mainly three proposed methods to carry out this task such as region growing, Hough transform and random sample consensus (RANSAC) (Fischler and Bolles, 1981). In a recent work, Tarsha-Kurdi, Landes and Grussenmeyer (2007) compared the Hough transform method with the RANSAC method for the automated detection of building roof planes and concluded that the RANSAC algorithm was more reliable in detecting the roof planes. In this study, we use the RANSAC algorithm to search the best plane from 3D points. The RANSAC algorithm is implemented by the following four steps:

1. To randomly select three points and calculate the parameters of the corresponding plane P, because the smallest number of three points is sufficient to determine the parameters of 3D plane model.
2. To detect every single building point and determine whether or not belonging to the calculated plane P with a predefined distance threshold τ.

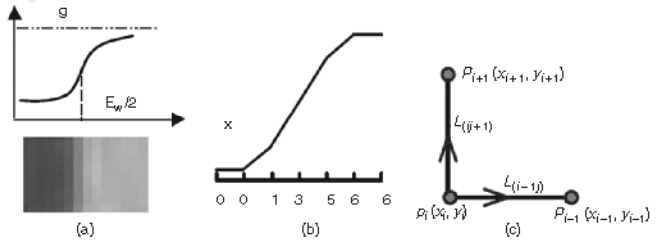


FIGURE 6.5 Least-squares template matching with the right-angle constraint: (a) edge profile template, (b) discrete template, (c) perpendicular condition.

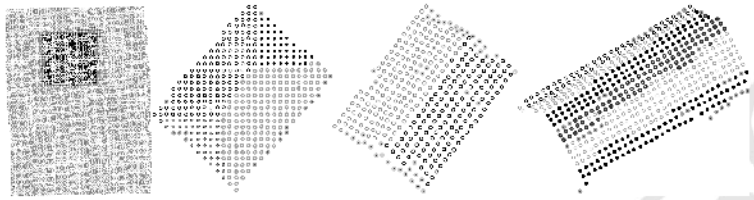


FIGURE 6.6 Results of roof detection by RANSAC.

3 To repeat Steps (1) and (2) M times. In each one, it compares the obtained result with the last one. If the new result is better, then it replaces the saved result with the new one.

4 To exclude the points belonging to the best plane from the building data when the best plane P is found.

In order to detect all roof surfaces within an individual building, the above process has to be repeated several times until the number of points is smaller than the given threshold δ (in this study, $\delta = 20$). Namely, if the number of points within a plane is below 20, these points are treated as noisy points and ready to be rejected. Defining a minimum number of points per surface helps to remove small surfaces that are not of further interest, e.g., chimneys. Beside the distance threshold τ and the minimum number of points δ, the RANSAC algorithm also uses the following three variables to control the plane model estimation process:

- ε, the probability that any selected data agrees with the plane model,
- α, ranging from 0.90 to 0.99, the minimum probability of finding at least one good set of observations in M trials, and
- M, the number of iterations, M, which can be defined as:

$$1 - (1 - (1 - \varepsilon)^m)^M = \alpha \tag{6.13}$$

Taking the logarithm of both sides of Equation 6.13, we have

$$M = \log(1 - \alpha) / \log(1 - (1 - \varepsilon)^m) \tag{6.14}$$

The RANSAC algorithm uses a pure mathematical principle to detect the best planes from a 3D point cloud. That means it looks for the maximum number of points which represent statistically the best planes without considering whether those building points are consecutive in 3D space or not. In other words, it detects a set of points which represents several roof planes or belongs to several planes. Therefore, we use a simple region-growing algorithm to find consecutive roof surfaces. It is necessary to build a TIN structure to search neighborhood for points within each individual building. Randomly starting from a point, a roof surface grows until it cannot find any neighbor point. Points being part of this roof surface are removed from the available points and the algorithm continues until all potential points are used. As shown in Fig. 6.6, several rooftop structures are successfully extracted from building points using the RANSAC algorithm.

6.2.3.4 *Building boundary refinement by aerial imagery*

By comparison with its vertical accuracy, the planimetric accuracy of lidar data is relatively lower due to the errors in the global positioning system (GPS), inertial measurement unit (IMU), mirror angles, and range measurements. However, lidar data cannot provide the linear edges of buildings because it is a set of sample points about the Earth's landscape. The aerial image compensates the weakness of lidar data. Locations of linear features of roofs within building boundaries are not affected by the distribution of points because they are estimated from intersection of two extracted roof planes. Thus, aerial image can be utilized to improve the quality of building boundary refinement.

The procedures of boundary refinement can be listed as follows:

1 To extract edge line segments from aerial image.

2 To project the building boundaries extracted from lidar data into the aerial image; see Fig. 6.7 for an example of a 2D projected boundary $\{B_1, B_2, B_3, B_4\}$.

3 To determine a buffer zone for each line segment of the boundaries- such as buffer of the segment B_1B_2 in Fig. 6.7. The buffer zone of the line segment of the building boundary is formed by expanding to its perpendicular direction.

4 To search for edge line segments extracted in a buffer zone and construct the new boundary by replacing the old line

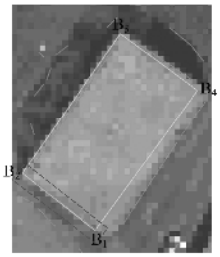


FIGURE 6.7 Refinement of building boundary.

segments of the boundary with the new correspondent ones from aerial image if it exists. If there are several edge line segments in a buffer zone, their slope and length will be analyzed to determine the new ones. If there is not one in the buffer zone, the old one is retained.

5 To refine the new boundary by least-squares template matching with orthogonal constraints.

6.2.3.5 Computation of building models

The adjacency relationship of roof surfaces will be set up to the reconstruction of 3D building model after detecting the roof surfaces by an adjacent matrix graph. Under the adjacent matrix relationship graph, the small adjacent roof surfaces were merged if its normal vectors meet the predefined threshold. Then line segments of boundary are treated as vertical walls to add into adjacent matrix graph, from which the ridge points will be decided by the intersection of adjacent surfaces and the building corners will be calculated. Finally, the 3D building models can be reconstructed.

Assume a set of roof surfaces $P_i(i = 0, 1 \ldots n-1)$ are detected, the adjacency matrix can be represented as shown in Fig. 6.8(a), where, n is the number of the roof surfaces, $R_{ij}(i = 0, 1, \ldots n-1;\ \ j = 0, 1, \ldots n-1)$ is the adjacency relationship between P_i and P_j, whose elements will be labelled to 1 if the roof surfaces P_i and P_j are adjacent to each other. Otherwise, it will be labelled to 0. In Fig. 6.8(b), the diagonal elements of the adjacent relationship matrix are set to 0 because it is symmetric. It can be reconstructed with the help of the TIN data structure. For example, given a point V_m within the surface P_i, find out all of its adjacent points $\{v_0, v_1, v_2, v_3, v_4, v_5\}$. As shown in Fig. 6.9, points v_0 and v_1 are located within surfaces P_k and P_j, respectively, and points v_2 and v_3 are both within surface P_m, so the value of R_{ij}, R_{ik} and R_{im} in the matrix equals to 1, where, i, j and k are the indices of the roof surfaces.

With the aid of adjacent matrix, the small adjacent roof surfaces are merged according to their normal vectors. In general, the threshold of merging surfaces is set to 5°. In other words, if the angle of the normal vectors of the two adjacent surfaces is smaller than this threshold, these two surfaces are merged. For reconstruction of a 3D building model, normally building corners are reconstructed first because they are the primitives of a computer-aided drafting (CAD) model. Thus, the new relationship between roofs and vertical walls is remodeled by adding the vertical walls into the adjacent matrix of the roof surfaces.

For each single line segment of a building boundary, we first find all roof surfaces, then expand along its perpendicular direction to form a buffer zone. According to the building points within the buffer zone belonging to which one of the roof surfaces, the adjacency of the vertical walls and roof surfaces can be detected. Figure 6.10(a) shows an example of a building with

$$R_{n,n}=\begin{bmatrix} R_{0,0} & R_{0,1} & \cdots & R_{0,j} & \cdots & R_{0,n-1} \\ R_{1,0} & R_{1,1} & \cdots & R_{1,j} & \cdots & R_{1,n-1} \\ \vdots & \vdots & & \vdots & & \vdots \\ R_{i,0} & R_{i,1} & \cdots & R_{i,j} & \cdots & R_{i,n-1} \\ \vdots & \vdots & & \vdots & & \vdots \\ R_{n-1,0} & R_{n-1,1} & \cdots & R_{n-1,j} & \cdots & R_{n-1,n-1} \end{bmatrix} \Rightarrow R_{n,n}=\begin{bmatrix} 0 & R_{0,1} & \cdots & R_{0,j} & \cdots & R_{0,n-1} \\ 0 & 0 & \cdots & R_{1,j} & \cdots & R_{1,n-1} \\ \vdots & \vdots & & \vdots & & \vdots \\ 0 & 0 & \cdots & 0 & \cdots & R_{i,n-1} \\ \vdots & \vdots & & \vdots & & \vdots \\ 0 & 0 & \cdots & 0 & \cdots & 0 \end{bmatrix}$$

(a) (b)

FIGURE 6.8 Adjacent matrix graph of roof surfaces.

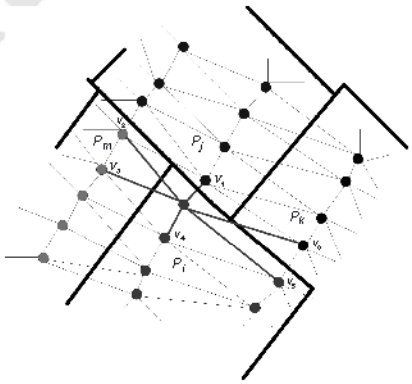


FIGURE 6.9 An example of searching neighborhood.

FIGURE 6.10 A hipped roof model.

a hipped roof. Figure 6.10(b) presents the adjacent matrix of the roof surfaces and vertical walls in Fig. 6.10(a), where $R_0 \sim R_4$ are the roof surfaces, $W_0 \sim W_4$ are the vertical walls. Under the adjacent matrix, the corner-surface matrix can be formed as shown in Fig. 6.10(c), whose purpose is to get the coordinates of the building corners. Typically, the coordinate of a spatial point needs at least three surfaces to solve. Thus, in the corner-surface matrix, each column represents a corner of the 3D building model. Corners located within the boundary can be calculated by the intersection of the roof surfaces. For example, corner A can be calculated by the intersection of the adjacent roofs R_0, R_1, and R_2. Corner B can be calculated by the roofs R_0, R_1, and R_3. In reality, the ridge line is horizontal, so the height value of the corner A is equal to that of the corner B. Similarly, corners $B_1 \sim B_4$ can also be calculated by the intersection of roofs and vertical walls. For instance, four adjacent surfaces (two roof surfaces R_1 and R_2, two vertical walls W_0 and W_1) are used to get the corner B_1. More attention needs to be paid to the heights of the corners $B_1 \sim B_4$. They should be the same value because they generally are horizontal in the real world. For a building with a hipped roof (see Fig. 6.10a), the coordinates of six corners with 14 unknown parameters are retrieved by the least-squares adjustment.

6.3 Results and discussion

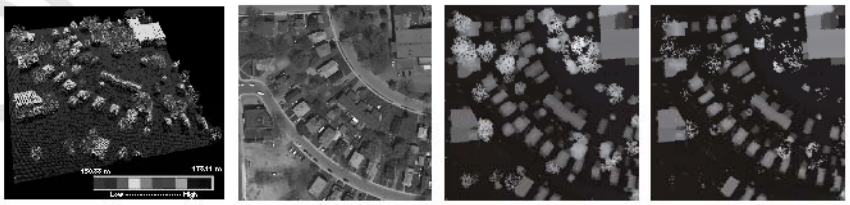
This section first describes the lidar point cloud data and color aerial image data used in this chapter and then presents the results of building extraction.

6.3.1 Datasets

The lidar datasets used in this study were collected over 1000 m above ground level by an Optech ALTM 3100 system. The system also carried a 4k × 4k digital camera. The datasets include the first and the last returns of the laser beam and the true color aerial imagery. Figure 6.11 shows a residential scene, which is a subset of a typical urban area in the City of Toronto, Ontario, Canada. As shown in Fig. 6.11(a) presents the raster-based DSM, containing a total of 105,298 points, which was interpolated with both the first and last pulse returns by the bilinear interpolation method. The width and height of grid equals to the ground sample distance (GSD) of the aerial image (0.5 m). The elevation of the study area ranges from 150.00 m to 178.11m. Besides buildings, there are several clusters of trees. Figure 6.11(b) shows the true color aerial image that was resampled to 0.5 m ground pixel. The majority of buildings appeared in the color image are with gable roofs or hipped roofs. Figure 6.11(c) and 6.11(b) illustrate the lidar range images of the first and last returns, respectively.

6.3.2 Results

Figure 6.12(a) shows the filtered results overlaid on the color aerial image. The results demonstrate that all the on-terrain points have been removed quite well. The parameter of the iterative distance in the filtering algorithm which is usually below



(a) (b) (c) (d)

FIGURE 6.11 Lidar range data and colour aerial image of the study area: (a) DSM, (b) true colour aerial image, (c) first return range image, and (d) last return range image.

the height of cars, and it gradually decreases with the increase of iterative times.

The classification accuracy relies heavily on the quality of segmentation. The abundant spectral information from color aerial image is beneficial to the classification. On the other hand, it increases the difficulty in segmentation due to spectral confusion between-class and spectral variation within-class. For example, the roof material of a building along a road sometimes is similar to that of the road. Figure 6.12(b) shows the materials of the building and ground are similar, so they will always be partitioned into a homogenous region in Fig. 6.12(c). However, the quality of segmentation is not guaranteed when the range image (see Fig. 6.12d) is used alone. For instance, if a tree is close to a building, it is very difficult to separate one from another, as shown in Fig. 6.12(e). Therefore, height information from lidar data as an additional channel can improve the quality of color image segmentation. Figure 6.13(a) shows the result of segmentation by fusing lidar data and the aerial image. Figure 6.13(b) presents the spatial discrete measurement result of lidar data by eigen-analysis, in which most of tree regions are described as "discrete" and "edges," while majority of building regions are presented as "planes."

Some parameters for the supervised object-oriented classification need to be predetermined. First, the parameters of the prior probability can be derived from the typical training set. The most important parameters in this study are the weights of features. The

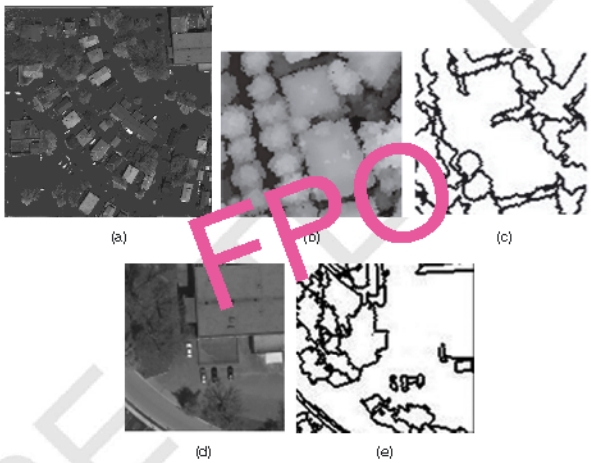


FIGURE 6.12 (a) Filtered result overlaid on the aerial image, (b) aerial image and (c) its corresponding segmentation results, (d) lidar range image and (e) its corresponding segmentation results.

FIGURE 6.13 (a) The segmentation result of the aerial image fused with the height and texture information from lidar data; (b) The spatial distribution result of lidar data by eigen-analysis.

way to choose weights is self-adaptive to features of each homogeneous region. Because features $(X_1(f), X_2(f), X_3(f), X_4(f))$ have already normalized, the weights used for each feature is its value

$$m_i = X_i(f)/(X_1(f) + X_2(f) + X_3(f) + X_4(f)) \qquad i = 1,2,3,4 \tag{6.15}$$

This method matches the observation that a larger percentage of a feature is more important than a smaller percentage of the feature. For each homogeneous region, the weights vary with different features in the region. For example, if the values of four features are $\{0.8, 0.8, 0.2, 0.1\}$ in a region, the correspondent weights are $\{0.42, 0.42, 0.1, 0.06\}$, respectively. Figure 6.14(a) shows the result of the object-oriented classification. By visual inspection of the aerial image, a few buildings were not separated from the adjoining trees, due to large crowns of those trees covering the major part of the buildings. Some small areas, which most likely belong to trees, were misclassified as buildings. These areas are so small that some of them can be eliminated by an area threshold. Then, the building blobs can be separated (see Fig. 14b and c).

In order to reconstruct the 3D building models, the building boundaries should be extracted as its vertical walls from the lidar range image. In the proposed algorithm, the boundary of a building is assumed to be a rectangular-shaped object. That is, the line segments of the boundary are either parallel or perpendicular from one to another. We first use the Douglas–Peuker algorithm to simplify these ragged building boundaries (see Fig. 6.14d). Then the simplified irregular boundaries can be regularized to rectangular-shaped ones using the least-square template matching with orthogonal constraints. Figure 6.15 shows the extracted building boundaries by simplification using the Douglas–Peuker algorithm (labelled as B in a circle) followed by regularization using the least-squares template matching with geometric right-angle constraints.

Figures 6.16 and 6.17 show the building reconstruction process, in which "A" represents roof boundaries extracted from lidar data, "B" represents roof boundaries and ridge lines only using the least-squares template matching, "C" represents roof boundaries and ridge lines by the least-square template matching with orthogonal constraints, and "D" represents line segments extracted from aerial image. Figure 6.16 shows one building with a gable roof (having two planar surfaces), in which (a) shows the points of the planer surfaces detected from the classified building points using the RANSAC algorithm, (b) shows the roof boundary (the "A" labels) projected onto the aerial image from the lidar-derived polygon, (c) shows the initial ridge line (the "B" labels) projected onto the aerial image detected from the intersection edges of neighboring surfaces together with final ridge line (the "C" labels) rectified using image information, and (d) shows the reconstructed 3D model of a single building with a gable roof. Figure 6.17 shows the reconstruction process of a building with a rooftop having four planar surfaces.

Figure 6.18(a) shows the building boundaries extracted from range image, which are then projected onto the color aerial image to improve the planimeric accuracy. The boundaries before and after refinement are presented by light-gray and darker lines, respectively. As shown in Fig. 6.18(a), the three individual detached houses are misclassified as a single house in the previous processes. Given the fact that these individual

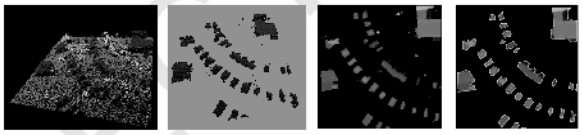


FIGURE 6.14 (a) The result of object-oriented classification; (b) The classified buildings' lidar points; (c) The classified buildings' range image. (d) The extracted building roof boundaries.

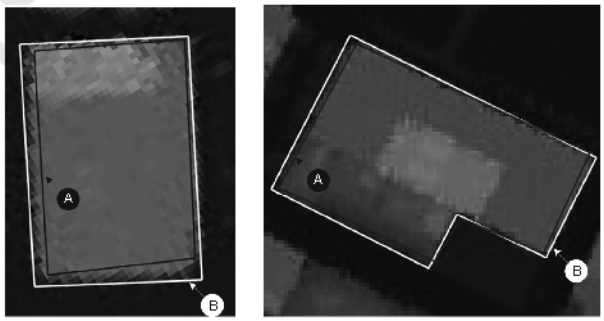
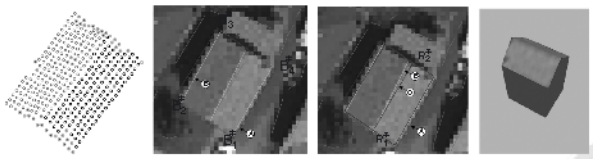
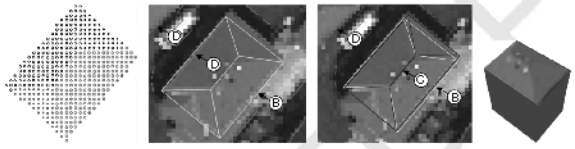


FIGURE 6.15 Extracted roof outlines. The "A" labels represent the result of Douglas–Peuker simplification. The "B" labels show the result of regularization using least-squares template matching with geometric constraint.



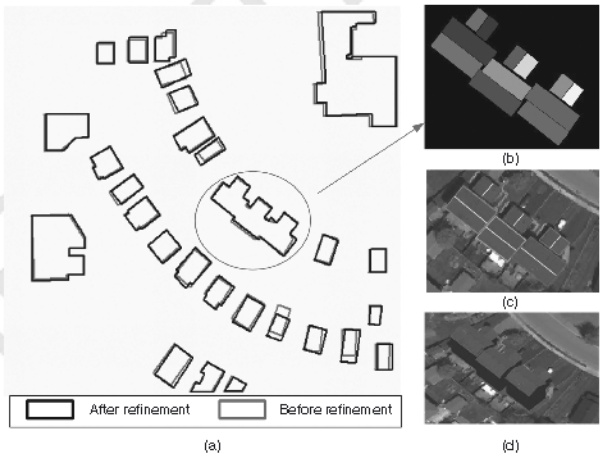
(a) (b) (c) (d)

FIGURE 6.16 The reconstructed building with a gable roof.



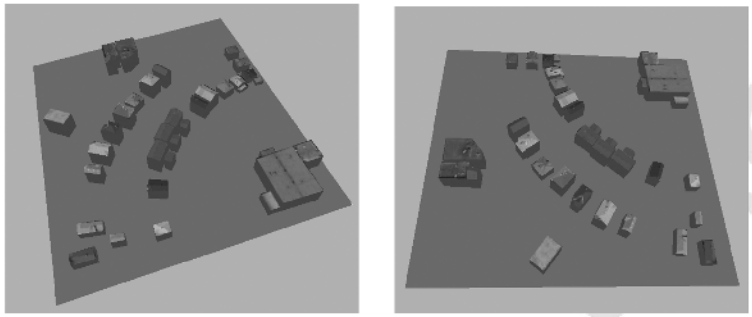
(a) (b) (c) (d)

FIGURE 6.17 The reconstructed building with a complex roof having four planar surfaces.



(a) (b) (c) (d)

FIGURE 6.18 (a) Refined building roof boundaries, (b) Detected planer surfaces of roofs, (c) accurately located rood boundaries, (d) 3D reconstructed buildings overlaid on the aerial image.



(a) (b)

FIGURE 6.19 Perspective views of the 3D building models with textured roofs reviewed from two different viewpoints.

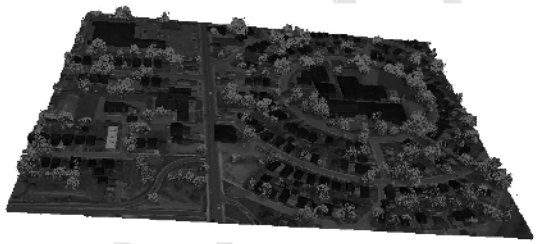


FIGURE 6.20 The created 3D city model with textured building models, trees and terrain.

buildings are different in height, lidar point clouds are used to separate them with a total of six planar surfaces detected using the RANSAC algorithm (see Fig. 6.18b). Then the outlines of these surfaces are generated, simplified and regularized. Finally, the individual building with a gable roof can be determined by the roof ridge line from the intersection of the two planar surfaces (see Fig. 6.18c). Figure 6.18(d) shows the three reconstructed building models overlaid on the color aerial image.

The color aerial image (orthoimagery) was also used as texture and draped onto both building rooftops and terrain surface. From a data-structure point view, orthoimagery is expressed as 2D raster data, which can be stored or manipulated as a special layer in a 3D geospatial system. This work focuses on a 3D scene with only three object classes: buildings, trees and terrain. The textured building roofs are required for creation of a photorealistic 3D city model towards fly- or walk-throughs and simulation (particularly for the planning and visualization of building models). Figure 6.19 shows the perspective reviews of the study area from two different viewpoints. Figure 6.20 shows the resulting 3D city model with buildings, trees and textured terrain. It presents useful opportunities for site visualization, which enhances community participation in decision-making.

Concluding remarks

In this chapter we have presented a comprehensive approach for the determination of 3D building models from lidar point cloud data fused with color aerial imagery. A two-step extraction strategy, building detection followed by building reconstruction, has been developed and implemented. Building detection is first done by filtering to separate on-terrain and off-terrain points, and further, to integrate information from color aerial imagery into an object-oriented classification process to catalog three classes (buildings, trees, and terrain).

To ensure completeness, it is advisable to initialize the very first step, namely the coarse selection of building regions, interactively. All subsequent steps (i.e., outline extraction and regularization, planar surface detection, and building model generation) are applied automatically. Possibly erroneous buildings can be improved by interactive post-processing (i.e., manual editing).

Compared to lidar point cloud data, the advantage of color aerial imagery is its higher accuracy in sharp linear features, which can be utilized to refine building boundaries. The rectangular-shaped buildings can be delineated well by using the least-squares template matching with orthogonal constraints.

Since our method is a data-driven approach, we attempt the reconstructed building models to approximate the given point clouds the best. Compared to the model-driven methods, the data-driven approach is more flexible. While the building model reconstructed using the model-driven method is restricted by a predefined catalog of buildings or building parts, a data-driven approach requires more flexible assumptions. A typical assumption in this study is that buildings can be modeled by a composition of planar surfaces. However, this might be a restriction, considering curved roof structures.

In our case study, due to the complexity and diversity of natural and man-made objects in urban areas, manual input is required in the classification stage, which reduces the automation level. Another disadvantage is that those buildings partially covered by trees cannot be reconstructed completely and correctly due to the failure of the roof detection, even these building areas had been classified correctly. Finally, this study assumes a building as a rectangular-shaped object only. Further investigations on methods for dealing with non-rectangular-shaped buildings are needed. Nevertheless, the results met the requirement of being accurate and reliable. Subsequent processing like texture mapping is possible. The resulting models are well structured and topologically correct and are, therefore, directly applicable for 3D urban visualization.

In addition, for those applications requiring realistic 3D city models, reconstructing photorealistic building facade models should be applied. This can be done from terrestrial laser point clouds and close-range images. Considering quick creation of such building facade models in a large urban area, a vehicle-borne mobile lidar mapping system would be an effective means for fast collection of terrestrial lidar point clouds along with color digital camera images of the roadside buildings. Moreover, accurate fusion of the point clouds acquired by both airborne and terrestrial mobile lidar mapping systems would open a new avenue to reconstruction of realistic 3D city models.

Acknowledgments

The work presented in this chapter was supported by a NSERC discovery grant and a Wiser Foundation research grant. The datasets used in this chapter were provided by Optech, Inc., Toronto, Canada. The authors would like to thank the anonymous reviewers for their valuable comments.

References

Ackermann, F. (1983) High precision digital image correlation. In *Proceedings of 39th Photogrammetric Week*, University of Stuttgart, pp. 231–243.

Alharthy, A. and Bethel, J. (2002) Heuristic filtering and 3D feature extraction from lidar data. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, **34**(3A+B), 6.

Axelsson, P. (1999) DEM generation from laser scanner data using adaptive TIN models. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, **33**(B4/1), 110–117.

Baltsavias, E.P. (1999) Airborne laser scanning – relations and formulas. *ISPRS Journal of Photogrammetry and Remote Sensing*, **54**, 199–214.

Bentley J.L. (1985) Multidimensional binary search trees used for associative searching. *Communications of the Association for Computer Machinery (ACM)*, **18**, 509–517.

Dorninger, P. and Pfeifer, N. (2008) A Comprehensive automated 3D approach for building extraction, reconstruction, and regularization from airborne laser scanning point clouds. *Sensors*, **8**(11), 7323–7343.

Douglas, D. and Peucker, T. (1973) Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *The Canadian Cartographer*, **10**(2), 112–122.

Elaksher, A.F. and Bethel, J.S. (2002) Building extraction using lidar data. *ACSM-ASPRS Annual Conference*, 19–26 April, Washington DC, 9 pp.

Filin, S. and Pfeifer, N. (2006) Segmentation of airborne laser scanning data using a slope adaptive neighborhood. *ISPRS Journal of Photogrammetry and Remote Sensing*, **60**(2), 71–80.

Fischler, M.A. and Bolles, R.C. (1981) Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, **24**, 381–395.

Gamba, P. and Houshmand, B. (2002) Joint analysis of SAR, lidar and aerial imagery for simultaneous extraction of land cover, DTM and 3D shape of buildings. *International Journal of Remote Sensing*, **23**(20), 4439–4450.

Gonzalez, R.C. and Woods, R.E. (2008) *Digital Image Processing*, 3rd edition, Prentice Hall, Upper Saddle River, NJ.

Gross, H., Thoennessen, U. and Hansen, W. (2005) 3D modeling of urban structures. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, **36**(3-W24).

Gruen A. (1985) Adaptive least squares correlation: a powerful image matching technique. *South African Journal of Photogrammetry and Remote Sensing*, **14** (3), 175–187.

Haala, N., Brenner, C. and Anders, K.-H. (1998) 3D urban GIS from laser altimeter and 2D map data. In *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, **32**(3/ W1), 321–330.

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

Haralick, R.M. and Shapiro, L.G. (1985) Image segmentation techniques. *Computer Vision Graphics Image Process*, **29**, 100–132.

Hershberger, J. and Snoeyink, J. (1992) Speeding up the Douglas–Peucker line-simplification algorithm. *Proceedings of the 5th Symposium on Data Handling*, 134–143.

Hofmann, A.D., Maas, H.-G., and Streilein, A. (2003) Derivation of roof types by cluster analysis in parameter spaces of airborne laser scanner point clouds, in *Proceedings of the ISPRS WG III/3 Workshop on 3-D Reconstruction from Airborne Laserscanner and InSAR Data*, Dresden, October 8–10.

◆ Q4

Hu, Y. (2003) Automated Extraction of Digital Terrain Models, Roads and Building Using Airborne Lidar Data, PhD Thesis, University of Calgary.

Hu, Y., Tao, V. and Collins, M. (2003). Automatic extraction of buildings and generation of 3D city models from airborne lidar data. *ASPRS Annual Conference*, 3–9 May, Anchorage, AK, 12 p.

Hug, C. and Wehr, A. (1997) Detecting and identifying topographic objects in imaging laser altimeter data. *International Archives of Photogrammetry, Remote Sensing, and Spatial Information Sciences*, **32**, 19–26.

Jwa, Y., Sohn, G., Tao, V. *et al.* (2008) An implicit geometric regularization of 3D building shape using airborne lidar data. *International Archives of Photogrammetry, Remote Sensing, and Spatial Information Sciences*, **36**(5), 69–76.

Kilian, J., Haala, N. and Eglich, M. (1996) Capture and evaluation of airborne laser scanner data. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, **31**(B3).

Kraus, K. and Pfeifer, N. (1998) Determination of terrain models in wooded areas with airborne laser scanner data. ISPRS Journal of Photogrammetry and Remote Sensing, 53, 193–203.

Lafarge, F., Descombes, X., Zerubia, J. *et al.* (2008) Automatic building extraction from DEMs using an object approach and application to the 3D-city modeling. *ISPRS Journal of Photogrammetry and Remote Sensing*, 63, 365–381.

Lee, I. and Schenk, T. (2002) Perceptual organization of 3D surface points. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, **34**(3), 6.

Li, J. (1999) *Informal Settlement Modeling using Digital Small-format Aerial Imagery*, PhD Thesis, University of Cape Town.

Li, Y., Li, J. and Chapman, M.A. (2010) Segmentation of SAR intensity imagery with a Voronoi tessellation, Bayesian inference and reversible jump Markov chain Monte Carlo algorithm. *IEEE Transactions on Geoscience and Remote Sensing*, **48**(4), 1872–1881.

Maas, H.-G. (1999) Potential of height texture measurement for the segmentation of airborne laser scanner data. *ISPRS Journal of Photogrammetry & Remote Sensing*, **54**(2/3), 245–261.

Maas, H.-G. and Vosselman, G. (1999) Two algorithms for extracting building models from raw laser altimetry data. *ISPRS Journal of Photogrammetry and Remote Sensing*, **54**(2/3), 153–163.

Matikainen, L., Hyyppä, J. and Hyyppä, H. (2003) Automatic detection of buildings from laser scanner data for map updating. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, **34**(3/W13).

Morgan, M. and Tempfli, K. (2000) **Automatic building extraction from airborne laser scanning data**. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, **33**(B4): 616–622.

Nardinocchi, C. and Forlani, G. (2001) Building detection and roof extraction in laser scanning data. In *Proceedings of 3rd International Workshop on Automatic Extraction of Man-Made Objects from Aerial and Space Images*, Ascona, Switzerland.

Roerdink, J.B.T.M. and Meijster, A. (2001) The watershed transform: definitions, algorithms and parallelization strategies. *Fundamenta Informaticae*, **41**(1–2), 187–228.

Rottensteiner, D.F. and Briese, C. (2002) A new method for building extraction in urban areas from high-resolution lidar data. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, **34** (3A/B), 295–301.

Rottensteiner, F. (2003) Automatic generation of high-quality building models from lidar data. *IEEE Computer Graphics and Applications*, **23**(6), 42–51.

Rottensteiner, F., Trinder, J., Clode, S. *et al.* (2005) Automated delineation of roof planes from lidar data. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 36(3/W19).

Rutzinger, M.B. Höfle, B., Pfeifer, N. *et al.* (2006) Object based analysis of airborne laser scanning data for natural hazard purposes using open source components. In *the First International Conference on Object-based Image Analysis*, Salzburg, Austria.

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

Sampath, A. and Shan, J. (2008) Building reconstruction from airborne lidar data based on clustering analysis. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, **37**(B3a), 279–284.

Shan, J. and Sampath, A. (2006) Urban terrain and building extraction from airborne lidar data. In *Urban Remote Sensing*, (eds Q. Weng and D. Quattrochi), CRC/Taylor & Francis, Boca Raton, pp. 21–45.

Shan, J. and Sampath, A. (2005) Urban DEM generation from raw lidar data: a labeling algorithm and its performance. *Photogrammetric Engineering and Remote Sensing*, **71**(2), 217–226.

Shan, J. and Toth, C. (2008) *Topographic Laser Ranging and Scanning: Principles and Processing*. CRC Press, London, UK.

Sithole, G. and Vosselman, G. (2004) Experimental comparison of filtering algorithms for bare-earth extraction from airborne laser scanning point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing*, **59**(1–2), 85–101.

Sonka, M., Hlavac, V. and Boyle, R. (2008) *Image Processing, Analysis, and Machine Vision*, 3rd edition, Brooks/Cole Publishing Company, Pacific Grove, California, pp. 800.

Tarsha-Kurdi, F., Landes, T. and Grussenmeyer, P. (2007) Hough-transform and extended RANSAC algorithms for automatic detection of 3D building roof planes from lidar data. *International Archives of Photogrammetry and Remote Sensing*, **36**(3/W52), 407–412.

Tóvari, D. and Vögtle, T. (2004) Object classification in laser-scanning data. In *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, **36**(8/W2), 45–49.

Tso, B. and Mather, P.M. (2001) *Classification Methods for Remotely Sensed Data*. Taylor & Francis, London, p. 75.

Vestri, C. and Devernay, F. (2001) Using robust methods for automatic extraction of buildings, in *2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'01)*, Vol. 1, pp. 133.

Vosselman, G. (1999) Building reconstruction using planar faces in very high-density data. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, **32**(3–2W5).

Vosselman, G. and Dijkman, S. (2001) 3D building model reconstruction from point clouds and ground plans, In *International Archives of Photogrammetry and Remote Sensing*, **34**(3/ W4), 37–44.

Wang, J. and Shan, J. (2009) Segmentation of lidar point clouds for building extraction, in *Proceedings of 2009 ASPRS Annual Conference*, Baltimore, Maryland, March 9–13.

Wang, Z. and Schenk, T. (2000) Building extraction and reconstruction from lidar data. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, **33**(B3), 958–964.

Weidner, U. and Forstner, W. (1995) Towards automatic building extraction from high resolution digital elevation models. *ISPRS Journal of Photogrammetry and Remote Sensing*, **50**(4), 38–49.

Yoon, T. *et al.* (1999) Building segmentation using an active contour model. In *ISPRS Conference on Sensors and Mapping from Space*, Hannover, 27–30 September.

Zhan, Q., Molenaar, M. and Tempfli, K. (2002) Building extraction from laser data by reasoning on image segments in elevation slices. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, **34**(3A+B), 305–308.

Zhang, K. and Whitman, D. (2005) Comparison of three algorithms for filtering airborne lidar data. *Photogrammetric Engineering and Remote Sensing*, **71**(3), 313–324.

Zhou, G. (2008) Urban 3D building model from lidar data and digital aerial images. In *Remote Sensing of Impervious Surfaces* (ed. Q. Weng), Taylor & Francis, London, pp. 251–267.