SUPERPIXEL-BASED OPTIMAL SEAMLINE DETECTION VIA GRAPH CUTS FOR PANORAMIC IMAGES

Li Li, Jian Yao†, Xiaohu Lu, and Jing Ren

School of Remote Sensing and Information Engineering, Wuhan University, P.R. China
†Email: jian.yao@whu.edu.cn

ABSTRACT

In this paper, we present a novel method for seamlessly mosaicking panoramic images based on superpixels in the graph cuts energy minimization framework. To effectively ensure that all seamlines are detected in the laterally continuous regions with the high image similarity and the low object dislocation, the energy functions adopted in graph cuts combine the pixel-level similarities of image characteristics, including color and gradient, and the texture complexity. Instead of finding the optimal solution of seamlines in overlap regions via graph cuts among the entire set of pixels, we find it among superpixels created from input images, which greatly improves the efficiency of the global graph cuts energy optimization because the number of elements in graph cuts dramatically decreases. Experimental results demonstrate that the superpixel-based method is capable of generating high-quality seamlines as the pixel-based method but greatly reduces the computation time.

Index Terms— Seamline Detection, Panoramic Images, Image Mosaicking, Superpixels, Graph Cuts

1. INTRODUCTION

Nowadays, as the development of Street View which provides panoramic views along streets in the world, the demand for high-quality panoramic images gradually becomes larger. Image mosaicking is the key technology to produce high-quality panoramic images, which is also an important and classical problem in the field of computer vision, which is used to blend a set of aligned images into a single image (mosaic image) as seamlessly as possible. Generally, the methods of image mosaicking can be divided into two main categories, smoothing transition and optimal seamline detection.

Smoothing transition methods try to make given seams invisible and remove stitching artifacts by smoothing color differences between input source images. Alpha blending [1,2] is a simple and fast smoothing transition method, and has been widely used in panoramic images mosaicking. It blended input images based on the weighting map where the weighting coefficient of each pixel varies as a function of the distance from the seam. But it can't completely remove stitching artifacts and avoid ghosting problems caused by object moving and small geometric misalignments in mosaic images. The gradient domain based method [3] has been proposed for image blending. It defined the cost functions in the gradient domain rather than in the intensity domain to evaluate the visibility of the seam, which can reduce color differences and smooth color transitions. It can produce high-quality mosaic images in many cases.

However, due to the impreciseness of camera calibration and the deviation of multi-camera projection centers, the geometric positions of corresponding pixels from different images may be different and the pixels of closer objects shift more than those of distant ones, especially for panoramic images of Street View simultaneously captured by multiple cameras whose projection centers are not the same. In addition, due to the inconsistence of the moment of exposure between multi-cameras, there are larger geometric inconsistencies in moving objects. Transition smoothing methods can deal with the color difference along the seam but can't handle the great geometric position difference. One kind of methods to solve this problem is to detect the optimal seamlines avoiding crossing majority of visually obvious foreground objects and most of overlap regions with the low image similarity. If the seams and stitching artifacts are still visible due to color differences, the smoothing transition technique can be further applied to solve it easily. In this paper, our work only focused on the optimal seamline detection, which is the first key step in producing high-quality panoramic images.

Optimal seamline detection methods search for the seamlines between input source images where intensity or gradient differences in the overlap regions are minimal. Many methods regarded the seamline detection as an energy optimization problem and solved it by minimizing the special energy function which is defined to represent the difference along the seamlines [4–7]. Kerschner [5] proposed an automated seamline detection method using twin snake. But this algorithm requires a high computation time and can't completely overcome the local minimum problem. Graph cuts [8] is the most popular energy optimization algorithm which is applied to detect seamlines. Kwatra et al. [6] first applied the graph cuts algorithm to detect the optimal seamlines for image and video synthesis. After that, Agarwala et al. [7] provided a framework to easily create a single composite picture by using graph cuts to choose good seamlines within the constituent images, which needed an intuitive user interaction for defining local and global objectives. Gracias et al. [4] combined the watershed segmentation and the graph cuts algorithm to detect the optimal seamlines. Their algorithm began with creating a set of watershed segments on the difference image of overlap regions followed by finding the solution via graph cuts between those segments instead of the entire set of pixels. However, it only considered the intensity difference when computing the cost of each pixel and the difference image has lost some necessary information for image segmentation.

To greatly reduce the computation cost, we formulated the optimal seamline detection as a graph cuts energy minimization problem in the superpixel domain instead of in the pixel domain in this paper. The superpixels were generated from the input source images instead of the difference images, which ensures the accuracy of image segmentation. Not only the color and gradient differences were considered into the cost of each pixel but also the texture complexity inspired by HOG (Histogram of Oriented Gradient) [9] was integrated. Based on those, the optimal seamlines are located along the boundaries of superpixels via graph cuts.

2. OUR METHOD

Before giving a detailed description of our method, we briefly review the superpixel algorithms and the graph cuts algorithm we used. Superpixels are an oversegmentation of an image and popularly used in many computer vision applications. We choose SLIC [10] and V-Cells [11] algorithms to compare the influence of different superpixel algorithms in the last seamline detection results. SLIC and VCells regard the superpixel segmentation as a pixel clustering problem. SLIC clusters pixels in the 5-D space defined by the L, a and b values of the CIELAB color space as well as the x and y coordinates of the pixels. VCells firstly divides an image into small patches with uniform size and shape, and then applies EWCVT-LNN [11] to cluster pixels.

Graph cuts [8] is an efficient energy optimization algorithm to solve labeling problems, and has been popularly applied in many fields of computer vision, such as image segmentation [12,13], stereo matching [14], and image blending [4,6,7]. The basic technique is to first construct a weighted graph where each edge weight cost represents the value of corresponding energy function, and then to find the minimum cut in this graph based on the max-flow or min-cut algorithm [15]. Let $\mathcal{P}$ be a set of elements, $\mathcal{N}$ be a set of all element pairs $\{p, q\}$ in the neighborhood, and $\mathcal{L}$ be a set of labels. The goal is to find a labeling f that assigns a label $f_p \in \mathcal{L}$ to each element $p \in \mathcal{P}$ by minimizing the energy function:

$$E(f) = \sum\nolimits_{p\in\mathcal{P}} D_p(f_p) + \sum\nolimits_{(p,q)\in\mathcal{N}} V_{p,q}(f_p, f_q), \quad (1)$$

where $D_p(f_p)$ denotes the cost of assigning the label f_p to the element p and $V_{p,q}(f_p, f_q)$ defines the cost of assigning the label f_p and f_q to the adjacent elements p and q, which are often called as *the data energy term* and *the smooth energy term*, respectively. If f_p and f_q is equal, the value of $V_{p,q}(f_p, f_q)$ would be 0.

2.1. Superpixel Segmentation

In our study, we assume that all input source images for mosaicking Street View panoramic images have been geometrically aligned into the same coordinate system as precise as possible. However, there exist always geometric misalignments between these images in different extents due to that they are captured from the scenes with large depth differences by different cameras without the precise same camera projection center. For two input images $\mathcal{I} = (\mathbf{I}_1, \mathbf{I}_2)$, we generate superpixles in the overlap region of $\mathbf{I}_1$ or $\mathbf{I}_2$, and project them onto another image, as shown in Fig. 1. These superpixels are represented as $\mathcal{S} = \{S_i\}_{i=1}^n$ where n is the number of superpixels. In this paper, all free parameters were left at their default values for SLIC and VCells superpixel algorithms.

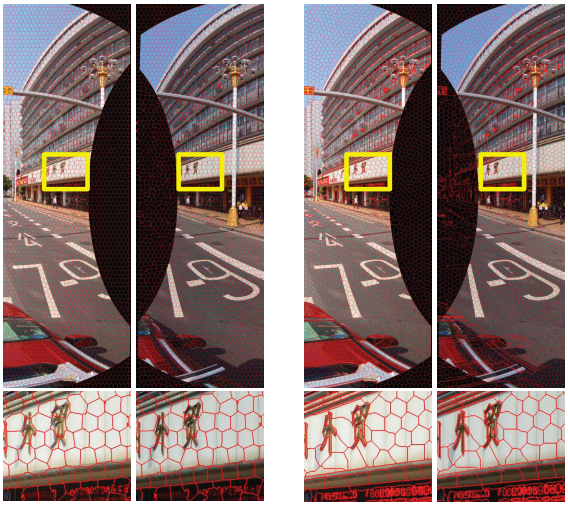
2.2. Energy Definition

The energy cost $C(\mathbf{p})$ of the pixel $\mathbf{p} = (x, y)$ of $\mathcal{I}$ is comprised of three terms: the color difference term $C_c(\mathbf{p})$, the gradient magnitude term $C_g(\mathbf{p})$ and the texture complexity term $C_t(\mathbf{p})$, which is defined as:

$$C(\mathbf{p}) = (C_c(\mathbf{p}) + C_g(\mathbf{p})) \times C_t(\mathbf{p}). \quad (2)$$

The color difference for the pixel $\mathbf{p}$ between two images is computed in the HSV (Hue, Saturation, Value) color space rather than in RGB, which is defined as:

$$C_c(\mathbf{p}) = w_h|H_1(\mathbf{p}) - H_2(\mathbf{p})| + w_s|S_1(\mathbf{p}) - S_2(\mathbf{p})|, \quad (3)$$



(a) SLIC (b) VCells

Fig. 1. The superpixels produced by SLIC and VCells in the left image and projected onto the right image. The image size is 1352×4069 and the number of superpixels is 3000.

where $H_1(\mathbf{p})$ and $S_1(\mathbf{p})$ denote the values of H and S channels of $\mathbf{p}$ in $\mathbf{I}_1$ and there are the same meanings for $H_2(\mathbf{p})$ and $S_2(\mathbf{p})$. The weight coefficients w_h and w_s are used to balance the influence of the differences at the H and S channels, which were set as $w_h = 1$ and $w_s = 0.1$ in this paper, respectively.

The gradient magnitudes of each pixel in the horizontal and vertical directions are obtained using the Sobel operator in the grayscale space. The gradient magnitude cost term $C_g(\mathbf{p})$ of the pixel $\mathbf{p}$ is defined as:

$$\begin{aligned} C_g(\mathbf{p}) =& |G_1^x(\mathbf{p}) - G_2^x(\mathbf{p})| + |G_1^y(\mathbf{p}) - G_2^y(\mathbf{p})| + \\ & \lambda(|G_1^x(\mathbf{p})| + |G_2^x(\mathbf{p})| + |G_1^y(\mathbf{p})| + |G_2^y(\mathbf{p})|), \end{aligned} \quad (4)$$

where λ is the balanced coefficient which was set as 0.25 in this paper, $G_1^x(\mathbf{p})$ and $G_1^y(\mathbf{p})$ denote the horizontal and vertical gradient magnitudes of $\mathbf{p}$ in $\mathbf{I}_1$, and there are the same meanings for $G_2^x(\mathbf{p})$ and $G_2^y(\mathbf{p})$.

Sometimes, some specific regions such as roads, sky and woodland are more suitable to be located in the seamlines due to that the image difference in these regions is not easy to be observed although there exist large color differences and large gradient magnitudes in such these regions. To solve this problem, we proposed a new criterion to distinguish those regions, which is inspired by HOG (Histogram of Oriented Gradient) feature descriptors [9]. The texture complexity $\Gamma(\mathbf{p})$ of the pixel $\mathbf{p}$ in overlap regions is calculated as follows. The gradient orientation $O(\mathbf{p})$ is computed firstly as: $O(\mathbf{p}) = \arctan(G^y(\mathbf{p})/G^x(\mathbf{p}))$ where $G^y(\mathbf{p})$ and $G^x(\mathbf{p})$ denote the gradient magnitude values of $\mathbf{p}$ in the vertical and horizontal directions, respectively. All the gradient orientations are converted into the range of $[0, 2\pi]$. Then, we compute the histogram of oriented gradient $H(\mathcal{N}_{k\times k}(\mathbf{p}))$ comprised of h ($h = 12$ was used in this paper) bins over the $k \times k$ ($k = 11$ was used in this paper) size window region $\mathcal{N}_{k\times k}(\mathbf{p})$ centered at the pixel $\mathbf{p}$. Based on the histogram of oriented gradient, the texture complexity at the pixel $\mathbf{p}$ is defined as:

$$\Gamma(\mathbf{p}) = 1 - \frac{\sum_{i=1}^{h} \min(H_i(\mathcal{N}_{k\times k}(\mathbf{p})), \bar{H}(\mathcal{N}_{k\times k}(\mathbf{p})))}{\sum_{i=1}^{h} H_i(\mathcal{N}_{k\times k}(\mathbf{p}))}, \quad (5)$$

where $H_i(\mathcal{N}_{k\times k}(\mathbf{p}))$ denotes the frequency of the i-th bin in

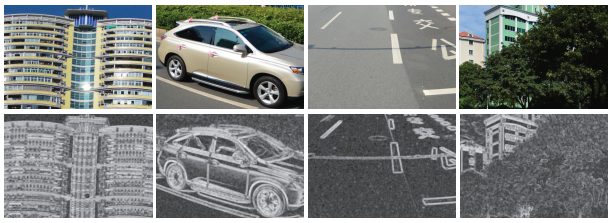


Fig. 2. The four typical regions (Top) happened in Street View panoramic images and their corresponding normalized texture complexity maps (Bottom) where the lighter regions indicate higher texture complexity.

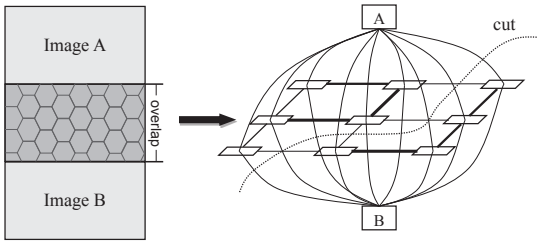


Fig. 3. An illustration example of the optimal seamline detection method via graph cuts. The thickness of lines between adjacent superpixels represents the value of the energy cost and the "cut" denotes the minimum cut, which means the optimal seamline.

$H(\mathcal{N}_{k\times k}(\mathbf{p}))$ and $\bar{H}(\mathcal{N}_{k\times k}(\mathbf{p}))$ represents the mean of frequencies of all bins, which is defined as $\bar{H}(\mathcal{N}_{k\times k}(\mathbf{p}))=\frac{1}{h}\sum_{i=1}^{h}H_i(\mathcal{N}_{k\times k}(\mathbf{p}))$.

Obviously, if the pixel $\mathbf{p}$ is located in the local region with poor texture like sky or strongly repetitive patterns like road or woodland, the frequencies of different bins in the histogram is approximately equal, so $\Gamma(\mathbf{p})$ is small and close to 0. In contrast, $\Gamma(\mathbf{p})$ is big and close to 1 if the frequencies of few bins are high and the rest are low.

Based on the definition of the proposed texture complexity, the texture complexity term $C_t(\mathbf{p})$ for the pixel $\mathbf{p}$ is defined as:

$$C_t(\mathbf{p}) = \Gamma_1(\mathbf{p}) + \Gamma_2(\mathbf{p}), \tag{6}$$

where $\Gamma_1(\mathbf{p})$ and $\Gamma_2(\mathbf{p})$ represent the texture complexity of $\mathbf{p}$ in $\mathbf{I}_1$ and $\mathbf{I}_2$. In this way, we can apply $C_t(\mathbf{p},\mathcal{I})$ to constrain the color difference and gradient magnitude in the regions with poor texture or strongly repetitive patterns without affecting other richly-textured regions. Fig. 2 shows the normalized texture complexity maps of several typical regions happened in Street View panoramic images.

2.3. Labeling via Graph Cuts

We formulate the optimal seamline detection as an energy minimization problem and use graph cuts to find the solution between superpixels, as shown in Fig. 3. The energy cost $E(\mathcal{I})$ comprises of the data energy term $E_{data}(\mathcal{I})$ and the smooth energy term $E_{smooth}(\mathcal{I})$, where the data energy term represents all energy costs for individual superpixels with one of input source images and the smooth energy term represents all energy costs between adjacent superpixels, which is defined as:

$$E(\mathcal{I}) = E_{data}(\mathcal{I}) + E_{smooth}(\mathcal{I}), \tag{7}$$

where the data energy term $E_{data}(\mathcal{I})$ is defined as:

$$E_{data}(\mathcal{I}) = \sum\nolimits_{S_i\in\mathcal{S}} \left(R_{S_i}(\mathbf{I}_1) + R_{S_i}(\mathbf{I}_2)\right), \tag{8}$$

where $R_{S_i}(\mathbf{I}_1)$ and $R_{S_i}(\mathbf{I}_2)$ represent the costs of assigning the label of the superpixel S_i to $\mathbf{I}_1$ and $\mathbf{I}_2$, which are defined as $R_{S_i}(\mathbf{I}_k) = 0$ if $S_i \in \mathbf{I}_k$ otherwise $R_{S_i}(\mathbf{I}_k) = \infty$ if $S_i \notin \mathbf{I}_k$, $k = 1, 2$. According to the above definition, for each superpixel S_i, its data energy only depends on whether it is inside the valid region of one image.

The smooth energy term $E_{smooth}(\mathcal{I})$ is defined as:

$$E_{smooth}(\mathcal{I}) = \sum\nolimits_{(S_i,S_j)\in\mathcal{N}(\mathcal{S})} \sigma_{i,j}\cdot E_{smooth}(S_i,S_j), \tag{9}$$

where $\mathcal{N}(\mathcal{S})$ represents the set of all neighour superpixel pairs in $\mathcal{S}$ and the coefficient $\sigma_{i,j} = 0$ if the labels of the superpixels S_i and S_j are the same otherwise $\sigma_{i,j} = 1$. $E_{smooth}(S_i, S_j)$ represents the smooth energy between two neighbor superpixels S_i and S_j, which is defined as:

$$E_{smooth}(S_i,S_j) = \max\nolimits_{\mathbf{b}\in\mathcal{B}(S_i,S_j)} \beta(\mathbf{b})\cdot C(\mathbf{b}), \tag{10}$$

where $\mathcal{B}(S_i, S_j)$ represents the set of all pixels in the common boundary of the superpixels S_i and S_j, and $C(\mathbf{b})$ is the energy cost of the pixel $\mathbf{b}$ defined in Eq. (2). $\beta(\mathbf{b})$ is used to reduce the influence of the noise pixels, which is defined as $\beta(\mathbf{b}) = 0$ if $C(\mathbf{b})$ is one of the biggest k costs along the common boundary otherwise $\beta(\mathbf{b}) = 1$, where $k = \min(\alpha \times |\mathcal{B}(S_i, S_j)|, 3)$, where $|\mathcal{B}(S_i, S_j)|$ denotes the size of the set $\mathcal{B}(S_i, S_j)$ and $\alpha = 0.05$ was used in this paper.

3. EXPERIMENTAL RESULTS

The images of outdoor scenes captured by an integrated multi-camera equipment with 6 Nikon D7100 cameras of 24 million pixels installed on a mobile vehicle platform to illustrate the performance of our method for mosaicking panoramic images. To mosaic a panoramic image, we first warped 6 original images into the same coordinate system with the image size of 12000×6000 by the reliable alignment. In total, 6 individual seamlines were detected from 6 camera views, consisting of 5 seamlines from 5 adjacent horizontal camera view pairs and 1 seamline between the top camera view and the horizontal camera ones. Our algorithm was implemented with C++ under Windows and tested in a computer with an Intel Core i7-4770 at 3.4GHz. All the images used in this paper and more experimental results are available at `http://cvrs.whu.edu.cn/projects/SSD/`.

To illustrate that our energy criteria defined in Section 2.2 is effective, we compared the seamline detection results by our energy criteria with the intensity difference used in [4], as shown in Fig. 4. From the seamline detection results especially in the detailed local regions, we found that our energy criteria obviously outperforms only intensity difference considered. The seamlines detected based on our energy criteria can avoid crossing cars, buildings and markings in the road. However, it failed for only using intensity difference.

Fig. 5 presents the seamline detection results of the pixel-based, superpixel-based and squared-grid-based methods. The SLIC superpixel algorithm was utilized to generate superpixels. Squared grids are an extreme case of superpixels. By comparing them, we found that seamlines detected by these methods always locate in the regions with high image similarity due to the constraint of our energy criteria, but there are great differences in details. We observed that seamlines detected by the superpixel-based method are similar with the pixel-based method without noticeable degradation of the seamline quality, and obviously outperform the squared-grid-based method because the squared grids don't have the information of segmentation. The total execution times of those methods including the

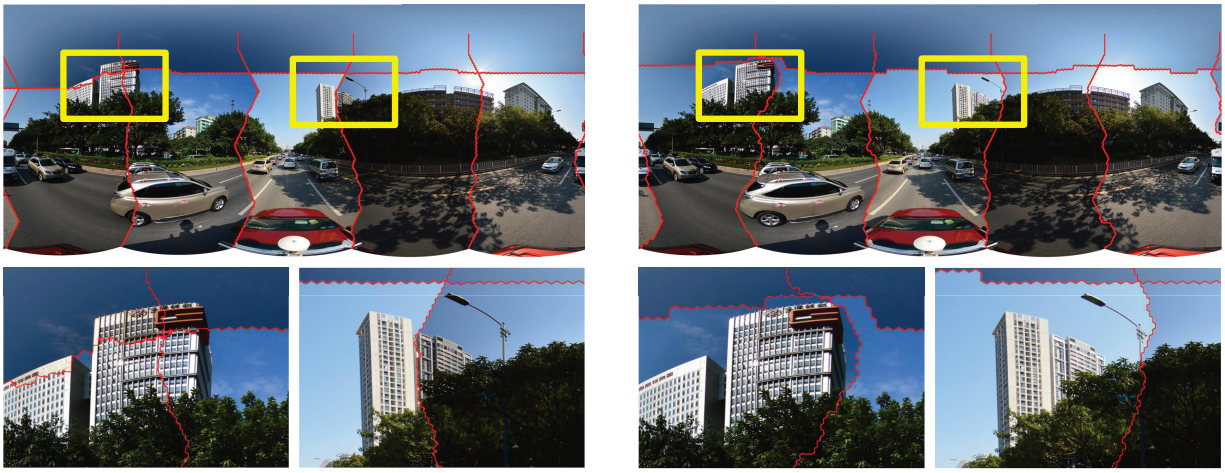


Fig. 4. The seamline detection results with the intensity difference used in [4] (Left) and our proposed energy criteria (Right). The SLIC superpixel algorithm was utilized to generate superpixels, and the number of superpixels is 3000 in each overlap region.

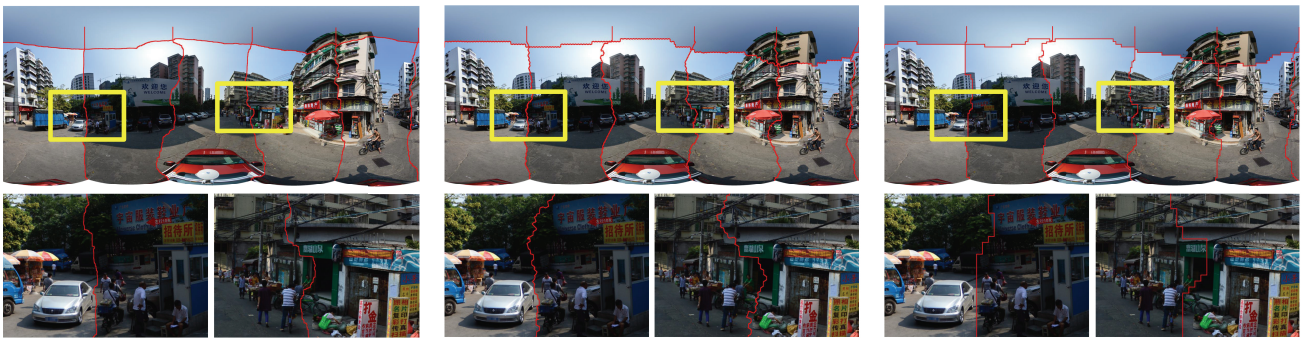


Fig. 5. Visual comparison of the pixel-based (Left), the superpixel-based (Middle) and the squared-grid-based (Right) methods. The number of superpixels used is 3000 in each overlap region.

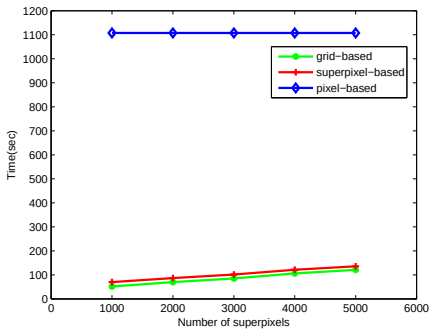


Fig. 6. The computation times of different methods with different numbers of superpixels or squared grids.

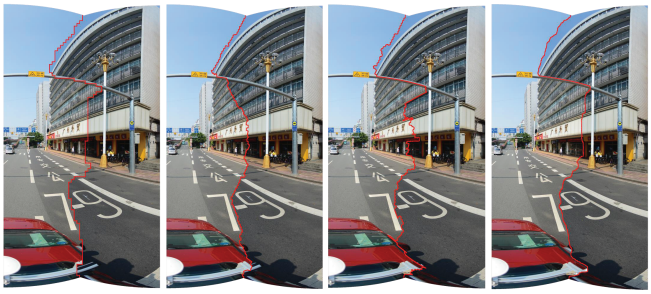


Fig. 7. Visual comparison of our proposed seamline detection method with different superpixel algorithms applied: Square-grid, SLIC, VCells and Pixel-based from left to right. The number of superpixels is 3000.

segmentation time for superpixels are plotted in Fig. 6. Obviously, our proposed superpixel-based method took much less computation time than the pixel-based method, which can greatly improve the efficiency.

From Fig. 1, we observed that superpixels produced by SLIC are more regular than VCells, but the boundaries of superpixels produced by VCells are more close to ground truth boundaries than SLIC. Fig. 7 shows the visual comparison of the seamline detected based on different superpixel algorithms, including squared-grid (an extreme case), SLIC and VCells, from which we observed that the detected seamline is more reasonable if the boundaries of superpixels are more close to ground truth.

4. CONCLUSION

This paper presented a novel superpixel-based optimal seamline detection algorithm for image mosaicking via graph cuts. Instead of using the simple intensity difference computing the cost of the corresponding pixels, we proposed a new energy criteria combining the texture complexity, the color difference and the gradient magnitude. Another contribution in this paper is that superpixels adopted greatly reduces the computation time of finding the solution compared with finding it over all pixels in the overlap regions. Experimental results on panoramic images have proved that the efficiency has significantly improved without noticeable degradation of the quality of seamlines.

5. REFERENCES

[1] Matthew Uyttendaele, Ashley Eden, and Richard Skeliski, "Eliminating ghosting and exposure artifacts in image mosaics," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2001.

[2] Yingen Xiong and Kari Pulli, "Mask based image blending and its applications on mobile devices," in *Sixth International Symposium on Multispectral Image Processing and Pattern Recognition*, 2009.

[3] Anat Levin, Assaf Zomet, Shmuel Peleg, and Yair Weiss, "Seamless image stitching in the gradient domain," in *European Conference on Computer Vision (ECCV)*. Springer, 2004.

[4] Nuno Gracias, Mohammad Mahoor, Shahriar Negahdaripour, and Arthur Gleason, "Fast image blending using watersheds and graph cuts," *Image and Vision Computing*, vol. 27, no. 5, pp. 597–607, 2009.

[5] Martin Kerschner, "Seamline detection in colour orthoimage mosaicking by use of twin snakes," *ISPRS journal of photogrammetry and remote sensing*, vol. 56, pp. 53–64, 2001.

[6] Vivek Kwatra, Arno Schödl, Irfan Essa, Greg Turk, and Aaron Bobick, "Graphcut textures: image and video synthesis using graph cuts," in *ACM Transactions on Graphics (ToG) (Proceedings of SIGGRAPH 2003)*, 2003.

[7] Aseem Agarwala, Mira Dontcheva, Maneesh Agrawala, Steven Drucker, Alex Colburn, Brian Curless, David Salesin, and Michael Cohen, "Interactive digital photomontage," in *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2004)*, 2004.

[8] Yuri Boykov, Olga Veksler, and Ramin Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 11, pp. 1222–1239, 2001.

[9] Navneet Dalal and Bill Triggs, "Histograms of oriented gradients for human detection," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005.

[10] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Süsstrunk, "SLIC superpixels," Tech. Rep., 2010.

[11] Jie Wang and Xiaoqiang Wang, "VCells: simple and efficient superpixels using edge-weighted centroidal voronoi tessellations," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 6, pp. 1241–1247, 2012.

[12] Yuri Boykov and Gareth Funka-Lea, "Graph cuts and efficient N-D image segmentation," *International journal of computer vision*, vol. 70, no. 2, pp. 109–131, 2006.

[13] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake, "Grabcut: Interactive foreground extraction using iterated graph cuts," in *ACM Transactions on Graphics (TOG) (Proceedings of SIGGRAPH 2004)*, 2004.

[14] Daolei Wang and Kah Bin Lim, "Obtaining depth map from segment-based stereo matching using graph cuts," *Journal of Visual Communication and Image Representation*, vol. 22, 2011.

[15] Yuri Boykov and Vladimir Kolmogorov, "An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 9, pp. 1124–1137, 2004.