Double Regions Growing Algorithm for Automated Satellite Image Mosaicking

Yihua Tan*, Chen Chen, Jinwen Tian
State Key Laboratory for Multi-spectral Information Processing Technology
Huazhong University of Science and Technology, Wuhan 430074, P. R. China

ABSTRACT

Feathering is a most widely used method in seamless satellite image mosaicking. A simple but effective algorithm – double regions growing (DRG) algorithm, which utilizes the shape content of images' valid regions, is proposed for generating robust feathering-line before feathering. It works without any human intervention, and experiment on real satellite images shows the advantages of the proposed method.

Keywords: Double regions growing algorithm, Mosaicking, Satellite Image, Remote sensing

1. INTRODUCTION

Image mosaicking is the fundamental task in GIS and remote sensing for getting panorama scenes. How to eliminate the visible seams in result images is one of the main difficulties. The existing approaches for eliminating seams can be broadly classified into two types [1]: one is to find optimal curve and stitch the mosaicked images directly in the overlap region, such as minimum error boundary cut by Efros [2] and watershed graph cut by Gracias [3]; the other one is blending the overlap parts of images, such as Flexible Image Blending [4], gradient domain blending [1] and feathering [5]. For large satellite images, the computational cost of some algorithms such as iteration in [1] is unacceptable. In addition, the images of adjacent scenes in practical often have great difference in intensity due to different weather, different imaging time, resulting that stitching the images directly along a curve could not eliminate the seam, wherever the curve is chosen. In the last few years, feathering is widely used in remote sensing image mosaicking and has been integrated into many kinds of commercial software such as ERDAS, which blends the images using a weighting function determined by the distance to a seam-line in the overlap region that we call feathering-line in this letter. Now most panorama images are produced under this general process: (i) color balance (histogram matching is often used); (ii) drawing a feathering-line and setting feathering distance; (iii) feathering with the feathering-line. Many methods in existing literatures have well solved the problems of (i) and (iii). However, how to choose the feathering-line and estimate the feathering distance for satellite images is still not solved in current mosaicking techniques and much human

*yhtan@hust.edu.cn; phone 86 027 87556301;

MIPPR 2011: Remote Sensing Image Processing, Geographic Information Systems, and Other Applications,
edited by Jianguo Liu, Jinwen Tian, Hongshi Sang, Jie Ma, Proc. of SPIE Vol. 8006, 80060K
 · doi: 10.1117/12.901786

operation is needed in practical projects. In this paper, we present a novel algorithm called double regions growing (DRG) algorithm for feathering-line generation, which could be applicable for any shape of the valid areas in input images and estimate the feathering distance automatically without any human interaction.

2. SATELLITE IMAGE MOSAICKING

Although color balance could largely reduce the difference of input images before mosaicking, visual artifacts still can be seen if we just cut and paste the images. To remove the visible seam caused by simple mosaicking, the prevailed method feathering provides a smooth combined image with high quality. Let I_1, I_2 be two adjacent images after rectification, and I be the mosaic result. Image feathering operation can be expressed as (in the mosaic coordinate system):

$$I(x,y) = w(x,y)\cdot I_1(x,y) + (1-w(x,y))\cdot I_2(x,y) \tag{1}$$

where $w(\cdot)$ is the weighting function defined in following equation

$$w(x,y) = \frac{\pm d(x,y) + MaxD}{2MaxD} \quad (d(x,y) \le MaxD) \tag{2}$$

Where $d(x,y)$ denotes the distance from (x,y) to the feathering-line, and its symbol is determined by the position relationship to the feathering-line (up or down), $MaxD$ denotes the max feathering distance.

The max feathering distance is decided by the shorter one of the distances from the feathering-line to both sides the overlap boundaries. So it is obvious that feathering with the bisector of the overlap region would be optimal in this case.

3. DOUBLE REGIONS GROWING ALGORITHM

3.1 Basic theory on feathering-line

However, the main problem for the above process is how to generate the feathering-line automatically. Firstly, invalid regions always exist in rectified images, such as the black background in Fig. 1a and 1b, which results that artificial seams will occur when weighting coefficients vary as the distance from the pixel to the centers of each image in [4]. Besides, odd shapes of the overlap areas occur frequently when the mosaicking images may have already been mosaicked from two or more images. In [5], the method of creating mask image with odd shape is not given. Some of the commercial software systems create feathering-line using the overlap region's geometry such as rectangle. But for more intricate shapes, it would be helpless. Apart from these, the resolution of satellite images is very low, resulting that choosing rivers, roads or objects' edges [3] for feathering-line is no longer applicable. So generating robust feathering-lines is urgently needed in automated seamless mosaicking on remote sensing.

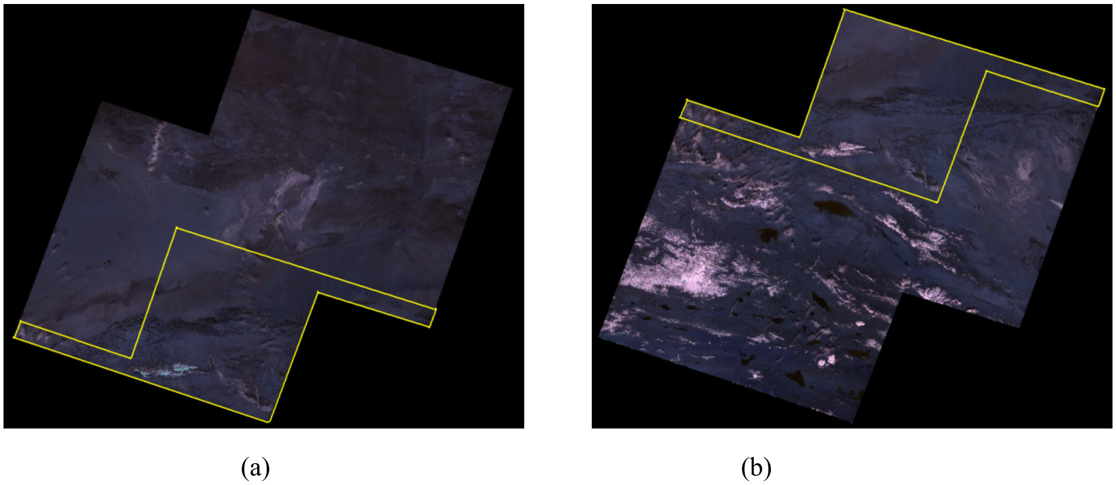


Fig. 1. Two HJ-1 satellite images with yellow polygon indicating the overlap region. (a) The upper image. (b) The lower image.

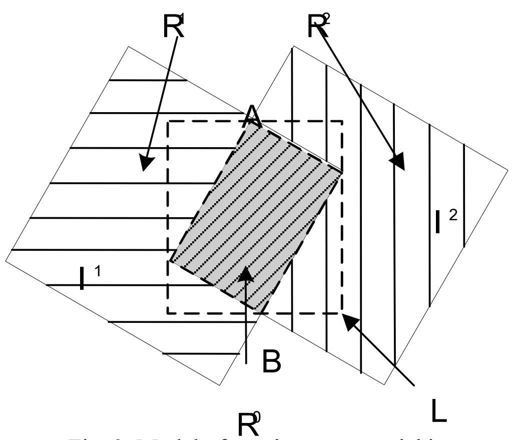


Fig. 2. Model of two images mosaicking.

A good feathering-line must have three properties to overcome the problems in satellite images mosaics: (1) passing through the intersection points of the two mosaicked images, such as point A and point B in Fig. 2, to ensure all of the overlap area to be feathered efficiently; (2) dividing the overlap region equally to provides both sides of the feathering-line having sufficient regions for feathering; (3) being created automatically for any shape of the valid overlap area. Our double regions growing algorithm is designed for these. We take two images mosaicking for example. Multiple images mosaics could be easily extended. Specifically, let I_1, I_2, be two adjacent images having been rectified as shown in Fig. 2. Let R_1, (R_2 resp.) be the valid region viewed exclusively in image I_1 (I_2 resp.), and let R_0 be the overlap valid region. Thus regions R_1, R_2 and R_0 are mutually exclusive. Region L is the boundary rectangle of R_0. Firstly, a state image that contain L and a little larger than L is created for both regions growing and feathering-line extracting. Then every pixel in R_1 and R_2 grow to its eight-neighbour that in R_0 respectively and simultaneously, until they meet at the middle of R_0. Finally, the feathering-line could be extracted where the two regions meet. This feathering-line could always satisfy the three conditions mentioned above. The feathering distance also could be estimated by the growing distance automatically. In order to make the program more efficient, it is necessary to use the intensity interpolation resampling logic (e.g., nearest-neighbour) to reduce the size of state image. It will significantly reduce the execution time and do not influence the result of the feathering-line. The feathering-line can be projected to the mosaic coordinate system after it is obtained from the state image. This algorithm could deal with images with odd overlap regions, even curved boundary, and is very fast and robust.

3.2 Growing role

In practical coding, we define four kinds of state for this algorithm:

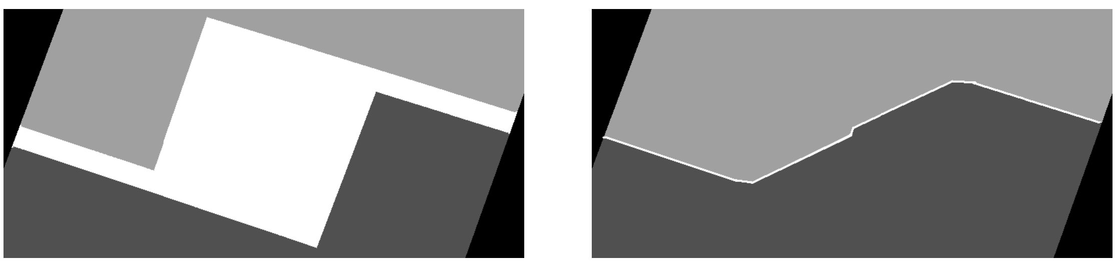
Sate A: the pixel belongs to I_1 (identified by the lower gray in Fig. 3);

State B: the pixel belongs to I_2 (identified by the higher gray in Fig. 3);

State U: the state of a pixel is underdetermined (identified by the white in Fig. 3);

State N: the pixel belongs to invalid image area (identified by the black in Fig. 3);

State M: the pixel locates in the feathering line (identified by the white line in Fig. 3(b)).



(a) (b)

Fig. 3. Initial state image and finally state image for mosaicking in Fig. 1. (a) The initial state image. (b) The final state image. The white line indicates the feathering-line.

For discrete form, double regions growing (DRG) is performed as follows:

Step 1.Initialize the state image

A state image is created by the states of the pixels, which is like a mask image of the overlap region L (Fig. 3(a)). The pixels in R_1 (R_2 resp.) are labeled as state A (B resp.), and pixels in R_0 are labeled as state U which means unknown and will change later. As in Fig. 3, only the states of pixels in R_0 will change (state U). Consequently, a state image of region L is initialized.

Step 2.Regions growing

For every pixel of state U, we count the number of each pixel labeled as A and B in its eight neighbours (for the pixels on the edge of the image, five neighbours or three neighbours replaced), which are denoted by *numA* and *numB*, respectively. For all pixels in state U, update them to be:

$$S_{(x,y)}^{[t]} = \begin{cases} A & if\ \ numA^{[t-1]} > numB^{[t-1]} \\ U & if\ numA^{[t-1]} = numB^{[t-1]} = 0 \\ B & otherwise \end{cases} \tag{3}$$

where $S_{(x,y)}^{[t]}$ is the state of pixel (x,y) at time t, with $S_{(x,y)}^{[0]}$ being set by the logic mentioned in Step 1:

$$S_{(x,y)}^{[0]} = \begin{cases} A & if\ (x,y) \in R_1 \\ B & if\ (x,y) \in R_2 \\ U & if\ (x,y) \in R_o \\ N & otherwise \end{cases} \tag{4}$$

Repeat this step and stop if no more pixel with state U exists. Finally, the image is composed of 3 kinds of regions. The total times of iteration S should be recorded which will be used to eliminate the feathering distance. As the number of pixels labeled in state U is finite, it's obvious that iteration will be convergent. Because of the same velocities of the two regions growing, they always arrive at the middle of the overlap region at the same time.

Step 3.Extract mid- feathering line

Similarly, *numA'* is the number of neighboring pixels labeled as A. For all the pixels of state B :

$$S_{(x,y)}^{[N]} = \begin{cases} M & if\ \ numA'^{[N-1]} > 2 \\ B & otherwise \end{cases} \tag{5}$$

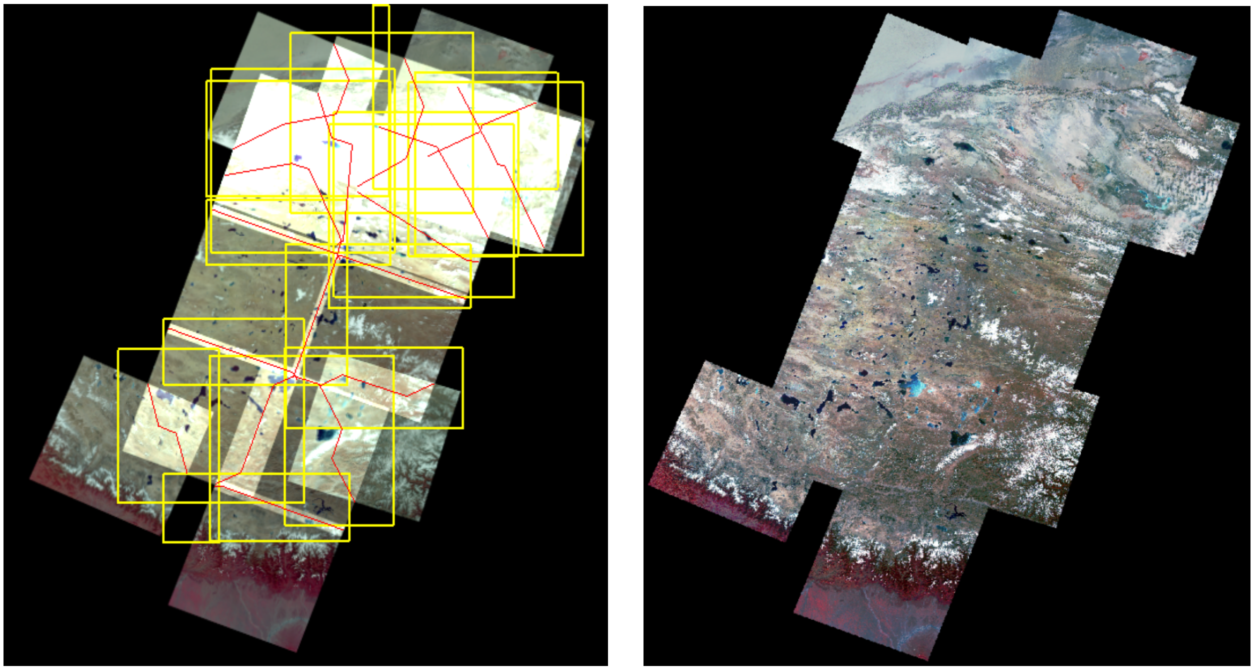
The feathering-line labeled by M is extracted then as in Fig. 3(b).

4. EXPERIMENTAL RESULTS AND DISCUSSION

We select 16 HJ-1 satellite CCD images (Fig. 4) for mosaicking, to demonstrate the use of proposed method. It is hard for human to draw these feathering-lines but easy for DRG as in Fig. 4(a). Mosaic can be divided to one-to-one models. We first rank all the images by their position. Every time, current image is mosaicked with all its upper images. And then histogram matching is used in every pair. Finally, feathering all the images with the produced feathering-line and estimated feathering distance. For HJ-1 satellite dataset, we choose 1/3 of the growing distance as *MaxD* in formula (2). Some commercial software could also create feathering-lines in Fig. 4's case by using the bounding rectangles of each image. But when this result image of odd bounding edge is mosaicked with another similar image, commercial software could not work anymore.

5. CONCLUSIONS

We propose a new method called double regions growing algorithm for satellite image mosaicking. Experiment shows our algorithm could produce robust feathering-lines for feathering. Our algorithm solves a practical problem in remote sensing and could be used for real engineering applications. We use the shape content of the images for an automatic feathering-line generation without any gray content. Then the future research is to combine gray content of the images to get better feathering-line.



(a) (b)

Fig. 4. Mosaicking 16 HJ-1 satellite images with DRG algorithm. (a) The original images with yellow boxes indicating the overlap regions and red lines indicating feathering-lines created by DRG. (b) The result of histogram matching and feathering, bands 4, 3, 2.

ACKNOWLEDGEMENTS

This project is supported by Hubei Province Key Laboratory of Intelligent Robot (HBIR201101).

REFERENCES

[1] Levin,A., Zomet,A., Peleg,S., and Weiss,Y., "Seamless image stitching in the gradient domain", Proc. ECCV, 377-389(2004).

[2] Efros,A.A., and Freeman,W.T. ,"Image quilting for texture synthesis and transfer", Proc. SIGGRAPH, 341–346(2001).

[3] . Gracias, N., Mahoor, M., Negahdaripour, S. and Gleason, A., "Fast image blending using watersheds and graph cuts", Image and Vision Computing, 27, 597–607(2009).

[4] Zhao, W., "Flexible Image Blending for Image Mosaicing with Reduced Artifacts", International Journal of Pattern Recognition and Artificial Intelligence, 20(4), 609–628 (2006).

[5] Burt ,P.J. ,and Adelson, E.H., "A multiresolution spline with application to image mosaics", ACM Trans. Graph., 2(4), 217-236(1983).