

Automatic Construction of 3-D Building Model From Airborne LIDAR Data Through 2-D Snake Algorithm

Jianhua Yan, Keqi Zhang, Chengcui Zhang, *Member, IEEE*,
Shu-Ching Chen, *Senior Member, IEEE*, and Giri Narasimhan

Abstract—The snake algorithm has been proposed to solve many remote sensing and computer vision problems such as object segmentation, surface reconstruction, and object tracking. This paper introduces a framework for 3-D building model construction from LIDAR data based on the snake algorithm. It consists of nonterrain object identification, building and tree separation, building topology extraction, and adjustment by the snake algorithm. The challenging task in applying the snake algorithm to building topology adjustment is to find the global minima of energy functions derived for 2-D building topology. The traditional snake algorithm uses dynamic programming for computing the global minima of energy functions which is limited to snake problems with 1-D topology (i.e., a contour) and cannot handle problems with 2-D topology. In this paper, we have extended the dynamic programming method to address the snake problems with a 2-D planar topology using a novel graph reduction technique. Given a planar snake, a set of reduction operations is defined and used to simplify the graph of the planar snake into a set of isolated vertices while retaining the minimal energy of the graph. Another challenging task for 3-D building model reconstruction is how to enforce different kinds of geometric constraints during building topology refinement. This framework proposed two energy functions, deviation and direction energy functions, to enforce multiple geometric constraints on 2-D topology refinement naturally and efficiently. To examine the effectiveness of the framework, the framework has been applied on different data sets to construct 3-D building models from airborne LIDAR data. The results demonstrate that the proposed snake algorithm successfully found the global optima in polynomial time for all of the building topologies and generated satisfactory 3-D models for most of the buildings in the study areas.

Index Terms—Light detection and ranging (LIDAR), snake algorithm, topology.

Manuscript received June 12, 2012; revised November 28, 2012, May 2, 2013, and October 10, 2013; accepted January 14, 2014. This work was supported by the Florida Hurricane Alliance Research Program sponsored by the National Oceanic and Atmospheric Administration.

J. Yan is with Amazon.com, Inc., Seattle, WA 98144-2734 USA (e-mail: yjhsjtu@google.com).

K. Zhang is with the Department of Environmental Studies and International Hurricane Research Center, Florida International University, Miami, FL 33199 USA (e-mail: zhangk@fiu.edu).

C. Zhang is with the Department of Computer and Information Sciences, The University of Alabama at Birmingham, Birmingham, AL 35294 USA (e-mail: zhang@cis.uab.edu).

S.-C. Chen is with the Distributed Multimedia Information System Laboratory, School of Computing and Information Sciences, Florida International University, Miami, FL 33199 USA (e-mail: chens@cs.fiu.edu).

G. Narasimhan is with the School of Computing and Information Sciences, Florida International University, Miami, FL 33199 USA (e-mail: giri@cs.fiu.edu).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TGRS.2014.2312393

I. INTRODUCTION

THREE-dimensional building models are of fundamental importance to many applications such as constructing virtual city models, creating urban landscape models, and assessing urban heat island effects [1]. High-resolution data for extracting 3-D building models often come from LIDAR measurements, aerial photographs, and high-resolution satellite images [2]. Airborne LIDAR systems generate voluminous and irregularly spaced 3-D point measurements of buildings scanned by the laser beneath the aircraft. Airborne LIDAR measurements are particularly useful for reconstruction of building models because the LIDAR technology provides direct measurements of horizontal coordinates (x and y) and elevations (z) of a building and avoids the shadow and relief distortions seen in aerial photographs and satellite images.

Many algorithms have been developed to automatically extract 3-D building models from the LIDAR measurements [3] in the past decade. Schwalbe [4] categorized these methods as model-driven and data-driven ones. Model-driven methods fit some primitive building models into the LIDAR measurements. In the model-driven method, building models are identified by fitting predefined models into the LIDAR measurements. For example, Maas and Vosselman [5] estimated parameters for primitive building models based on the invariant moment analysis. Brenner [6] extended this method to complex buildings by splitting a building into simple primitives first and then fitting individual primitives using point clouds. You and Lin [7] constructed building models and analyzed the quality of models by integrating LIDAR data and topographic maps. However, data sources like building models or topographic maps for a study area are not always available in advance, which limits the application of the model-driven method.

Some of data-driven methods construct building models by integrating LIDAR data and other data sources. For example, Awrangzeb *et al.* [8] generated building models based on masks obtained from LIDAR and multispectral imagery. Xiao *et al.* [9] applied image matching techniques to reconstruct building models from LIDAR data and oblique airborne imagery. The application of these methods is limited since data sources other than LIDAR are not always available. Other data-driven methods grouped building measurements first for different roof planes. Then, the raw 2-D topology of each building separating neighboring facets is derived. Raw 3-D building models can be directly created from the raw 2-D topologies and identified

roof planes. For example, Sampath and Shan [10] separated roof facets using surface normal of each roof point. Kim and Shan [11] selected level sets for roof facet segmentation. Both of these two methods construct building models based on segmented roof facets directly. However, the quality of such building models may be poor because a 2-D topology is often noisy due to irregularly spaced LIDAR measurements. A refinement of 2-D topology and roof plane parameters is often needed before high-quality building models can be derived. Many geometric constraints have been proposed to regularize and refine the 2-D topology. Zhang *et al.* [2] proposed some operations to adjust and enforce parallelism on building footprints. However, this method cannot be extended to 2-D topology. Gruen and Wang [12] enforced seven constraints by using least-square adjustments with manual grouping involved, which limits the application of this method. Other methods realized the difficulty in enforcing multiple constraints simultaneously and only enforced important constraints. For example, Vosselman [13] enforced parallelism on building topology adjustment. However, their assumption is too restrictive and cannot be applied to buildings with edges oblique to both dominant directions.

To overcome limitations mentioned in the aforementioned methods, this paper introduces a new data-driven model for 3-D building model construction from LIDAR data through the snake algorithm that can enforce multiple constraints simultaneously and easily. The snake algorithm was first introduced by Kass *et al.* in 1988 [14], and it usually starts with an initial 1-D contour, 2-D surface, or even-dimensional (3-D) volume [15]–[18] close to a target model and then gradually deforms the contour or surface with the goal of minimizing energy functions so that the resulting contour or surface best matches the target boundary or topology of the object [16], [19]. Significant research efforts have been placed on applying the snake algorithm to determine a surface or a contour having optimal properties in the past decade [20]. Applications of the snake algorithm include remote sensing imagery analysis, geometric modeling, and tracking of nonrigid objects. For example, the snake algorithm is one of the most frequently used methods to extract building models and footprints from imagery and LIDAR [21]–[23] in the remote sensing area. The critical steps for applying snake algorithm include defining an energy function fit for the given task and minimization of the energy function.

A traditional snake algorithm utilizes dynamic programming to find the global minima by analyzing a collection of admissible solutions. In dynamic programming, constraints are often placed on the set of allowable solutions, thus reducing the computational complexity. For example, in case of 1-D active contours, the set of admissible solutions are, in fact, the set of all allowed curves that connect the start point with the end point. Unlike the variational method, dynamic programming can be directly applied to a discrete grid without approximations. Amini *et al.* [24] devised a time-delayed discrete dynamic programming algorithm to minimize the energy for 1-D active contours with only the first-order energy terms involved, where each energy term depends on at most two vertices of the contour. The discretization of the contour energy $E(C)$ is represented by $E(C) = E_1(v_1, v_2) + E_2(v_2, v_3) + \dots + E_{n-1}(v_{n-1}, v_n)$, where $C = \{v_1, \dots, v_n\}$ is the set of vertices forming a

1-D contour. Each contour point v_i is allowed to only take on d possible values. Instead of using exhaustive enumeration to find the minimum of $E(C)$, the discrete dynamic programming method computes the global minimum in a much more efficient way by breaking the global optimization problem down into simpler subproblems. However, the existing dynamic programming method [20], [24], [25] is inherently restricted to problems with 1-D topology such as a contour. The dynamic programming method outlined for 1-D topology cannot be extended to 2-D cases [26]. The 2-D planar snake problems, such as the construction of building models, cannot be solved efficiently through existing dynamic programming methods.

This paper proposes a novel snake algorithm, named “2-D snake algorithm,” to minimize the energy function associated with 2-D planar topology through the graph reduction technique. The graph reduction technique consists of a set of graph operations capable of reducing a 2-D planar topology to isolated vertex/vertices while keeping the minimum energy of the topology. Meanwhile, this paper proposes two energy functions for 2-D topology adjustment by enforcing multiple geometric constraints, such as parallel and deviation constraints. The proposed snake algorithm and the energy functions are applied to construct 3-D building models from LIDAR measurements [27], and their effectiveness is verified thoroughly.

This paper is an extension of a short proceeding which appeared in the 2007 IEEE International Conference on Computer Vision and Pattern Recognition [28]. The new content in this paper includes more detailed descriptions of the four graph reduction operations, a generalization of the type IV reduction operation in [28] to allow the algorithm to work on general instead of a subset of 2-D snake problems. This paper is organized as follows. Section II illustrates a framework for 3-D building model construction based on the proposed snake algorithm. Section III describes the 2-D snake algorithm based on graph reduction techniques and the energy functions proposed for 3-D building model construction. Section IV presents the experimental results of the algorithm on 3-D building model construction from LIDAR measurements, and Section V concludes this paper.

II. CONSTRUCTION OF 3-D BUILDING MODEL FROM LIDAR DATA THROUGH 2-D SNAKE ALGORITHM

Fig. 1 illustrates a framework of constructing 3-D building models from LIDAR measurements through the 2-D snake algorithm. The framework consists of eight major steps. To facilitate the discussion, an example as shown in Fig. 3 is selected to illustrate the process of the framework and the results from each step of the framework. More details about the framework can be found in [3] and [29].

A. Separation of Ground and Nonground Points

The first step in the framework is to separate ground measurements from nonground ones (e.g., buildings and vegetation). We selected the progressive morphological filter [30] for this task because this filter separated the ground from nonground measurements sufficiently well for the sample data sets used in our experiment. Other filters can also be used in this step

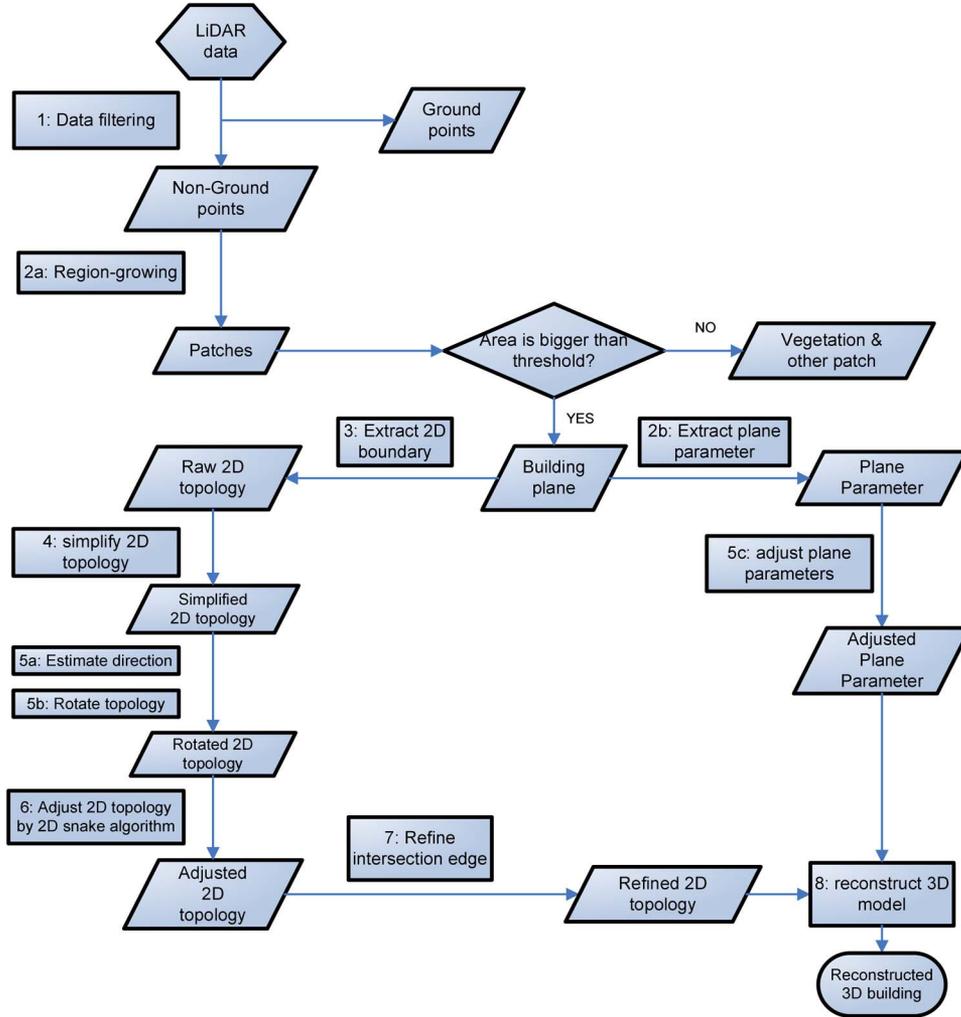


Fig. 1. Proposed framework for reconstruction of 3-D building models from LIDAR measurements. Each rectangular block represents one step or a substep of the framework. The integer inside each rectangular block corresponds to the processing order of the step in the framework.

if they produce a good classification. Fig. 3(a) demonstrates raw nonground LIDAR measurements overlaid on the aerial photograph.

B. Building Measurement Identification

The second step further separates building measurements from vegetation measurements using a region-growing algorithm based on a plane-fitting technique [2] first (step 2a). The rationale behind this algorithm is to group nonground measurements, which are located on the same planes, into patches. Building patches are much larger in size because measurements for a roof facet are almost always located in the same plane, while vegetation patch sizes are small due to large local variations in elevations for irregular LIDAR data. The plane equation describing each roof patch is derived in step 2b and represented by the following equation:

$$z = ax + by + c \quad (1)$$

where (a, b, c) are the plane parameter for one roof patch. To facilitate this discussion, measurements for each roof facet of a building are assigned a unique positive integer label starting from the label “1.” Fig. 3(b) shows the labels of facets from the

building in Fig. 3(a). There are totally seven facets identified for the building.

C. Two-Dimensional Topology Extraction

The third step identifies the raw 2-D topology of each building, which consists of footprints and internal boundaries between adjacent roof facets. The boundaries between roof facets form a set of connected polygons that are the projections of roof facets on a horizontal plane. In order to obtain the boundaries of a roof facet, the mesh c_s covering the data set further goes through a mesh refinement process and a boundary tracking step [29]. Fig. 3(c) demonstrates the identified boundary points (i.e., those with the label “-1”), and Fig. 3(d) illustrates the raw 2-D topology for the building in Fig. 3(a).

D. Two-Dimensional Topology Simplification

The fourth step simplifies the raw 2-D topology by applying an edge simplification algorithm. The raw 2-D topology is noisy because of the interpolation of irregularly spaced LIDAR measurements, segmentation errors introduced by the region-growing step, etc. The raw 2-D topology needs to be simplified so that the successive operations of the framework can be

applied. Here, the Douglas–Peucker algorithm [31] is selected for topology simplification since it generates high-quality results and its efficient implementation is available. Fig. 3(e) demonstrates the simplified 2-D topology for the building in Fig. 3(a).

E. Building Direction Estimation

The fifth step derives the building direction based on the simplified 2-D topology (step 5a). We select the algorithm proposed in [2] since it is robust and can handle complicated buildings with lots of edges oblique to the building direction. Next, the simplified 2-D topology is rotated clockwise according to the estimated direction (step 5b) so that the 2-D topology is aligned with the building direction for the convenience of successive adjustments. Meanwhile, the parallel and perpendicular property of each roof facet is enforced as proposed by [29] by adjusting the plane parameters based on the estimated building direction (step 5c). Since the building in Fig. 3(a) is estimated to be horizontally placed, it is unnecessary to rotate its 2-D topology in this step.

F. Two-Dimensional Topology Adjustment

The sixth step of the framework is to adjust the rotated 2-D topology by applying the proposed 2-D snake algorithm as detailed in Section III. We notice that the positions of most vertices and edges on the 2-D topology from step 5 are still distorted. An adjustment/rectification of the edges and the vertices is needed to derive accurate 3-D building models. The objective of topology adjustment is to enforce parallelism on the 2-D topology while keeping the adjusted topology as close to its original location as possible. We define direction energy E_{Dir} to enforce the parallelism constraint and deviation energy E_{Dis} to limit the deviation of the adjusted topology from its original position. More details of these two energy functions are discussed in Section III. The energy functions are gradually minimized starting with a simplified 2-D topology such as the one in Fig. 3(e). Finally, the topology corresponding to the minimum of the energy function is derived to represent the target topology. The key to this procedure is to develop an effective algorithm for energy minimization. The 2-D snake algorithm based on graph reduction is proposed to find a global minimum for a problem with 2-D topology, and the details of the algorithm are presented in Section III. Fig. 3(f) shows the 2-D topology after applying the proposed 2-D snake algorithm to the topology in Fig. 3(e), from which we can see that most of the edges on the 2-D topology are adjusted to be parallel or perpendicular to the building direction.

G. Adjustment of Edges at Intersections

The topology from the sixth step could be further refined in the seventh step for some buildings, such as residential houses that mainly consist of nonhorizontal roof facets. The edges between building facets can be classified into two categories: intersection and step edges. A step edge separates either two parallel planes or two intersecting planes with a height discontinuity [Fig. 2(a) and (b)]. An intersection edge separates two adjoining roof planes with height continuity [Fig. 2(c)]. Obviously, all edges of the footprint outline are step edges.

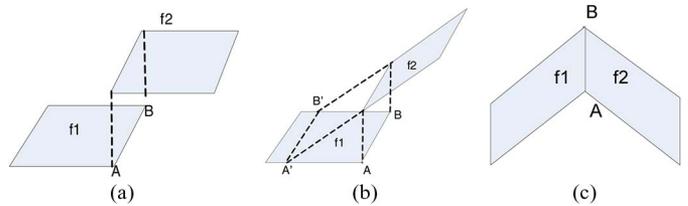


Fig. 2. Two types of edge (AB) between two roof facets. (a) Step edge separating two parallel planes. (b) Step edge separating two intersection planes with height discontinuity. (c) Intersection edge separating two intersection planes with height continuity.

The height values of a vertex on an intersection edge from two adjoining roof facets may be different because the proposed 2-D snake algorithm does not enforce the height continuity constraint. Fig. 3(g) demonstrates the reconstructed 3-D building model based on the topology from the 2-D snake algorithm directly. We can see many fake walls caused by the false height discontinuity between neighboring roof facets in the model. To remove this inconsistency, this step will enforce the height continuity constraint for vertices on intersection edges. However, before any adjustment operation is performed, intersection edges have to be identified. We determine an intersection edge using the following equation:

$$DH(e(v, w)) = \sum_{p \in \overline{vw}} |h_1(p) - h_2(p)| / n \quad (2)$$

where \overline{vw} is the set of grid cells containing the edge e . n is the number of grid cells in the set, and p is one grid cell in the set. $h_1(p)$ and $h_2(p)$ are the elevation of p on its two adjoining planes, respectively. If $DH(e)$ is less than a predefined threshold T_{Step} , edge e is classified as an intersection edge as shown in Fig. 2(c). Otherwise, it is a step edge as shown in Fig. 2(a) or (b). In our framework, T_{Step} is conservatively set to $2\Delta h_T$. Δh_T is the average elevation measurement error of LIDAR points. Since the height deviation of a point from the fitting plane can reach Δh_T in the worst situation, the height difference between two neighboring roof planes with height continuity can reach $2\Delta h_T$.

This step will check each vertex whether it is located on any intersection edge. If a vertex is located on two intersection edges, it will be replaced by the joint point of the three roof facets forming these two intersection edges. If a vertex is located on an intersection edge and a step edge, it will be replaced by the joint point of the two roof facets forming the intersection edge and the wall where the step edge is located. Fig. 3(h) shows the 2-D topology after refining several intersection edges in Fig. 3(f). Vertex 4 is replaced by the joint point of roof facets 1, 4, and 5 [see the facet labels in Fig. 3(c)]. Similar operations are applied to vertices 5, 9, and 10. We can see that the refined 2-D topology better approximates the real topology.

H. Three-Dimensional Building Model Construction

The 3-D building model will be constructed based on the adjusted 2-D topology and the plane parameters in previous step. ArcGIS 3-D engine is selected to visualize 3-D building models in our framework. For each roof facet, vertices forming its boundary are identified, and their x and y coordinates can be

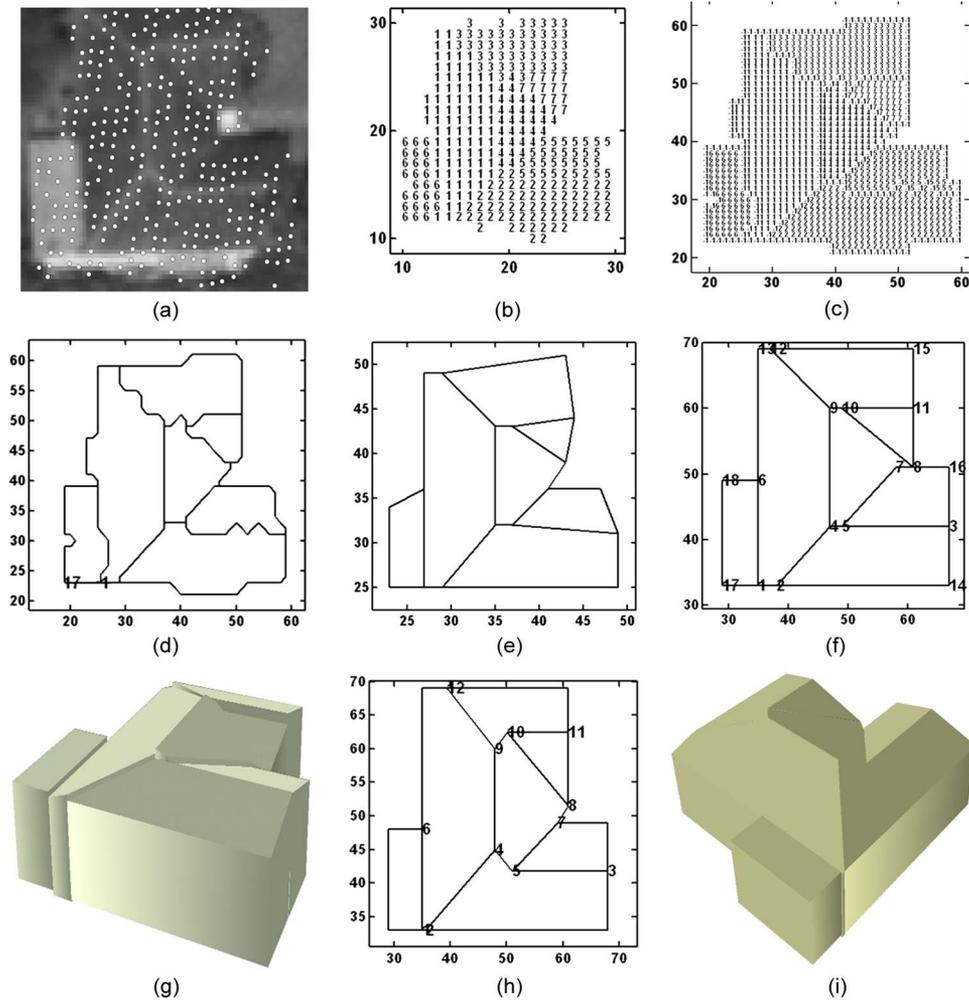


Fig. 3. Example illustrating the process of 3-D building model reconstruction based on the 2-D snake algorithm. (a) Raw LIDAR points overlaid on the aerial photograph. (b) Segmented roof facets which are labeled by different positive integers. (c) Points forming the boundaries of 2-D polygons are labeled with “-1.” (d) Raw 2-D topology of the building. (e) Simplified 2-D topology using the Douglas–Peucker algorithm. (f) Adjusted 2-D topology by applying the 2-D snake algorithm on the 2-D topology in (e). (g) Three-dimensional building model reconstructed based on the topology from (f). (h) Refined 2-D topology by replacing intersection edges with the joint point of neighboring roof facets. (i) Three-dimensional building model reconstructed in ArcGIS based on the refined topology from (h).

derived from the refined 2-D topology directly. z coordinates of these vertices can be calculated according to the plane (1). Vertices with x , y , and z coordinates are saved as a closed polygon in a shape file. The same operation is applied to every facet, and finally, the shape file will contain multiple closed polygons, and each of them corresponds to one roof facet’s boundary. For example, the shape file for the building in Fig. 3(c) contains seven polygons. The polygon for the facet with label “6” is formed by vertices 17, 18, 6, and 1. ArcGIS 3-D engine can load a shape file and generate the corresponding 3-D building model. Fig. 3(i) displays the constructed 3-D models for the building in Fig. 3(a).

III. TWO-DIMENSIONAL SNAKE ALGORITHM

A. Introduction of the 2-D Snake Algorithm

The 2-D snake algorithm was initially proposed by Yan [28]. The goal of the 2-D snake algorithm is to find the state assignment for each vertex that would minimize the energy of 2-D topology represented by a graph $G = (V, E)$. Let $Q = \{1, 2,$

$\dots, d\}$ be a set of possible states for any vertex $v \in V$. In our framework for 3-D building model reconstruction, each state of a vertex on the building topology corresponds to one possible position that the vertex could be adjusted to. The energy of vertex v in state $i \in Q$ is denoted by $EV_G(v, i)$. The energy of an edge $e = (u, v) \in E$ of graph G , given that vertex u is in state i and that vertex v is in state j , is denoted by $EE_G(e, i, j)$. Let S be a state assignment function for each vertex $v \in V$, i.e., $S : V \rightarrow Q$. For a given state assignment S , the energy of the entire graph G is the sum of the energies of the vertices and the edges of G

$$EG(G) = \sum_{e=(u,v) \in E} EE(e, S(u), S(v)) + \sum_{v \in V} EV(v, S(v)). \quad (3)$$

Note that the energy of an edge only depends upon two vertices to which the edge is connected. The specific forms of the energy function EV and EE rely on the particular application. Minimizing the total cost (energy) generates the

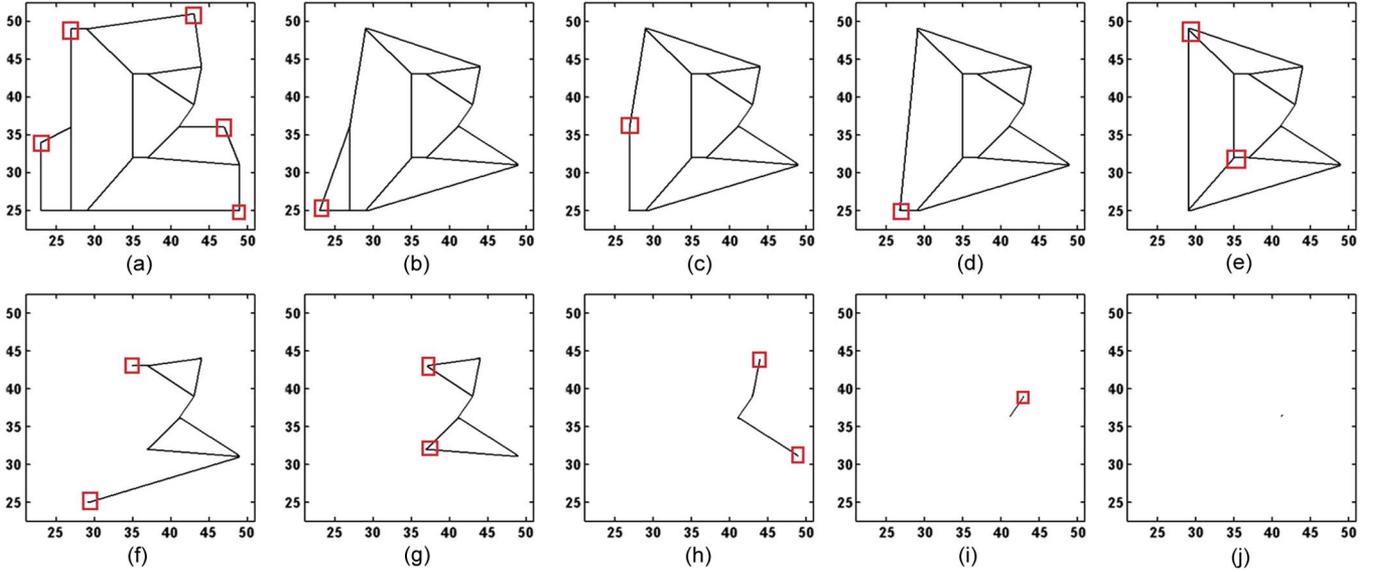


Fig. 4. Example showing the graph reduction process for the 2-D topology as shown in Fig. 3(e).

optimal topology that best fits the given object. A brute-force implementation of minimizing the energy specified in (3) is to try each of the d^n combinations of state assignments for n number of vertices of the graph. This would take time that is exponential to the number of vertices and is not tolerable for snake problems with complicated topology. The 2-D snake algorithm finds the minimum energy of a graph by progressively simplifying the graph using graph reduction operations. Each reduction operation will simplify the original graph while retaining the minimum energy of the graph. Four graph reduction operations, as listed in the following, are utilized in the 2-D snake algorithm, and examples and more details about them can be found in the Appendix.

- 1) Type I operation: eliminate vertices of degree two—This operation reduces a graph by replacing each vertex v of degree two (i.e., the number of edges incident to v is two) and its two incident edges with one single edge that connects the two vertices (v' and v'') to which v originally connects. The new edge e will hold the energy of v and its two incident edges. We store in a look-up table the state assignments of v that yield the minimum energy of e for each possible pair of state assignments of v' and v'' .
- 2) Type II operation: eliminate parallel edges—This operation reduces a graph by replacing all of the parallel edges between two vertices with one single edge e . This new edge will hold the energy of all of the replaced parallel edges.
- 3) Type III operation: eliminate vertices of degree one—This operation eliminates all vertices of degree one and their incident edges. The energy of an eliminated vertex v and its incident edge e will be stored in the vertex v' it originally connects to via e . The state assignment of v that minimizes the sum of its energy and that of e , in correspondence with each state assignment of v' , will be stored in the look-up table.
- 4) Type IV operation: eliminate a pair of vertices of higher degrees—This operation eliminates a selected pair of

vertices of higher degrees (> 2) and their incident edges, when none of type I–III operations can be applied further. The energy of those eliminated edges is absorbed by the vertices that are adjacent to the eliminated vertices, and the energy of the eliminated pair of vertices is absorbed by a random vertex selected from the subgraph that connects the two.

Fig. 4 demonstrates the graph reduction process for the 2-D topology as shown in Fig. 3(e). Vertices marked by red rectangles are involved in one graph reduction operation and removed. The original graph in Fig. 4(a) is reduced to a single vertex in Fig. 4(j). Type I operations (eliminate vertices of degree two) can be found in Fig. 4(a)–(d) and (g). In Fig. 4(b), a type I operation is first performed to replace a degree-2 vertex v (marked by a red rectangle) with an edge that connects the two vertices v originally connects to, introducing a new edge parallel to the one that already exists connecting the two vertices. A type II operation (eliminate parallel edges) immediately follows to replace the two parallel edges with one single edge. Type III operations (eliminate vertices of degree one) can be found in Fig. 4(f), (h), and (i). One type IV operation (eliminate a pair of vertices of higher degrees) is collected at Fig. 4(e). The state that yields the minimum energy for the single vertex at Fig. 4(j) is determined first. The states of the remaining vertices yielding the minimal energy can be determined by retrieving the corresponding states from their look-up tables created by the 2-D snake algorithm (see the Appendix for examples of look-up tables).

B. Energy Functions for Building 2-D Topology Adjustment

We propose two energy functions to enforce geometric constraints, such as parallel and deviation constraints. They are used for building 2-D topology adjustment in our proposed building model construction framework. Given an edge $e = (v, w)$ joining two vertices v and w on the topology and given that v' and w' are some possible states for vertices v and w ,

respectively, we define the direction energy E_{Dir} for the edge $e' = (v', w')$ as

$$E_{\text{Dir}}(e' = (v', w')) = \begin{cases} 0 & |v_x - w_x| \leq T \\ & \& v'_x = w'_x \\ 0 & |v_y - w_y| \leq T \\ & \& v'_y = w'_y \\ 0 & |v_x - w_x| > T \& \\ & |v_y - w_y| > T \& \\ & |v'_y - w'_y| > T \& \\ & |v'_x - w'_x| > T \\ |v - w| & \text{others.} \end{cases} \quad (4)$$

A nonadjustable oblique edge e is a line whose projections on the x - and y -axes are larger than a threshold T . All of the other edges are considered adjustable ones, and they will be adjusted to be either horizontal or vertical. A zero value is assigned to the direction energy function if an adjustable horizontal, adjustable vertical, or nonadjustable edge e is adjusted to a horizontal, vertical, or oblique edge e' , respectively. Otherwise, a penalty value proportional to the length of the edge is assigned to the adjusted edge e' . As a result of applying this energy function, the adjustable horizontal and vertical edges tend to align with the dominant directions, and nonadjustable oblique edges tend to remain oblique.

The deviation energy function E_{Dis} is defined as the sum of the distance values between the points on the adjusted edge (v', w') and that of the original edge (v, w)

$$E_{\text{Dis}}(e' = (v', w')) = \sum_{p \in v'w'} D_{\text{chess}}(v, v', w, w') \quad (5)$$

where p is a point on the edge (v', w') . The smaller the deviation energy of the adjusted edge e' , the closer e' is to the original edge e . In order to compute E_{Dis} , a distance transform [29] is applied on the 2-D topology to derive a gray scale image whose pixel intensity indicates the distance between the position of that pixel and the nearest edge in the topology. The edges between roof facets are rasterized using the same grid mesh for 2-D topology extraction in the previous section. As shown in Fig. 5, the distance values of edge cells are initialized as 0, and the distance values of their direct neighbor cells are assigned 1. The distance values of the direct neighbors of cells with distance values of 1 are assigned 2 and so on. Only the distance values of points in an area close to the edges in the original 2-D topology need to be calculated since each vertex on the 2-D topology is only allowed to move within a small window $W_v^*W_w$. The total energy for each adjusted edge e' is determined as follows:

$$E(e' = (v', w')) = C_{\text{Dir}} * E_{\text{Dir}}(e') + C_{\text{Dis}} * E_{\text{Dis}}(e') \quad (6)$$

where C_{Dir} and C_{Dis} are the weights for the two energy terms. The snake algorithm tests different states for each adjustable vertex on the 2-D topology and finds an optimal combination of vertex states that minimizes the sum of energy for all of the edges on the 2-D topology.

Fig. 5 demonstrates the distance values in the gray scale image after applying distance transform on the 2-D topology in Fig. 3(d). The distance value is calculated up to 3. These two

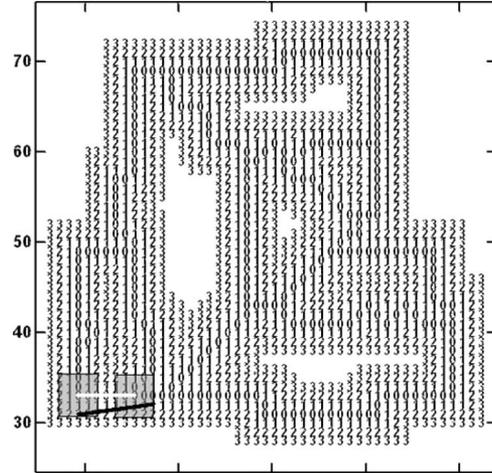


Fig. 5. Distance values derived by applying distance transform on the 2-D topology of the building in Fig. 3. Label “0” indicates the edge cells in between roof facets, and labels “1,” “2,” and “3” represent the distance values from the nearest edges. The x and y coordinates are in meters.

gray windows represent the adjustable positions for vertices 17 and 1 in Fig. 3(d), respectively. The white segment represents the original state of the edge connecting these two vertices. The black line represents one possible state of the edge, where E_{Dis} is $(2 + 2 + 2 + 2 + 1 + 1 + 1 + 1) = 12$ according to (5) and E_{Dir} is around 8 according to (4).

IV. EXPERIMENTS FOR THE 3-D BUILDING MODEL CONSTRUCTION FRAMEWORK

A. Data Processing

The proposed algorithm has been applied to reconstruct 3-D building models from airborne LIDAR measurements to examine its effectiveness. The test data site is located around the campus of Florida International University (FIU), Miami, FL, USA, covering 6 km² of low relief topography. Two data sets are analyzed in our experiment. One is located at the FIU campus and has 67 institutional buildings. Another one is next to the FIU campus and has 211 residential and commercial buildings. The LIDAR data for building extraction with an average point spacing of 1 m were collected in August 2003.

Initial topology of buildings is extracted from LIDAR measurements. The edges extracted are noisy, and most of the critical corner vertices were not in correct positions in the initial topology due to the influence of irregularly spaced point LIDAR measurements. The process for refining building topology involves adjusting the initial topology by changing the admissible states of vertices by minimizing the defined energy functions. The admissible states of vertices are determined by the spatial resolution and errors of LIDAR measurements. Therefore, the refinement of building topology from LIDAR measurements provides an ideal case to test the proposed algorithm. The thresholds used in our experiments for 3-D building reconstruction are listed in Table I. A sensitivity analysis showed that small changes in most thresholds have little impact on the final results. Aerial photographs and field investigation were used to help evaluate the reconstructed building models. The aerial photographs were collected in 1999 at a resolution of 0.3 m. The focus of the experiment is to examine whether

TABLE I
PARAMETERS FOR RECONSTRUCTING BUILDING MODELS

Cell size c_s for Snake algorithm	0.5 m
Threshold to identify segmentation edges T_Step	0.4 m
Threshold for projection (T)	2 m ($4 c_s$)
Weight for direction energy (C_{Dir})	1
Weight for deviation energy (C_{Dis})	5
Window size for vertex adjustment (W_v)	5

the minimum energy of a connected building topology can be derived within a reasonable amount of time and how effective the proposed energy functions can be applied to construct 3-D building models.

B. Results

Here, 66 of 67 buildings in the data set at the FIU campus and 210 of 211 buildings in the data set next to the FIU campus are successfully reduced by the proposed algorithm without type IV operations involved, which means that over 98% of buildings from both data sets can be processed by type I, II, and/or III operations only. The remaining two buildings have very complicated topologies and are successfully refined through the 2-D snake algorithm with one type IV reduction operation involved in each case. It took about 2 and 3 min for a personal computer with a 2.8-GHz processor and 2-GB RAM to complete the entire building reconstruction process for the data set of the FIU campus and the data set next to the FIU campus, respectively, which demonstrates the efficiency of the proposed snake algorithm in calculating the minimum energy of building topologies in real-world cases.

Most buildings in the data set at the FIU campus consist of flat roof planes, and their 3-D building model can be constructed by the footprint and the height of the building. The quality of these reconstructed models is evaluated by comparing the footprint of models with the actual footprint. The result indicates that the footprint extracted based on the snake algorithm is much better than that from the adjusting operations in Zhang [2]. Fig. 6 compares the footprints of several flat buildings in the FIU main campus extracted by these two methods. Both of these two methods produce very similar geometric shapes. After careful visual inspection, we found that the snake-based algorithm generates more accurate details. For example, compared with Fig. 6(a), Fig. 6(b) has more satisfactory results at eight edges enclosed by red ovals, although it has worse results at three edges enclosed by green ovals. Table II shows the quantitative comparison results between the footprint results adjusted by these two methods. The first and second rows show the building IDs and the number of edges of each building topology. Error1 and Error2 represent the numbers of edges incorrectly adjusted by the adjusting method and the proposed 2-D snake algorithm, respectively. There are 12 buildings in this area, and the total number of errors from the 2-D snake algorithm is 11, which is 50% improvement compared with the total errors (22) from the adjusting operation method.

The quality of the reconstructed models for the data set next to the FIU campus is evaluated by visual inspection since no ground-truth 3-D building models for this area are available. Fig. 7(a) and (b) shows the extracted 2-D topologies and 3-D building models from this data set. It is very difficult to quantify

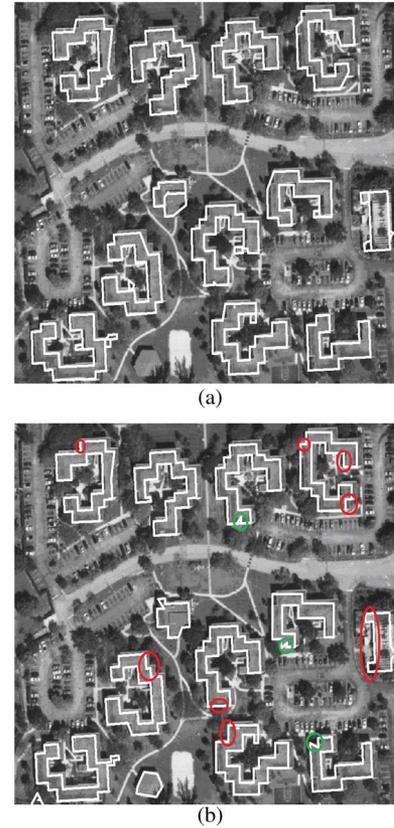


Fig. 6. Footprints of buildings in the FIU main campus adjusted by (a) adjusting operations and (b) the proposed snake algorithm and energy functions.

TABLE II
COMPARISON OF FOOTPRINTS ADJUSTED BY ADJUSTING OPERATIONS AND THE PROPOSED SNAKE ALGORITHM

Building Id	Number of Edges	Error1	Error2
1	26	2	1
2	28	0	1
3	26	0	1
4	26	3	0
5	10	2	2
6	26	4	1
7	26	2	0
8	16	3	2
9	8	2	1
10	28	2	1
11	30	2	0
12	16	0	1

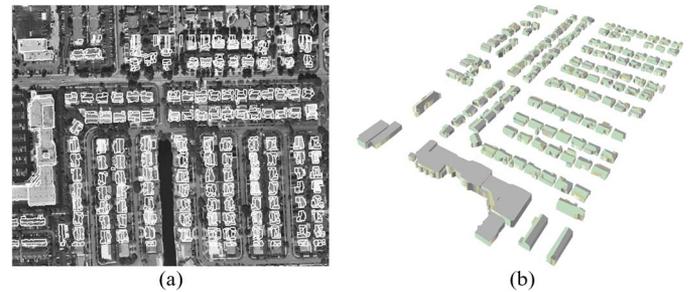


Fig. 7. (a) Two-dimensional topology overlaid on an aerial photograph. (b) Reconstructed 3-D models for buildings next to the FIU campus.

the errors of the algorithm for reconstructing 3-D building models since no ground-truth data with a higher accuracy are available, and digitizing 3-D building models manually from LIDAR measurements is not feasible. We qualitatively examined the

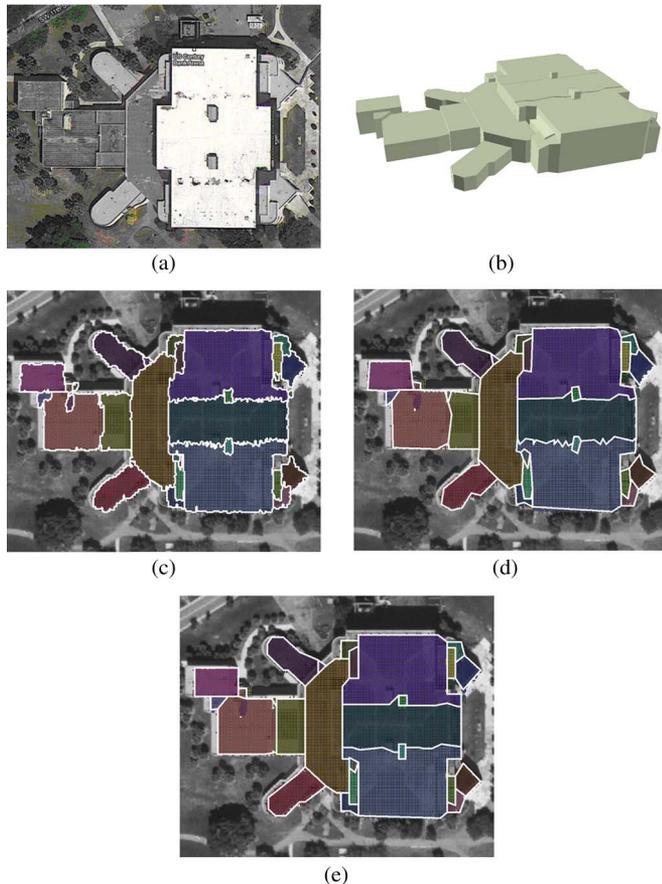


Fig. 8. Derivation of a complex 3-D building model from LIDAR measurements. (a) Aerial photography of the building. (b) Reconstructed 3-D building model. (c) Raw 2-D topology. (d) Simplified 2-D topology through the 2-D snake algorithm. LIDAR measurements for various flat roof facets are displayed with different colors.

quality of the extracted building models by comparing the constructed 3-D building models with the LIDAR measurements and the aerial photographs in ArcGIS. The results indicated that most buildings (196/211) were reconstructed properly by the proposed snake algorithm. The remaining building models are not constructed with a satisfactory quality since the building facet segmentation results at those areas are poor.

Fig. 8 demonstrates the 3-D building model reconstruction process for a complicated building. The 2-D topology of the building roof consists of 46 vertices and 67 boundaries. Although the topology looks very complicated, it contains only 1 type IV reduction operation, and the minimum energy can be determined in less than 1 min. Meanwhile, all of the roof facets of the building are flat, and we can skip the step to adjust intersection edges. The reconstructed 3-D building model through the 2-D snake algorithm is shown in Fig. 8(b) and is visually satisfactory.

Some buildings, such as the residential house in Fig. 3(a), mainly consist of nonhorizontal roof planes. The 2-D topology adjusted through the 2-D snake algorithm needs to be refined in order to derive satisfactory 3-D building models. Fig. 3(f) illustrates the 2-D topology derived by applying the proposed 2-D snake algorithm on the building in Fig. 3(a). We can see that edges connecting vertices 4 and 5 are misclassified as horizontal ones. Some intersection edges, such as that connect-

ing vertices 2 and 4 in Fig. 3(f), are adjusted away from its exact location. The height continuity is violated in some places, leading to some fake walls as visualized in the constructed building model shown in Fig. 3(g). Since the edges connecting vertices 9–10 and 9–12 are classified as intersection edges, vertex 9 is replaced by the joint point of three planes with labels 1, 3, and 4 [see labeled planes in Fig. 3(b)]. Vertices 10, 4, and 5 are adjusted in a similar way. Since vertex 11 is located on both the footprint of the building and the intersection edge connecting vertices 10 and 11, it is replaced by the joint point of the planes with labels 3 and 7 and the wall where the footprint is located. Fig. 3(i) demonstrates the constructed 3-D building model after replacing intersection edges. Comparing Fig. 3(i) with Fig. 3(g), we can see that, with additional topology refinement by replacing intersection edges, the resulting 3-D building model constructed fits closer to the real building in Fig. 3(a).

Accurate segmentation of roof facets is critical for the extraction of 2-D topologies and reconstruction of sophisticated building models. Theoretically, the high density of LIDAR measurements will increase the accuracy of segmentation of building patches because more points are available for parameter estimation. Numerical experiments also demonstrated that segmentation accuracy will be enhanced as the density of LIDAR measurements increases. Some reconstructed models, such as the 3-D model in Fig. 8(e), did not generate result good enough for certain edges between roof facets within a building footprint since the accuracy of segmentation is restricted by the low density of LIDAR measurements. Further improvement in segmentation is needed for reconstruction of better building models.

The window size for vertex adjusting W_v also has certain impact on the quality of reconstructed building models. With a large window size, each vertex on the topology has more allowable states for adjustment, and the refined topology will likely be more visually pleasing. However, the snake algorithm has to take more time in deriving the minimum energy of the 2-D topology. Also, the commission and omission errors between the refined and raw 2-D topologies tend to increase because a vertex could be adjusted to positions further away from its original position. Therefore, there is a tradeoff between achieving visually pleasant results and minimizing commission and omission errors. In our experiment, we selected a window size large enough to achieve a reasonably good quality of refined topology while keeping commission and omission errors low. Energy functions used by the snake algorithm have a direct effect on the refined 2-D topology. New energy functions can be easily added into the 2-D snake algorithm to enforce more geometric constraints simultaneously.

V. CONCLUSION

Many algorithms have been developed to automatically extract 3-D building models from LIDAR measurements in the past decade. However, all of them have different kinds of limitations. For example, some algorithms need involvement of data source other than LIDAR, such as topographic maps. Some algorithms generate building models with poor quality since the 2-D building topologies extracted are always noisy.

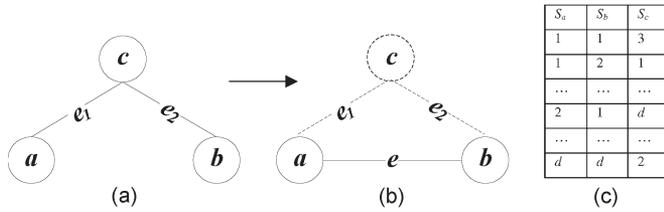


Fig. 9. Example of type I operation. (a) Original graph. (b) Graph after being reduced by this operation. The vertex and the two edges marked by dashed lines are removed and replaced by a new edge e . (c) Associated state look-up table.

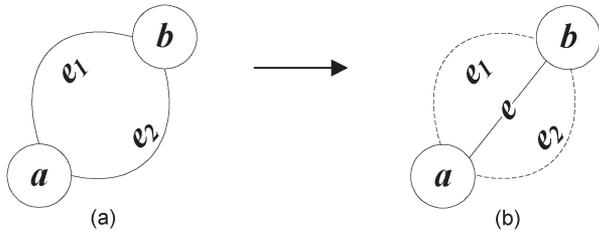


Fig. 10. Example of type II operation to reduce the two parallel edges connecting vertices a and b . (a) Original graph. (b) Graph after being reduced by this operation. The edges marked by dashed line are removed and replaced by a new edge e .

To overcome these limitations, a framework is introduced in this paper to construct 3-D building models from LIDAR data through a novel 2-D snake algorithm. This paper has three major technical contributions. First, this framework enables a systematic and automatic way of refining noisy building topologies and generating building models with a much better quality. Second, this paper proposes a “2-D snake algorithm” to address the challenging issue from traditional snake algorithm, which is to find the minimum energy of a complicated graph instead of a 1-D contour. Third, this paper proposes two energy functions for building topology adjustment. They enforce multiple geometrical constraints, such as deviation and parallel constraints, in building topology adjustment easily and simultaneously, which is very difficult to achieve in existing algorithms for 3-D building model construction.

The effectiveness of the framework has been verified by applying the algorithm to constructing various institutional, commercial, and residential buildings. Over 98% of building topologies have been successfully processed by the 2-D snake algorithm with only type I, II, or III operations, and their global optima have been found in polynomial time. The remaining 2% of buildings’ topologies can be processed with only one type IV operations involved, which still takes polynomial time. This demonstrates that the proposed 2-D snake algorithm is very effective and efficient in deriving the minimum energy for the building topology adjustment which is the most critical step in the framework for building model construction. Both the quantitative and qualitative experimental results demonstrate that the framework is effective in constructing a 3-D model for various kinds of buildings.

APPENDIX A TYPE IV OPERATION

Details about type I–III operation can be found in [28] by Yan *et al.* Figs. 9–11 demonstrate examples of each type, respectively.

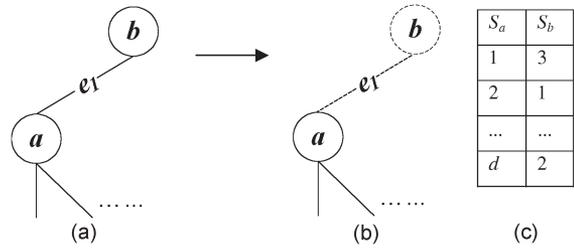


Fig. 11. Example of type III operation. (a) Original graph. (b) Graph after being reduced by this operation. The vertex and the edge marked by dashed lines are removed. (c) Associated state look-up table.

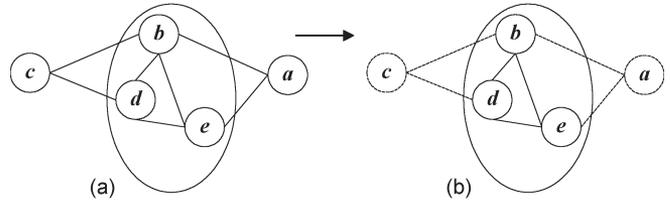


Fig. 12. Example of type IV operation. (a) Original graph. (b) Graph after being reduced by the operation. The vertices and the edges marked by dashed lines are removed.

Originally, the 2-D snake algorithm is restricted by type IV reduction operation proposed by Yan *et al.* [28] and can be applied only to some kind of graphs. A new type IV operation is proposed in this paper to enable the 2-D snake algorithm to work on any kind of graphs. For a given pair of vertices a and c , as shown in Fig. 12(a), we will remove a and c and the edges adjacent to them in a type IV operation. Assume that a connects with vertices a_1, \dots, a_t , and c connects with vertices c_1, \dots, c_g .

We denote the reduced graph by $G_4 = (V_4, E_4)$, where $V_4 = V \setminus \{a, c\}$ and $E_4 = E \setminus \{(a, a_1), \dots, (a, a_t), (c, c_1), \dots, (c, c_g)\}$.

If S_a and S_c are the states of vertices a and c , respectively, in graph G , then the energy of a vertex $v \in V_4$ will be updated as follows if v connects to either a or c :

$$EV_{G_4}(v, S_v) = EV_G(v, S_v) + EE_G(e = (v, u), S_v, S_u) \quad (7)$$

where S_v is the state of vertex v and v connects to $u \in \{a, c\}$. If v connects to both a and c , its energy will be updated as follows:

$$EV_{G_4}(v, S_v) = EV_G(v, S_v) + EE_G(e = (v, a), S_v, S_a) + EE_G(e = (v, c), S_v, S_c). \quad (8)$$

Meanwhile, one random vertex $w \in V_4$ will be selected to hold the energy of vertices a and c . If vertex a connects with c , w will also take the energy of edge (a, c) , and its energy will be updated as follows:

$$EV_{G_4}(w, S_w) = EV_G(w, S_w) + EV_G(a, S_a) + EV_G(c, S_c) + EE_G(e = (a, c), S_a, S_c). \quad (9)$$

Otherwise, the energy of vertex w will be updated as follows:

$$EV_{G_4}(w, S_w) = EV_G(w, S_w) + EV_G(a, S_a) + EV_G(c, S_c) \quad (10)$$

where S_w is a state of vertex w . Since there are d^2 possible state combinations of (S_a, S_c) , we need to calculate the minimum

energy of the reduced graph G_4 under d^2 possible state combinations (S_a, S_c) in order to derive the minimum energy of the original graph G . Similar to other reduction operations, graphs G and G_4 have the same minimum energy.

With type I–IV operations, we have developed the following 2-D snake algorithm to reduce a graph G and derive the state assignment list $min_states = (s_1, s_2, \dots, s_k, \dots, s_{|V|}), s_k \in Q$ that minimizes the total energy of G . This algorithm consists of three major steps. The first step applies and records all of the reduction operations used to reduce a graph G to a set of isolated vertices and stores them in array *operations*. The second step calculates the minimum energy min_energy of the graph G , and the last step returns the state assignment min_states of vertices that minimizes the energy of G . Details about each step are listed as follows.

Step 1) *Collect reduction operations*: This step starts with an iterative process of collecting type I, II, and III operations. It iteratively scans each vertex v of graph G , applies any type I, II, and III operations applicable to v to the current graph, and records the applied operations in the array *operations*. The iteration terminates if no type I–III operation can be further applied to the graph.

If the graph has been reduced to a set of isolated vertices, the algorithm exits the current step. Otherwise, a pair of vertices will be selected based on certain criteria, and a type IV reduction operation based on the selected vertices is applied to further reduce (simplify) the graph and recorded in the array *operations*. Then, the algorithm goes back to the beginning of this step and continues to collect new type I–IV reduction operation until no new operation can be applied.

Step 2) *Determine the minimum energy of G* : This step recursively calculates the minimum energy of the graph G based on the information stored in the operation array *operations*. It checks the operations in the array *operations* one by one from the beginning. For a type I, II, or III operation, the graph is deterministically reduced, and its energy is updated accordingly. For a type IV operation, the energy of the current graph at each possible state combination of the pair of vertices removed in this operation will be calculated recursively and compared to return the minimum. For each possible state combination, the graph is reduced, and its energy is updated accordingly.

Once all of the operations in the operation array have been checked, the graph should be reduced to a set of isolated vertices. The minimum energy of the graph is the sum of the minimum energy of all of the vertices, which is further compared with the global minimum energy min_energy found so far. If the minimum energy of the current graph is smaller, it will replace min_energy , and the corresponding states of vertices removed in every type IV operation will be recorded in the array min_states .

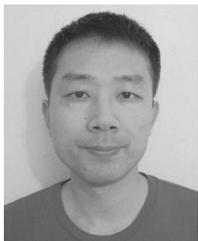
Step 3) *Determine the state assignment of vertices that minimizes the energy of G* : After step 2, vertices removed in all type IV operations have determined their corresponding states in min_states . To decide the states of the remaining vertices in min_states , this step scans again each reduction operation in the array *operations* from the start and applies them to the original graph G one by one. For a type I, II, or III operation, the graph's energy is updated accordingly, and the corresponding state look-up table is constructed. For each type IV operation, the energy of the graph is updated accordingly, given the known states of the pair of vertices removed in the operation which are stored in array min_states .

After all of the reduction operations have been processed, the current graph is reduced to a set of isolated vertices. The state of every isolated vertex that yields the minimum energy value is determined by comparing its energy under each possible state and saved to min_states . The states of the remaining vertices are retrieved by checking the look-up tables in the order reverse to the insertion order of their corresponding operations.

REFERENCES

- [1] J. R. Jensen, *Remote Sensing of the Environment*, 2nd ed. Upper Saddle River, NJ, USA: Prentice-Hall, 2000.
- [2] K. Zhang, J. Yan, and S.-C. Chen, "Automatic construction of building footprints from airborne LIDAR data," *IEEE Trans. Geosci. Remote Sens.*, vol. 44, no. 9, pp. 2523–2533, Sep. 2006.
- [3] K. Zhang, J. Yan, and S.-C. Chen, "A framework for automated construction of building models from airborne LIDAR measurements," in *Topographic Laser Ranging and Scanning: Principles and Processing*, J. Shan and C. Toth, Eds. New York, NY, USA: Taylor & Francis, 2008.
- [4] E. Schwalbe, "3D building model generation from airborne laser scanner data using 2D GIS data and orthogonal point cloud projections," in *Proc. ISPRS WG III/3, III/4, V/3 Workshop Laser Scanning*, Enschede, The Netherlands, Sep. 12–14, 2005, pp. 209–214.
- [5] H. Maas and G. Vosselman, "Two algorithms for extracting building models from raw laser altimetry data," *ISPRS J. Photogramm. Remote Sens.*, vol. 54, no. 2/3, pp. 153–163, Jul. 1999.
- [6] C. Brenner, "Modelling 3D objects using weak CSG primitives," in *Proc. Int. Arch. Photogramm. Remote Sens.*, 2004, vol. 35, no. B3, pp. 1–6.
- [7] R.-J. You and B.-C. Lin, "A quality prediction method for building model reconstruction using LIDAR data and topographic maps," *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 9, pp. 3471–3480, Sep. 2011.
- [8] M. Awrangjeb, M. Ravanbakhsh, and C. S. Fraser, "Automatic detection of residential buildings using LIDAR data and multispectral imagery," *ISPRS J. Photogramm. Remote Sens.*, vol. 65, no. 5, pp. 457–467, Sep. 2010.
- [9] J. Xiao, M. Gerke, and G. Vosselman, "Building extraction from oblique airborne imagery based on robust facade detection," *ISPRS J. Photogramm. Remote Sens.*, vol. 68, pp. 56–68, Mar. 2012.
- [10] A. Sampath and J. Shan, "Segmentation and reconstruction of polyhedral building roofs from aerial LIDAR point clouds," *IEEE Trans. Geosci. Remote Sens.*, vol. 48, no. 3, pp. 1554–1567, Mar. 2010.
- [11] K. Kim and J. Shan, "Building roof modeling from airborne laser scanning data based on level set approach," *ISPRS J. Photogramm. Remote Sens.*, vol. 66, no. 4, pp. 484–497, Jul. 2011.
- [12] A. Gruen and X. Wang, "News from cybercity-modeler," in *Proc. of 3rd International Workshop on Automatic Extraction of Man-Made Objects from Aerial and Space Images*, Monte Verita, Ascona, Switzerland, Jun. 10–15, 2001, pp. 93–101.
- [13] G. Vosselman, "Building reconstruction using planar faces in very high density height data," in *Proc. Int. Archive Photogramm. Remote Sens.*, 1999, vol. 32, no. 32W5, pp. 87–92.

- [14] D. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *Int. J. Comput. Vis.*, vol. 1, no. 4, pp. 321–331, Jan. 1988.
- [15] G. Subsol, J. P. Thirion, and N. Ayache, "A scheme for automatically building three-dimensional morphometric anatomical atlases: Application to skull atlas," *Med. Image Anal.*, vol. 2, no. 1, pp. 37–60, Mar. 1998.
- [16] D. Terzopoulos, A. Witkin, and M. Kass, "Constraints on the deformable models: Recovering 3D shape and nonrigid motions," *Artif. Intell.*, vol. 36, no. 1, pp. 91–123, Aug. 1988.
- [17] J. P. Thirion, "Image matching as a diffusion process: An analogy with Maxwell's demons," *Med. Image Anal.*, vol. 2, no. 3, pp. 243–260, Sep. 1998.
- [18] T. Chan and L. Vese, "Active contours without edges," *IEEE Trans. Image Process.*, vol. 10, no. 2, pp. 266–277, Feb. 2001.
- [19] V. Caselles, R. Kimmel, G. Sapiro, and C. Sbert, "Minimal surfaces based object segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 4, pp. 394–398, Apr. 1997.
- [20] J. Montagnat, H. Delingette, and N. Ayache, "A review of deformable surfaces: Topology, geometry, deformation," *Image Vis. Comput.*, vol. 19, no. 14, pp. 1023–1040, Dec. 2001.
- [21] S. Ahmadi, M. V. Zoej, H. Ebadi, H. Moghaddam, and A. Mohammadzadeh, "Automatic urban building boundary extraction from high resolution aerial images using an innovative model of active contours," *Int. J. Appl. Earth Observ. Geoinf.*, vol. 12, no. 3, pp. 150–157, Jun. 2010.
- [22] J. Peng, D. Zhang, and Y. Liu, "An improved snake model for building detection from urban aerial images," *Pattern Recognit. Lett.*, vol. 26, no. 5, pp. 587–595, Apr. 2005.
- [23] K. Karantzas and N. Paragios, "Recognition-driven two dimensional competing priors toward automatic and accurate building detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 47, no. 1, pp. 133–144, Jan. 2009.
- [24] A. A. Amini, T. E. Weymouth, and R. C. Jain, "Using dynamic programming for solving variational problem in vision," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 12, no. 9, pp. 855–867, Sep. 1990.
- [25] D. Geiger, A. Gupta, L. A. Costa, and J. Vlontzos, "Dynamic programming for detecting, tracking, matching deformable contours," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 17, no. 3, pp. 294–302, Mar. 1995.
- [26] Y. Boykov, O. Weksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 11, pp. 1222–1239, Nov. 2001.
- [27] J. Shan and C. Toth, *Topographic Laser Ranging and Scanning: Principles and Processing*. Boca Raton, FL, USA: CRC Press, 2008.
- [28] J. Yan, K. Zhang, C. Zhang, S.-C. Chen, and G. Narasimhan, "A graph reduction method for 2D snake problems," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Minneapolis, MN, USA, Jun. 17–22, 2007, pp. 1–6.
- [29] J. Yan, "A Framework for Automatic Feature Extraction From Airborne Light Detection and Ranging Data," Ph.D. dissertation, Florida Int. Univ., Miami, FL, USA, 2007.
- [30] K. Zhang *et al.*, "A progressive morphological filter for removing non-ground measurements from airborne LIDAR data," *IEEE Trans. Geosci. Remote Sens.*, vol. 41, no. 4, pp. 872–882, Apr. 2003.
- [31] D. Douglas and T. Peucker, "Algorithms for the reduction of the number of points required to represent a digitized line or its caricature," *Can. Cartogr.*, vol. 10, no. 2, pp. 112–122, Dec. 1973.



Jianhua Yan received the Ph.D. degree from the School of Computer Science, Florida International University, Miami, FL, USA, in 2007.

He is currently with the Transaction Risk Management Team, Amazon.com, Seattle, WA, USA. His research interests include image processing, pattern recognition, data mining, and geographic information systems.



Keqi Zhang received the Ph.D. degree from the University of Maryland, College Park, MD, USA, in 1998.

He joined the Department of Earth and Environment, Florida International University, Miami, FL, USA, as an Assistant Professor in 2003, where he was promoted to Associate Professor in 2007 and Full Professor in 2013. He is the author and coauthor of 80 journal papers, book chapters, conference proceedings, and technical reports. His research interests include coastal response to sea level rise and storm impact, airborne light detection and ranging remote sensing, storm surge modeling, and mapping coastal hazards using geographic information systems.



Chengcui Zhang (M'00) received the Ph.D. degree in computer sciences from Florida International University, Miami, FL, USA, in 2004. She is an Associate Professor with the Department of Computer and Information Sciences, The University of Alabama at Birmingham, Birmingham, AL, USA. She is the author and coauthor of over 130 research papers focusing on multimedia data mining, multimedia information retrieval, multimedia databases, bioinformatics, and geographic information systems.

Prof. Zhang has served on more than 70 international conferences and workshops and taken a leadership role at the 2014 IEEE International Conference on Multimedia and Expo, the 2012–2013 IEEE International Conference on Information Reuse and Integration, 2012 IEEE International Symposium on Multimedia, and 2010 IEEE International Conference on Multimedia and Ubiquitous Engineering (MUE). In addition, she is a frequent panelist for National Science Foundation panels and an Editorial Board member of several international journals.



Shu-Ching Chen (M'95–SM'04) received the M.S. degrees in computer science, electrical engineering, and civil engineering and the Ph.D. degree in electrical and computer engineering from Purdue University, West Lafayette, IN, USA, in 1992, 1995, 1996, and 1998, respectively.

He is currently a Full Professor with the School of Computing and Information Sciences, Florida International University, Miami, FL, USA. His main research interests include distributed multimedia database management systems, multimedia data mining, and disaster information management.

Dr. Chen is a Fellow of SIRI. He was a steering committee member of the IEEE TRANSACTIONS ON MULTIMEDIA from 2011 to 2013. He received the 2011 ACM Distinguished Scientist Award and the Best Paper Award from the 2006 IEEE International Symposium on Multimedia.



Giri Narasimhan received the B.Tech. degree in electrical engineering from the Indian Institute of Technology, Bombay, India, and the Ph.D. degree in computer science from the University of Wisconsin, Madison, WI, USA.

He is a Professor with the School of Computing and Information Sciences, Florida International University (FIU), Miami, FL, USA. He is also currently the Associate Dean for Research and Graduate Studies with the College of Engineering and Computing, FIU. He has been on the program committee of

numerous international conferences and is on the editorial board of three international journals. His research has been funded by the US National Science Foundation, US National Institutes of Health, state agencies, and industry. He has more than 100 refereed publications to his credit in the form of journal publications, conference proceedings, and book chapters. He is the coauthor of a monograph entitled *Geometric Spanner Networks* (Cambridge University Press) and has coedited two conference proceedings. His research interests are in the design and analysis of algorithms and bioinformatics.