Journal of Zhejiang University-SCIENCE C (Computers & Electronics)
ISSN 1869-1951 (Print); ISSN 1869-196X (Online)
www.zju.edu.cn/jzus; www.springerlink.com
E-mail: jzus@zju.edu.cn

Intelligent optimization of seam-line finding for orthophoto mosaicking with LiDAR point clouds*

Hong-chao MA†1, Jie SUN2
(1*School of Remote Sensing, Wuhan University, Wuhan 430079, China*)
(2*State Key Lab for Surveying, Mapping and Remote Sensing, Wuhan University, Wuhan 430079, China*)
†E-mail: hchma@whu.edu.cn

Received July 2, 2010; Revision accepted Dec. 13, 2010; Crosschecked Feb. 28, 2011

Abstract: A detailed study was carried out to find optimal seam-lines for mosaicking of images acquired by an airborne light detection and ranging (LiDAR) system. High ground objects labeled as obstacles can be identified by delineating black holes from filtered point clouds obtained by filtering the raw laser scanning dataset. An innovative A* algorithm is proposed that can automatically make the seam-lines keep away from these obstacles in the registered images. This method can intelligently optimize the selection of seam-lines and improve the quality of orthophotos. A simulated grid image was first used to analyze the effect of different heuristic functions on path planning. Three subsets of LiDAR data from Xi'an, Dunhuang, and Changyang in Northwest China were obtained. A quantitative method including pixel intensity, hue, and texture was used. With our proposed method, 9.4%, 8.7%, and 9.8% improvements were achieved in Dunhuang, Xi'an, and Changyang, respectively.

Key words: Light detection and ranging (LiDAR), Filter, A* algorithm, Mosaicking, Seam-line
doi:10.1631/jzus.C1000235 **Document code:** A **CLC number:** P237.3

1 Introduction

Airborne Light detection and ranging (LiDAR) systems have been widely developed in the past decade, with the capability of acquiring 3D spatial data quickly and with reasonably high precision over a given area (Baltsavias, 1999a). A modern commercial LiDAR system has a charge-coupled device (CCD) camera as one component of its payload (Baltsavias, 1999b) and therefore, can acquire photogrammetric images during its operation. Such a system can produce not only a large scale digital terrain model (DTM) with high accuracy, seen as the primary use of a LiDAR system, but also high quality digital ortho-maps (DOM) which, in some applications, could be as important a final product as the DTM (Axelsson, 1999). One of the critical steps for DOM generation from a given image dataset is the finding of seam-lines for image mosaicking. A bad seam-line can cause serious errors in the result, such as discontinuous roads, rivers, bridges, and the breaking of buildings or other man-made objects (Kerschner, 2001). In the process of automatically finding a seam-line, firstly, an initial seam-line is often selected, which usually crosses the obviously different gray areas in overlapping regions. Strategies are then employed for optimization. However, many factors can cause gray differences, for instance, illumination differences, lens distortion, and relief displacement. In a real aerial photogrammetric project, all images are usually acquired during the same time period, so illumination differences are small. In addition, color balancing is often performed to eliminate color differences between images. However, since a true DOM is not produced, the relief displacement of high man-made objects remains. Such displacements often make a high man-made object tilt in different directions on adjacent images, leading to large gray differences in the overlapping area. A seam-line should

---

* Project supported by the National Basic Research Program (973) of China (No. 2009CB724007) and the National High-Tech R & D Program (863) of China (No. 2006AA12Z101)
© Zhejiang University and Springer-Verlag Berlin Heidelberg 2011

keep away from such areas. Of all the main factors causing gray differences, relief displacements of high man-made objects are the most serious, especially in urban areas. Thus, in this study we focus on keeping the seam-line away from high man-made objects.

Traditional methods for finding a seam-line (Shiren *et al.*, 1989; Fernandez and Marti, 1999) are usually based on the minimum averaged differences of gray values in overlapping areas through which seam-lines pass. Schickler and Thorpe (1998) used a correlation coefficient for measuring similarity between two contiguous pixels along a seam-line. Geometric criteria were also used in their approach. Fernandez and Marti (1999) applied a weighted graph search algorithm for seam-line optimization. Davis (1998) used the Dijkstra algorithm to search for the shortest path as the optimal seam-line, but the efficiency of the algorithm was low. Kerschner (2001) proposed a twin snake algorithm to find a seam-line. In this algorithm, a more sophisticated similarity measure is defined that makes use not only of color, but also of texture. Two snakes start from the opposite edges of the overlapping area, and then pull each other together to become one optimal seam-line according to an energy function. This algorithm cannot overcome the local minima problem completely, and it is computationally expensive. In addition, large radiometric and color differences, especially for multi-temporal imagery, may require fully manual processing. Chon and Kim (2006) adopted dynamic programming on the converted cost space to find the optimal seam-line. But this algorithm has a ‘distance effect’; that is, with the increase in the length of the seam-line, the computational cost increases dramatically.

In general, traditional methods can produce a good seam-line if the quality of all adjacent images is satisfactory and the gray difference is small enough in overlapping areas. However, large gray differences in overlapping areas often occur due to various factors. Suboptimal seam-lines will be generated if traditional methods are used in such scenarios. Automatically finding high quality seam-lines for image mosaicking remains an active research topic.

Seam-line finding by the above methods is based solely on pixel information. In this paper, a novel LiDAR point clouds based seam-line optimization method is proposed.

The schematic workflow of our method consists of three steps:

1. Filtering non-ground points and marking high ground objects such as man-made buildings as obstacle areas.
2. Interpolating the filtered point clouds to generate a highly precise DTM from which real obstacle areas can be extracted. Water bodies are removed during the interpolation.
3. Generating an initial seam-line network using positioning and orientation system (POS) data. The optimized seam-line is then found by using an improved A^* algorithm with assistance of the obstacle marked DTM. Interpolating the obstacle marked DTM eliminates all the obstacles in the original DTM dataset, which is used for DOM generation. The raw images are then ortho-rectified with obstacle-removed DTM and POS data. The mosaicking process is then performed with the optimized seam-lines.

2 Extraction of obstacle areas

The raw dataset generated by a LiDAR system with a CCD camera contains raw point clouds, POS data, and images with camera parameters. To identify tall objects such as man-made buildings from the point clouds, the first step is filtering. Filtering is a process in which ground points are distinguished from non-ground points, after which the point clouds are filtered out leaving empty holes in the scene. We used inverse distance to a power (IDP) interpolation (Davis, 1986) to convert the irregular discrete ground points to a grid DTM with obstacle areas marked. With the help of the interior and exterior orientation elements of the CCD camera, aerial images can be registered to the same coordinate system as DTMs. Therefore, the obstacle areas marked on the DTM can be projected into the image space. In this way, the obstacle areas in the image corresponding to those in the DTM can be delineated. The procedure for obstacle area extraction from images using LiDAR point clouds is outlined in Fig. 1 and described in more detail below.

2.1 Point cloud filtering

LiDAR point cloud filtering is one of the key preprocessing steps in the application of LiDAR data. The primary objective of filtering is to filter out non-ground data from the raw dataset for DTM

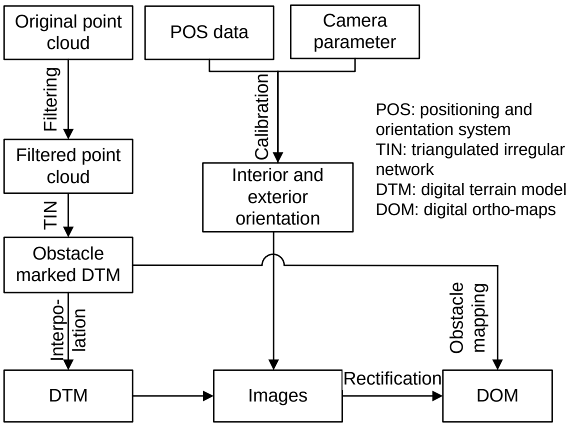

Fig. 1 Flowchart for obstacle area extraction

generation. It is also an essential step in identification and classification of other man-made objects.

Though an essential problem, LiDAR data filtering has remained an active research topic since the advent of LiDAR technology. Various methods exist both in the academic community and in the engineering field. Sithole and Vosselman (2004) have classified the existing methods into the following categories:

1. Surface based filters: The filters in this group start by either initially assuming that all points belong to the ground surface and then removing those points that do not fit, or rebuilding the ground by classifying more and more points as belonging to it. The former case is represented by the robust interpolation proposed by Kraus and Pfeifer (1998; 2001) and its extensions, while the typical method of the latter case is the triangulated irregular network (TIN) progressive densification proposed by Axelsson and some other extensions and variants (Axelsson, 1999).

2. Morphological filters: A structural element describing admissible height differences as a function of the horizontal distance is used in the erosion operation. The smaller the distance between a ground point and its neighboring points, the less the height difference accepted between them. The structural element is positioned at each candidate point and this point is identified as an off-terrain point if one or more height differences to its neighbors are above the admissible height difference (Zhang and Whitman, 2005).

3. Segmentation-based or clustering filters: This group of filters works on segments. The basic idea is that the processing of laser scanning point clouds can be strengthened by first aggregating information (i.e., building homogeneous segments) and then analyzing segments rather than individual points (Brovelli *et al.*, 2002). Segmentation can be performed either in object space or in feature space. Many segmentation methods have been borrowed from image processing and therefore, generally speaking, these filtering methods work on large entities, i.e., not on single points or pixels, and are less influenced by noise.

Note that none of these filters can use human-machine interactive software toolkits to avoid the need to visually distinguish ground points from non-ground points. As a result, more than 60% of the total time required for post-processing is spent on visual classification (Sithole and Vosselman, 2003).

In this study, TIN progressive densification filtering is used to filter raw point clouds automatically, and then visual classification is carried out using software toolkits developed by the authors.

Fig. 2 shows one scene of an intensity image of unclassified point clouds from Xi’an, China. Fig. 3 shows the filtered ground class image corresponding to Fig. 2. In Fig. 3, most non-ground points, such as those from vegetation, buildings, etc., have been removed and are displayed by the pseudo-color mode; that is, different classes correspond to different colors. Black regions are left by filtering point clouds out from high buildings or other man-made objects and therefore can be marked as obstacle areas.

Fig. 2 An intensity image of unclassified point clouds from Xi’an, China

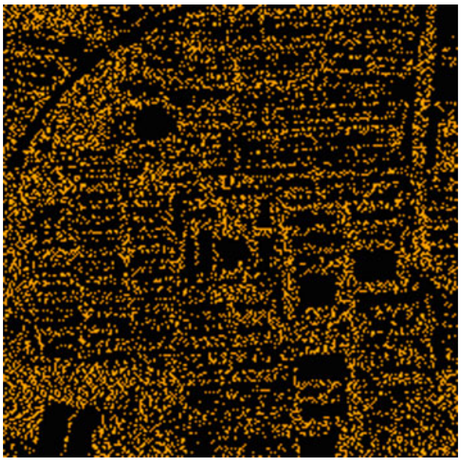

Fig. 3 Filtered ground class image corresponding to Fig. 2 (filter by class, remove non-ground points, and then display by class)

As LiDAR point clouds have an irregular distribution in nature, the ground points shown in Fig. 3 are rasterized by interpolation to generate the DTM, from which continuous edge lines of tall objects can be extracted (Priestnall *et al.*, 2000).

2.2 Water body removal and obstacle area delineation

Almost all the laser sources in a LiDAR system work in the near-infrared spectrum, which is mostly absorbed by water, leaving a water body looking like an empty hole in the point clouds. When this scenario occurs, the water bodies are often falsely marked as obstacle areas, which could affect seam-line optimization. Therefore, water bodies should be removed from the obstacle areas.

Since one of the key steps in DTM generation is gridding the point clouds, we first study the properties of the points within a grid mesh before interpolation. If these properties can be determined by the method described below that the points from a given grid mesh are reflected from a water body, then these points are labeled as water and the points in the grid are gridded and interpolated using IDP with a large searching radius to guarantee enough neighbors for interpolation. In our experiments, the size of the searching radius was 100 times that of the point spacing. Otherwise, the points in the grid are gridded and interpolated using IDP with a small searching radius, which is a little larger than the average point spacing.

Any point clouds reflected from water bodies have the following three properties: (1) the point density is low; (2) the elevation value remains almost the same over a water body; (3) the echo intensity is weak. These three properties are used as criteria for water body delineation.

1. Calculation of point density

Suppose $V=\{v_1, v_2, \ldots, v_N\}$ denotes the raw LiDAR dataset, where N is the number of points. $v(x, y, z)$ denotes a point with three coordinate values. $A=\{a_1, a_2, \ldots, a_M\}$ represents the dataset that has been divided into M blocks. Vb_i denotes all the points in the ith block. Counting the total number of points in a given block, the following criterion can then be used to determine whether the given block is sparse or not: if $\mathrm{Count}(v|v\in \mathrm{Vb}_i)<\alpha$, then the block is sparse, where α is a predefined threshold value set to a quarter of the grid area multiplied by the point density in the experiment.

2. Calculation of elevation similarity

Elevation similarity is calculated using the mean square error (MSE) of elevation values. Let D_i be the elevation MSE of block I; the block will be labeled as a water body while $D_i<\beta$, where β is the predefined threshold value for the determination of a water body, which was set to 0.01 in the experiments. A large β would have no effect on the constraint.

3. Calculation of echo intensity

Theoretically speaking, laser pulses will be absorbed by clear water due to the near-infrared property of a laser source. But echoes reflected from water bodies are often detected by a laser detector because of water waves or matter suspended in the water that can reflect part of the pulse energy. However, due to the weak reflectivity of water, the intensity values of the points that are reflected from water are much lower than those from land. The intensity values are converted into digital numbers (DN) (Fig. 4), and then the intensity information is displayed as a gray-scale image. A smaller DN represents weak reflection; therefore, the corresponding pixel can most probably be labeled as water. A predefined value is given and those points whose intensity values are less than the threshold value are labeled as water. In our experiment, the threshold was set to 20. A large value would increase the water area falsely.

Fig. 5 shows the workflow for the interpolation of point clouds to generate a grid DTM, with water bodies removed on the basis of the three properties of points reflected from water.

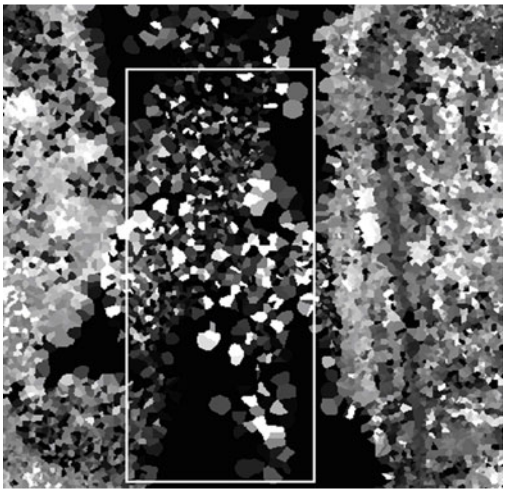

Fig. 4 Echo intensity in water

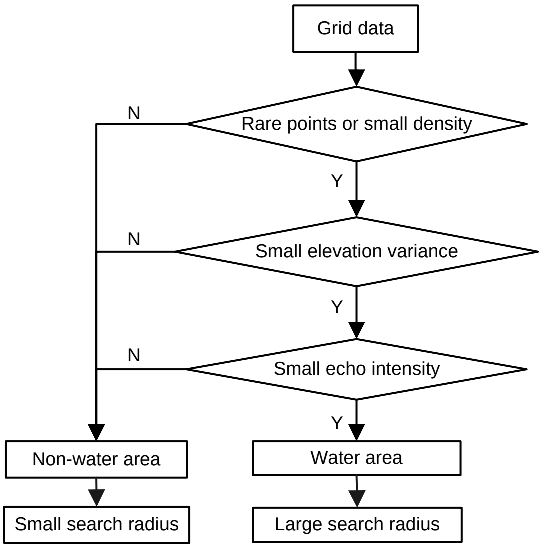

Fig. 5 Flow chart for grid interpolation

The workflow consists of the following steps. Firstly, the raw LiDAR dataset is partitioned into equal-sized grids. The size of the grid is determined by the requirement that in the calculation of elevation similarity mentioned above, there should be no fewer than 10 points in order to meet the statistical robustness required by the calculation of MSE. Taking this requirement into account, the side length of the grid is four times that of the point spacing. Then the K nearest points around the center of the grid are used to calculate the average elevation of the grid, followed by water body determination using the three criteria described above. If the grid corresponds to a water area, then interpolation is performed with a large enough search radius. Otherwise, a small search radius is used for interpolation.

Since the point clouds are discrete in nature and in most cases a point cannot hit on the edge of a ground object exactly, the resulting obstacle area may be a little bit smaller than what it should be. A binary morphological dilation operator is used to overcome this problem. In our experiment, the obstacle areas were dilated to a range equal to the average point spacing with a 3×3 square structure element.

Fig. 6 shows the interpolation result of point clouds data regardless of water areas. The water regions are falsely labeled as obstacle areas since in most regions there is a lack of return echoes, which is similar to obstacle areas caused by man-made objects. Fig. 7 shows the interpolation result using our algorithm with a larger search neighborhood after the water bodies were identified. The black parts represent ground objects and are labeled as obstacle areas. The next step is to project the obstacle areas from the DTM into the image space. This requires that the image be ortho-rectified, which is an easy task when the camera parameters and the DTM data covering the image region are given. The image pixels can be divided into walkable pixels and obstacle pixels after registration with the DTM (Fig. 8). Walkable pixels are those that can be reached from the start node of a seam-line, no matter what kind of path the seam-line passes through, while obstacle pixels are those that can not. Thus, the seam-line finding problem has now been converted to an optimal path planning problem.

Fig. 6 Interpolation regardless of water area

The water regions are falsely labeled as obstacle areas since in most regions there is a lack of return echoes, similar to obstacle areas caused by man-made objects

Fig. 7 Interpolation using our algorithm with a larger search neighborhood after the water bodies are identified
Black parts represent ground objects and are labeled as obstacle areas

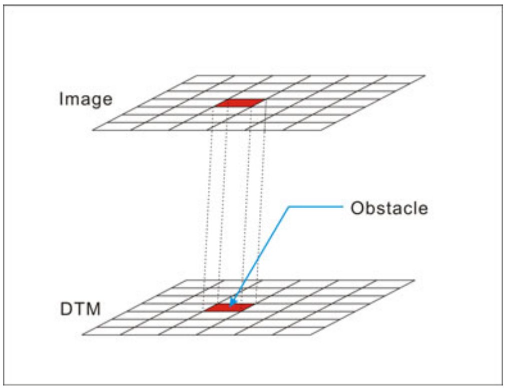

Fig. 8 Image obstacle area extraction from digital terrain model (DTM)

3 Improved A* seam-line path programming algorithm

Initial seam-lines are generated by simple methods before optimization is performed. These seam-lines are always straight lines, which cannot avoid obstacle areas automatically. In the following sections, we first introduce the generation of initial seam-lines. Then the standard A* algorithm is provided and an improved version to optimize each initial seam-line to satisfy the mosaicking is described in detail.

3.1 Initial seam-line generation

Since the quality of each orthophoto is always better in the vicinity of its nadir point, the central area of each orthophoto should be used as much as possible during the mosaicking process. A strategy to guarantee that each orthophoto contributes its central area as much as possible is to generate the initial seam-line from the perpendicular bisector of the two nadir points of each adjacent image pair. In the LiDAR system, the coordinate values of the nadir point of an image can be obtained from the POS system directly. Voronoi diagrams can be adopted to generate the initial seam-lines of the whole area (Aurenhammer, 1991).

Although the initial seam-line network can be easily generated, some seam-lines may cross ground objects. Such phenomena should be avoided if a visually high quality image is demanded. The initial seam-line network needs to be optimized following two principles:

Principle 1 The line should keep away from high ground objects.

Principle 2 The optimized path should be near to the initial seam-line.

An improved A* algorithm is proposed and used for the optimization problem. The details of the algorithm are presented in the next subsections.

3.2 Introduction of the A* algorithm

The A* algorithm (Hart *et al.*, 1968) is generally considered to be the best path finding algorithm. It differs from the depth-first search and breadth-first search algorithms in that a heuristic function relating to the problem domain is used to evaluate the cost at each step during the searching process (Cherkassky *et al.*, 1996).

The cost function of the A* algorithm can be expressed as

$$f(n)=g(n)+h(n), \tag{1}$$

where $f(n)$ is the evaluation function, $g(n)$ is the length of the shortest path from the start node to node n, and $h(n)$ is a heuristic function whose value equals the length of the shortest path from node n to the goal node.

The A* algorithm uses an open list and a closed list to maintain the nodes. Detailed steps of the algorithm are as follows, where F, G, and H are short for $f(n)$, $g(n)$, and $h(n)$, respectively (Lester, 2005).

1. Add the start node to the open list.

2. Repeat the following:

(1) Look for the lowest F cost node in the open list. It is referred to as the current node.

(2) Switch to the closed list.

(3) For each of the eight nodes adjacent to the current node,

i) If it is not walkable or if it is on the closed list, ignore it.

ii) If it is not on the open list, then add it to the open list, and make the current node the parent of this node. Record the F, G, and H costs of the node.

iii) If it is on the open list already, check to see if this path to that node is better, using G cost as the measure. A lower G cost means that this is a better path. If so, change the parent of the square to the current square, and recalculate the G and F scores of the square. If the open list is sorted by the F score, then resort it to account for the change.

(4) Stop when the target node is added to the closed list, in which case the path has been found, or if the target node is not found, and the open list is empty. In this case, there is no path.

3. Save the path. Working backwards from the target node, go from each node to its parent node until the start node is reached. That is the path.

3.3 An improved A* algorithm

After proposing the A* algorithm, many extensions and variants were developed, such as the extent A algorithm (Pohl, 1973), special A algorithm (Harris, 1973), B (Martelli, 1977), A_ε^* (Pearl and Kim, 1982; Pearl, 1984), A_ε (Gelperin, 1977; Ghallab and Allard, 1983), C (Bagchi and Mahanti, 1983), BF* (Dechter and Pearl, 1985; 1988), B′ (Mero, 1984), IDA* (Korf, 1988), D (Mahanti and Ray, 1988), A** (Dechter and Pearl, 1985; 1988), and the SDW algorithm (Koll and Kaindl, 1992). Each algorithm is problem-oriented and the traditional A* algorithm provides generic principles for algorithm improvement. In seam-line optimization, we need an improved A* algorithm, which is suitable for the two optimization principles.

Hart *et al.* (1968) have proved that the A* algorithm has the capacity for finding the shortest path between a given pair of start and end nodes. It is also suitable for finding an optimal seam-line on a raster image labeled with walkable regions and obstacle regions. In other words, it can satisfy Principle 1 in Section 3.1. The remaining issue is how to make the optimized seam-line as close to the initial seam-line as possible.

Aerial images are usually stored in raster format. In this context, the standard $g(n)$ and $h(n)$ can be calculated by the Manhattan distance, which moves only along the horizontal and vertical directions and ignores the obstacles. The Manhattan distances of $g(n)$ and $h(n)$ are defined as

$$g(n)=D\left(|n.x-\text{start}.x|+|n.y-\text{start}.y|\right), \quad (2)$$

$$h(n)=D\left(|n.x-\text{goal}.x|+|n.y-\text{goal}.y|\right), \quad (3)$$

where $(n.x, n.y)$ are the coordinates of point n on the grid, start and goal represent the start point and end point, respectively, and D is the size of the grid.

The key issue is choosing an appropriate cost function $f(n)$. Since in our research we aimed to find not only the shortest path but an optimal path, which can automatically bypass the obstacle regions and get as close as possible to the initial seam-line, a heuristic function $h(n)$ that can accomplish this aim was designed.

Assume that the center point of the 3×3 grid is the current node n, and that the right node (right) and the downside node (down) of n are two adjacent nodes relatively close to the target node in the open list (Fig. 9). $h_{\text{right}}(n)$ and $h_{\text{down}}(n)$ are defined as

$$h_{\text{right}}(n)=D\left(|n.(x+1)-\text{goal}.x|+|n.y-\text{goal}.y|\right), \quad (4)$$

$$h_{\text{down}}(n)=D\left(|n.x-\text{goal}.x|+|n.(y+1)-\text{goal}.y|\right). \quad (5)$$

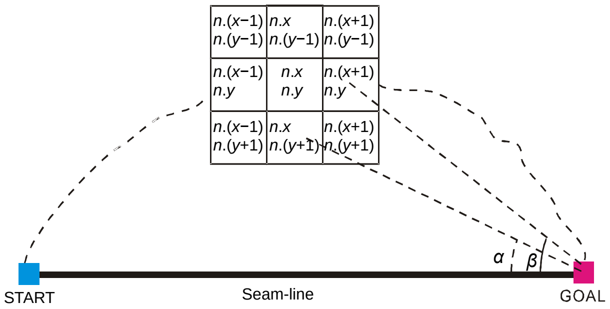

Fig. 9 State diagram of node *n*

In the context of a four-connected neighborhood raster dataset, one pixel downward movement from right and one pixel rightward movement from down will reach the same state, namely

$$h_{\text{right}}(n) = h_{\text{down}}(n). \tag{6}$$

Since right and down are both child nodes of *n*,

$$g_{\text{right}}(n) = g_{\text{down}}(n), \tag{7}$$

$$f_{\text{right}}(n) = f_{\text{down}}(n). \tag{8}$$

When the same two minimum values are found in the open list, the path finding method will try to extend to two different child nodes according to the A* algorithm described in Section 3.2. Since we need an optimized seam-line as close to the initial seam-line as possible, in Fig. 9, down is closer to the initial seam-line. This motivates us to introduce a restraint to the algorithm. Suppose the current path reaches point *n*. Then, with the ‘as close to the initial seam-line as possible’ principle, at the next node the path should extend down. For calculation simplicity, we modify *h*(*n*) to

$$\begin{aligned} h(n) = {} & D\left(|n.x - \text{goal}.x| + |n.y - \text{goal}.y|\right) \\ & + |\mathbf{vector}_1 \times \mathbf{vector}_2| \cdot \eta, \end{aligned} \tag{9}$$

where $|\mathbf{vector}_1 \times \mathbf{vector}_2| \cdot \eta$ is the additive distance variable. $\mathbf{vector}_1$ is the vector connecting node *n* to the target node:

$$\mathbf{vector}_{1\text{down}} = \left(n.x - \text{goal}.x,\ n.(y-1) - \text{goal}.y\right), \tag{10}$$

$$\mathbf{vector}_{1\text{right}} = \left(n.(x+1) - \text{goal}.x,\ n.y - \text{goal}.y\right). \tag{11}$$

$\mathbf{vector}_2$ is the vector connecting the start node to the goal node of the path:

$$\mathbf{vector}_2 = (\text{goal}.x - \text{start}.x,\ \text{goal}.y - \text{start}.y). \tag{12}$$

Because the magnitude of the cross product can be interpreted as the positive area of the parallelogram spanned by the two vectors, the following inequality always holds:

$$\left|\mathbf{vector}_{1\text{down}} \times \mathbf{vector}_2\right| < \left|\mathbf{vector}_{1\text{right}} \times \mathbf{vector}_2\right|. \tag{13}$$

According to Eqs. (1)–(3), (9), and (13), Eq. (14) can be deduced:

$$f_{\text{right}}(n) > f_{\text{down}}(n). \tag{14}$$

To make sure the restraint variable has a small effect on *f*(*n*), the influence factor of the distance variable η is usually set to a small value to guarantee $|\mathbf{vector}_1 \times \mathbf{vector}_2| \cdot \eta < 1$. In our experiments, the value of η was 0.00001.

4 Experiments and discussion

Experiment 1 To analyze the effect of different heuristic functions on path planning and to test our improved algorithm, a simulated image with 20×20 pixels was first used.

The left middle pixel is the start point while the right middle pixel is the end point (Fig. 10). The vertical black area in the middle represents obstacle areas. The initial seam-line is a straight line connecting the start and end nodes. Two optimized seam-lines are shown in Figs. 11 and 12.

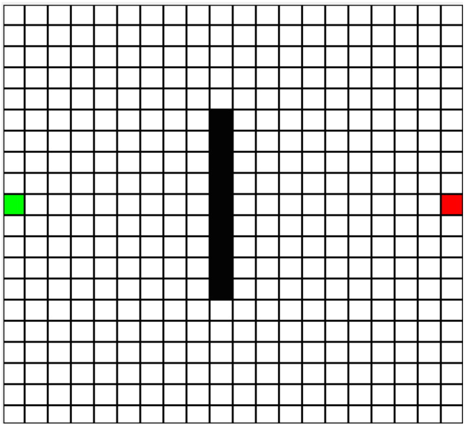

Fig. 10 20×20 path-finding simulation environment
White block is walkable, and black block is an obstacle; left middle pixel is the start point, and right middle pixel is the end point

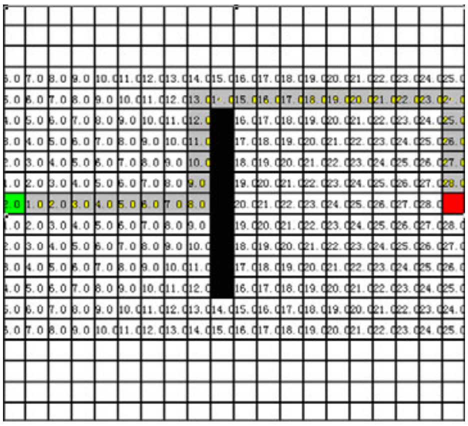

Fig. 11 The path found using the standard A* algorithm
White block is walkable, black block is an obstacle, and gray block is the path found

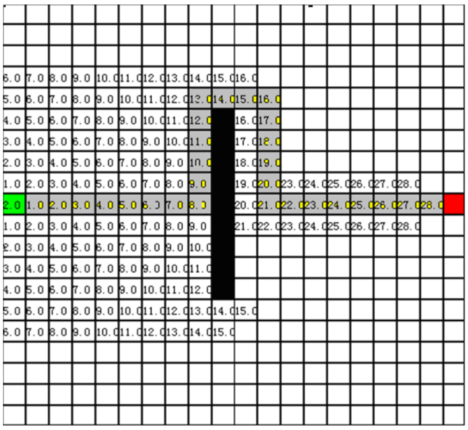

Fig. 12 The path found using the improved A* algorithm
White block is walkable, black block is an obstacle, and gray block is the path found

The gray pixels in Fig. 11 form the seam-line path generated by the standard A* algorithm. Though it keeps away from the obstacle areas intelligently and has the shortest path compared to others, this path does not get close enough to the initial seam-line. Thus, the standard A* algorithm can satisfy only the first principle for seam-line generation. In Fig. 12, the path labeled with gray pixels is generated by our method. The path not only bypasses the obstacle but also gets close enough to the initial seam-line. Thus, our method can meet the two principles for seam-line generation quite well.

Experiment 2 The dataset acquired by the small-footprint airborne LiDAR system, Leica50II (Leica, Swiss), was used. The characteristics of the dataset are listed in Table 1. We selected three datasets covering Xi'an, Dunhuang, and Changyang, all in Northwest China.

The LiDAR datasets from the real projects listed in Table 1 were used to verify our method. Dunhuang is a small town, which lacks high buildings and other infrastructures, while Xi'an is a metropolis full of high buildings and other man-made objects. Changyang is a mountain town, which has a complex terrain. Images over downtown areas of these cities were used for the experiment. Raw images need to be ortho-rectified before the optimization process is performed. Three ancillary datasets are necessary for image ortho-rectification: exterior and interior elements of each image and DTM data over the area corresponding to that covered by the image. Exterior and interior elements can be obtained from POS data and the camera calibration file which is provided by the camera manufacturer, while the DTM dataset is generated from corresponding LiDAR point clouds. In our proposed method, depending on the different interpolators used, two kinds of DTM datasets can be generated: one is obstacle-marked for seam-line optimization, the other is conventional for ortho-rectification.

Table 1 Characteristics of the dataset*

Parameter	Value/Description		
	Xi'an, China, 2009.5	Dunhuang, China, 2009.10	Changyang, China, 2008.9
Range (km×km)	4×8	5×10	6×9
Flying height (m)	1300	1300	1200
Laser point density (points/m^2)	1.1	1.2	1.2

* The following are the same for all the three cases: LiDAR type, Leica ALS50II; camera, Leica RCD105; image resolution, 0.31 m; scan angle, 60°; forward overlap, 70%; side overlap, 30%; overlap rate, 30%

The twin snake method was used as a conventional seam-line finding method, and its energy function was used for judging the effect of our method. Kerschner (2001) proposed a quantitative measure:

$$E_{\text{pho}} = aE_{\text{I}} + bE_{\text{H}} + cE_{\text{T}}, \tag{15}$$

$$E^* = \int_0^1 E_{\text{pho}}(v(s))\text{d}s, \tag{16}$$

where E_{pho} is the photometric energy of each pixel on the seam-line. It includes optimality criteria such as color similarity and texture similarity. E_{I} is the difference of intensity, E_{H} is the normalized difference of hue, and E_{T} is the difference of texture. a, b, and c are weighting coefficients. In the experiment, all three terms were weighted equally and E^* was calculated for each pixel $v(s)$ and integrated over the whole length s of the final seam-line. The E^* of the twin snakes' result was used as the optimality criterion, and its value was between 0 and 255. The smaller is the value, the greater is the similarity of the color and texture. Details of the calculation method of E_{pho} can be found in Kerschner (2001). An evaluation method based on the maximum and minimum E^* values obtained by the twin snake method is proposed:

$$\psi = \frac{E_{\text{w}}^* - E_{\text{p}}^*}{E_{\text{w}}^* - E_{\text{b}}^*}, \tag{17}$$

where ψ represents the optimality of the proposed method, $E_{\mathrm{w}}^{*}-E_{\mathrm{b}}^{*}$ is the effective range of E^{*}, E_{w}^{*} and E_{b}^{*} denote the maximum and minimum E^{*} values of the seam-line respectively, which represent the twin snakes' final optimization result, and E_{p}^{*} is the E^{*} value of the seam-line optimized by our proposed method.

Figs. 13–15 show the different image mosaicking results of Dunhuang city. Fig. 13 shows the initial seam-line of each image generated by the Voronoi diagram. It is obvious that the seam-lines cross through many man-made objects. Fig. 14 shows the optimized result from the twin snake method. Fig. 15 shows the result of our proposed method.

The threshold determination for the three criteria for water area removal can be summarized based on this experiment. A precision ratio R is defined to indicate the performance of water area removal:

$$R=\Delta S / S_{\text{image}}, \tag{18}$$

$$\Delta S=\max(S_{\text{point}}-S_{\text{non\_water}},\ 0), \tag{19}$$

where S_{image} denotes the area of the water body in the reference image, while S_{point} and $S_{\text{non\_water}}$ denote the areas of water and non-water bodies respectively, in the point clouds.

Based on the relationship between the precision ratio and the parameters in the three criteria for water body removal, the threshold determination of the parameters is as follows:

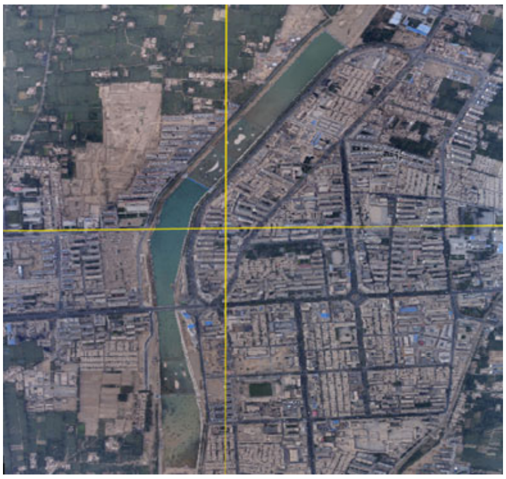

Fig. 13 Initial seam-lines generated by the Voronoi diagram (Dunhuang, China)

Fig. 14 Optimized result generated using the twin snake method (Dunhuang, China)

Fig. 15 Optimized result generated using our proposed method (Dunhuang, China)

1. The relationship between the precision ratio and α in criterion 1 (Section 2.2) is shown in Fig. 16, where N denotes the blocks' average number of points. Clearly, the optimal value of α should be a quarter of the blocks' average number of points. A suboptimal value would decrease the precision ratio.

2. The optimal value of β in criterion 2 (Section 2.2) can be determined in the same manner according to the relationship between the precision ratio and β (Fig. 17). A suboptimal value could decrease the precision ratio. In this particular case, the optimal β was 0.01.

3. Likewise, the threshold in criterion (3) can be determined by the relationship between the precision ratio and the gray threshold (Fig. 18). In our experiment, it was 20.

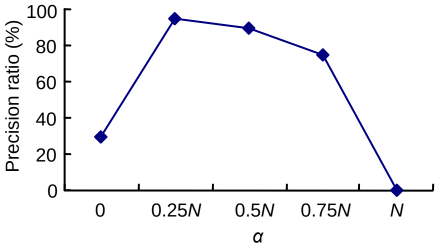

Fig. 16 Relationship between the precision ratio and predefined threshold *α* that determines whether the grid data are rare or of small density

N denotes the blocks' average number of points

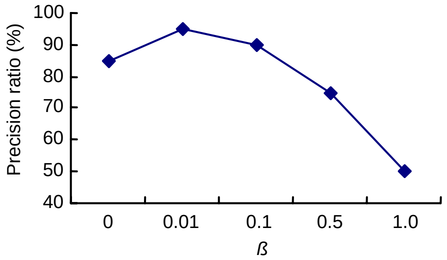

Fig. 17 Relationship between the precision ratio and predefined threshold *β* that determines the elevation MSE

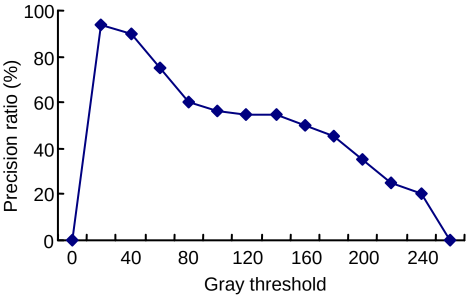

Fig. 18 Relationship between the precision ratio and the gray threshold

Images from Xi'an were used to test our method when applied to a large volume dataset. Fig. 19a is the mosaicking result from 21 aerial images. The black lines are initial seam-lines. Fig. 19b is the zoom-in display of the box area in Fig. 19a, showing many buildings are crossed through by seam-lines.

Fig. 20a is the mosaicking result after each seam-line has been optimized using our proposed method. Fig. 20b is the zoom-in display of the box area in Fig. 20a, showing the expected result that the seam-lines bypass obstacles and get close enough to the initial seam-lines.

Finally, images from Changyang were used to verify the proposed method in a complex terrain condition. The top and bottom of the image represent hilly areas while the middle is a residential area (Fig. 21). The initial seam-line crosses through trees and buildings. Fig. 22 is the result obtained using our proposed method. The seam-line bypassed most of the trees and buildings.

Fig. 19 Initial seam-lines (black lines) (Xi'an, China)

(a) Mosaicking result from 21 aerial images; (b) Zoom-in display of the box area in (a)

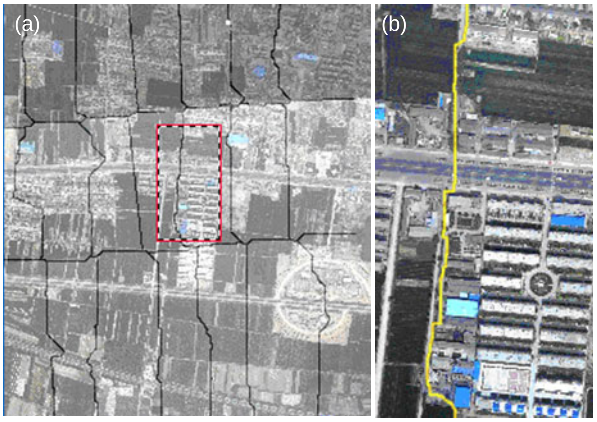

Fig. 20 Result obtained using our proposed method (Xi'an, China)

(a) Mosaicking result after each seam-line has been optimized using our proposed method; (b) Zoom-in display of the box area in (a)

A quantitative method is proposed to compare the differences between the initial seam-lines and the seam-lines generated by the proposed method and the twin snake method. Table 2 shows the computational results. Compared with the initial result, the E^* of our proposed method shows improvements of 9.4%, 8.7%, and 9.8% in Dunhuang, Xi'an, and Changyang respectively, which confirms the effectiveness of our method. It outperforms the twin snake approach in terms of computational speed. However, the E^* is a little larger than that of the twin snake method, which

Fig. 21 Initial seam-line in a complex terrain condition (Changyang, China)
Top and bottom are hilly areas; middle is a residential area

Fig. 22 Result obtained using our proposed method (Changyang, China)

means that our proposed method is suboptimal in terms of color balance and texture continuity. Nevertheless, it is found to be visually acceptable, and there are no obvious discontinuities in Figs. 15, 20, or 22 seen with the naked eye. The number of crossed man-made objects decreases from 35, 126, and 8 to 10, 31, and 1 in Dunhuang, Xi'an, and Changyang respectively when our proposed method is used. When a roof's color, or the texture of a man-made object is similar to that of the vicinity and its relief displacement is small, the seam-line will cross a man-made object when using the twin snake approach, while, using our proposed method, most of the man-made objects were bypassed if they were marked on the DTM. The experiment proved that our proposed method can effectively optimize the initial seam-line.

Table 2 Comparison of optimality between different methods

	E^*			Number*			Time (s)		
	DH	XA	CY	DH	XA	CY	DH	XA	CY
Initial	37.5	41.2	36.5	35	126	8			
E_w^*	103.6	117.3	109.5						
E_b^*	25.7	32.4	26.7	17	47	4	271	7365	135
Ours	30.1	33.8	28.4	10	31	1	56	97	40
ψ	9.4%	8.7%	9.8%						

* Number of man-made objects crossed by seam-lines. DH=Dunhuang, XA=Xi'an, CY=Changyang. E^* is the integrated photometric energy of each pixel on the seam-line. The photometric energy includes optimality criteria such as color similarity and texture similarity. E_w^* and E_b^* are the maximum and minimum E^* values of the seam-line, respectively. ψ is the optimality of the proposed method

5 Conclusions

An improved A* algorithm combining LiDAR point clouds for orthophoto seam-line optimization is proposed in this paper. The method is able to bypass the obstacle areas intelligently and the resulting path can get close enough to the initial seam-lines. It is based on accurate registration between images and DTM data, which is actually an easy task since both the image and DTM dataset will have been accurately geo-referenced when an airborne LiDAR system is considered for the raw dataset acquisition. The image resulting from the use of our method has high visual quality.

Although our method can bypass obstacle areas intelligently and successfully, some key points still require attention. First, there are not yet any automatic algorithms that can filter the non-ground points completely, especially when the terrain is complex. While in our proposed method, the extraction of obstacle areas is effected by the precision of point cloud filtering, and manual assistance is sometimes needed. Second, exterior and interior elements of an image and camera parameters need to be accurately calibrated to obtain a high quality registration result between images and DTM data.

References

Aurenhammer, F., 1991. Voronoi diagrams: a survey of a fundamental geometric data structure. *ACM Comput. Surv.*, **23**(3):345-405. [doi:10.1145/116873.116880]

Axelsson, P., 1999. Processing of laser scanner data algorithms and applications. *ISPRS J. Photogr. Remote Sens.*, **54**(2-3): 138-147. [doi:10.1016/S0924-2716(99)00008-8]

Bagchi, A., Mahanti, A., 1983. Search algorithms under different kinds of heuristics—a comparative study. *J. ACM*, **30**(1):1-27. [doi:10.1145/322358.322359]

Baltsavias, E.P., 1999a. Airborne laser scanning: existing systems and orther resources. *ISPRS J. Photogr. Remote Sens.*, **54**(2-3):164-198. [doi:10.1016/S0924-2716(99)00016-7]

Baltsavias, E.P., 1999b. A comparison between photogrammetry and laser scanning. *ISPRS J. Photogr. Remote Sens.*, **54**(2-3):83-94. [doi:10.1016/S0924-2716(99)00014-3]

Brovelli, M.A., Cannata, M., Longoni, U.M., 2002. Managing and Processing LiDAR Data within GRASS. Proc. Open Source GIS-GRASS Users Conf., unpaginated CD-ROM.

Cherkassky, B.V., Goldberg, A.V., Radzik, T., 1996. Shortest paths algorithms: theory and experimental evaluation. *Math. Program.*, **73**(2):129-174. [doi:10.1007/BF02592101]

Chon, J., Kim, H., 2006. Determination of the optimal seam-lines in image mosaicking with the dynamic programming (DP) on the converted cost space. *LNCS*, **4029**:750-757.

Davis, J., 1998. Mosaics of Scenes with Moving Objects. Proc. IEEE Computer Society Conf. on Computer Vision and Pattern Recognition, p.354-360. [doi:10.1109/CVPR.1998.698630]

Davis, J.C., 1986. Statistics and Data Analysis in Geology. Wiley, New York, p.656.

Dechter, R., Pearl, J., 1985. Generalized best-first search strategies and the optimality of A*. *J. ACM*, **32**(3):505-536. [doi:10.1145/3828.3830]

Dechter, R., Pearl, J., 1988. The Optimality of A*. *In*: Kanal, L., Kumar, V. (Eds.), Search in Artificial Intelligence. Springer-Verlag, p.166-199.

Fernandez, E., Marti, R., 1999. GRASP for seam drawing in mosaicking of aerial photographic maps. *J. Heurist.*, **5**(2):181-197. [doi:10.1023/A:1009633811636]

Gelperin, D., 1977. On the optimality of A*. *Artif. Intell.*, **8**(1):69-76. [doi:10.1016/0004-3702(77)90005-4]

Ghallab, M., Allard, D., 1983. A_ϵ—an Efficient near Admissible Heuristic Search Algorithm. Proc. 8th Int. Joint Conf. on Artficial Intelligence, **2**:789-791.

Harris, L.R., 1973. The Bandwith Heuristic Search. Proc. 3rd Int. Joint Conf. on Artificial Intelligence, p.23-29.

Hart, P.E., Nilsson, N., Raphael, B., 1968. A formal basis for the heuristic determination of minimum cost paths. *IEEE Trans. Syst. Sci. Cybern.*, **4**(2):100-107. [doi:10.1109/TSSC.1968.300136]

Kerschner, M., 2001. Seam-line detection in colour orthophoto mosaicking by use of twin snakes. *ISPRS J. Photogr. Remote Sens.*, **56**(1):53-64. [doi:10.1016/S0924-2716(01)00033-8]

Koll, A., Kaindl, H., 1992. A New Approach to Dynamic Weighting. Proc. 10th European Conf. on Artificial Intelligence, p.16-17.

Korf, R.E., 1988. Optimal Path-Finding Algorithms. *In*: Kanal, L., Kumar, D. (Eds.), Search in Artificial Intelligence. Springer-Verlag, p.223-267.

Kraus, K., Pfeifer, N., 1998. Determination of terrain models in wooded areas with airborne laser scanner data. *ISPRS J. Photogr. Remote Sens.*, **53**(4):193-203. [doi:10.1016/S0924-2716(98)00009-4]

Kraus, K., Pfeifer, N., 2001. Advanced DTM generation from LiDAR data. *Int. Arch. Photogr. Remote Sens. Spat. Inform. Sci.*, **34**(3/W4):23-30.

Lester, P., 2005. A* Path Finding for Beginners. Available from http://www.policyalmanac.org/games/aStarTutorial.htm [Accessed on Sept., 2010].

Mahanti, A., Ray, K., 1988. Network Search Algorithms with Modifiable Heuristics. *In*: Kanal, L., Kumar, D. (Eds.), Search in Artificial Intelligence. Springer-Verlag, p.200-222.

Martelli, A., 1977. On the complexity of admissible search algorithms. *Artif. Intell.*, **8**(1):1-13. [doi:10.1016/0004-3702(77)90002-9]

Mero, L., 1984. A heuristic search algorithm with modifiable estimate. *Artif. Intell.*, **23**(1):13-27. [doi:10.1016/0004-3702(84)90003-1]

Pearl, J., 1984. Heuristics: Intelligent Search Strategies for Computer Problem Solving. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

Pearl, J., Kim, J.H., 1982. Studies in semi-admissible heuristics. *IEEE Trans. PAMI*, **4**(4):392-400.

Pohl, I., 1973. The Avoidance of (Relative) Catastrophe, Heuristic Competence, Genuine Dynamic Weighting and Computational Issues in Heuristic Problem Solving. Proc. 3rd Int. Joint Conf. on Artificial Intelligence, p.12-17.

Priestnall, G., Jaafar, J., Duncan, A., 2000. Extracting urban features from LiDAR digital surface models. *Comput. Envir. Urban Syst.*, **24**(2):65-78. [doi:10.1016/S0198-9715(99)00047-2]

Schickler, W., Thorpe, A., 1998. Operational Procedure for Automatic True Orthophoto Generation. Int. Archives of Photogrammetry and Remote Sensing, p.527-532.

Shiren, Y., Li, L., Peng, G., 1989. Two-dimensional seam-point searching in digital image mosaicking. *Photogr. Eng. Remote Sens.*, **55**(1):49-53.

Sithole, G., Vosselman, G., 2003. Automatic Structure Detection in a Point-Cloud of an Urban Landscape. Proc. 2nd GRSS/ISPRS Joint Workshop on Remote Sensing and Data Fusion over Urban Areas, p.67-71. [doi:10.1109/DFUA.2003.1219959]

Sithole, G., Vosselman, G., 2004. Experimental comparison of filter algorithms for bare-Earth extraction from airborne laser scanning point clouds. *ISPRS J. Photogr. Remote Sens.*, **59**(1-2):85-101. [doi:10.1016/j.isprsjprs.2004.05.004]

Zhang, K., Whitman, D., 2005. Comparison of three algorithms for filtering airborne LiDAR data. *Photogr. Eng. Remote Sens.*, **71**(3):313-324.