Blending Zone Determination for Aerial Orthimage Mosaicking

Chao-Hung Lin, Bo-Heng Chen, Bo-Yi Lin, and Han-Szu Chou

Department of Geomatics, National Cheng Kung University, Taiwan

E-mail: linhung@mail.ncku.edu.tw

Abstract

Creating a composed image from a set of aerial images is a fundamental step in orthomosaic generation. One of the processes involved in this technique is determining an optimal seamline in an overlapping region to stitch image patches seamlessly. Most previous studies have solved this optimization problem by searching for a one-pixel-wide seamline with an objective function. This strategy significantly reduced pixel mismatches on the seamline caused by geometric distortions of images but did not fully consider color discontinuity and mismatch problems that occur around the seamline, which sometimes cause mosaicking artifacts. This study proposes a blending zone determination scheme with a novel path finding algorithm to reduce the occurrence of unwanted artifacts. Instead of searching for a one-pixel-wide seamline, a blending zone, which is a *k*-pixel-wide seamline that passes through high-similarity pixels in the overlapping region, is determined using a hierarchical structure. This strategy allows for not only seamless stitching but also smooth color blending of neighboring image patches. Moreover, the proposed method searches for a blending zone without the pre-process of highly mismatched pixel removal and additional geographic data of road vectors and digital surface/elevation models, which increases the usability of the approach. Qualitative and quantitative analyses of aerial images demonstrate the superiority of the proposed method to related methods in terms of avoidance of passing highly mismatched pixels.

Keywords: image mosaicking, optimal path determination, dynamic programming

1. Introduction

Seamless image mosaicking has gained attention in the fields of photogrammetry and image processing because of the increasing use of digital aerial imagining system. Composing multiple aerial images into a single mosaicking image is necessary to generate an image that covers an entire region. The visual transitions between overlapping images should also be imperceptible. Aerial images are generally orthorectified with a digital elevation model (DEM) or digital surface model (DSM) before mosaicking. The tilt and relief displacement can be corrected. However, uncontained or inaccurate objects in DEM or DSM cannot be orthorectified correctly, which may cause geometric distortions and color variations around the seamline of adjacent images taken at different times and from different angles (Wan et al., 2013). Naïve approaches namely, image stacking and linear color blending, have been proven inappropriate in such cases. Thus, considerable research attention has been devoted to seamless and smooth image mosaicking.

The main step in image mosaicking is seamline determination. Previous methods (Botterill et al., 2010; Chon et al., 2010; Fernandez et al., 1998; Kerschner, 2001; Mills and McLeod, 2013; Pan et al., 2015; Wan et al., 2013; Yu et al., 2012) search for a seamline with the optimal value of a defined objective function and stitch images using the determined seamline. Kerschner (2001) proposed the twin snake method, which uses two snakes or called active lines that start from the opposite borders of an overlapping area. These two snakes attract each other and change their shapes dynamically. The optimal seamline is determined when these two snakes merge. A technique that minimizes local mismatches is proposed by Chon et al. (2010). The highly mismatched pixels are removed in preprocessing to alleviate geometric distortion, and a shortest path algorithm is applied to search for the optimal seamline in the remaining pixels. Wan et al. (2012, 2013) and Wang et al. (2012) focused on ways to prevent from seamlines crossing objects and buildings that have higher elevations than the ground. A seamline that follows the centerlines of roads or the skeletons of overlapping regions is determined with the aid of vector road information. Pan et al. (2012) integrated the differences of pixel gradients and colors in the objective function to avoid seamlines passing through object edges. Yu et al. (2012) introduced an integrated objective function that considers color differences, edge features, textures, and image saliency. The features are combined as a weighted sum to represent the overall cost of a pixel in the image. Recently, several advanced algorithms such as change detection and image matching are used with pathing finding algorithm to avoid seamlines passing objects. Pan et al. (2014, 2015) utilized segmentation and region change rate in seamline determination. Mean shift algorithm and change detection are performed to select the preferred regions, and Dijkstra's algorithm with differential cost is adopted to determine an optimal seamline in the defined regions. Pang et al. (2016) adopted semi-global matching method to identify obstacle and non-obstacle areas prior to the seamline determination. Li et al. (2016) used graph cut algorithm for multiple image mosaicking. In addition, Pan et al. (2009 and 2014) and Mills and McLeod (2013) presented systems that generate seamline networks using graph-based approaches with the aid of Voronoi diagrams. Similarly, Chen et al. (2014) guided the seamline network passing through low-altitude areas based on the elevation information in DSM. The abovementioned methods yield good results. However, color discontinuity and seam artifact may still exist around the seamline after mosaicking and color blending because that only one-pixel-wide seamline is considered in the optimization problem.

To ease the artifact problem, a *k*-pixel-wide blending zone is considered in image mosaicking. In the proposed method, a semi-optimal blending zone is extracted efficiently to stitch the neighboring image patches and smoothen color transition in the overlapping region. The blending zone is determined in a coarse-level rather than the finest-level of image pyramids to consider computational efficiency. Moreover, inspired by the method of Chon et al. (2010), which removes highly mismatched pixels with an optimal threshold value prior to the determination of a seamline, a new objective function with an extension of Dijkstra's algorithm is proposed to search for a seamline that passes though homogenous regions rather than the shortest seamline that may pass highly mismatched pixels. The proposed method has the advantages of searching for a seamline without the requirements of high-cost pixel thresholding and additional geographic data, compared with the related methods (Chen et al., 2014; Chon et al., 2010; Wang et al., 2012; Wan et al., 2013). Experimental results confirmed that the obtained blending zone tends to pass broad homogenous channels, such as wide roads, which can alleviate the problem of discontinuous color transition in the color blending step. The remainder of this paper is organized as follows. Section 2 presents the objective function and blending zone determination method. Section 3 discusses the experimental results. Section 4 presents the conclusions, and discusses limitations of the proposed method and future work.

2. Blending Zone Determination

2.1 Objective Function

The measurement of pixel mismatch in the objective function is used in previous studies to search for a one-pixel-wide seamline. However, the pixel mismatches that occur around the seamline are not considered and sometimes cause artifacts after color blending. By considering both seamline determination and transition smoothness, which is the color transition from one image path to another around the determined seamline, the objective function is defined to search for a *k*-pixel-wide blending zone. Using the measurement of pixel color difference, the objective function measuring the cost of a blending zone **Z** is defined as

$$\psi(\mathbf{Z}) = \frac{1}{N_z} \sum_{p \in \mathbf{Z}} |I^a(p) - I^b(p)|, \tag{1}$$

where N_z denotes the number of pixels in zone **Z**, and $I^a(p)$ and $I^b(p)$ are the intensities of the corresponding pixels in overlapping regions of the images I^a and I^b, respectively. Note that other definitions of pixel costs, such as pixel gradient (Pan et al., 2012) and differential cost (Pan et al., 2015), can be integrated into or replace Eq. (1).

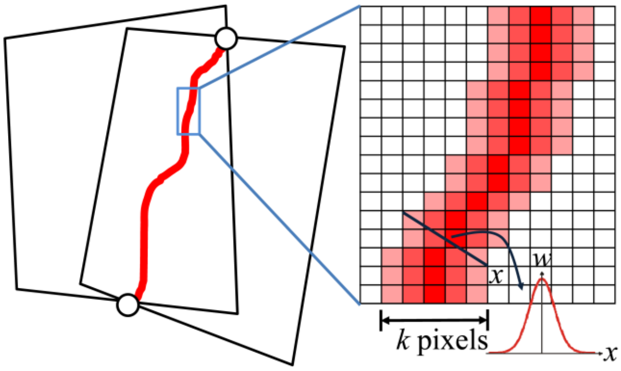

Figure 1. Illustration of the blending zone. The blending zone is a k-pixel-wide seamline with a Gaussian weighting function for color blending and mosaicking. The pixels in the centerline of the blending zone have the highest weight.

Considering color discontinuity around the seamline, the previous studies (Brown and Lowe 2007; Wang and Ng 2012) applied a color blending algorithm with weighting function to the neighborhood of the determined one-pixel-wide seamline. The weighting function is generally defined as the inverse distance from a pixel to the seamline. By integrating the weighting function employed in color blending into the objective function, Eq. (1) is reformulated as

$$\psi(\mathbf{Z}) = \frac{\sum_{p\in\mathbf{Z}}(|I^a(p)-I^b(p)|\times w_p)}{N_z\times\sum_{p\in\mathbf{Z}} w_p}, \tag{2}$$

where w_p is the weight of pixel p, which is defined as the inverse Gaussian of the distance between p and the centerline of the determined zone, as shown in Figure 1. Solving the optimization in Eq. (2) requires an exhaustive search that is computationally intensive. Specifically, the weight calculation depends on the centerline of the determined zone; however, the zone determination requires information on pixel weights. To reduce the computational cost, the pixel differences multiplying weights in Eq. (2) is reduced to a convolution scheme. The convolution scheme is an integral of the point-wise multiplication of the pixel differences and the Gaussian weighting function. In the implementation, a $k \times k$ Gaussian kernel, denoted as G, is used, and the pixels in the centerline of the zone are applied to the convolution, that is,

$$\psi(\mathbf{Z}) \cong \psi(\mathbf{S}) = \frac{1}{N_S}\sum_{p\in\mathbf{S}}(G \circledast |I^a(p) - I^b(p)|), \tag{3}$$

where **S** is the one-pixel-wide centerline of zone **Z**. N_S denotes the number of pixels in **S**, and

“*” represents the convolution operator. In this manner, the resulting cost of seamline **S** (i.e.,$\psi(\mathbf{S})$) approximates to the cost of zone **Z** calculated in Eq. (2) (i.e., $\psi(\mathbf{Z})$), and a semi-optimal *k*-pixel-wide blending zone can be obtained.

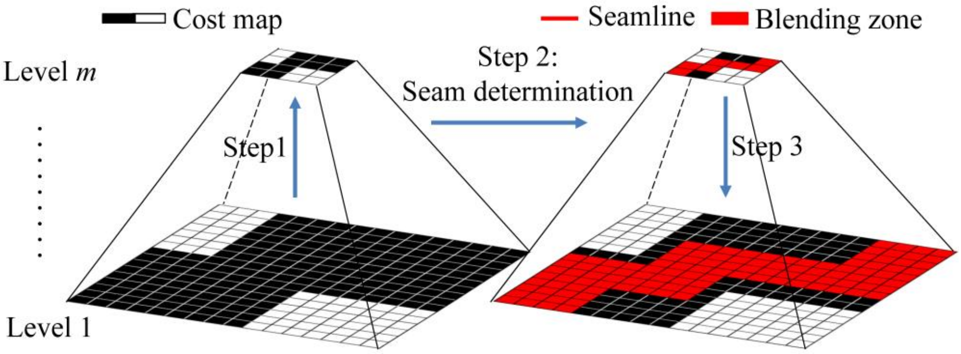

Figure 2. Illustration of the hierarchical structure in the proposed blending zone determination. The blending zone is obtained by projecting the one-pixel-wide seamline in level *m* to level 1 of the image pyramid.

The optimization equation in Eq. (3) is solved using a hierarchical structure, as illustrated in Figure 2, to further consider computational efficiency. The smoothing filter with Gaussian kernel is applied to the overlapping images. That is, the image Gaussian pyramids for images I^a and I^b are constructed, and a seamline is determined in a specified level of the image pyramids, which are denoted by I_m^a and I_m^b. Hence, Eq. (3) is simplified as

$$\psi(\mathbf{S}_m) = \frac{1}{N_{S_m}} \sum_{p \in \mathbf{S}_m} |I_m^a(p) - I_m^b(p)|, \tag{4}$$

where N_{S_m} is the number of pixels in the seamline $\mathbf{S}_m$ located in the *m*-th level of the image pyramid (denoted as L_m). After extracting seamline $\mathbf{S}_m$ using a path finding algorithm, which will be described in Section 2.2, the blending zone can be obtained by projecting the seamline $\mathbf{S}_m$ from L_m to L_1 (the finest level) of the image pyramid, that is,

$$\psi(\mathbf{Z}^*) \cong \psi(\mathbf{Z}) = \frac{\sum_{p \in \mathbf{Z}} (|I_1^a(p) - I_1^b(p)| \times w_p)}{N_{\mathbf{Z}} \times \sum_{p \in \mathbf{Z}} w_p}, \tag{5}$$

where $\mathbf{Z}^*$ is the zone obtained by projecting $\mathbf{S}_m$ from L_m to L_1 of the pyramid. $I_1^a(p)$ and $I_1^b(p)$ denote the corresponding pixels in L_1 of the image pyramids. Thus, a semi-optimal blending zone can be obtained, and the cost of the obtained blending zone approximates to the cost of the optimal blending zone, that is, $\psi(\mathbf{Z}) \cong \psi(\mathbf{Z}^{*})$.

2.2 Path Finding Using Greedy Strategy

Following the previous study (Chon et al., 2010), the two intersecting pixels of the borders of overlapping images are selected as starting and ending pixels of the seamline (see Figure 1). An extension of the standard shortest path algorithm called Dijkstra's algorithm is performed to search for the optimal seamline in a specified level of the cost map calculated based on color difference. The overlapping region is regarded as an eight-connectivity graph, as illustrated in Figure 3, which means that the pixels adjacent to (i, j) are (i-1, j-1), (i-1, j), (i-1, j+1), (i, j-1), (i, j+1), (i+1, j-1), (i+1, j), and (i+1, j+1). Given the starting pixel *s* and the ending pixel *e*, Dijkstra's algorithm finds the shortest path, that is, the minimal-cost path, between these two pixels using greedy strategy. This algorithm repeatedly selects a pixel *v*, which is nearest to the starting pixel *s*, from the unselected pixel set, removes pixel *v* from the unselected pixel set, and declares the distance to be the actual shortest distance from *s* to *v*. The adjacent pixels of *v*, denoted as *w*, are then checked to determine if a shorter path can be found by passing through pixel *v* followed by the relevant outgoing edges of pixel *v*, that is,

$$sDist(w) = \min\{sDist(w),\ sDist(v) + Cost(w)\}, \forall w \in Neigh(v), \tag{6}$$

where $Neigh(v)$ is the adjacent pixel set of *v*, $sDist(w)$ represents the cost of the shortest path from the starting pixel *s* to *w*, and $Cost(w)$ denotes the cost of pixel *w*, that is, $|I_m^a(w) - I_m^b(w)|$. The iteration is terminated when all pixels in the overlapping region are visited. Next, the shortest path is extracted by backtracking the path from the ending pixel to the starting pixel.

The shortest path is a path with minimal total cost. However, the shortest path may pass through high-cost and highly mismatched pixels because this path prefers a shorter path with minimal total cost to a lengthy path with low-cost pixels. For instance, the shortest path generated by Dijkstra’s algorithm in Figure 4 passes through the pixel with the cost of 12 [pixel (2, 2)], which is the pixel with the highest cost in the test data. The high-cost pixel corresponds to high mismatch. People more easily perceive seam artifacts from a path containing highly mismatched pixels despite its shortness, compared with that from a lengthy path with low-cost pixels (Chon et al., 2010).

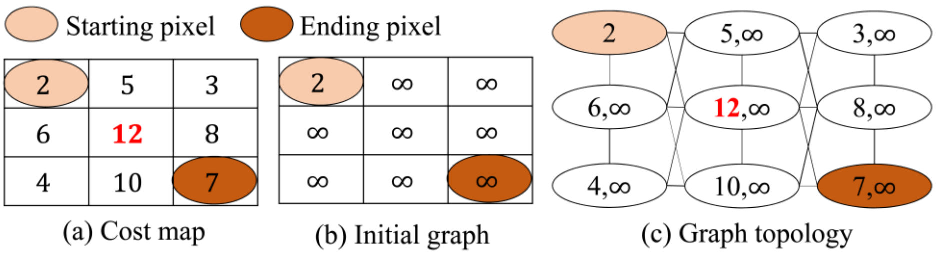

Figure 3. Illustration of graph topology in seamline determination. (a) A simulated cost map of size 3×3. (b) The initial cost of (a). The pixel costs are initially set to infinity except the starting pixel. (c) An eight-connectivity graph containing the information of (a) and (b).

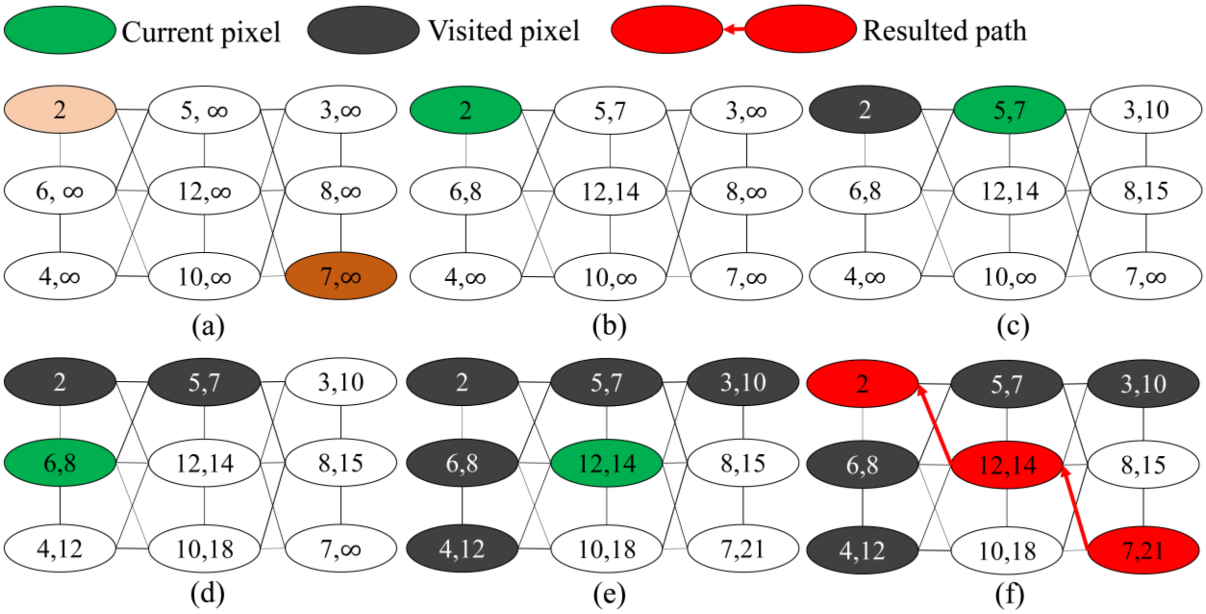

Figure 4. Illustration of Dijkstra’s algorithm. (a)–(e) Steps of the algorithm. (f) Shortest path result. The two numbers shown in each node represent the pixel cost and the total cost from the starting pixel to this pixel.

To solve this problem, Chon et al. (2010) proposed the removal of high-cost pixels with a determined threshold in the preprocessing, and Dijkstra’s algorithm is then applied to search for the optimal seamline in the remaining pixels. The threshold used to filter out high-cost pixels is vital for a successful path search. A seamline exists after thresholding, whereas the number of highly mismatched pixels is minimized in the remaining regions. Searching for the optimal threshold requires an iteration procedure that contains the steps of pixel thresholding and shortest path determination. Specifically, the algorithm iteratively probes for a threshold until a path between the starting and ending pixels can be found and the cost of path from Dijkstra's algorithm is minimal. This approach successfully improves Dijkstra's algorithm in terms of high-cost pixel avoidance and reduces the possibility of passing highly mismatched pixels. However, iteratively performing Dijkstra's algorithm is computationally insensitive for images of large sizes. In addition, the effect of removing significant mismatches is reduced for images of urban areas because low-mismatch paths cannot be found. The absence of these paths is because urban areas generally contain many ground objects with large relief displacements.

This study proposed a simple and efficient path finding algorithm without the determination of optimal thresholds and the preprocess of high-cost pixel removal. The basic idea is to consider path length in the path search optimization and adopt average-cost minimization instead of total-cost minimization in the objective function. The new objective function prefers a lengthy path that contains local low-cost pixels rather than a shorter path that contains high-cost pixels. Therefore, the problem in Dijkstra's algorithm can be alleviated effectively. Moreover, the path finding algorithm with this objective function is a non-iterative process and has computational efficiency, which is similar to Dijkstra's algorithm. This idea is realized by reformulating Eq. (6) as

$$avgDist(v) = min\left\{avgDist(w), \frac{avgDist(w) \times L(w) + Cost(v)}{L(w)+1}\right\}, \forall w \in Neigh(v), \qquad (7)$$

where $L(w)$ is the path length from the starting pixel s to the ending pixel w. $avgDist(v)$ represents the minimal average-cost of the path from s to w, which is equal to $sDist(v)/(L(w)+1)$ that means the shortest path distance $sDist(v)$ is normalized by the current path length $L(w) + 1$.

A comparison of path finding using the objective functions in Eqs. (6) and (7) is shown in Figure 5. The results indicate that a seamline from the proposed objective function can avoid passing locally high-cost pixels. However, the selected seamline with minimal average-cost tends to be circuitous in low-cost regions, making this seamline inappropriate for mosaicking. To solve this problem, the greedy search strategy is replaced by a breadth-first search strategy, which is described in Section 2.3.

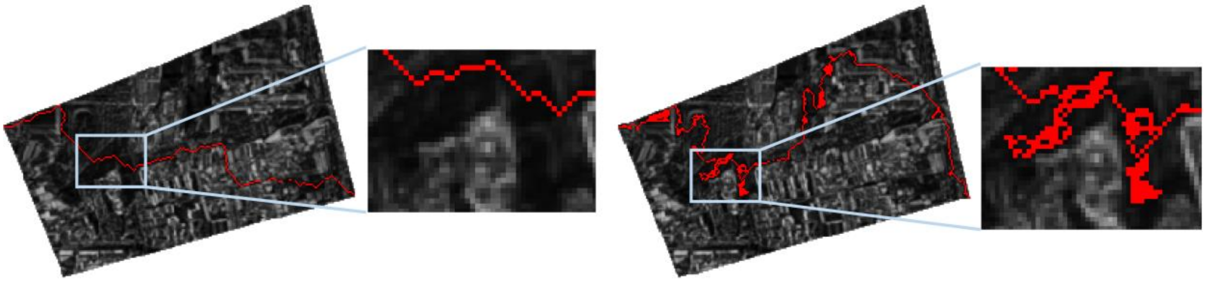

Figure 5. Comparison of different objective functions with greedy traversal strategy. Left: Total cost minimization in Eq. (6). Right: Average cost minimization in Eq. (7).

2.3 Path Finding Using Breadth-first Traversal Strategy

The greedy traversal strategy with objective function in Eq. (7) enables the seamline to move forward and backward to the ending pixel, which potentially making the path circuitous. To solve this problem, a breadth-first traversal, which traverses the graph in level order, is utilized. The method traverses the adjacent pixels in the same level before moving to the next level. Hence, the moving direction of the seamline from the starting to the ending pixels is restricted to the same level or forwarded on the graph structure, which means that path circuitousness is significantly reduced because of the avoidance of a backward traversal. An illustration of path finding using breadth-first traversal is shown in Figure 6. With this traversal strategy, the proposed method can select a semi-optimal seamline in terms of average-cost minimization rather than an optimal one with circuitous problems. An example of the proposed path finding algorithm is provided in Figure 7. The simulated data in Figure 3 is tested, where the starting node is pixel (1,1). The nodes in the first and second levels of the traversal are marked by blue polygons in Figures 7(d) and 7(e), respectively. This level-order traversal strategy causes the determined seamline to move forward toward the ending node, that is, pixel (3, 3). The determined path is longer than that of Dijkstra's algorithm in Figure 4, but the path does not pass the pixel with the highest cost, that is, pixel (2,2). Moreover, the circuitous problem shown in Figure 5 is avoided.

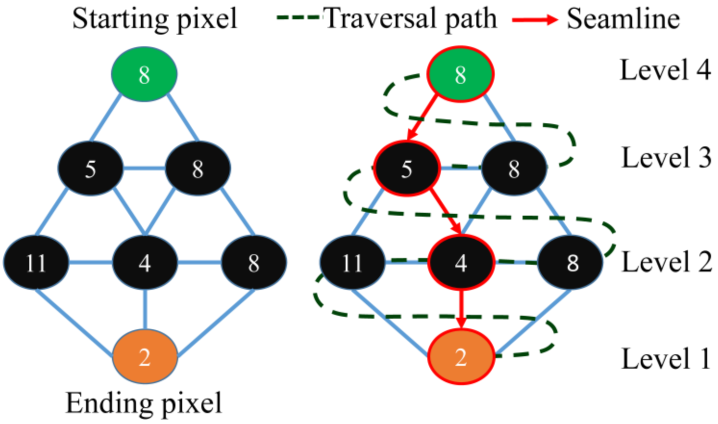

Figure 6. Illustration of breadth-first search. Left: Tested graph. Right: Seamline determination using breadth-first search. The dashed line denotes the search path and the red path represents the selected seamline.

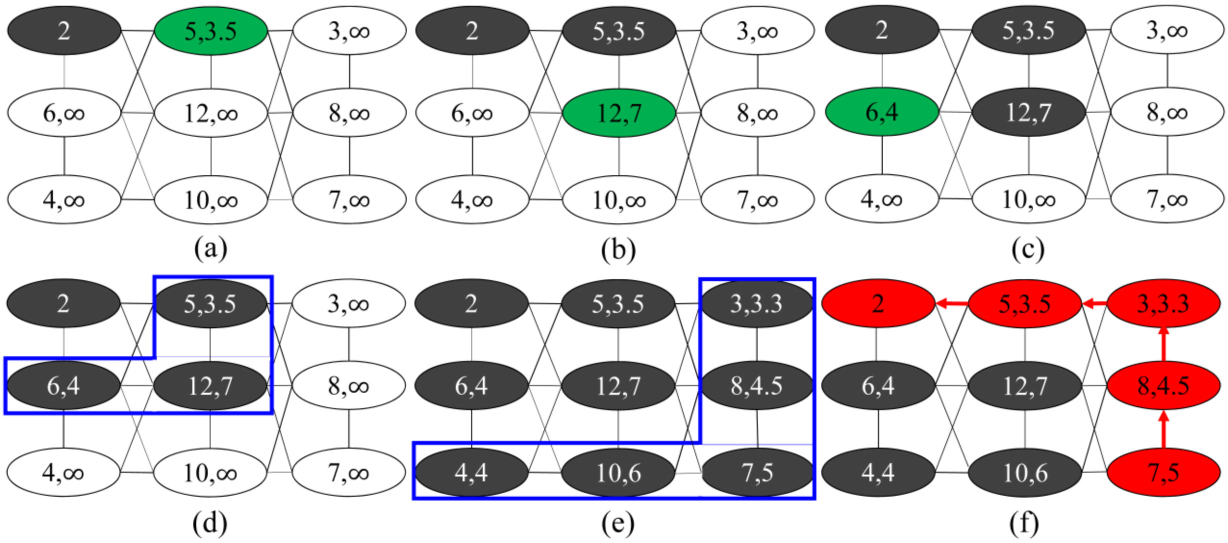

Figure 7. Illustration of the proposed path finding algorithm using breadth-first traversal. (a)–(e) Steps of the algorithm. The nodes in the polygons in (d) and (e) are the nodes in

the first and second levels of the graph structure, respectively.

The following pseudo-code briefly describes the proposed optimal path finding algorithm. The input to the algorithm is a set of pixels in the overlapping region, an adjacency list of graph, and a set of pixel costs. Initially, the shortest distance of the starting pixel to itself is set to the pixel cost, that is, avgDist[s]←Cost[s], and all other pixels are set to infinity to indicate that those pixels are not yet processed. The algorithm literately selects a pixel, denoted as *v*, from the unselected pixel set through function *BreathFirstSelection*() using the breath-first search strategy. Subsequently, *v* is removed from pixel set **V**, and Eq. (7) is performed until pixel set **V** is empty.

```
Input: V (the set of pixels 1..n)
/*s: the starting pixel; e: the ending pixel; Cost[1..n]: the pixel cost; avgDist[1..n]: the average
cost of the shortest path from the starting pixel to the other pixels.*/
Procedure PathFinding() {
    for all pixel i in V, avgDist[i]←∞;
    avgDist[s]←Cost[s];
    repeat
        v←BreathFirstSelection (V);
        // select the pixel v under breath-first search order.
        remove v from V;
        perform Eq. (7);
    until V is empty;
    backtrack the path from ending pixel e to starting pixel s and output the path;
}
```

2.3 Seamline Projection and Image Stitching

Once the seamline is determined in L_m of the image pyramid, this seamline is projected back to L_1 to form a blending zone. That is, the obtained one-pixel-wide seamline $\mathbf{S}_m$ in L_m

of the image pyramid corresponds to the k-pixel-wide blending zone $\mathbf{Z}^*$ in L_1, where $k = 2^m + 1$. Moreover, the centerline of seam-zone $\mathbf{Z}^*$ is extracted for image patch stitching. After the determination of blending zone, the colors of corresponding pixels in the overlapping region are blended to generate a smooth color transition in the determined blending zone. To reuse the image pyramid and Gaussian smoothing results, the multi-band color blending approach (Brown and Lowe, 2007) is adopted. The image patches in the blending zone are decomposed into several frequency-band components using a low-pass filter with a Gaussian kernel and the difference of Gaussian smoothing results. Different frequency bands of the overlapping image patches are then combined with an inverse-distance weighting function. Low-frequency components are mixed over a large spatial range, and fine details in the high-frequency components are blended over a short spatial range. Therefore, the ghost effects on the mismatched pixels can be eased further under this color blending scheme. Image stitching with the determined blending zone and color blending can generate a visually smooth transition and seamless mosaicking.

3. Experimental Results

3.1 Experimental Data

Two datasets containing 16 and 21 aerial images were tested, as shown in Table 1. Dataset A was acquired in 2012 using an unmanned aerial vehicle (UAV) at an average altitude of 400 meters with approximately 60% forward overlap and 40% side overlap. An UVA equipped with Canon IXUS 220HS with 24 mm focus length and 1.54 μm pixel size is used to acquire data. The ground resolution is 0.15 meters, and the image size is 4000 × 3000 pixels. Dataset B was acquired in 2013 at average altitude of 900 meters with approximately 80% forward overlap and 60% side overlap. A manned aircraft equipped with Sony A850 with 20 mm focus length and 5.95 μm pixel size is used to acquire data. The ground resolution is 0.1 meter, and the image size is 6048 × 4032 pixels. The images in Datasets A and B cover the urban and suburban areas in Kaohsiung and Tainan, Taiwan, respectively. The images in Dataset A have relatively higher density of buildings than that in Dataset B. Before mosaicking, all images were orthorectified to the Taiwan Datum 1997 for horizontal datum and Taiwan Vertical Datum 2001 for vertical datum with ground control points and DEM. Note that the ground objects in the UAV images exhibit large shifting or the so-called parallax effects. These parallax effects are attributed to the low altitude of the flying height and the presence of tall buildings in Dataset A. The buildings and parallax effects in the UAV images increase the difficulty of mosaicking.

Table 1. Study areas and experimental data.

Dataset	Image size (pixel)	Ground resolution (m)	Aircraft	Altitude (m)	Acquisition date	Study area
A	4000 × 3000	0.15	UAV	400	Nov., 2012	Kaohsiung, Taiwan
B	6048 × 4032	0.1	Manned aircraft	900	Nov., 2013	Tainan, Taiwan

3.2 Parameter Setting

The width of the blending zone, k, is the main parameter in the proposed approach. To test the sensitivity of mosaicking quality to this parameter, the proposed approach was tested using various values of k. The experimental results are shown in Figure 8. If a small k is used, e.g., k=9, the blending zone has a high possibility of passing through narrow alleys; however, the blending zone may also pass through a narrow homogenous channel in a high-cost region. If a large k is used, e.g., k=129, the blending zone prefers passing through wide roads and broad homogenous regions; however, the blending zone may miss alleys in urban areas. This experiment shows that the value of parameter k corresponds to the width of low-cost channels and roads. Searching for the optimal value of k without additional data and information is difficult and inefficient because the setting of this parameter is dependent on the input images. Therefore, the parameter k is turnable in the implementation. The parameter k is identical (k=33 and the number of levels in the image pyramid is 5) in all experiments for fair comparisons.

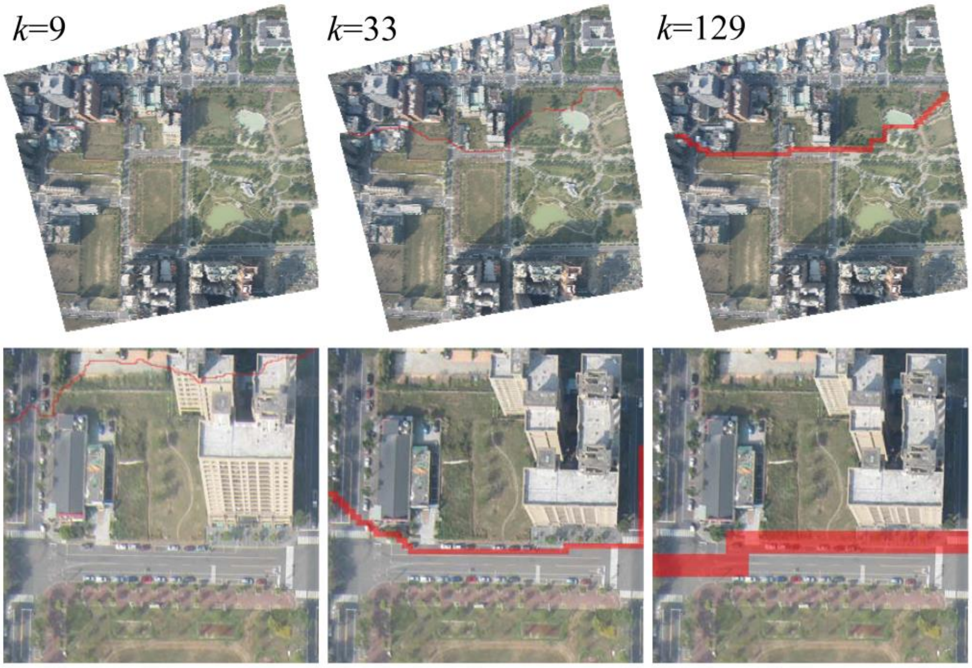

Figure 8. Comparison of blending zones using different parameter values, including *k*=9 (left), 33 (middle), and 129 (right). Top: Mosaicking results using the determined zones. The zone is displayed by transparent line segment. Bottom: Close-up views of the determined zones.

3.3 Evaluation of Path Finding Algorithm

The proposed algorithms were implemented by C++, and all experiments were conducted using a PC with a quad-core 2.6 GHz processor and 8 GB RAM. A single thread is adopted in the implementation, that is, only one CPU core is used. For an image pair in Dataset A, the average computation times for a one-pixel-wide seamline (calculated in L_1 of the image pyramid) and a 33-pixel-wide blending zone (calculated in L_5 and then the seamline is projected to L_1) are 3.35 s and 0.67 s, respectively. Hence, computation time for blending zone determination is improved by 80% under the hierarchical structure. Moreover, the complex optimization problem for a k-pixel-wide seam zone in Eq. (2) is reduced to Eq. (4), in which a seamline is determined from a coarse-level image and then projected to the finest-level of the pyramid. The search space and size of the seamline, that is, the number of pixels in a seamline, are reduced significantly. The computational time of the proposed path finding algorithm increases linearly with the number of the calculated pixels, that is, $N/2^m$ in L_m. Therefore, the algorithm time complexity is $O(N/2^m)$, which is the same with Dijkstra's algorithm, where N represents the number of pixels in L_1. In the method of Chon et al. (2010), the algorithm is performed in L_1 to remove the high-cost pixels while ensuring the existence of a seamline. The time complexity of this algorithm is $O(cN)$, where c is the number of iterations in the optimal threshold determination. The complexity analysis indicates that the proposed method does not increase computational loading compared with Dijkstra's algorithm while improving the seamline and blending zone results.

An objective function with breath-first traversal strategy is proposed to search for a seamline which contains local low-cost pixels rather than a shorter seamline that may have local high-cost pixels. To validate the effectiveness of this method, comparisons with Dijkstra's algorithm (DA), constrained Dijkstra's algorithm (CDA), which is proposed by Chon et al. (2010), and the shortest path algorithm with breadth-first traversal (SPB) were conducted. The path-finding approaches are compared using the images in L_5. The comparison results are shown in Figures 9–11. The seamlines generated by DA, CDA, SPB, and the proposed method are represented in the cost maps by the colors blue, green, yellow, and cyan, respectively. The proposed method can deal with color images for color orthomosaicks. For a visual comparison, the gray cost maps representing the color differences and pixel costs are shown in Figures 9-11. To check whether the determined seamlines can avoid passing through highly mismatched pixels, the high-cost pixels (cost>20) on the seamline are marked as red. From the results, the seamline generated by Dijkstra's algorithm is shorter than that generated by the proposed approach. However, the mismatch visualization shows that the seamline of the minimal total cost contains many high-cost pixels, that is, the pixels marked by red, which may result in color discontinuity. In contrast, the proposed approach considers the path length and generates a lengthy path that potentially avoids passing through high-cost pixels, which results in seamless mosaicking. CDA generally performs better than DA because of the removal of high-cost pixels prior to seamline determination. However, large relief displacements and high-density buildings in the tested datasets make it difficult to remove all highly mismatched pixels while finding a path. Thus, the result of CDA is slightly better than that of DA. Comparing SPB and the proposed method, SPB uses minimal cost, whereas the proposed method adopts minimal average-cost as the objective function. Therefore, the length of seamline generated by SPB is between that of the proposed method and that of DA.

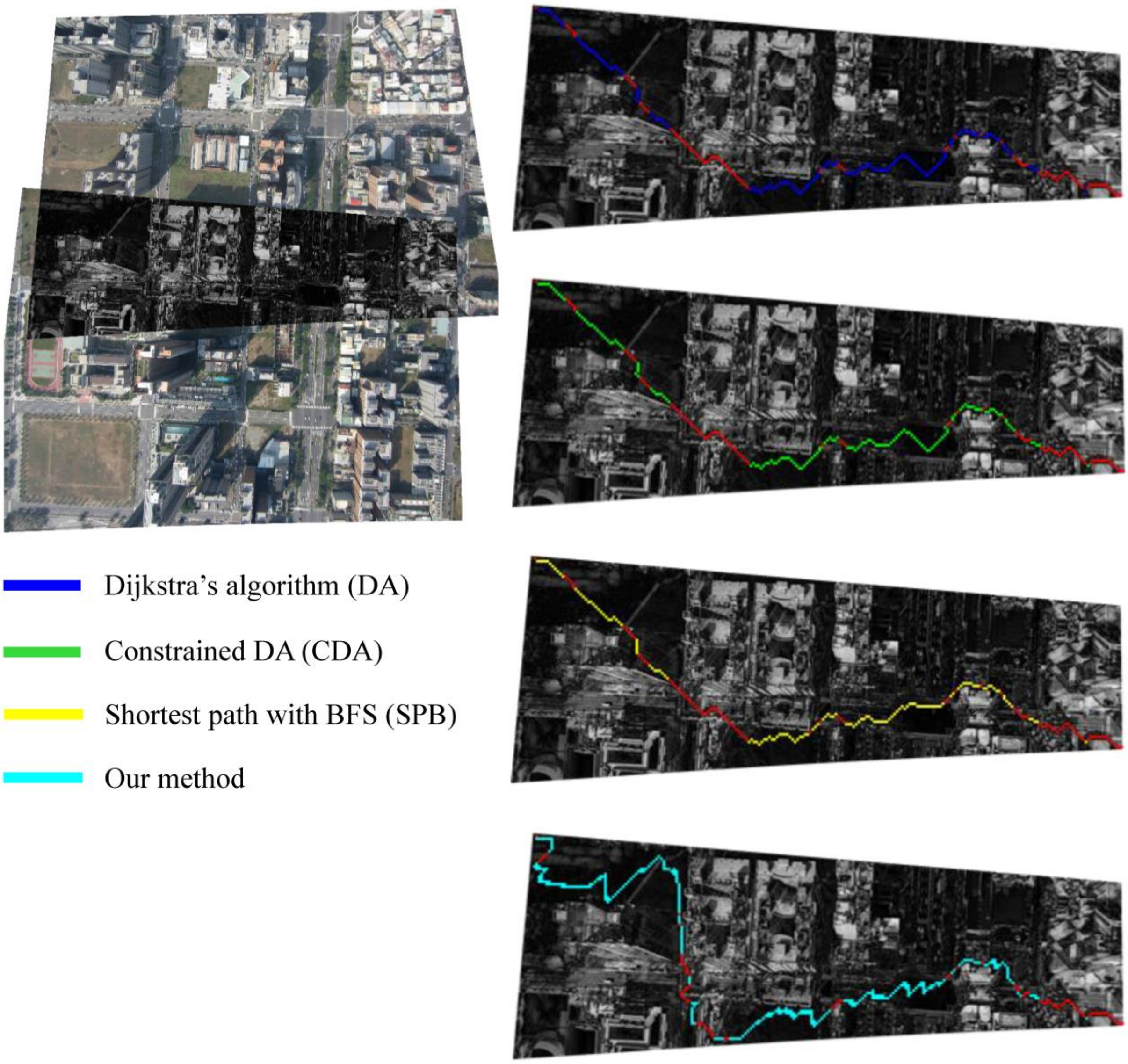

Figure 9. Comparison of the seamlines generated by DA, CDA, SPB, and the proposed method using an image pair in Dataset A. Left: Tested image pair. The overlapping region is visualized by pixel cost in gray levels. Right: Generated seamlines. The high-cost pixels are displayed on the seamline in different color (red).

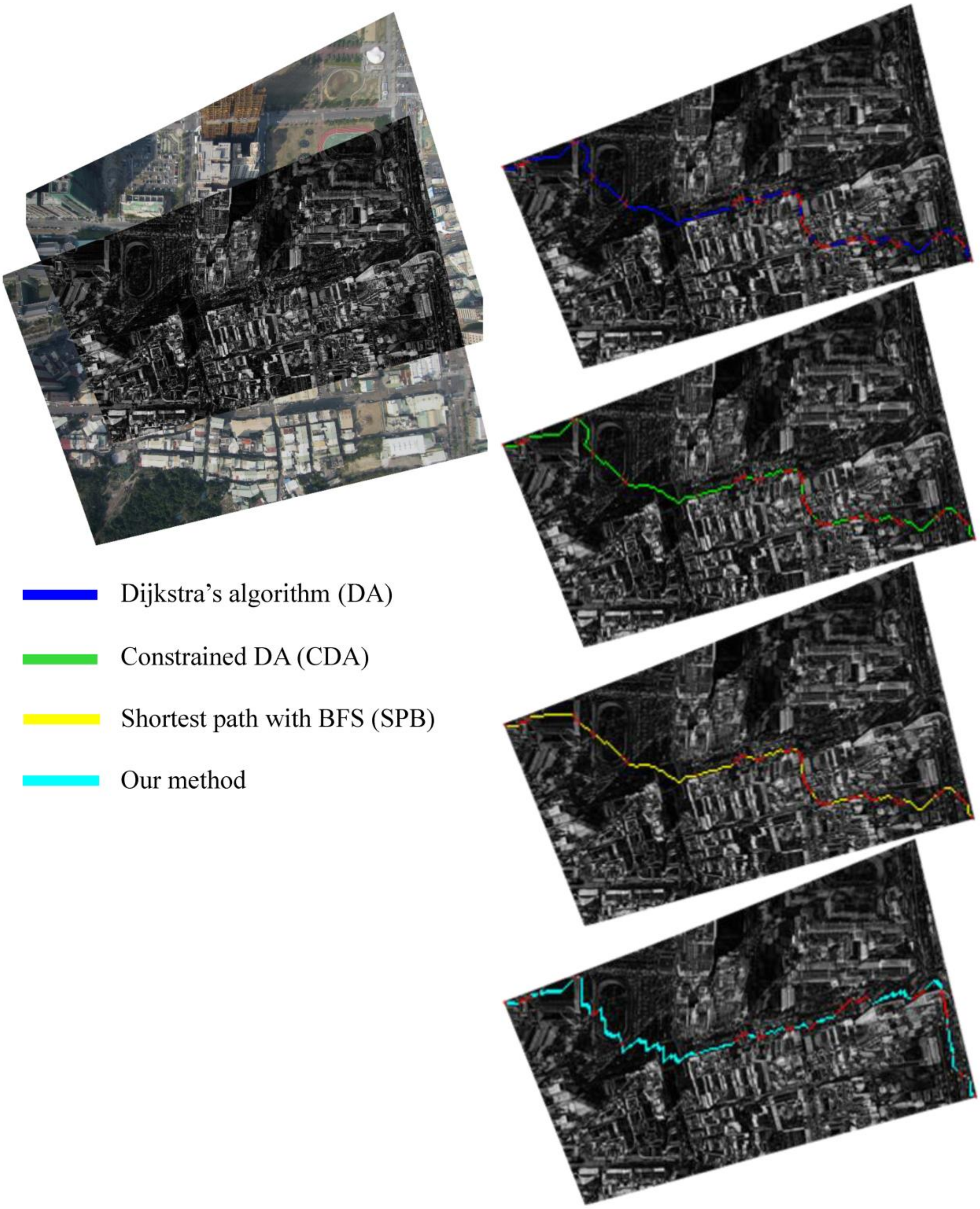

Figure 10. Comparison of the seamlines generated by DA, CDA, SPB, and the proposed method using an image pair in Dataset A. Left: Tested image pair. The overlapping region is visualized by pixel cost in gray levels. Right: Generated seamlines.

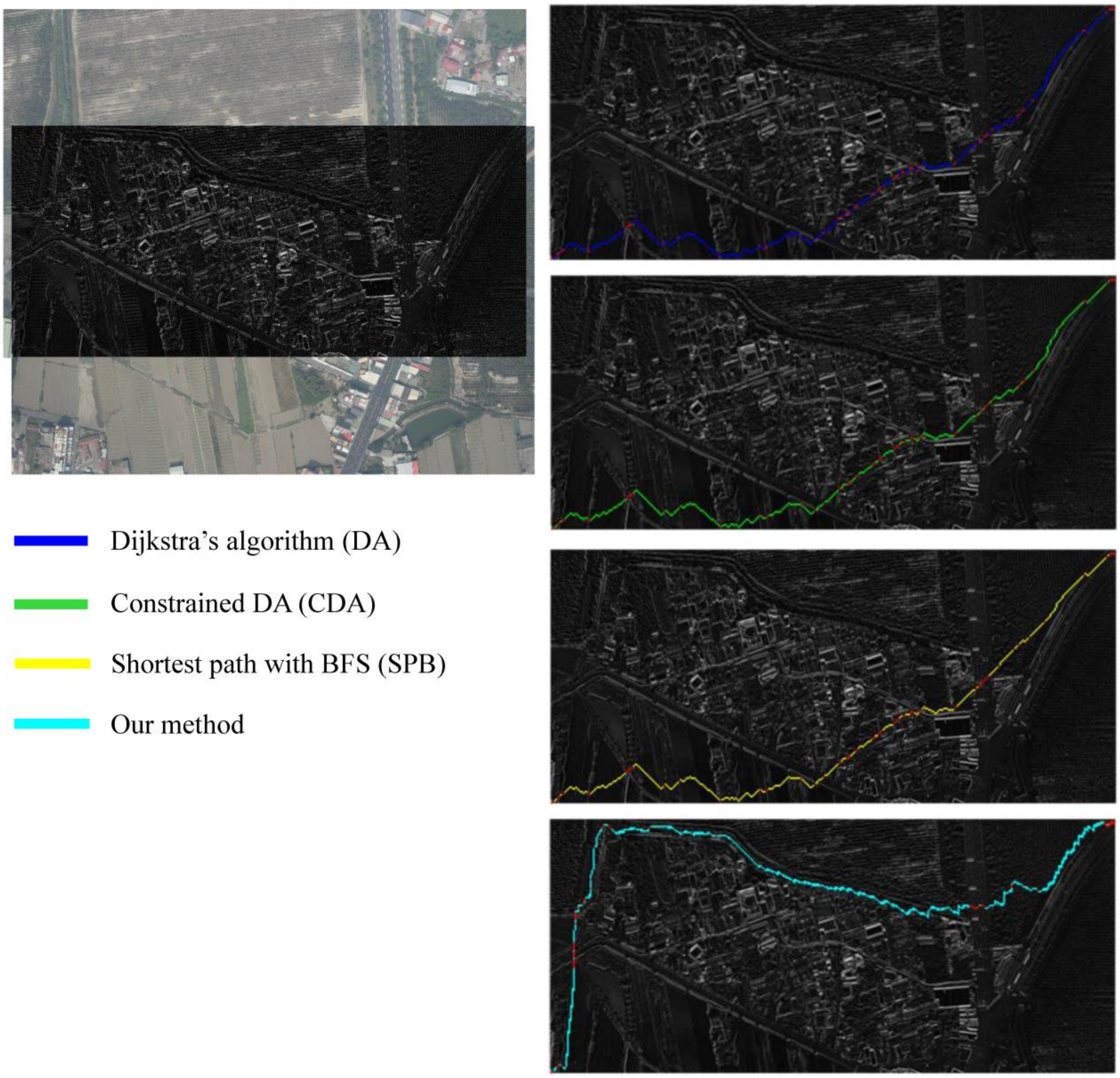

Figure 11. Comparison of the seamlines generated by DA, CDA, SPB, and the proposed method using an image pair in Dataset B. Left: Tested image pair. The overlapping region is visualized by pixel cost in gray levels. Right: Generated seamlines.

Two seamline quality measurements, called high-cost distortion (HD) and high-cost percentage (HP), are adopted to evaluate the pixel mismatch of a seamline in addition to the general statistics, average, standard deviation, and maximum of the pixel costs. HD is defined as the average cost of the top 10% high-cost pixels in the determined seamline, and HP refers to the percentage of pixels whose cost is higher than a specified threshold. The threshold is set to 20 in the experiments. HD and HP can reveal the highly mismatched information of a seamline. The measurement of the total cost was not used in the experiment because of the different lengths of seamlines in different test data. The statistical comparison results are shown in Table 2. The results show that the generated seamline has the lowest HD and HP. Hence, the proposed method has good performance on the highly mismatched pixel avoidance in seamline determination. Moreover, the lowest average and standard deviation of the pixel costs of the results also demonstrate the effectiveness of the proposed objective function. In addition, the proposed path

Table 2. Comparison of determined seamlines in Dataset A. The average (denoted by Avg.), standard deviation (Std.) and maximum (Max.) of the pixel costs, high-cost distortion (HD), and high-cost percentage (HP) are shown in the Table.

Dataset	Method	Avg.	Std.	Max.	HD	HP (%)	Length
Figure 9	SPB	19.03	16.29	69	71.98	38.96	238
	DA	18.52	16.33	69	71.98	38.96	243
	CDA	18.42	16.23	69	70.05	38.30	240
	Ours	13.25	13.42	69	58.53	20.91	406
Figure 10	SPB	18.25	17.13	79	62.67	34.70	263
	DA	17.56	16.06	59	63.54	35.03	273
	CDA	17.51	15.98	51	67.09	36.80	272
	Ours	14.45	14.73	98	54.91	22.52	420
Figure 11	SPB	15.82	10.52	60	45.59	17.66	378
	DA	15.49	10.44	60	45.59	17.66	385
	CDA	15.41	10.40	60	45.28	17.58	380
	Ours	13.81	7.80	60	39.31	9.24	757

finding method is efficient without the iterative process of optimal threshold determination for high-cost pixel removal compared with the method of (Chon et al., 2010). Compared with the related methods (Wan et al., 2012; Wang et al., 2012; Wan et al., 2013; Chen et al., 2014), the proposed method is easy to use because it does not require information of road vectors or elevation information in DSM. The generated seamline tends to pass low-cost roads in urban areas without any prior information (Figures 9–11). Hence, the visual comparisons in Figures 9–11 and the statistical comparison in Table 2 show that the proposed path finding method, which searches for a seamline using breadth-first-search traversal and the objective function of average-cost minimization, outperforms the related methods.

3.4 Evaluation of Blending Zone Determination

The comparison between *k*-pixel-wide blending zone and one-pixel-wide seamline using the proposed path finding algorithm was conducted to evaluate the effectiveness of the proposed blending zone. To allow for a fair comparison, the seamline is generated in L_1 of the pyramid and is then thickened to a zone of the same width with the blending zone calculated from L_5 of the pyramid. The comparison results are shown in Table 3. The results show that the blending zone calculated from L_5 is better than the seamline calculated from L_1 based on the measurements, HD, average, standard deviation, and maximum of the pixel costs. Visual

Table 3. Comparison between blending zones generated in L_5 and thickened zone from the seamline in L_1. "#Pixels" represents the number of pixels in blending zone.

Dataset	Blending Zone from L_5					Thickened Zone from Seamline in L_1				
	Avg.	Std.	Max.	HD	# Pixels	Avg.	Std.	Max.	HD	# Pixels
Figure 9	14.10	17.81	203	58.53	11.19e+4	25.14	33.80	228	110.20	10.70e+4
Figure 10	13.94	17.29	177	54.90	11.52e+4	20.15	25.14	207	81.27	11.39e+4
Figure 11	14.57	11.91	172	39.31	20.61e+4	21.24	21.74	220	71.98	18.66e+4

comparisons in Figure 12 show that the seamline tends to pass narrow and homogenous channels, whereas the blending zone prefers broad and local low-cost channels. Therefore, the blending zone can alleviate mismatch and color discontinuity problems occurring around a seamline when image blending is performed after seamline determination. Table 3 also indicates that HD in the blending zone is improved (from HD=71.98 to HD=39.31), compared with the thickened seamline.

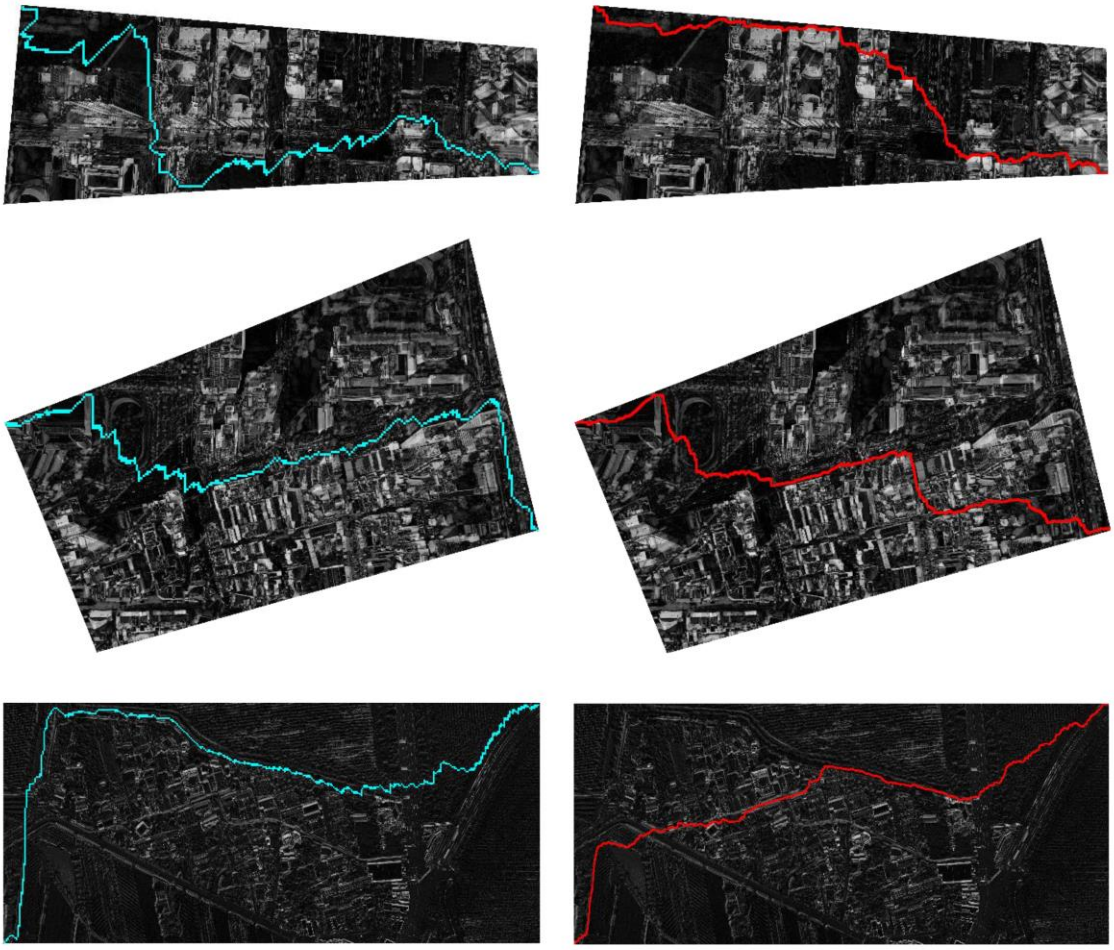

Figure 12. Visual comparison of the blending zone calculated from L_5 (left) and thickened seamline from L_1 (right) of the image pyramid. The blending zones in the left figures and seamlines in the right figures are displayed by line segments.

4. Conclusions, Limitation and Future Work

An image mosaicking scheme was introduced in this study. Considering both seamline determination and color blending, a semi-optimal blending zone is extracted instead of a seamline. This blending zone is used in image patch stitching and color blending of overlapping patches, which can ease pixel mismatch and color discontinuity problems efficiently. The blending zone is determined in the hierarchical structure with the proposed path finding algorithm. The proposed approach can obtain a semi-optimal path that contains local low-cost pixels rather than the shortest and minimal-cost path that sometimes contains high-cost pixels. The experiments demonstrate that local high-cost pixels are avoided without the preprocessing of high-cost pixel removal and additional information, such as road vectors and DSM. Moreover, the qualitative and quantitative analyses of the UAV orthoimages demonstrate the superiority of the proposed method over related mosaicking methods in terms of seamless mosaicking. Moreover, the iteration process for optimal threshold determination is avoided, and the complex blending zone optimization problem is simplified through the extraction of seamline in a coarse-level of the image pyramid. Search space and seamline size are reduced, thereby reducing the computational cost.

At present, the proposed approach has difficulties in efficiently and automatically finding an appropriate value for parameter *k*, which is used to determine the width of the blending zone, because this parameter varies with the width of local homogenous regions in the input images. In the future, we plan to develop an adaptive approach for parameter *k* determination. The parameter value is determined based on the widths of local low-cost regions in an image pair. We also plan to apply the proposed path finding algorithm to seamline networks for multiple image mosaicking. Moreover, the out-of-core external mosaicking will be implemented using multi-core CPU and GPU, the information reconstruction technique (Lin et al., 2013) will be applied to the aerial images, and the mosaicking scheme integrated to generate a cloud-free image.

Acknowledgments

The authors would like to thank the anonymous reviewers for their valuable comments and suggestions. We are also grateful to Prof. Jiann-Yeou Rau for providing images. This work was supported by the Ministry of Science and Technology (Contracts MOST 104-2628-E-006-003-MY3 and MOST 104-2628-M-006-002-MY3), Taiwan.

References

Adelson, E. H., Anderson, C. H., Bergen, J. R., Burt, P. J., Ogden, J. M., 1984. Pyramid method in image processing. RCA Engineer 29 (6), 33-41.

Botterill, T., Mills, S., Green, R., 2010. Real-time aerial image mosaicking. 25th International Conference of Image and Vision Computing New Zealand, 8-9 Nov., Queenstown, pp. 1-8.

Brown, M. and Lowe, D. G., 2007. Automatic Panoramic Image Stitching using Invariant Features. International Journal of Computer Vision 74(1), 59-73.

Burt, P. J., Adelson, E. H., 1983. A multiresolution spline with application to image mosaics. ACM Transactions on Graphics 2 (4), 217-236.

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 Sensing 6(12), 12334-12359.

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 Journal of Photogrammetry and Remote Sensing 65 (1), 86-92.

Fernandez, E., Garfinkle, R., Arbiol, R., 1998. Mosaicking of aerial photographic maps via seams defined by bottleneck shortest paths. Operations Research 46 (3), 293-304.

Kerschner, M., 2001. Seamline detection in colour orthoimage mosaicking by use of twin snakes. ISPRS Journal of Photogrammetry and Remote Sensing 56 (1), 53-64.

Li, L., Yao, J., Lu, X., Tu, J., and Shan, J., 2016. Optimal seamline detection for multiple image mosaicking via graph cuts. ISPRS Journal of Photogrammetry and Remote Sensing 113, 1-16.

Lin, C.-H., Tsai, P.-H., Lai, K.-K., Chen, J.-Y., 2013. Cloud Removal from Multitemporal Satellite Images Using Information Cloning. IEEE Transactions on Geoscience and Remote Sensing 51(1), 232-241.

Mills, S., McLeod, P., 2013. Global seamline networks for orthomosaic generation via local search. ISPRS Journal of Photogrammetry and Remote Sensing 75, 101-111.

Pan, J., Wang, M., Li, D., Li, J., 2009. Automatic generation of seamline network using area Voronoi diagrams with overlap. IEEE Transactions on Geoscience and Remote Sensing 47 (6), 1737-1744.

Pan, J., Wang, M., 2011. A seam-line optimized method based on difference image and gradient image. IEEE 19th International Conference on Geoinformatics, 24-26 June, Shanghai, pp. 1-6.

Pan, J., Zhou, Q., Wang, M., 2014. Seamline Determination Based on Segmentation for Urban Image Mosaicking. IEEE Geoscience and Remote Sensing Letters 11(8), 1335-1339.

Pan, J., Wang, M., Li, J., Yuan, S., Hu, F., 2015. Region change rate-driven seamline determination method. ISPRS Journal of Photogrammetry and Remote Sensing 105, 141-154.

Pang, S., Sun, M., Hu, X., and Zhang, Z., 2016. SGM-based seamline determination for urban orthophoto mosaicking. ISPRS Journal of Photogrammetry and Remote Sensing 112, 1-12.

Peleg, S., 1981. Elimination of seams from photomosaics. Computer Graphics and Image Processing 16 (1), 90-94.

Uyttendaele, M., Eden, A., Szeliski, R., 2001. Eliminating ghosting and exposure artifacts in image mosaics. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. II:509-II:516.

Wan, Y., Wang, D., Xiao, J., Wang, X., Yu, Y., Xu, J., 2012. Tracking of vector roads for the determination of seams in aerial image mosaics. IEEE Geoscience and Remote Sensing

Letters 9 (3), 328-332.

Wan, Y., Wang, D., Xiao J., Lai, X. and Xu, J., 2013. Automatic determination of seamlines for aerial image mosaicking based on vector roads alone. ISPRS Journal of Photogrammetry and Remote Sensing 76, 1-10.

Wang, D., Wan, Y., Xiao, J., Lai, X., Huang, W., Xu, J., 2012. Aerial Image Mosaicking with the Aid of Vector Roads. Photogrammetric Engineering and Remote Sensing 78(11), 1141-1150.

Wang, W. and Ng, M. K., 2012. A variational method for multiple-image blending. IEEE Transactions on Image Processing 21(4), 1809-1822.

Yu, L., Holden, E. J., Dentith, M. C., Zhang, H., 2012. Towards the automatic selection of optimal seam line locations when merging optical remote-sensing images. International Journal of Remote Sensing 33 (4), 1000-1014.