ISPRS Journal of Photogrammetry and Remote Sensing 125 (2017) 207–224

Contents lists available at ScienceDirect

ISPRS Journal of Photogrammetry and Remote Sensing

journal homepage: www.elsevier.com/locate/isprsjprs

Using vector building maps to aid in generating seams for low-attitude aerial orthoimage mosaicking: Advantages in avoiding the crossing of buildings

Dongliang Wang [a,b], Wei Cao [a,*], Xiaoping Xin [b,*], Quanqin Shao [a,*], Matthew Brolly [c], Jianhua Xiao [d], Youchuan Wan [e], Yongjun Zhang [e]

[a] Key Laboratory of Land Surface Pattern and Simulation, Institute of Geographic Sciences and Natural Resources Research, Chinese Academy of Science, Beijing 100101, China
[b] National Hulunber Grassland Ecosystem Observation and Research Station, Institute of Agricultural Resources and Regional Planning, Chinese Academy of Agricultural Sciences, Beijing 100081, China
[c] School of Environment and Technology, University of Brighton, Brighton BN2 4GJ, UK
[d] Wuhan Surveying and Mapping Institute, Wuhan 430022, China
[e] School of Remote Sensing and Information Engineering, Wuhan University, Wuhan 430079, China

ARTICLE INFO

Article history:
Received 19 November 2015
Received in revised form 31 December 2016
Accepted 23 January 2017

Keywords:
Aerial image mosaicking
Vector building map
Seam detection

ABSTRACT

A novel seam detection approach based on vector building maps is presented for low-attitude aerial orthoimage mosaicking. The approach tracks the centerlines between vector buildings to generate the candidate seams that avoid crossing buildings existing in maps. The candidate seams are then refined by considering their surrounding pixels to minimize the visual transition between the images to be mosaicked. After the refinement of the candidate seams, the final seams further bypass most of the buildings that are not updated into vector maps. Finally, three groups of aerial imagery from different urban densities are employed to test the proposed approach. The experimental results illustrate the advantages of the proposed approach in avoiding the crossing of buildings. The computational efficiency of the proposed approach is also significantly higher than that of Dijkstra's algorithm.

1. Introduction

Orthoimages have increasingly become a popular visualization product and planning tool for integrating the rich information content of images with geometric properties of maps (ground projection). An advantage is that they can be easily combined and employed within a geographic information system (Kerschner, 2001). However, the coverage area of an individual orthoimage is typically very small; thus, it is necessary to mosaic several individuals to combine and create larger images to cover greater geographic regions. This process is implemented in many applications, e.g., environmental monitoring and disaster management (Díaz-Varela et al., 2015; Li and Shao, 2014; Pan et al., 2014a). For mosaicking orthoimages (referred to as "images" from here on), the seam-based mosaicking method is commonly used (Ai et al., 2011; Mills and McLeod, 2013; Pan et al., 2009, 2014b; Soille, 2006; Wang et al., 2016). In this method, the seams between the images to be mosaicked must be determined in such a way as to minimize the visual transition. This includes consideration of color and geometric characteristic transition, from one image to the next, before combining. In this paper, we concentrate on determining seams to minimize geometric characteristic transition.

Most existing seam detection methods are based on raster data to prevent seams from crossing high difference areas, where geometric, radiometric, and color characteristics may significantly vary for some objects (Choi et al., 2015). Examples of these are high-rise buildings. Many scholars have also presented automatic methods for selecting seams, which can then be checked using interactive, computer-assisted methods. Milgram (1977) defined the "best" seam as that which automatically minimizes visual

* Corresponding authors at: A11 Datun Road, Chaoyang District, Beijing 100101, China.
E-mail addresses: wangdongliang@igsnrr.ac.cn (D. Wang), caowei@igsnrr.ac.cn (W. Cao), xinxiaoping@caas.cn (X. Xin), shaoqq@igsnrr.ac.cn (Q. Shao), M.Brolly@brighton.ac.uk (M. Brolly), xjhwhkc@163.com (J. Xiao), zhangyj@whu.edu.cn (Y. Zhang).

http://dx.doi.org/10.1016/j.isprsjprs.2017.01.015
0924-2716/

confusion and employed an image of the absolute difference between two overlapping image windows to determine the least-weight path. Soille (2006) proposed a morphological image compositing algorithm. The scope of this algorithm is to position seams along salient image structures to diminish their visibility in the output mosaic even in the absence of radiometric corrections. Kerschner (2001) proposed a method called the "twin snake technique" for the determination of the optimal seam. This algorithm uses two lines starting from the opposite borders on the overlapping area. The two lines attract each other, with the optimal seam determined when the two lines merge. Chon et al. (2010) presented a method limiting the level of maximum difference in the seam selection process using normalized cross-correlation. They first determined this desired level of maximum difference and then applied Dijkstra's algorithm to find the optimal seam. Chen et al. (2014) proposed a method guiding seams toward low areas based on the digital surface model (DSM). This method generates an initial path network by using a morphological algorithm to process the orthoimage elevation synchronous model (OESM) data and then uses Dijkstra's algorithm to determine the least-cost path from the initial network. The performance of this method depends on the accuracy of DSM. Pan et al. (2015) presented a seam determination method based on region change rate (RCR). The RCR is defined as the percentage of changed pixels in the segmented region. The seams are then designed to pass through the connected regions with minimized maximum RCR value. Pang et al. (2016) proposed a semi-global matching (SGM)-based method to guide seam determination. In this method, the Hilditch thinning algorithm is used to generate the skeleton line of the non-obstacle regions. Dijkstra's algorithm is then used to determine the optimal path on the skeleton network. Lin et al. (2016) used a semi-optimal blending zone instead of a seamline for image patch stitching and color blending. This method efficiently eases pixel mismatch and color discontinuity problems.

The methods outlined above are raster-based methods and are commonly time-consuming and ineffective due to the large size of the aerial images to be mosaicked. In scenarios such as this, it becomes impossible to recognize and avoid the crossing of some high difference objects, such as buildings. The areas of high mismatch are typically related to the objects that are not contained in the DEM and thus the DEM cannot be used to rectify correctly (Chon et al., 2010). Compared to the high difference areas recognized from images, vector surveying and mapping data, including various man-made or natural surface features, such as roads, buildings and mountains, are more accurate and easier to follow for urban areas. To avoid recognizing high difference objects for image mosaicking, Wan et al. (2013) proposed a vector-road-based method using vector roads alone to generate candidate seams. This method of tracking vector roads within the overlapping area avoids the recognition of roads from images and thus proves extremely effective, particularly when vector roads are available. However, the vector-road-based method has its limitations. In some urban or rural areas, roads are sparse or unavailable. Conversely, in many of these cases, vector buildings are available. To extend the application range of the vector-based seam detection method, this paper proposes an approach using vector building maps to generate candidate seams. Contrary to the vector-road-based method, which tracks vector roads to generate seams, the proposed vector-building-based method is designed to prevent the extracted seams from crossing vector buildings.

The structure of the remainder of this paper is as follows. The definitions for generating candidate seams and the proposed method for generation of optimal seams using vector building maps are given in Sections 2 and 3, respectively; experimental results and summary are reported and discussed in Section 4. Concluding remarks are presented in Section 5.

2. Definitions for generating candidate seams

An ideal seam between adjacent images is one where the geometric and color characteristics of each pixel pair from adjacent images are the same. Achieving the ideal seam may be impossible, but the geometric and color differences can at least be minimized (Wan et al., 2013). Generally, buildings cannot be rectified correctly and thus appear as "high difference areas" in adjacent orthoimages. It is natural that candidate seams should be designed to bypass the buildings where the geometric differences are lower, although color differences may still exist. The feathering operation (Chon et al., 2010) can then be applied along the seams to achieve color balance after the geometric alignment but lies beyond the scope of this study. A feasible approach to achieving that purpose is to follow the centerlines between buildings.

Based on this idea, we propose that candidate seams preferentially follow the occlusion-free land centerlines, where the building interspaces are wider and the building heights are lower. In some areas where no other occlusion-free land centerlines exist, candidate seams are designed to follow the straight skeleton of the overlapping area between the adjacent images to be mosaicked. This is the worst-case scenario, as some high difference areas may be crossed. Fig. 1 shows a simple diagram for determining candidate seams from vector buildings. Fig. 1(a) shows the potential subpaths, consisting of occlusion-free land centerlines and the straight skeleton of the overlapping area. The thickness of the solid lines represents the valid widths of potential subpaths. Thicker solid lines indicate lower passing cost (weight) of potential subpaths. The widest occlusion-free land centerlines, shown as the thickest solid lines, have the greatest valid widths. However, skeleton segments, shown as the dotted lines, have the minimum valid widths. The centerlines between buildings can be considered equivalent to the street centerlines. The straight skeleton of the overlapping area is a method of creating some self-connected straight-line segments within the overlapping area to connect some scattered paths for constructing candidate seams. A detailed introduction and the generation of the potential subpaths, including straight skeletons and centerlines, are discussed in Section 3. The candidate seam with the lowest weight is determined from the potential subpaths, as shown in Fig. 1(b). The candidate seam tracks two skeleton segments and three centerline segments.

To ensure that the potential subpaths follow the wider centerlines, we need to obtain the valid widths of potential subpaths. These potential subpaths include the centerlines and the skeleton of the overlapping area, which in turn are used to define functions to describe the weights of potential subpaths. This ensures that the lowest-weight candidate seams can be determined using Dijkstra's algorithm. Definitions used to determine candidate seams are given as follows.

2.1. Valid widths of potential subpaths

The valid width of a potential subpath is defined as the width of the region unshaded by its surrounding buildings. Generally, the occlusion distance or projection difference of a building in an image is proportional to the height of the building and the distance from the building to the principal point of the image. The valid width can be obtained through subtracting the total projection difference of the surrounding buildings from the minimal distance between the surrounding buildings. More than two buildings may affect the potential subpath. To minimize the complexity, we assume that only two surrounding buildings affect the potential subpath. Readers can derive functions to satisfy more complex cases, but this is less important because the impact radius of a building is typically limited. For example, for a 2-km long image

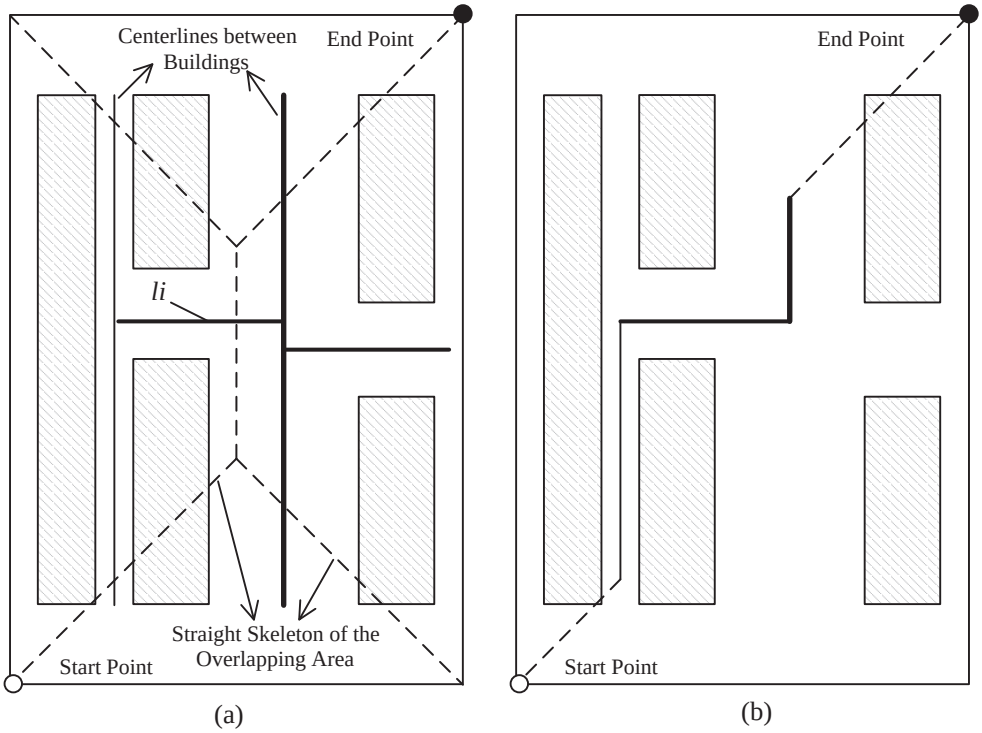

Fig. 1. Diagram for determining candidate seams from vector buildings: (a) potential subpaths consisting of centerlines between buildings (the solid lines) and the straight skeleton (the dotted lines) of the overlapping area. (b) Candidate seam determined from potential subpaths, consisting of two skeleton segments and three centerline segments. The outer rectangle indicates the overlapping area of adjacent images. The small rectangles filled with downward diagonal lines denote buildings. l_i denotes the ith potential subpath. Start and end points denote the start and end points of the candidate seam.

captured at a height of 2 km, a 100-m high building has a maximum projection difference of 50 m. In this scenario, this building may affect the potential subpaths within 50 m, which represents the same impact as a 10-m high building to the potential subpath within 5 m. Therefore, the effects of the nonadjacent buildings can be ignored.

Fig. 2 shows the geometry of photographic imaging. In the figure, l_i denotes the ith potential subpath (the centerline segment between buildings 1 and 2); $d(l_i)$ denotes the minimal distance between buildings 1 and 2; $vadW(l_i)$ denotes the valid width of the occlusion-free area; O_1 and O_2 denote the principal points of the left image and the right image, respectively; h_1 and h_2 denote the heights of buildings 1 and 2, respectively. The valid width of the potential subpath l_i can be expressed as:

$$vadW(l_i) = \begin{cases} 0, & \text{if } l_i \text{ is a skeleton segment} \\ d(l_i) - \text{projD}(h_1, h_2), & \text{if } l_i \text{ is a centerline} \end{cases} \quad (1)$$

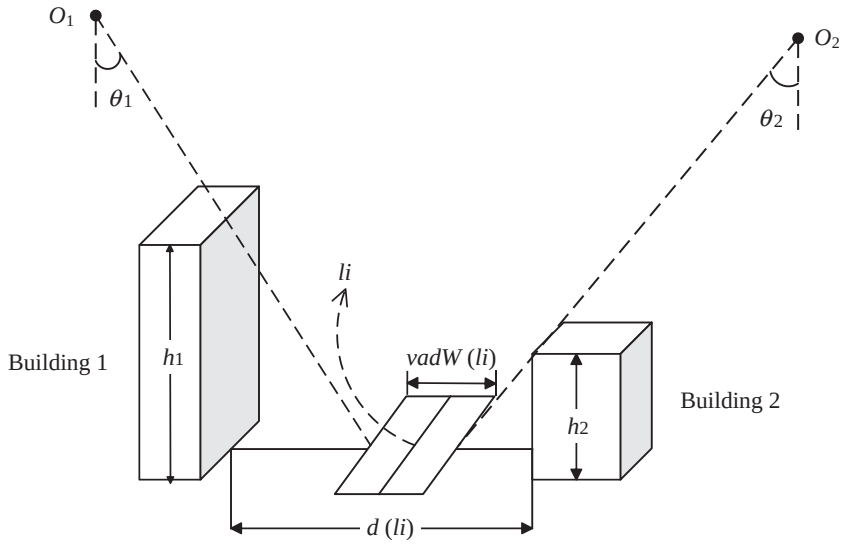

Fig. 2. Geometry of photographic imaging. l_i denotes the ith potential subpath. O_1 and O_2 denote the principal points of the left image and the right image, respectively. The small diamond area between buildings 1 and 2 denotes the occlusion-free area.

In the equation, the valid widths of skeleton segments are set as zero because these segments typically cut surrounding buildings. Those of centerlines are set as the difference between $d(l_i)$ and $\mathrm{projD}(h_1, h_2)$. $\mathrm{projD}(h_1, h_2)$ is the sum of projection differences of buildings 1 and 2. Then,

$$\mathrm{projD}(h_1, h_2) = \tan\theta_1 * h_1(l_i) + \tan\theta_2 * h_2(l_i) \quad (2)$$

In Eq. (2), θ_1 and θ_2 are the look angles at buildings 1 and 2, respectively; $\tan\theta_1 * h_1(l_i)$ and $\tan\theta_2 * h_2(l_i)$ are the effective projection differences of buildings 1 and 2, respectively.

Eq. (1) shows that when $\mathrm{projD}(h_1, h_2) \geqslant d(l_i)$, valid widths of centerlines between adjacent buildings $vadW(l_i) \leqslant 0$. This means that the centerlines with valid widths $\leqslant 0$ m are not defined as potential seam paths and must be removed from the potential set of subpaths before seam determination.

2.2. Weights of potential subpaths

After defining the valid widths of potential subpaths, the weight function of each potential subpath must be built to allow the use of Dijkstra's algorithm. This algorithm will then find the lowest-weight seam from these potential subpaths. Dijkstra's algorithm requires that: (i) the path graph is connected and (ii) the path weights are strictly positive. Dijkstra's algorithm can then be used to find the lowest-weight paths between nodes in the graph. Therefore, the path weights must be positive. Moreover, the weight function must decrease with path width because priority must be given to tracking wide occlusion-free land centerlines over narrow ones. The requirements can ensure that (i) the centerlines with greater valid widths have lower weights and (ii) the weight function can distinguish potential subpaths in valid widths. Otherwise, Dijkstra's algorithm cannot follow wider occlusion-free land centerlines.

As an example, suppose that M potential subpaths form the potential subpath set. The weight of tracking the potential subpath l_i is written as:

$$Tcost(l_i) = \kappa(l_i) \times len(l_i) \quad (3)$$

where $\kappa(l_i)$ indicates the contribution of the width information to the weight and $len(l_i)$ is the length of the potential subpath l_i, which represents the contribution of the length information to the weight. $\kappa(l_i)$ is expressed as:

$$\kappa(l_i) = \begin{cases} \frac{1}{c \times minW}, & \text{if } l_i \text{ is a skeleton segment} \\ \frac{1}{vadW(l_i)}, & \text{if } l_i \text{ is a centerline} \end{cases} \quad (4)$$

In Eq. (4), if the potential subpath l_i is a centerline, $\kappa(l_i)$ is set as the reciprocal of the valid width of the centerline $vadW(l_i)$; if the potential subpath l_i is a skeleton segment, $\kappa(l_i)$ is set as the reciprocal of $(c \times minW)$. $minW$ is defined as the minimal valid width of all centerlines (excluding skeleton segments). c is a positive constant. If these conditions are not adhered to, $\kappa(l_i)$ is not a legal expression. To ensure that all centerlines have lower weights than those of skeleton segments, c must be less than 1. Thus, c can be any real number between 0 and 1. An experimental analysis of the relationship between c and the quality of seams given in Section 4.1 shows that when $0 < c < 1$, the change of c has no influence on the quality of seams. To simplify the definition of path weights, c is fixed as

$$c = 0.1 \quad (5)$$

This assigned value indicates that tracking of a skeleton segment pays ten times the weight of the narrowest centerline. In this situation, it is highly unlikely for the candidate seam to track a skeleton segment when other occlusion-free land centerlines exist.

In Eq. (3), the potential subpaths with great valid widths are assigned low weights. On the one hand, the weight function meets the requirements of Dijkstra's algorithm, as mentioned above. On the other hand, the weight function considers the contributions of the interspaces between buildings, the heights of the buildings, the distances between buildings to the principle points, and the path lengths. In Eq. (4), the reciprocal function is used to construct the weight of the width information to the weight. The result of this is that tracking of narrow potential subpaths must pay more weight than wide ones. This weight assignment is expected to ensure that almost all seams follow wide occlusion-free land centerlines. For example, if the valid width of a centerline that buildings do not shade is not greater than 0 m, the centerline is then removed from the potential subpath set. This condition means that the centerline is not considered a potential seam path. For a centerline l_i, if $vadW(l_i)$ is 100 m (suppose that this is the widest centerline), then $Tcost(c, l_i)$ is set as $0.01 * len(l_i)$. If $vadW(l_i)$ is 1 m (suppose that this is the narrowest centerline), then $Tcost(c, l_i)$ is set as $1 * len(l_i)$. However, if the potential subpath is a skeleton segment, $Tcost(c, l_i)$ is set as $10 * len(l_i)$, which means that tracking a skeleton segment must pay 10 times the weight of the narrowest centerline or 1000 times the weight of the widest centerline.

3. Methodology for generating optimal seams

Based on the candidate seam definition described in Section 2, a vector-building-based approach for generation of the optimal seams from building maps is now proposed. Typically, the individual seams between two or more adjacent images must be determined and optimized before mosaicking (Pan et al., 2009; Wan et al., 2013). As an example and to minimize complexity, extraction of the seam of a pair of aerial orthoimages is described. These two images and the corresponding vector building map are part of the first group of test data. Detailed information regarding the data can be found in Section 4.

The work flow is presented in Fig. 3. First, the overlapping region of adjacent images is extracted, followed by the second step of extraction of the straight skeleton of the overlapping area of adjacent images. Third, the centerlines between the buildings within the overlapping region of adjacent images are extracted based on the constrained Delaunay triangulation skeletonization algorithm (Morrison and Zou, 2007; Nguyen Minh et al., 2009; Tang and You, 2003). Fourth, the centerlines between the buildings are overlaid with the skeleton of the overlapping region to build a weighted graph. Dijkstra's algorithm (Dijkstra, 1959) is then applied to find the lowest-weight candidate seam in the weighted graph. The lowest-weight path is considered a candidate seam. Finally, the candidate seam is refined using its surrounding pixels and the raster-based Dijkstra's algorithm (Chon et al., 2010; Pan et al., 2014a). The refined seam is employed as the final seam. These five steps are presented in more detail in Sections 3.1–3.5.

3.1. Extraction of the overlapping area polygon of adjacent images

The image data typically include null and non-null value pixels, and the valid regions indicate those covered by non-null value pixels. The valid regions of the two adjacent images, consisting of non-null value pixels, are first extracted using the Moore neighbor contour tracing algorithm (Pradhan et al., 2010). The reason that the four contours of each image are not directly used to compose the boundary polygon of each image is to avoid unnecessary background holes (null value pixels) appearing in the final mosaic images. The procedure is detailed in Wang et al. (2012). After extraction of the vector valid regions of the two adjacent images, the overlapping area of adjacent images is obtained through polygon clipping (Vatti, 1992), as shown in Fig. 4.

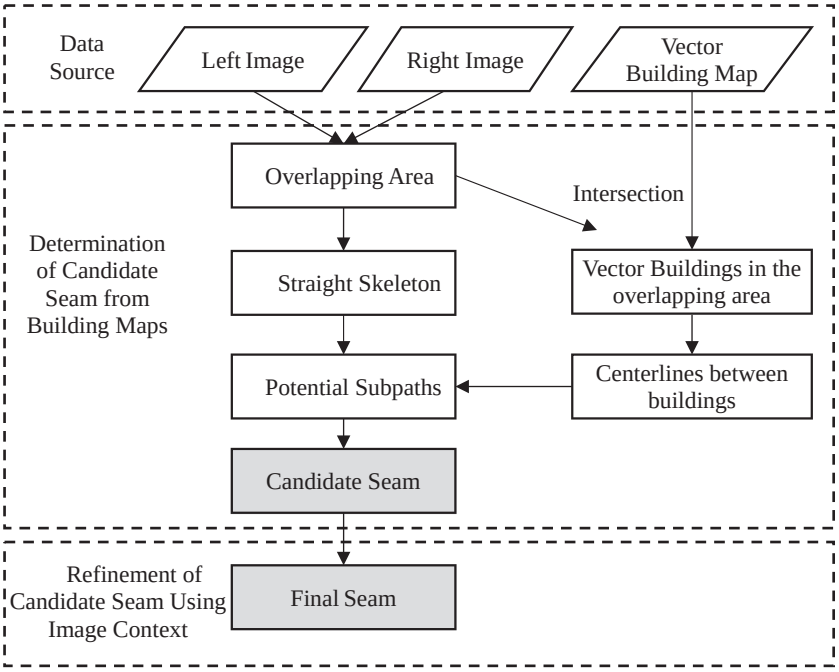

Fig. 3. Workflow for generation of the optimal seam between two images.

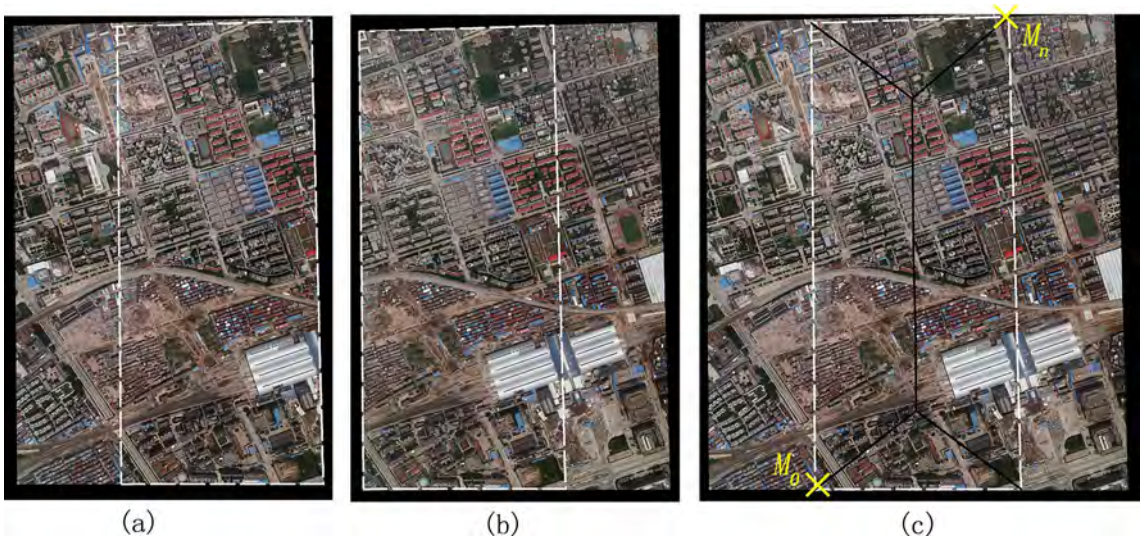

Fig. 4. Left (a) and right (b) are images to be mosaicked. The overlapping region and its skeleton are shown in (c). The overlapping areas are also indicated by the dotted boxes in (a) and (b). Two intersections of the polygons of the left and right images are marked as "M_0" and "M_n", respectively.

The raster image data are no longer used for extraction of the candidate seams but are reincorporated to refine the candidate seams using the image context described in Section 3.5.

3.2. Extraction of the straight skeleton for the overlapping area of adjacent images

The skeletons have long been recognized as an important tool in computer graphics, computer vision and medical imaging. The straight skeleton of a polygon, similar to the medial axis of a polygon, can be described as a set of ridge lines in the likeness of a building roof, as shown in Fig. 1(a). The skeleton of the overlapping area of adjacent images is used to chain the centerlines between buildings together in the overlapping area. The straight skeleton of the overlapping region can be extracted using the constrained Delaunay triangulation skeletonization algorithm (Morrison and Zou, 2007; Nguyen Minh et al., 2009; Tang and You, 2003). The generation steps can be found in Wang et al. (2012). In most cases, any kind of line, even a straight-line connecting the start point and end point of the candidate seam, can be used to chain the centerlines between buildings together to build the best candidate seam. However, using the skeleton as the connector is one of the optimum solutions, as the skeleton of the overlapping area is inherently always located within the overlapping area. Thus, the skeleton can replace the candidate seam with more centerlines between buildings. Fig. 4(c) shows the extracted straight skeleton for the overlapping area. Two intersections of the polygons of the two images are marked as M_0 and M_n and will be considered as the start and end points of the candidate seam.

3.3. Generation of the centerlines existing between buildings from vector building maps

To minimize visual transition, the candidate seam must avoid crossing salient objects, such as buildings. Minimization can be achieved by tracking the occlusion-free land centerlines. Occlusion-free land centerlines, similar to street centerlines, are defined as the medial axes of the occlusion-free areas located between the buildings (Fig. 1). The extraction of the centerlines from a vector-building map is an area-line conversion. Well-known methods for area-line conversion include triangulation, water lining, and straight skeletons (Selvi et al., 2010). Considering the operational speed, a Delaunay-triangulation-based straight skeleton method (Roberts et al., 2005) is employed to extract centerlines existing between buildings. The extracted centerlines are approximate substitutions of the occlusion-free land centerlines.

The steps for extracting the centerlines from the vector building maps are similar to those for extracting the straight skeleton of the overlapping region of adjacent images. The difference between these two procedures is the choice of triangles: for the former, the triangles falling outside the polygon (the overlapping area) are removed; however, for the latter, the triangles falling in the polygons (building polygons) are removed, while the triangles outside the polygons (building polygons) remain. The steps for generating the centerlines existing between buildings are as follows:

The vector building map is clipped by the overlapping area of adjacent images to ensure that all clipped vector building polygons fall in the overlapping area of adjacent images.

Vertices on vector building polygons are degenerated into Delaunay triangulation (DT) points; the Bowyer-Watson Delaunay triangulation algorithm (Chen and Ai, 2004; Morrison and Zou, 2007) is then run on DT points. A Delaunay triangulation network (DTN) is then built.

The constrained Delaunay triangulation network (CDTN) sub-algorithm (Morrison and Zou, 2007) is run to transform DTN into CDTN, ensuring that no triangle in DTN crosses the polygon edges. Subsequently, all triangles that fall in building polygons are removed and triangles in the space polygon between buildings remain.

Skeletons of vacant land polygons between buildings are extracted based on CDTN. Skeletons are a set of straight-line segments, each connecting the mid-points of internal edges or the centroids of the constrained Delaunay triangles (Nguyen Minh et al., 2009; Wang et al., 2012). The skeletons of vacant land polygons are referred to here as "centerlines between buildings".

Fig. 5(a) shows the vector buildings within the overlapping area and the centerlines extracted from vector buildings.

3.4. Determination of candidate seam from centerlines existing between buildings

After generation of the centerline network, the candidate seam is determined using Dijkstra's algorithm (Dijkstra, 1959). Candidate seams should preferentially follow the wider centerlines, where geometric and color differences are lower. The best candidate seam is defined as the least-weight path in the centerline network that connects the start and end points. The four key steps for determining the candidate seam from the centerline network are as follows:

Step 1: The start and end points of the candidate seam are chosen. There are several methods for choosing start and end points of the candidate seam in the overlapping area. Hsu et al. (2002) presented a local-global method based on ordinary Voronoi diagrams of frame centers. This method was used to place start and end points of the candidate seam, but cannot be ensured to generate seams lying in the overlapping area of adjacent images. This may generate gaps that cannot be covered by any image. To reduce these drawbacks in mosaicking, Pan et al. (2009) presented a method based on the concept of 'Area Voronoi Diagrams with Overlap' (AVDO) to ensure that start and end points of the candidate seam can be placed in the

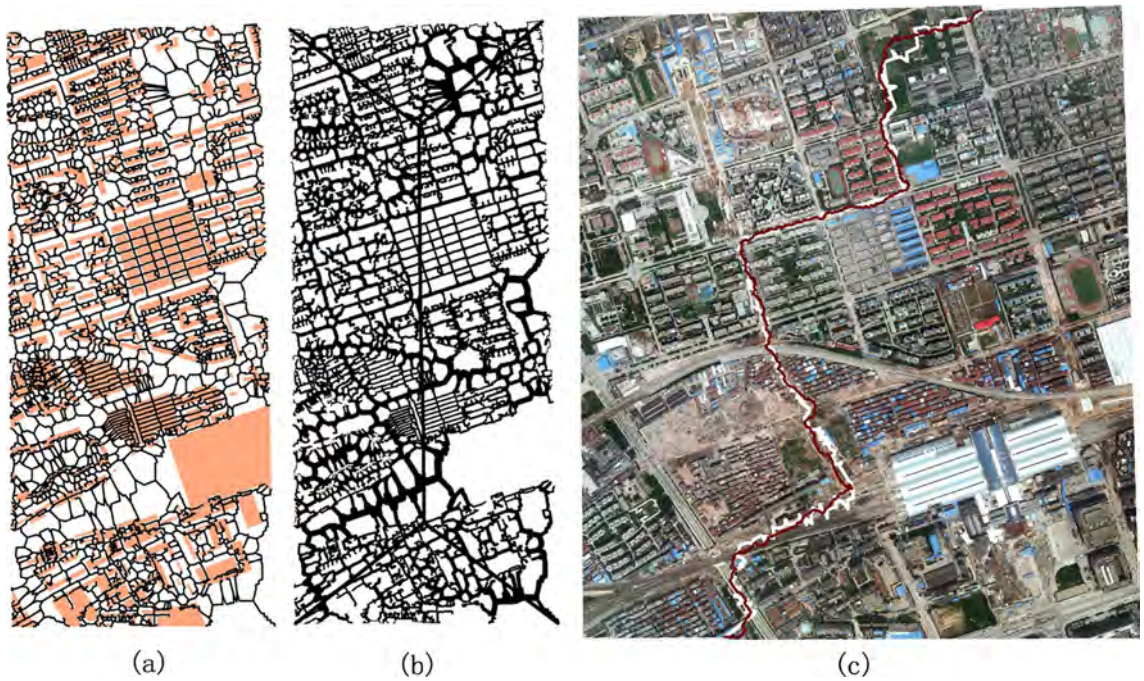

Fig. 5. (a) Vector buildings within the overlapping area (cantaloupe polygons) and centerlines extracted from vector buildings (thin polylines). (b) Weighted undirected graph $G(V, E)$ created from centerlines (their passing weights shown using varying thickness) and the straight skeleton of the overlapping area (the straight lines with constant thickness). (c) Candidate seam generated from the weighted undirected graph (white polyline) and refined seam (red polyline). Details of the mosaic are given in Figs. 8–10. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

overlapping area of adjacent images. However, the start and end points of the candidate seam may be placed in areas of high mismatch. To this end, Mills and McLeod (2013) proposed a method to replace the heuristic (fixed) placement of the seam network junctions using a local search for optimal positions. This provides greater flexibility in selecting optimal seams. In this study, we choose two intersections between the two adjacent image polygons as the start and end points of the candidate seam. Note that if there are two or more intersections, the two intersections with the longest distance are automatically chosen as start and end points of the candidate seam to ensure that the candidate seam is a simple polyline.

Step 2: The straight skeleton of the overlapping area of adjacent images is obtained using the method presented in Section 3.2. The centerlines existing between buildings are extracted using the method presented in Section 3.3, after vector buildings falling in the overlapping area are obtained.

Step 3: A weighted undirected graph G (V, E) is created for determination of candidate seams. This graph creation method begins with the creation of a null set of virtual paths. Next, the skeleton extracted in the second step and the centerlines between buildings extracted in the third step are added to the set of potential subpaths. Then, these potential subpaths, including skeleton segments and occlusion-free land centerlines, are split at intersections using the line-splitting algorithm (Zhao et al., 2004). This line-splitting processing is equated with the use of function "planarize lines" in ArcMap. Finally, the weights are assigned to these potential subpaths according to (3). If the set of the intersections and end points of virtual paths is considered as the set of vertices V and the set of the various potential subpaths is considered as the set of edges E, a weighted undirected graph G (V, E) can be built to find the least-weight candidate seam, as shown in Fig. 5(b). The thickness of the polylines represents weights of the centerlines and skeleton lines.

Step 4: Dijkstra's algorithm (Dijkstra, 1959) is applied to find the lowest-weight path from the start point to the end point in $G(V, E)$. The total weight of tracking a candidate seam CS that contains several candidate subpaths (i.e., potential seam paths) can be written as

$$f(CS) = \sum Tcost(l_i), \quad l_i \in SL \in W \tag{6}$$

where $f(CS)$ is the sum of the weights of tracking these candidate subpaths and W is the overlapping area of adjacent images to be mosaicked. The smaller the weight of a potential subpath, the greater the potential of the subpath l_i being part of a seam.

The lowest-weight (optimal) path is then obtained through minimizing $f(CS)$ using Dijkstra's algorithm; the weight is

$$f(BS) = \min[f(CS)], \quad l_i \in SL \in W \tag{7}$$

Fig. 5(b) and (c) shows the weighted undirected graph and the candidate seam generated from the centerline network, respectively.

3.5. Refinement of candidate seam by considering surrounding pixels

The best candidate seam generated from vector building maps in Section 3.4 chooses paths by avoiding the crossing of buildings existing on the vector map. They do not consider any image content information, as in the raster-based mosaicking method (Agarwala et al., 2004; Chon et al., 2010; Kerschner, 2001). Consequently, this may lead to the crossing of some salient objects, such as cars and trees on roads, or some new buildings not updated in real-time on the vector map. Therefore, the candidate seam should be considered as an intermediate production. To generate an optimal seam for seamlessly combining images, the candidate seam must be refined using a local search incorporating image data within the overlapping area. The refinement procedure is given as follows:

Step 1: Limit search area for the candidate seam. Generally, searching a 5-m buffer area from the initial candidate seam could easily take the candidate seam over cars and trees on roads. In this study, the search area is defined as a buffer area along the candidate seam with varying buffer distance. If the local valid width of the candidate seam is lower than 10 m, then the corresponding subpath is likely a centerline between dense buildings. The search distance is then set as 50 m to avoid crossing buildings (users can increase the search distance when necessary). If the local valid width of the candidate seam is between 10 m and 60 m, then the corresponding subpath is likely a street centerline. The search distance is then set as 5 m to avoid crossing cars or trees on streets. However, if the local valid width of the candidate seam is greater than 60 m (the maximum street width of China), then the corresponding subpath is likely a centerline between sparse buildings. Some newly built buildings may emerge. The search distance is set as 50 m. Overall, the buffer distances are expressed as:

$$BufDis(l_i) = \begin{cases} 5, & \text{if } 10 \leqslant vadW(l_i) < 60 \\ 50, & \text{if } vadW(l_i) < 10 \text{ or } vadW(l_i) > 60 \end{cases} \tag{8}$$

where $BufDis(l_i)$ is the buffer distance for the ith subpath on the candidate seam.

Step 2: Refine the candidate seam in the search area using surrounding pixels within the corresponding overlapping area. Assume that m images (F_1, F_2,..., F_m) have overlaps with the overlapping area W containing the candidate seam and that the refinement is conducted in the search area. The total weight of the final seam FS passing through a certain number of pixels can be defined as Eq. (9), as seen in (Wang et al., 2012):

$$Tcost(FS) = \sum Tcost(u, v), \quad (u, v) \in FS \in D \in W \tag{9}$$

where D indicates the search area defined in the first step and $Tcost(u, v)$ indicates the weight for the final seam FS passing through the $pixel(u, v)$. $Tcost(u, v)$ is defined as the difference between the maximum and minimum luminance values among the m overlapping images at pixel (u, v). Explicitly:

$$Tcost(u, v) = \max_{j=1\cdots m} L_j(u, v) - \min_{j=1\cdots m} L_j(u, v) \tag{10}$$

where $L_j(u, v)$ is the luminance image of the jth image at pixel (u, v), which can be defined as

$$L_j(u, v) = 0.3P_j^R(u, v) + 0.59P_j^G(u, v) + 0.11P_j^B(u, v) \tag{11}$$

where $P_j^R(u, v)$, $P_j^G(u, v)$, and $P_j^B(u, v)$ indicate the pixel (u, v) values in the red, green, and blue bands of the jth image, respectively, and 0.3, 0.59, and 0.11 are the weights of red, green, and blue bands used to build the luminance image, respectively.

Following this, the best choice for the candidate seam is to minimize $Tcost(FS)$ to obtain the optimal seam:

$$Tcost(FS) = \min[Tcost(FS)] \tag{12}$$

In Eq. (12), $\min[Tcost(FS)]$ is the minimum of $Tcost(FS)$. This is similar to minimizing $f(CS)$ in Eq. (7), achieved by running the raster-based Dijkstra's algorithm (Chon et al., 2010; Pan et al., 2014a).

Fig. 5 shows the refined seam (the red polyline). Note: this section only shows the procedure of generating seams. Details of the mosaic are given in Figs. 8–10.

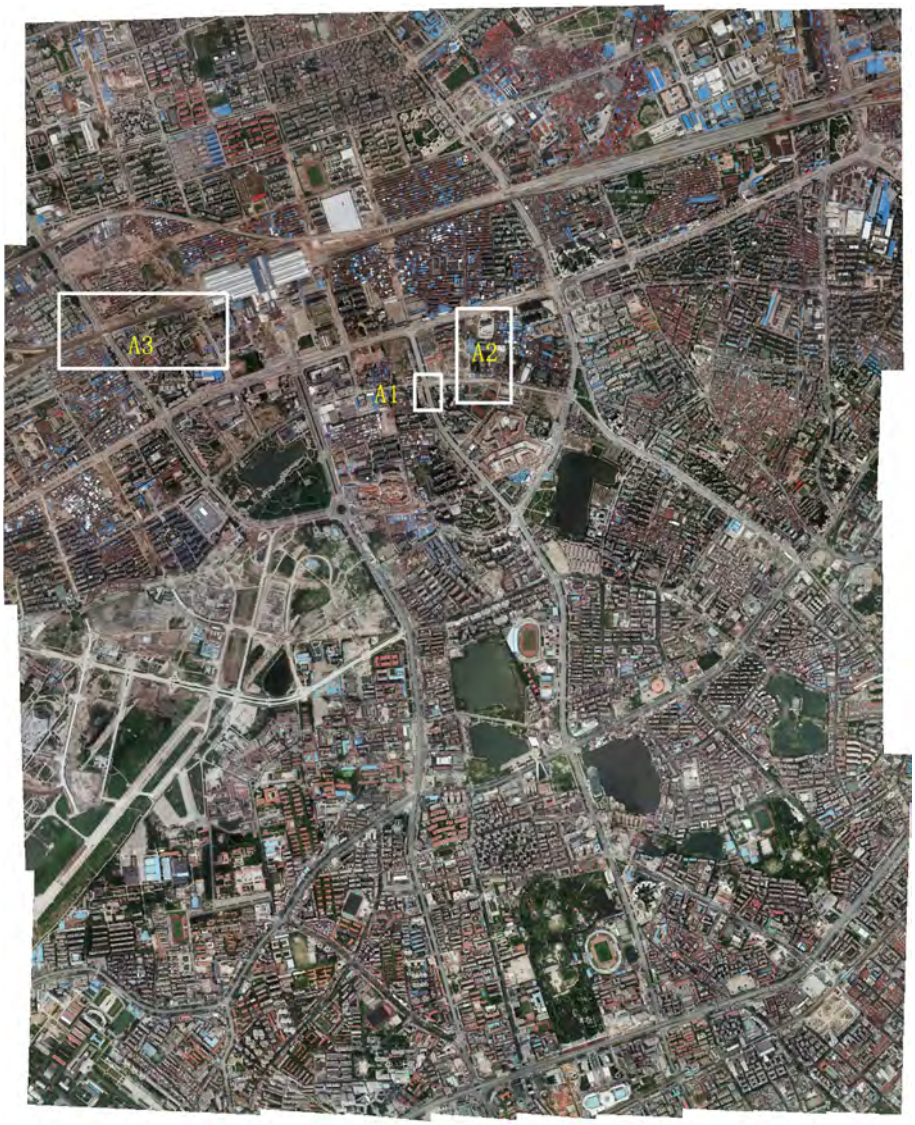

Fig. 6. Final mosaic of 36 images using the building map. Details of three sub areas (A1, A2, and A3) are shown in Figs. 8–10.

4. Experimental results and discussion

Three groups of aerial orthoimages are employed in the following experiments. The first, second, and third groups of images are from a downtown area representing dense high-rise buildings and roads, a suburban area representing many low-rise buildings and roads, and a rural area where a few flat houses and minimal roads are found, respectively. Visual inspections of the mosaicking results, weights and efficiency of the proposed method are also compared with those of two alternate methods: the vector-road-based method (Wang et al., 2012) and Dijkstra's method (Kerschner, 2001). Before mosaicking, these images have been orthorectified to the World Geodetic System of 1984 (WGS-84) using ground control points and a digital elevation model (DEM) with a resolution of 5 m. The dominant geographic features of study areas are Jianghan plain (elevation 20–50 m).

The algorithms described above were developed with C++ in Visual Studio 2010. The mosaicking process was conducted under 64-bit Windows 7 running on a laptop with an Intel(R) Core(TM) i5 Duo 2.2-GHz processor, 4-GB internal memory, and a hard disk with 750-GB capacity, 8-MB cache, and 5400 r/min speed.

4.1. Mosaicking thirty-six images from downtown area

The first test image data consist of thirty-six colored aerial images generated from four photographic strips. These images

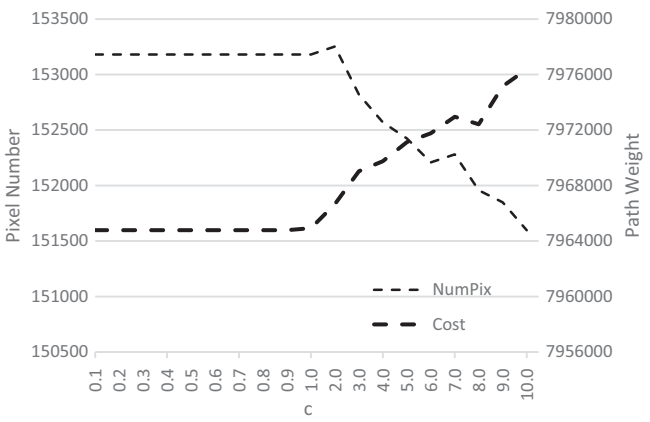

Fig. 7. The pixel numbers and path weights of the candidate seams for increasing c.

cover a study area containing many high-rise buildings from the Hankou district, a downtown area of Wuhan, China. Each photographic strip contains nine images. These images were also captured using DMC (Mumtaz and Palmer, 2013), with approximately 65% forward-overlap and 35% side-overlap. The spatial resolution of each image is 0.2 m and the image size is 6528 × 9856 pixels. The vector building map used in this test has a horizontal accuracy of 0.5 m and height accuracy of 1 m.

Using the method presented in Section 3, seams of these images are determined. Furthermore, the effective mosaic polygons (EMPs) (Pan et al., 2009) are formed for each image from the seams using the turn-left algorithm (Liang et al., 2005). The details of the EMP construction method are shown in (Wang et al., 2012). Each EMP defines the useful part(s) of each image. The final mosaic image is generated through filling the EMPs with the corresponding image context. Fig. 6 shows the final mosaic using the proposed method.

According to Eq. (4), the quality of the seams is dependent on c, which is used to define the weight of the skeleton of the overlapping area. We first use half of this group of images to analyze the influence of c on the passed pixel numbers of seams and the path weights. Path weights are calculated using Eq. (9). Fig. 7 shows the pixel numbers and weights of candidate seams. Fig. 7 shows that when $c > 1$, increasing c presents a lower possibility of centerlines

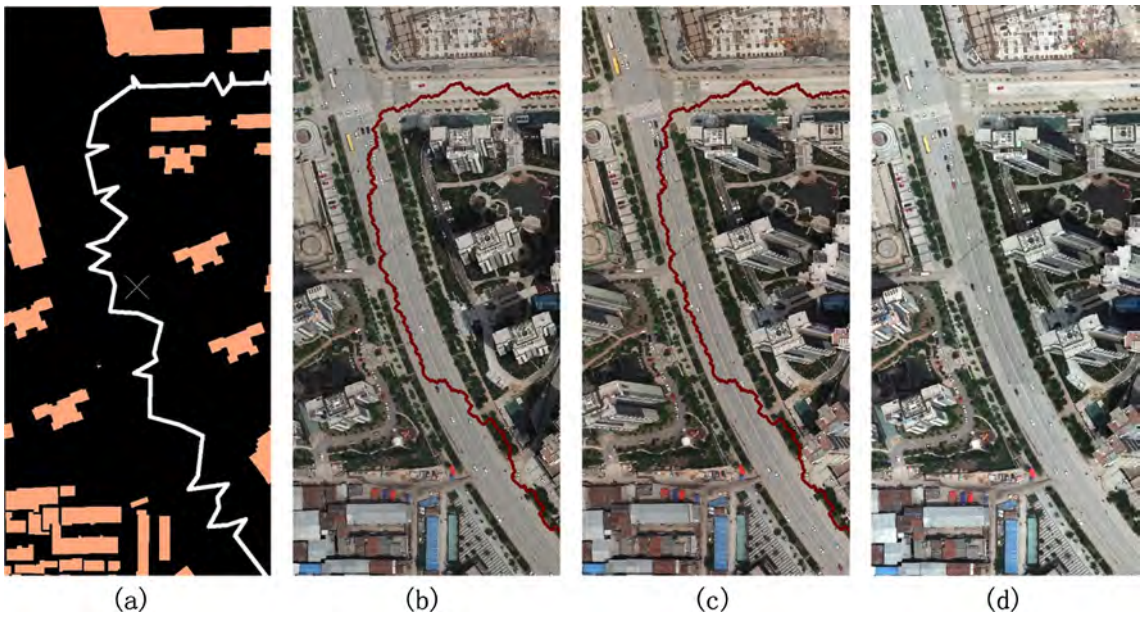

Fig. 8. Sub area A1 showing seams determined using the proposed vector-building-based method: (a) vector buildings overlapping candidate seam, (b) left image overlapping refined seam, (c) right image overlapping refined seam, and (d) result mosaicked using refined seam.

Fig. 9. Sub area A2, selected for comparison of the mosaics and seams determined using the three examined methods. (a) Seams determined using Dijkstra's algorithm (blue polyline), vector-road-based method (green polyline) and vector-building-based method (red polyline). (b) Mosaicking result of Dijkstra's algorithm. (c) Mosaicking result of vector-road. (d) Mosaicking result of vector-building. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

being chosen as parts of seams. This generally results in decreasing seam lengths and increasing path weights. The pixel numbers of candidate seams reduce from 153,181 to 151,597; the path weights of candidate seams increase from 7,964,785 to 7,976,377; however, when $0 < c < 1$, increasing c has no influence on seam quality. This result shows that c can be assigned any real number between 0 and 1, as mentioned in Section 2.2. To simplify the definition of path weights, c is fixed as 0.1 in the following experiments.

Fig. 8 shows details of the seams determined using the proposed method. Fig. 8(a)–(d) shows the vector building map overlapping the candidate seam, the left image overlapping the refined seam, the right image overlapping the refined seam, and the result mosaicked using the refined image without feathering, respectively. The results show good alignment. After visual inspection, the initial candidate seam lies between buildings and follows the wide roads to avoid crossing buildings. However, the initial candidate seam is jogged because the Delaunay triangulation algorithm (Roberts et al., 2005) was used for the generation of the centerlines between buildings. The centerlines consist of straight-line segments, each connecting the midpoints of the internal edges or the centroids of the Delaunay triangles. The refined seam is optimized to bypass smaller salient objects, such as trees and cars, and is thus more seamless than the candidate seam.

Using the same images, mosaicking is also conducted using two alternate methods: the vector-road-based method (Wang et al., 2012) and Dijkstra's method (Kerschner, 2001). Two selected regions of seams and mosaicking results corresponding to the

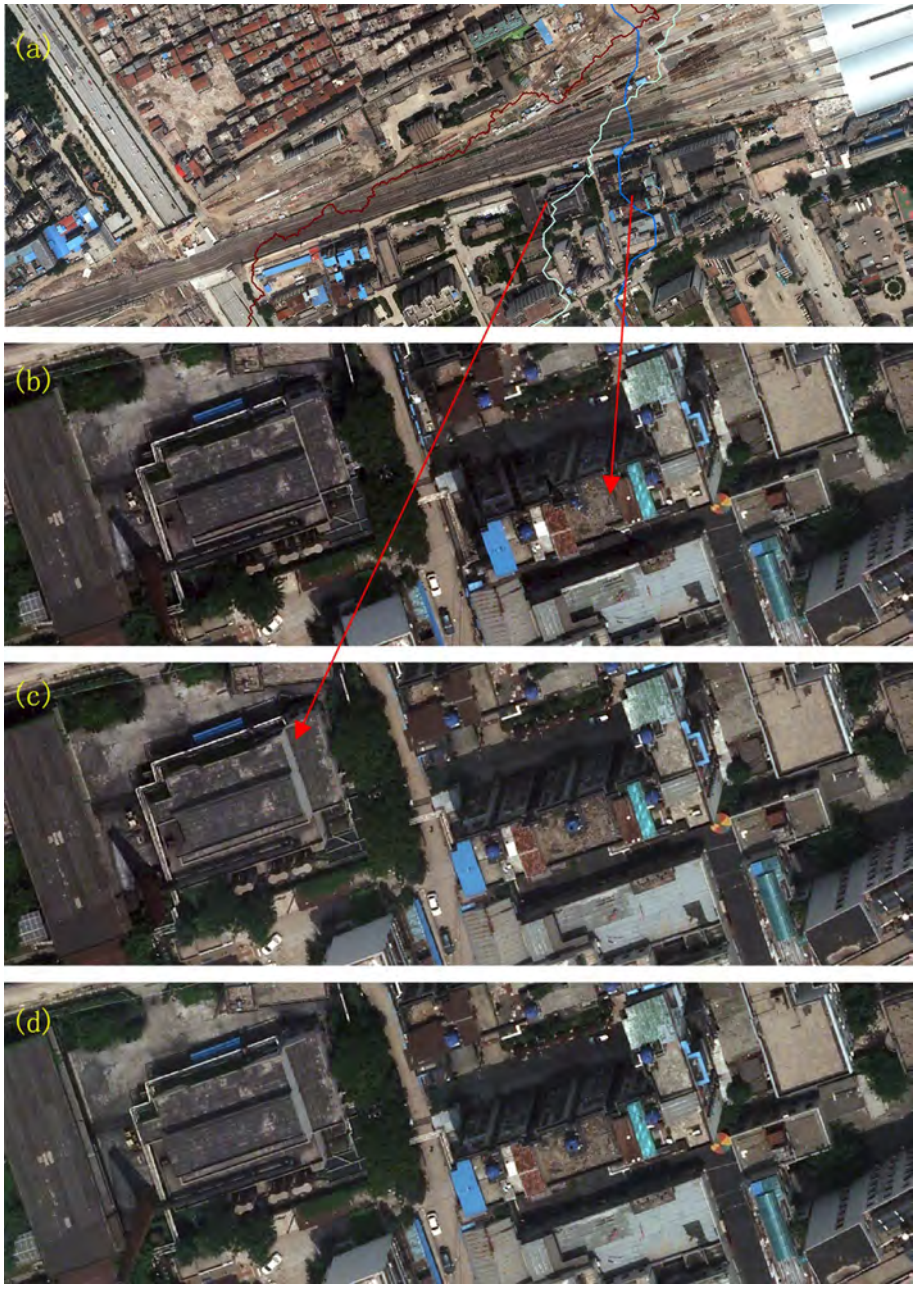

Fig. 10. Sub area A3, selected for comparison of the mosaics using the seams determined using the three examined methods. (a) Seams determined using Dijkstra's algorithm (blue polyline), vector-road-based method (green polyline), and vector-building-based method (red polyline), respectively. (b) Mosaicking result of Dijkstra's algorithm. (c) Mosaicking result of vector-road. (d) Mosaicking result of vector-building. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

three methods are shown in Figs. 9 and 10, respectively. The seams generated using Dijkstra's algorithm (blue polylines), vector-road-based method (green polylines), and vector-building-based method (red polylines) are shown in Figs. 9(a) and 10(a). The details of mosaics are shown in Figs. 9(b)–(d) and 10(b)–(d). The performance of the vector-building-based method is similar to that of the vector-road-based method when vector roads are available (Fig. 9) but crosses less buildings in the absence of or with low occurrence of roads (Fig. 10). For example, the railway in Fig. 10 (a) is not included in the vector road network. The vector-building-based method can still follow the wide centerlines (where the railway is located) between buildings, whereas both the vector-road-based method and Dijkstra's method cut many buildings. Dijkstra's method chooses the path with the lowest intensity difference as seams. However, the weights depend only on the length of the seam and the degree of color mismatch (Chon et al., 2010). The color match is sometimes worse on roads than on roofs. With this method, it is extremely difficult to locate salient features and then avoid crossing them. The mosaicking results from this method are thus barely satisfactory, as the seams cut the most number of buildings.

In Table 1, the numbers of buildings that seams crossed, the weights of the extracted seams calculated using Eq. (9), and the associated computation times are presented for the three methods. The candidate seams extracted using Dijkstra's algorithm can be described as the straight lines connecting two intersections of the polygons of adjacent images. As seen in Table 1, these candidate seams cross the most buildings (1839) and thus have the highest path weights (17,951,532). Even after refinement, when the refined seams have the lowest weight (921,486), they still cross the most buildings (807). The candidate seams extracted using the proposed method are jogged, as shown in Fig. 8(a), and have longer paths than those of the vector-road based method. As expected, the proposed method generates candidate seams with higher weights (14,024,805) than those of the vector-road-based method (9,418,726). However, the refined seams have lower weights (1,261,564) than those of the vector-road-based method (2,443,554). According to the definition, it is easy to understand that Dijkstra's algorithm generates the lowest-weight refined seams. The vector-road based method relies heavily on the vector road network to determine candidate seams. Its seams cross many salient objects (candidate seams: 252 buildings, refined seams: 152 buildings), especially in the absence of enough vector roads. Consequently, the vector-road-based method generates higher-weight seams. However, the proposed method can generate good candidate seams by using the vector-building map to reduce the crossing of buildings even in the absence of roads. The seams cross the least number of buildings (candidate seams: 192, refined seams: 24). The crossed buildings generally appear in image edges and no other lands are available to be chosen to be crossed. The crossed buildings also include those not updated into the building maps. Ultimately, this novel aspect reduces weights.

The numbers of crossed buildings and weights described above indicate that the proposed vector-building-based method could obtain final seams of higher quality than the other methods examined. However, the proposed method requires more computation time for generation of the candidate seams. The vector-road-based method requires 1 s to extract 35 candidate seams. The proposed method, however, spent approximately 348 s. This is due to the increased computational time associated with the generation of centerlines between buildings. The time complexity of the algorithm used in this study for generation of the centerlines is $O(N^2)$, where N is the vertex number of all building polygons (approximately 10,000 building polygon vertices per overlapping area in this study). Both the proposed and vector-road-based methods employ the same refinement algorithm, i.e., raster-based Dijkstra's algorithm, and thus require similar times (92 and 98 s) to refine 35 seams. Contrary to the other examined methods, Dijkstra's algorithm only connects the two intersections of the polygons of adjacent images to form the candidate seams. Consequently, the computation time for extracting candidate seams approaches 0 s, but additional time (2158 s) is required to search for 35 seams globally. Overall, the total computation time of the vector-building-based method is approximately 4 times greater than that of the vector-road-based method, but still only 1/5 that of Dijkstra's algorithm.

The time and memory requirements of the proposed method also vary with the number of images to be mosaicked. The algorithms are run on subsets of the images containing 3, 9, 18, 27, and 36 images. Fig. 11 shows the time required for generation of candidate seams and refined seams. It can be seen that the times required both for generation of candidate seams and refined seams are roughly linear in the number of images, which is expected. Generating a candidate seam takes an average of approximately 10 s. Refining the candidate seam takes an average of another 3 s. The peak memory usage also varies with the number of images, as shown in Fig. 12. When the number of images to be mosaicked is less than nine, constant memory usage is exhibited both for the

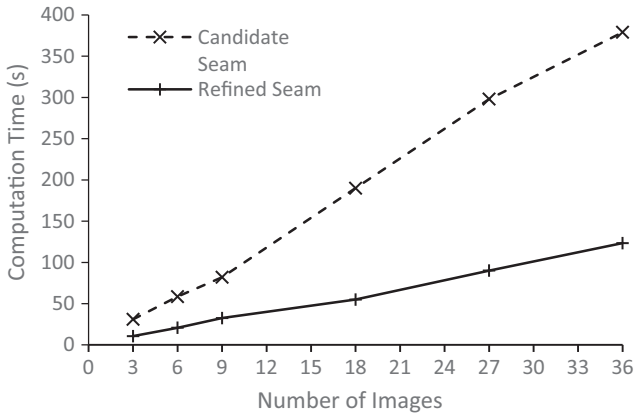

Fig. 11. Time required to process scenes with a varying number of images. Times shown for generation of candidate seams and refined seams.

Table 1
Comparison of the three examined methods using thirty-six images from downtown area.

		Dijkstra's	Vector-road	Vector-building
Crossed buildings	35 Candidate seams	1839	252	192
	35 Refined seams	807	152	24
Weights	35 Candidate seams	17,951,532	9,418,726	14,024,805
	35 Refined seams	921,486	2,443,554	1,261,564
Times (s)	Extracting 35 candidate seams	0[a]	1	348
	Refining 35 seams	2158	98	92

[a] Note that Dijkstra's algorithm does not depend on vector data. Candidate seams are the straight lines connecting the two intersections of the polygons of adjacent images; thus, their lengths are shortest. Computation time for extracting candidate seams approaches 0 s.

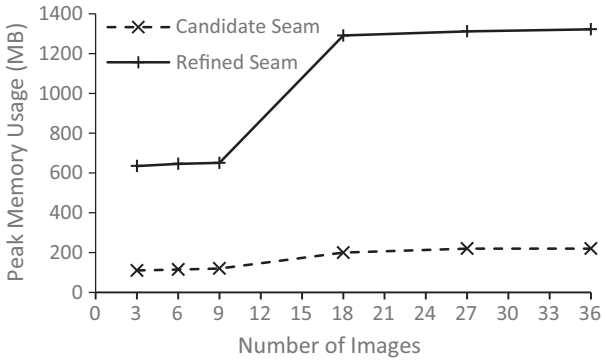

Fig. 12. Memory required to process scenes with a varying number of images. Memory usage shown for generation of candidate and refined seams.

generation of candidate seams and refined seams. However, when the number of images to be mosaicked is more than nine, the memory required is approximately doubled in both cases. This is because the greatest memory consumption for generation and refinement of seams is searching for least-weight paths. This is linear in the area of the overlapping area of adjacent images or photographic strips. When the images to be mosaicked come from two or more photographic strips (the number of images increases to nine), the double memory must be utilized to search for least-weight paths in the overlapping area of adjacent photographic strips (the area is approximately double that of the overlapping area of adjacent images in this example). When the number of photographic strips is greater than one, the peak memory usage no longer increases significantly with the number of photographic strips. This is because the maximum area of the overlapping areas no longer increases. Generating candidate seams between adjacent images consumes approximately 110 MB. That of refining candidate seams between adjacent images is approximately 635 MB. This memory is almost completely assigned to the weight graph structures, which are currently held in memory throughout the process. For larger scenes, the memory requirements may exceed the available RAM on computers. In such cases, the weight graph structures could easily be cached on a disk because their construction is independent of one another and their use for determining optimal seams is local.

4.2. Mosaicking six images from suburban area

The second example presented is a set of six aerial images from the Qingshan district, a suburb of Wuhan, China. The landscape consists of open lands with a few low-rise buildings (typically lower than 20 m) distributed throughout. The images were extracted from two photographic strips. The images were captured

Fig. 13. Final mosaic of six images using the building map. Details of three sub areas (A1, A2, and A3) are shown in Figs. 14–16.

Fig. 14. Sub area A1, showing details of refined seam determined using the proposed method: (a) upper image overlapping refined seam, (b) lower image overlapping refined seam, and (c) mosaicking result without feathering.

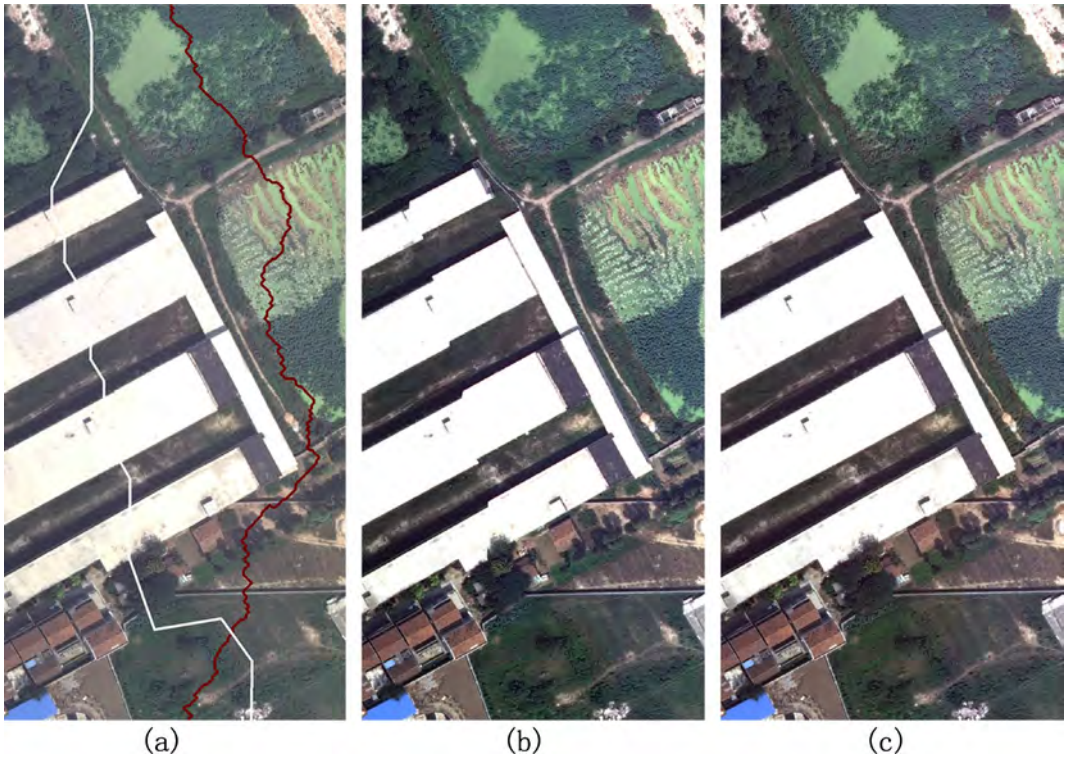

Fig. 15. Sub area A2, showing example of mosaic when new buildings are excluded from the building map: (a) left image overlapping candidate and refined seams, (b) mosaicking result using candidate seam, and (c) result mosaicked using refined seam.

using an Ultracam-D camera (Erfanifard et al., 2014) with approximately 30% forward-overlap and 35% side-overlap. The resolution of each image is 0.5 m and image size is 7020 × 12,430 pixels. The vector building map used in this test is a generalized building map, where adjacent buildings are merged into a polygon with a horizontal accuracy of 5 m and height accuracy of 5 m.

These six images from the suburban area are mosaicked using the proposed method. The vector building map was unable to exactly match these images, particularly where some small-scale buildings are newly built, but experiments show that this has no obvious influence on the final mosaic. Fig. 13 shows the mosaicking result of six images using the vector building map. Fig. 14 shows the details of a refined seam determined using the proposed method. The refined seams successfully avoided the crossing of buildings.

In addition to small salient objects, the refined seams can avoid crossing many new buildings not included in the building map. New buildings are common in this area due to recent development policies. Fig. 15 shows an example of a mosaic when new buildings are not updated in the building map. Fig. 15(a) shows the candidate and refined seams. The candidate seam cuts some new buildings. Using this candidate seam to generate the mosaic, the left parts of the buildings are clearly offset from the right parts, as shown in Fig. 15(b). However, the refined seam relocates the candidate seam to follow the boundaries of the new buildings to generate a seamless mosaic, as shown in Fig. 15(c). This example proves the necessity of refining candidate seams. Users can increase search buffer distances to bypass the larger salient objects using the proposed refinement method described in Section 3.5 to allow for effective refinement, but we do not guarantee that every attempt will be successful because Dijkstra's algorithm tends to produce a shorter path even with some high-cost pixels (salient objects).

The three methods are also compared based on this group of data. A selected region for the comparison is shown in Fig. 16. Both the vector-building-based method and vector-road-based method successfully follow roads. However, Dijkstra's algorithm crossed some buildings. This is another clear example that indicates that the vector-building-based method and vector-road-based method have better performance than Dijkstra's algorithm when vector roads are available.

Using this group of data, the numbers of buildings that seams crossed, the weights of the extracted seams calculated using Eq. (9), the computation times, and the peak memory usage are presented in Table 2. The results show good agreement with those of the first test. Dijkstra's algorithm generates the lowest-weight refined seams (921,486) but uses the most time (316 s) to generate the candidate and seams, and the final seams cross the most buildings. The vector-road-based method takes the least time (16 s) to generate the candidate and seams, but the refined seams have the highest weights (324,137) and cross the second most buildings. Overall, the proposed method generates the most accurate seams, crossing the least number of buildings (13), though the weights of the refined seams (205,234) and the computation time (62 s) are not the least. Due to the use of the width-varied search area in (8), the proposed method also utilizes the least memory compared with the other methods examined.

4.3. Mosaicking one hundred and ten images from rural area

The final example presented is a set of 110 Unmanned Aerial Vehicle (UAV) images from Xiangjiawei Village, a rural area of Hongshan district, Wuhan, China. These images cover a study area dominated by farmland. Only a few flat houses (typically lower

Table 2
Comparison of the three examined methods using six images from suburban area.

	Dijkstra's	Vector-road	Vector-building
Buildings crossed	183	32	13
Weights	163,558	324,137	205,234
Times (s)	316	16	62
Peak memory usage (MB)	2120	1260	640

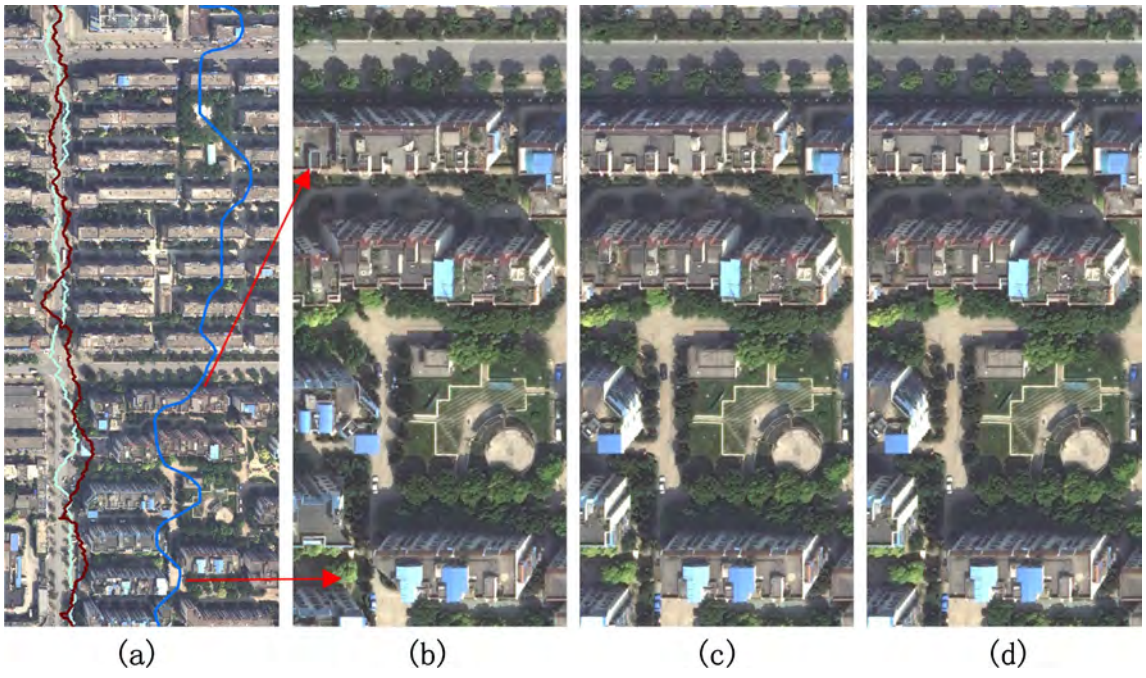

Fig. 16. Sub area A3, selected for comparison of mosaics and seams using six images from suburban area. (a) Left image overlapping the seams determined using Dijkstra's algorithm (blue polyline), vector-road-based method (green polyline) and vector-building-based method (red polyline). (b) Dijkstra's algorithm. (c) Vector-road. (d) Vector-building. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

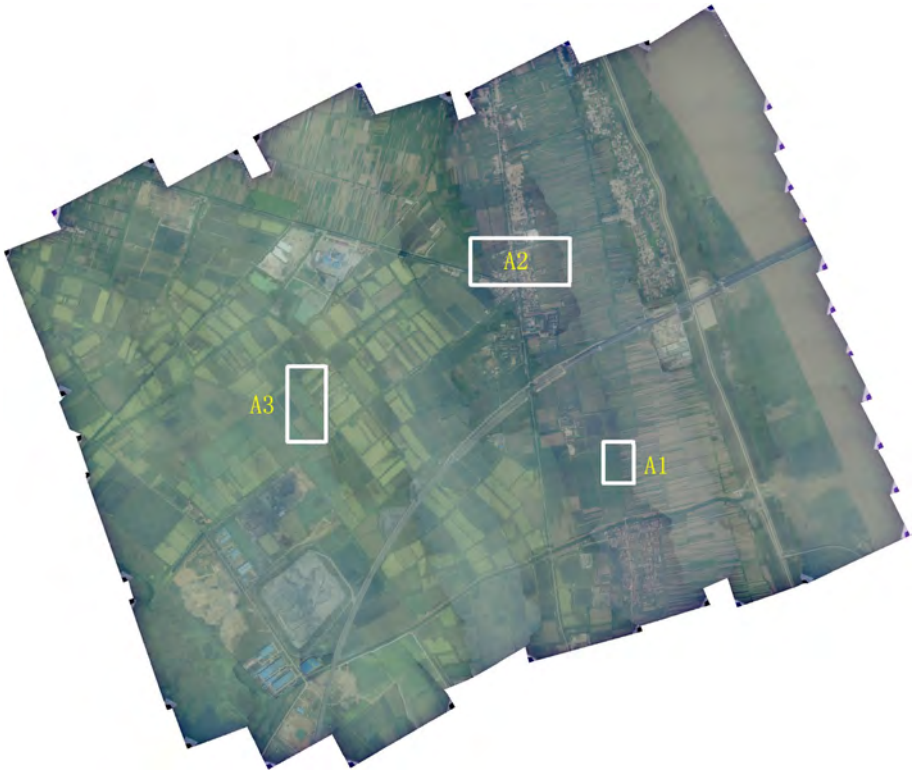

Fig. 17. Final mosaic of 110 images using building map. Details of three sub areas (A1, A2, and A3) are shown in Figs. 18–20.

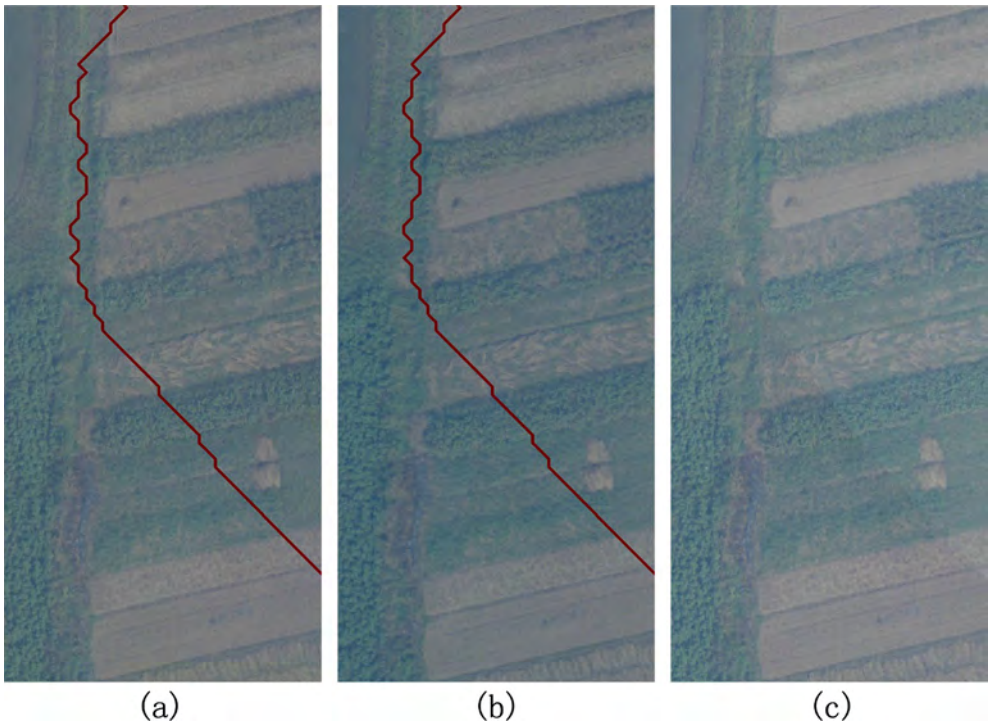

Fig. 18. Sub area A1, showing details of refined seam determined using the proposed method: (a) left image overlapping refined seam, (b) right image overlapping refined seam, and (c) mosaicking result without feathering.

than 6 m) and three roads are distributed in the area. The images are extracted from 11 photographic strips. The images were captured using a Leica M camera with approximately 65% forward-overlap and 35% side-overlap. The resolution of each image is 0.5 m and the image size is 5952 × 3976 pixels. The vector building map used in this test has a horizontal accuracy of 3 m and height accuracy of 2 m. Fig. 17 shows the final mosaic using the vector-building map.

Due to the sparse distribution of buildings, many candidate seams only consist of skeleton segments. However, the mosaicking results have no obvious geometric differences on the final mosaic. Fig. 18 shows details of a refined seam determined using the proposed method. Fig. 18(a)–(c) shows the left image overlapping the refined seam, the right image overlapping the refined seam, and the mosaicking result without feathering, respectively. In Fig. 18(c), the texture characteristics match very well, although color characteristics vary significantly for farmland. This example proves the effectiveness of the seam detection method in rural settings.

As a comparison, mosaicking was conducted for these images using all examined methods. The seams generated using Dijkstra's algorithm (blue polylines), vector-road-based method (green polylines), and vector-building-based method (red polylines) are shown in Figs. 19(a) and 20(a). The details of mosaics are shown in Figs. 19(b)–(d) and 20(b)–(d). The proposed method avoids the crossing of buildings in the regions with dense buildings. However, for the regions without buildings and roads, the proposed method generated the same seams as the vector-road-based method. The seams of Dijkstra's algorithm are almost the same as those of these two vector-based methods. The seams consistently cut farmlands where geometric characters align very well, as shown in Fig. 20.

The numbers of crossed buildings, the weights of the extracted seams, and the computation times are presented in Table 3. Similar to the former tests, Dijkstra's algorithm also generates the lowest-weight refined seams and uses the most memory and time to generate the candidate and refined seams. The vector-road-based method takes the least time to generate the candidate and seams. The proposed method generates the most satisfying seams, crossing only 5 buildings. However, weights of refined seams, running time, and peak memory usage are approximately equal to those of the vector-road-based method. This is because few vector buildings and roads are available for extracting seams and many skeleton segments are directly chosen as parts of seams. The comparison shows that the proposed method has similar or better performance for mosaicking images in rural areas, where both roads and buildings are extremely sparse.

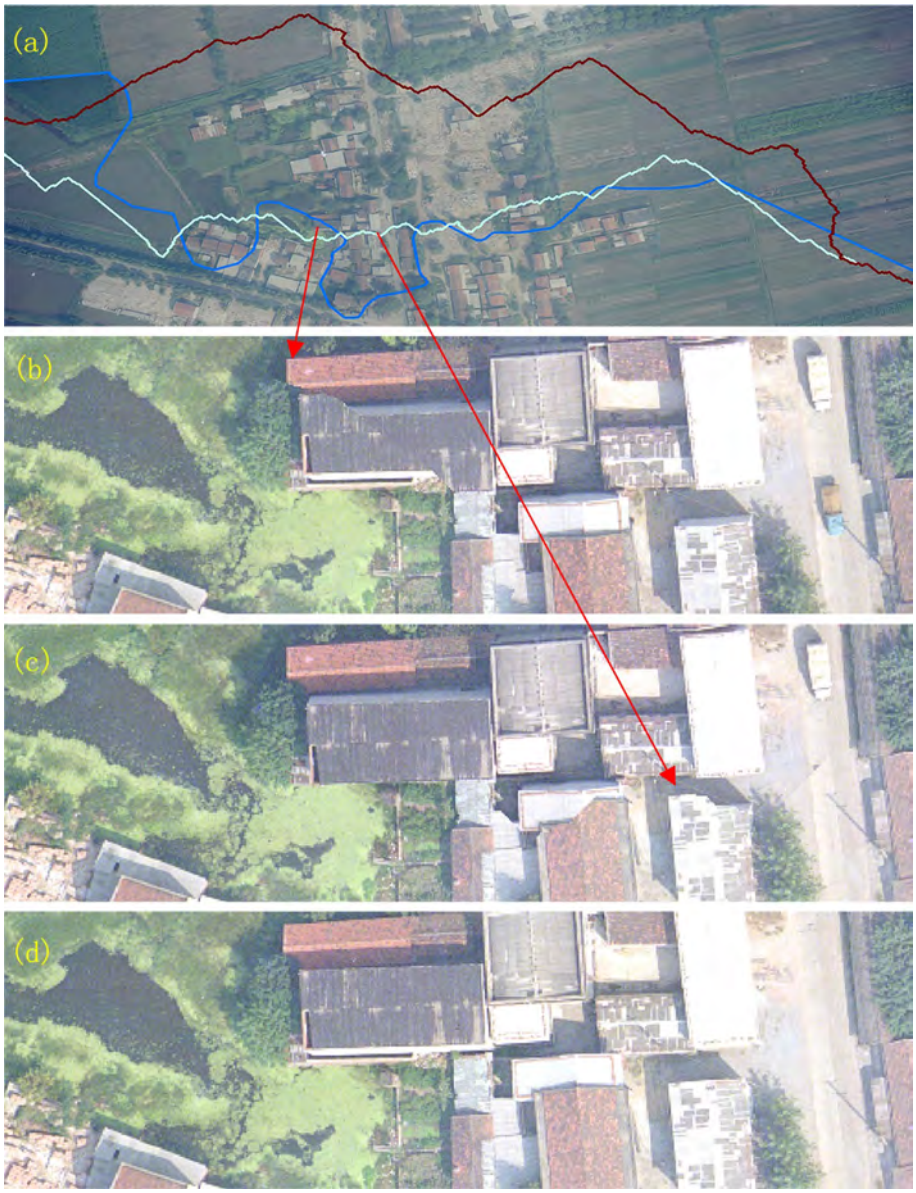

Fig. 19. Sub area A2, selected for comparison of mosaics and seams using 110 images from rural area. (a) Upper image overlapping the seams determined using Dijkstra's algorithm (blue polyline), vector-road-based method (green polyline), and vector-building-based method (red polyline). (b) Dijkstra's algorithm. (c) Vector-road. (d) Vector-building. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

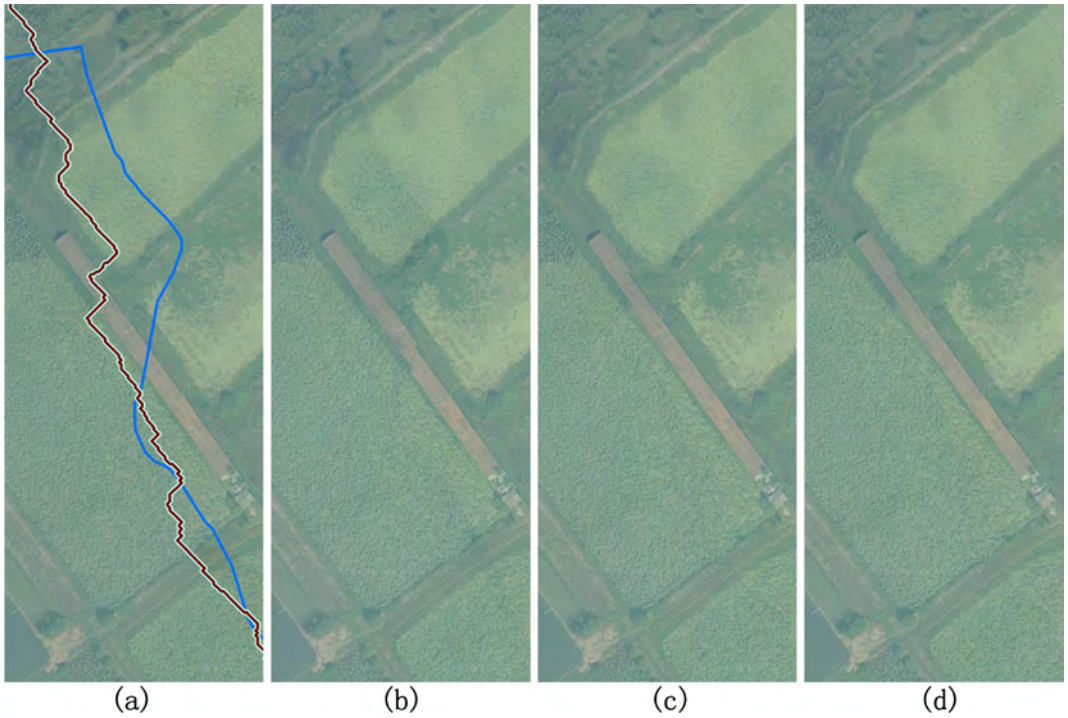

Fig. 20. Sub area A3, selected for comparison of mosaics and seams using 110 images from rural area. (a) Left image overlapping seams determined using Dijkstra's algorithm (blue polyline), vector-road-based method (green polyline) and vector-building-based method (red polyline). (b) Dijkstra's algorithm. (c) Vector-road. (d) Vector-building. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 3
Comparison of the three examined methods using 110 images from rural area.

	Dijkstra's	Vector-road	Vector-building
Buildings crossed	32	25	5
Weights	1,812,632	2,338,543	2,625,473
Times (s)	7560	695	742
Peak memory usage (MB)	2420	680	640

5. Conclusions

In this study, a novel method using vector building maps to generate seams is proposed. Different from the vector-road-based method, which tracks vector roads to generate seams, the proposed method is designed to follow the centerlines between buildings and to reduce the crossing of salient objects on images, particularly high-rise buildings.

The experimental results of this study demonstrate that the vector-building-based method has considerable potential to improve the efficiency and quality of mosaicking, especially for the areas where vector buildings are dense. The proposed method ensures that seams avoid crossing buildings, trees and cars as much as possible by placing them along the centerlines between buildings. Compared with Dijkstra's algorithm and the vector-road-based method, the proposed vector-building-based method generates higher quality of seams for images within downtown and suburban areas, where buildings are dense, and generates the same quality of seams for images from rural settings, where buildings are sparse or nonexistent. The memory requirements of this method are also the least. The vector-building-based method is a novel, efficient and effective development. Its use would also improve other things outside of showing images, including surveying and mapping, image recognition and classification, land use, land cover, and biomass estimation studies. For example, it is impossible to measure accurately the area of a building from a mosaic where seams cross the building. However, there are several potential pathways to improve.

First, the performance of the proposed method relies on the level of correlation or matching degree between vector buildings and the images to be mosaicked. To ensure this, vector building maps need to be updated in real time. Generally, large-scale construction projects are becoming less common in more and more cities in the world, particularly in China. Therefore, discrepancies between vector building maps and images are similarly becoming less of an issue. When new construction areas do dominate the resultant overlapping area, the proposed method will be invalid for determination of candidate seams. In such a scenario, the proposed method will strongly rely on Dijkstra's algorithm. Users can expand the search area for refining candidate seams, but some salient objects may also be crossed because Dijkstra's algorithm does not limit the crossing of the maximum-cost pixels, as discussed in Section 3.5. Some advanced strategies, such as the AVDO-based seam network refinement approach (Pan et al., 2014a) and the graph weighting strategy (Yu et al., 2012), could be applied to improve the resultant seams. The former combines the bottleneck model and Dijkstra's algorithm for refinement of the seam network. The latter considers all variables related to image appearance (i.e., color, edge and texture), image saliency and location constraints to find optimal seams. The influence of matching degree on mosaicking results is also analyzed based on three groups of test data. Experiments show that building maps with a horizontal accuracy of 5 m and height accuracy of 5 m are accurate enough to perform seam determination, but more data with lower accuracy should be analyzed to test the proposed method further.

Furthermore, the total processing time of the proposed novel method is less than that of Dijkstra's algorithm, but it remains much greater than that of the vector-road-based method for mosaicking images from areas with densely distributed buildings, such as downtown and suburban areas. This is because generation of the centerlines between buildings is very time-consuming. The

time complexity is $O(N^2)$. The computation time for generation of centerlines between buildings can be reduced in two known ways: (1) Incorporate a higher efficiency Delaunay triangulation algorithm, e.g., the divide-conquer Delaunay triangulation algorithm (Chew, 1989), whose time complexity is $O(N \log N)$. This technique can dramatically reduce the computation time without changing generated centerlines. (2) Combine adjacent buildings to reduce computation vertices (N). This approach can reduce computation time without changing the algorithm but may generate different or invalid centerlines. Combining these processes may provide an optimal balance between computation time and quality.

As the vector-road-based method has already been confirmed efficient for seam detection in the previous work, the proposed use of a vector-building-based method in combination with vector-road and other vector data may produce seams of higher quality. This will be addressed in our future work.

Acknowledgments

This work was done with financial support from the National Natural Science Foundation of China (41501416), China Agriculture Research System (CARS-35), the National Key Technology R&D Program of China (2013BAC03B00), and the open foundation of the Key Laboratory of Precise Engineering & Industry Surveying of the National Administration of Surveying, Mapping & Geoinformation (PF2015-17; F2013-14). The orthoimages and vector building map used in this study were provided by the Wuhan Surveying and Mapping Institute, China.

References

Agarwala, A., Dontcheva, M., Agrawala, M., Drucker, S., Colburn, A., Curless, B., Salesin, D., Cohen, M., 2004. Interactive digital photomontage. ACM Trans. Graph. 23, 294–302.

Ai, H.B., Zhang, L., Wang, L., 2011. Automatic mosaicking method for large block of orthophotos. Mippr 2011: Parallel Processing of Images and Optimization and Medical Imaging Processing 8005, 8.

Chen, Q., Sun, M., Hu, X., Zhang, Z., 2014. Automatic seamline network generation for urban orthophoto mosaicking with the use of a digital surface model. Remote Sens. 6, 12334–12359.

Chen, T., Ai, T., 2004. Automatic extraction of skeleton and center of area feature. Geomat. Inform. Sci. Wuhan Univ. 29 (443–446), 455.

Chew, L.P., 1989. Constrained Delaunay triangulations. Algorithmica 4, 97–108.

Choi, J., Jung, H.-S., Yun, S.-H., 2015. An efficient mosaic algorithm considering seasonal variation: application to KOMPSAT-2 satellite images. Sensors 15, 5649–5665.

Chon, J., Kim, H., Lin, C.-S., 2010. Seam-line determination for image mosaicking: a technique minimizing the maximum local mismatch and the global cost. ISPRS J. Photogram. Remote Sens. 65, 86–92.

Díaz-Varela, R., de la Rosa, R., León, L., Zarco-Tejada, P., 2015. High-resolution airborne UAV imagery to assess Olive tree crown parameters using 3D photo reconstruction: application in breeding trials. Remote Sens. 7, 4213.

Dijkstra, E.W., 1959. A note on two problems in connexion with graphs. Numer. Math. 1, 269–271.

Erfanifard, Y., StereĹ, K., Behnia, N., 2014. Parameter optimization of image classification techniques to delineate crowns of coppice trees on UltraCam-D aerial imagery in woodlands. J. Appl. Remote Sens. 8 (083520-083520).

Hsu, S., Sawhney, H.S., Kumar, R., 2002. Automated mosaics via topology inference. IEEE Comput. Graph. Appl. 22, 44–54.

Kerschner, M., 2001. Seamline detection in colour orthoimage mosaicking by use of twin snakes. ISPRS J. Photogram. Remote Sens. 56, 53–64.

Li, X., Shao, G., 2014. Object-based land-cover mapping with high resolution aerial photography at a county scale in midwestern USA. Remote Sens. 6, 11372.

Liang, X.W., Liu, Z.Q., Chen, Y.J., 2005. An algorithm of polygon auto-construction based on angle changing tendence. J. Image Graph. 6, 785–789.

Lin, C.-H., Chen, B.-H., Lin, B.-Y., Chou, H.-S., 2016. Blending zone determination for aerial orthimage mosaicking. ISPRS J. Photogram. Remote Sens. 119, 426–436.

Milgram, D.L., 1977. Adaptive techniques for photomosaicking. IEEE Trans. Comput. C-26, 1175–1180.

Mills, S., McLeod, P., 2013. Global seamline networks for orthomosaic generation via local search. ISPRS J. Photogram. Remote Sens. 75, 101–111.

Morrison, P., Zou, J.J., 2007. Triangle refinement in a constrained Delaunay triangulation skeleton. Pattern Recogn. 40, 2754–2765.

Mumtaz, R., Palmer, P., 2013. Attitude determination by exploiting geometric distortions in stereo images of DMC camera. IEEE Trans. Aerosp. Electron. Syst. 49, 1601–1625.

Nguyen Minh, N., Hoang Van, K., Nguyen Vinh, N., 2009. A Fast Algorithm for Constructing Constrained Delaunay Triangulation.

Pan, J., Wang, M., Li, D., Li, J., 2009. Automatic generation of seamline network using area voronoi diagrams with overlap. IEEE Trans. Geosci. Remote Sens. 47, 1737–1744.

Pan, J., Wang, M., Li, J., Yuan, S., Hu, F., 2015. Region change rate-driven seamline determination method. ISPRS J. Photogram. Remote Sens. 105, 141–154.

Pan, J., Wang, M., Ma, D., Zhou, Q.H., Li, J.L., 2014a. Seamline network refinement based on area voronoi diagrams with overlap. IEEE Trans. Geosci. Remote Sens. 52, 1658–1666.

Pan, J., Zhou, Q., Wang, M., 2014b. Seamline determination based on segmentation for urban image mosaicking. IEEE Geosci. Remote Sens. Lett. 11, 1335–1339.

Pang, S., Sun, M., Hu, X., Zhang, Z., 2016. SGM-based seamline determination for urban orthophoto mosaicking. ISPRS J. Photogram. Remote Sens. 112, 1–12.

Pradhan, R., Kumar, S., Agarwal, R., Pradhan, M.P., Ghose, M., 2010. Contour line tracing algorithm for digital topographic maps. Int. J. Image Process. (IJIP) 4, 156–163.

Roberts, S.A., Hall, G.B., Boots, B., 2005. Street Centreline Generation With an Approximated Area Voronoi Diagram. Springer-Verlag Berlin, Berlin.

Selvi, H.Z., Oztug Bildirici, I., Yerci, M., 2010. Triangulation method for area-line geometry-type changes in map generalisation. Cartogr. J. 47, 157–163.

Soille, P., 2006. Morphological image compositing. IEEE Trans. Pattern Anal. Mach. Intell. 28, 673–683.

Tang, Y.Y., You, X.G., 2003. Skeletonization of ribbon-like shapes based on a new wavelet function. IEEE Trans. Pattern Anal. Mach. Intell. 25, 1118–1133.

Vatti, B.R., 1992. A generic solution to polygon clipping. Commun. ACM 35, 56–63.

Wan, Y.C., Wang, D.L., Xiao, J.H., Lai, X.D., Xu, J.Z., 2013. Automatic determination of seamlines for aerial image mosaicking based on vector roads alone. ISPRS J. Photogram. Remote Sens. 76, 1–10.

Wang, D.L., Wan, Y.C., Xiao, J.H., Lai, X.D., Huang, W.L., Xu, J.Z., 2012. Aerial image mosaicking with the aid of vector roads. Photogram. Eng. Remote Sens. 78, 1141–1150.

Wang, M., Yuan, S., Pan, J., Fang, L., Zhou, Q., Yang, G., 2016. Seamline determination for high resolution orthoimage mosaicking using watershed segmentation. Photogram. Eng. Remote Sens. 82, 121–133.

Yu, L., Holden, E.J., Dentith, M.C., Zhang, H.K., 2012. Towards the automatic selection of optimal seam line locations when merging optical remote-sensing images. Int. J. Remote Sens. 33, 1000–1014.

Zhao, P., Feng, C., Li, B.L., 2004. Efficient algorithm for line clipping against general polygon. J. Southwest Jiaotong Univ. 39, 64–68.