A STUDY OF IMAGE REGISTRATION AND COLOR BLENDING FOR CLOSE-RANGE AND AERIAL IMAGES

Han-Szu Chou and Chao-Hung Lin***

** Graduate student, Department of Geomatics, National Cheng Kung University,*
701, No.1, Daxue Rd., East Dist., Tainan City, Taiwan; Tel: + 886+936-677-048;
E-mail: joanne78312@hotmail.com

*** Professor, Department of Geomatics, National Cheng Kung University,*
701, No.1, Daxue Rd., East Dist., Tainan City, Taiwan; Tel: + 886+6-2757575#63836;
E-mail: linhung@mail.ncku.edu.tw

KEY WORDS: Color blending, Image matching, Image registration, Image mosaicing

ABSTRACT: In this paper, an image mosaicing system is introduced. The proposed system consists of three main steps, feature matching, image registration, and color blending. In the first step, the approach of scale-invariant feature transform is adopted to describe the extracted features for matching. In the step of image registration, a least-square-based approach is adopted to align the matched features obtained in the first step. In the last step, the images are seamlessly mosaicked together by a multi-band color blending algorithm. The experimental results on close-range images and large-scale aerial images show that the proposed system can smoothly blend the colors on boundaries and thus seamlessly mosaic the images.

1. INTRODUCTION

There are many researches talking about image registration and color blending. The three main steps are feature matching, image registration, and color blending. In the first step, the approach of SIFT algorithm we chose can help us finding keypoints from two target images, then doing feature matching between these keypoints of two images. The reason we chose this approach SIFT algorithm is it is invariant to image scaling, translation and rotation. In the second step, we use parameter-transformation which also called affine transformation to achieve image registration. We obtain parameters of transformation matrix by calculating keypoints between images we obtained from first step. In the last step, color blending which is the main process in this paper. We use multi-band color blending algorithm to realize color blending. Multi-band blending algorithm use scale image pyramid to make image stitching more smoothing.

2. FEATURE MATCHING

The first step is to extract and match features by using SIFT algorithm [1] which is invariant to image scaling, translation and rotation, and partially invariant to illumination change, affine transformation. When matching the aerial images, the property of partial affine-transformation-invariance helps us to find features easier.

2.1 Keypoint localization

Before processing feature matching between two images, we can capture some key points with maxima or minima value that is calculated by comparing the value of points of its neighbors for each image. To achieve rotation invariance and to take efficiency into account, the pixels located at maxima and minima of a difference of Gaussian function applied in scale space are selected as key points. For two-dimensional image computation, 2D Gaussian function is separated into two parts and computed by two-phase 1D Gaussian function in horizontal and vertical directions. At the first time comparison, a pixel is compared with its 8 neighbors at the same scale level in the pyramid. We get maxima or minima locations at the scale level, then comparing the closest pixel at next lower scale level. If it still lower or higher than its 8 neighbors, then repeating this test for level above.

2.2 Keypoint descriptor

For each key location, we give a stable location, scale, and orientation by computing the sum of gradient of its circle neighbors' differences. We call the major orientation descriptor which can describe the feature character. For achieve image rotation invariant, we rotate each feature point to its major orientation during feature matching process. There are numbers of approaches we can find orientation descriptor for keypoints. Lowe [1] was found that for each image sample, $I(x, y)$, at this scale, the gradient magnitude, $m(x, y)$, and orientation, $\theta(x, y)$, is precomputed using pixel differences:

$$m(x,y) = \sqrt{(I(x+1,y) - I(x-1,y))^2 + (I(x,y+1) - I(x,y-1))^2} \tag{2.1}$$

$$\theta(x,y) = \tan^{-1}\left(\frac{I(x,y+1) - I(x,y-1)}{I(x+1,y) - I(x-1,y)}\right) \tag{2.2}$$

After getting the location, scale, and orientation of all keypoints, we produce stable descriptor which is invariant to image scaling, translation, and rotation. The gradient information is rotated to line up with the orientation of the keypoint and then weighted by a Gaussian with variance of 1.5 * keypoint scale. This data is used to create a set of histograms over a window centred on the keypoint. Keypoint descriptors uses a set of 16 histograms, aligned in a 4x4 grid, each with 8 orientation vectors, one for each of the main compass directions and one for each of the mid-points of these directions. These results in a feature vector containing 4*4*8 or 128 elements to describe the feature vectors.

2.3 Indexing and matching

There are many approaches to match feature, but most of them are cost lots of time to compare every keypoint. Lowe [2] was found that obtained by comparing the distance of the closest neighbor to that of the second-closest neighbor by Euclidean algorithm. Then calculate their distance ratio and define a threshold γ to filter unobvious feature.

$$\frac{P_i - P_{i1}}{P_i - P_{i2}} < \gamma \tag{2.3}$$

Follow Lowe's SIFT source code there is a default value 0.6, then we use 0.6 as the threshold value which unit is ratio.

3. IMAGE REGISTRATION

The approach we use to transform image is parameter transformation. We get the transform parameters or transform matrix between images by the feature points we obtain from SIFT algorithm. For this case, we chose affine transform matrix which is constructed to 7 parameters contain 1 scale, 3 offsets, and 3 rotations. For image transform, only need to request 1 scale, 2 offsets, and 1 rotation parameters to get a 2x2 matrix:

$$\begin{bmatrix} X_i \\ Y_i \\ 0 \end{bmatrix} = \begin{bmatrix} \Delta X \\ \Delta Y \\ \Delta Z \end{bmatrix} + S_{ij} \begin{bmatrix} 1 & \theta_z & -\theta_y \\ -\theta_z & 1 & \theta_x \\ \theta_y & -\theta_x & 1 \end{bmatrix} \begin{bmatrix} X_j \\ Y_j \\ 0 \end{bmatrix} \Rightarrow \begin{bmatrix} X_i \\ Y_i \end{bmatrix} = \begin{bmatrix} \Delta X \\ \Delta Y \end{bmatrix} + S_{ij} \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} X_j \\ Y_j \end{bmatrix} \tag{3.1}$$

In image transform we don't need to consider axis Z direction, only rotate in one direction. Then using the keypoint locations we get from SIFT, we translate the affine transform matrix and obtain the new equation:

$$\begin{bmatrix} L_{x1} L_{x2} \cdots L_{xn} \\ L_{y1} L_{y2} \cdots L_{yn} \end{bmatrix} = \begin{bmatrix} a & b & c \\ -b & a & d \end{bmatrix} \begin{bmatrix} R_{x1} R_{x2} \cdots R_{xn} \\ R_{y1} R_{y2} \cdots R_{yn} \\ 1 \quad 1 \quad \cdots \quad 1 \end{bmatrix} \tag{3.2}$$

Where $a = S_{ij}\sin\theta$, $b = S_{ij}\cos\theta$, $c = \Delta X$, $d = \Delta Y$, and define this equation as L=AR. We can get parameter matrix from $A=LR^{-1}$ and obtain pseudo inverse R by pinv() function in MATLAB.

4. COLOR BLENDING

The approach we stitching two color images is multi-band blending algorithm which calculate the Gaussian and Laplacian function by 3x3 window. We process a range of pixels along the straight stitching line. Following P. J.

Burt and E. H. Adelson [3], we use the Laplacian pyramids of the input images as the multi-band decomposition, and we approximate them by differences of Gaussians. Assume original image was represented by I, $G_i(I)$ represent the image I was computed by Gaussian at the level i, and Li(I) related with the difference between level i Gaussian image and level i+1 Gaussian image represent the level i Laplacian image. Then we can represent the relationship by following equations:

$$\begin{aligned} G_1(I) &= I \\ G_2(I) &= G_1(I) * g_\delta \\ G_3(I) &= G_2(I) * g_\delta \\ &\vdots \\ G_b(I) &= G_{b-1}(I) * g_\delta \end{aligned} \tag{4.1}$$

$$\begin{aligned} L_1(I) &= I - G_2(I) \\ L_2(I) &= G_2(I) - G_3(I) \\ L_3(I) &= G_3(I) - G_4(I) \\ &\vdots \\ L_{b-1}(I) &= G_{b-1}(I) - G_b(I) \\ L_b(I) &= G_b(I) \end{aligned} \tag{4.2}$$

To produce Gaussian image we convolute each pixel with Gaussian kernel g_δ by every 3x3 window. We defined g_δ as follow:

$$g_\delta = \frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} \tag{4.3}$$

After the Gaussian and Laplacian image pyramid was build, there are b levels or we can said b bands image was build. The sum of all Laplacian image can reconstruct back to the original image:

$$\begin{aligned} L_1(I) &= I - G_2(I) \\ L_2(I) &= G_2(I) - G_3(I) \\ &\vdots \\ L_{b-1}(I) &= G_{b-1}(I) - G_b(I) \\ +)\quad L_b(I) &= G_b(I) \\ \hline \sum_{i=1}^{b} L_i(I) &= I \end{aligned} \tag{4.4}$$

Now we need to define the new image mosaic function with different weight for every pixel on mosaicing line. The result of mosaicing image, C(I), the weight of each pixel x of image I_1 and I_2, $w_1(x)$, $w_2(x)$, and the relation function can write as follow:

$$C(I) = \sum_{i=1}^{b}\sum_{x=1}^{n} L_i(I_1) \times w_{1,i}(x) + L_i(I_2) \times w_{2,i}(x) \tag{4.5}$$

We define every pixel x as linear function w(x) from 0 to 1, and the interval value of weight for each pixel are varied with stitching buffer size. And the function as follow:

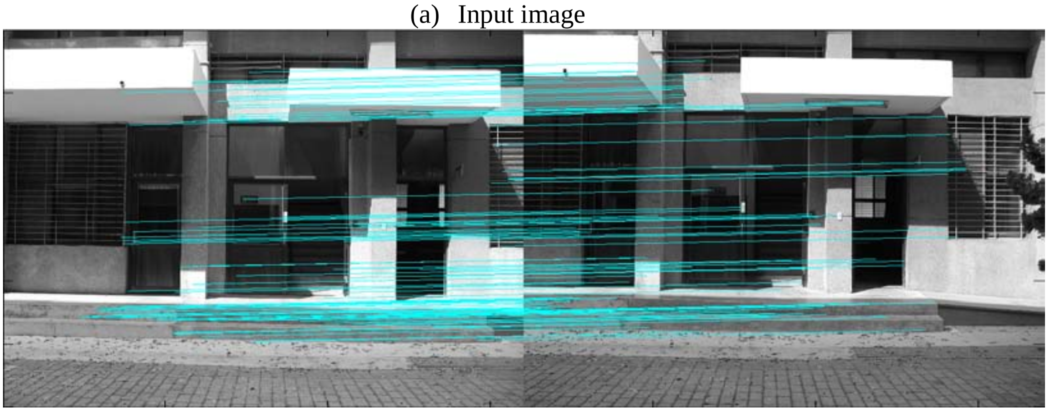
$$\begin{cases} w_1(x) = \dfrac{x}{buffer - size} \\ w_2(x) = 1 - w_1(x) \end{cases} \quad (4.6)$$

Later we need to consider the 3 color bands color image blending. We divide image to R(red), G(green), and B(blue) 3 matrix to save in memory. After blending images for 3 color bands respectively, we combine 3 color bands in one file. Then get a color image which is constructed by two neighbor scene images.

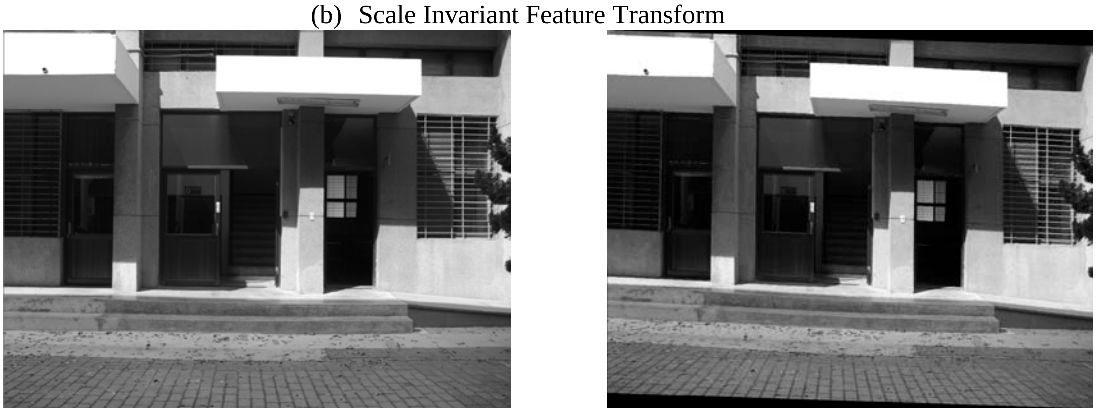
5. EXPERIMENTAL RESULTS

We can see the result in Figure 1 show that close-range image blending. Because the SIFT algorithm only support 1 color band feature matching, we only test 1 band image to show the blending. In Figure 2 we try to use aerial images to mosaic together. Different to Figure 1 case is vertical mosaicing line, Figure 2 and Figure 3 are horizontal mosiacing line. Then we need to change our blending function to stitch. The second difference is we blend image in 3 color bands in Figure 2 and Figure 3. As we saw, they are good result for us.

(a) Input image



(b) Scale Invariant Feature Transform

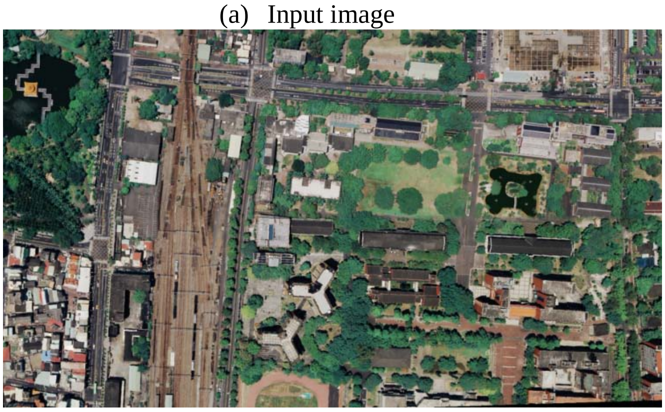


(c) Image registration

(d) Output image

Figure 1. The right and left of original image showed in (a). After SIFT algorithm process we obtain (b). Latter we using parameter transform to registration image. Final we use multi-band blending stitch two image to one image.

(a) Input image



(b) Output image

Figure 2. (a) shows two original images; and (b) shows our blending result.

(a) Input image

(b) Output image

Figure 3. (a) shows two original images; and (b) shows our blending result.

6. CONCLUSION

This paper has presented a system for closest-range and aerial images stitching. We do image matching with SIFT algorithm first, second using parameter-transformation for image registration. Last, we use multi-band color blending to blend color images. Combine these three main approach in one program, we can build it to a smoothly image stitching automatically system.

Possible future directions would be to change the approach parameter-transformation to buddle adjustment for a more accuracy image registration process. And add optimal mosaicing line for avoid high building oblique in aerial images.

7. REFERENCES

[1] David G. Lowe, September 1999, Object Recognition from Local Scale-Invariant Features, In: International Conference on Computer Vision, pp. 1150-1157.

[2] David G. Lowe, January 5, 2004, Distinctive Image Features from Scale-Invariant Keypoints, In: International Journal of Computer Vision, 60(2), pp.91-110.

[3] P. J. Burt and E. H. Adelson., October 1983. A Multiresolution Spline With Application to Image Mosaics. ACM Transactions on Graphics (TOG), 2(4), pp.217-236.

[4] M. Brown and D. G. Lowe, October 2003, Recognising Panoramas, In: ICCV 2003 Proceedings of the Ninth IEEE International Conference on Computer Vision – Vol.2, pp. 1218-1225.

[5] C Allene, Jean-Philippe Pons, Renaud Keriven, 2008, Seamless image-based texture atlases using multi-band blending, International Conference on Pattern Recognition - ICPR , pp. 1-4