A Parallel Method for Aerial Image Stitching Using ORB Feature Points

Guodong Wang, Zhengjun Zhai
School of Computer Science, Northwestern Polytechnical University
Xi'an, PR, China
Wgdong029@163.com

Bangdao Xu,Yue Cheng
Computing Technique Research Institute, Aviation Industry Corporation
Xi'an, PR, China
Wgdong029@163.com

***Abstract*—Aviation image stitching requires real-time processing, while traditional key point descriptor generating a floating-point feature vector, thus the processing efficiency in embedded hardware platform such as DSP, FPGA is not satisfactory. Recently ORB feature point is proposed, which has binary vector for feature descriptor, it greatly speeds up processing procedure for feature extraction and matching. In order to calculate the homography matrix between stitching sequence robustly, best of 2nd nearest matcher, cross-validation and RANSAC estimation are adopted. Registered images still have some color deviation in the same pixel position. If the traditional α-blending method is employed without consideration of the position information on the edge of the image, the stitching artifacts at image edges which largely affect the visual effects will be produced. A position-weighted image fusion algorithm which takes the location information of image pixels into consideration is also presented in this paper, so the image can be naturally stitched and the problem of artifacts is solved. The proposed algorithm is insensitive to complex noise presented in input image data. Furthermore, we propose a novel parallel framework for image stitching based on recent proposed ORB feature descriptor which is realized on a multicore DSP platform: TMS320C6678. With the implement of the parallel design, computing speed is obviously improved and real-time image stitching for airborne embedded application is realizable.**

Keywords—image-stitching，ORB，image registration，image fusion，Parallel computing

I. INTRODUCTION

The image-stitching has been developed for a long history in the area of the computer version and the image processing [1], this technology uses multiple image which has the overlap area to produce a high resolution panoramic image, which have been widely applied in disaster prevention and control, quality monitoring, digital entertainment and other fields. Especially in the enhancement of the aviation visual, the panoramic images, come from the stitch of the aerial images, are used in the aircraft cruising, take-off and landing tasks, which gives a lot of help to the pilot to understand the information comprehensively. As two key steps called parameter estimation and panorama generation in the image stitching, parameter estimation step calculates the geometric parameters of the image registration based on the feature points and the panorama generation step gets the naturally stitched image from the registration parameters.

In the computer vision field, image stitching has been fully investigated, and a lot of method has been proposed for different applications. These methods often run in general computing platforms such as Intel or AMD CPUs, and the platform characteristics commonly not considered. But in airborne application, we employ some embedded computing platform to run our stitching application in nearly real time. The choice of the embedded platform is very important as well as a well-designed stitching algorithm that is suitable for such platform.

As seen in Fig. 1, the TM320C6678 DSP is a 40-nm-based fixed and floating point SoC of TI's KeyStone architectures, integrating eight CorePac C66 cores, each operating at a frequency of 1.0 to 1.25 GHz. A single chip performance is 320 GMACS / 160 GFLOPS with a power consumption of 10W. Its main advantages are as follows:

1) High-performance and highly integrated.
2) Low power consumption.
3) Multicore and easy communication features.
4) Peripheral scalable large-capacity DDR and multi-core shared RAM.

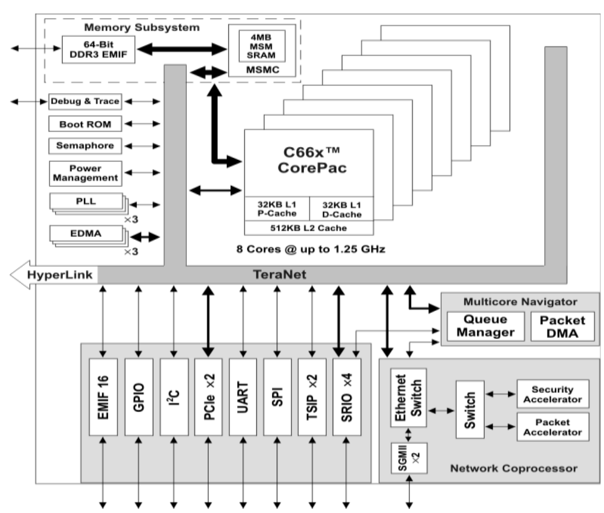


Fig. 1. Architecture of TMS320C6678

In this paper, we proposed an ORB feature point based method for aerial image stitching. A position-weighted image fusion algorithm which takes the location information of image pixels into consideration is also further presented in this paper. By using a parallel designing of proposed method on multicore DSP platform TMS320C6678, the stitching shows nearly real time performance.

The rest of the paper is organized as follows: Section 2 presents the related work. Section 3 presents the method of feature extraction and mating. Section 4 present our proposed algorithm for image fusion. Section 5 present a parallel strategy for acceleration we designed. Section 5 shows the realized results on TMS320C6678. Finally, conclusion and future work are discussed.

II. Related Work

Image-stitching based on pixel brightness by optimizing the similarity between pixels directly in the early, which matched the most similar part between the overlap of the adjacent images. Though the method above is accurate in the subpixel level, we can get a better result by matching the feature points which distribute sparsely in the image in recent research. Firstly, angular point or line is stable and can be identified in different images. Secondly, as the feature points is sparse in the images so that the whole-pixel calculation of the similarity between the images is unnecessary and the matching efficiency is increased especially which comes to the need for the processing of the aerial images. Thirdly, image-stitching algorithm based on feature points is insensitive to the geometrical features of imaging equipment, it can restore the geometric features of the image to some extent even if the imaging parameters such as focal length of the camera is changed, which is useful for the matching. Last but not least, the algorithm can search the matching target automatically without consideration of the position information of the images before, so that the panoramic image can be stitched from an unordered image sequence. All in all, image-stitching algorithm based on the feature points is widely used.

Above all, how to find stable feature points in the images is the first step and researchers have been paid a great effort on it. Forsner [2] and Harris [3] get the feature points by using Gaussian function to convolute local images, so that Hessian matrix and image characteristics in gray can be calculated from a series of differential Gaussian filter joining algebra operations directly. Due to the differentiable characteristics of Gaussian function and there is no particular order between convolution step and derivation step, a difference step of the image area after Gaussian filter is equivalent to the convolution between the original image and the Gaussian function after derivation.

There is usually some rotation relationship between images, in order to find a feature which is independent of rotation or zoom, Lowe [4] proposed a SIFT feature points, the feature vector can be described through gradient histogram formed by feature point neighborhood. Hager [5] fit a two-dimensional Gaussian kernel, the rotation invariant features are described by calculating the weighted features around an area. The feature vectors mentioned above are all using floating point Numbers. Leutenegger [6] group come up with a simple and effective feature called BRISK recently, use a FAST key point detector to find some features as alternatives, then feature vectors are formed by the binary comparison in a small neighborhood of the candidate feature. As binary description only takes a bit of data in the computer, this algorithm is simple and efficient.

As BRISK feature is sensitive to rotation and noise, the scale is changed in different images, by define a main direction, Rublee [7] proposed a Rotation invariant BRISK feature, called ORB, based on pixel block binary comparison instead of pixel value comparison in BRISK feature in order to increased ability to resist noise. Though the scale-variant problem is still exists in ORB, as the scale of the aerial images changes little in general, it can meet the demand of image-stitching, so we choose ORB method in our paper.

As the final step in image-stitching, registration error is inevitable image fusion, which is caused by the noise, distortion, exposure during image registration process. At the same time, color and position deviation would be caused by different camera which shoot the same scene. In order to solve the problems mentioned above, some algorithms are proposed such as α -blending method, double linear differential method, laplacian pyramid method [8] and so on, these methods help to smooth the area of the stitching image and make it more natural accord to the human observation habit. α-blending method or the improved one are widely used as simple and efficient solution in aerial image stitching. α-blending method is a weighted average of the image pixels from two sources. Color deviation in the edge of the stitching area would cause artifacts, which affect the quality of the image fusion. Based on the α-blending method, we proposed a position-weighted image fusion algorithm which takes the location information of image pixels into consideration, so the image can be naturally stitched and the problem of artifacts is solved.

III. FEATURE EXTRACTION AND MATCHING

A. The extraction of ORB feature points

ORB feature based on the FAST feature to extract the feature conform to the requirements, As FAST feature points take consideration of the information on the edge of the image and are greatly influenced, a Harris filter is necessary. So the maximum response feature points are picked out, which have an angular feature but not the edge one.

Firstly, the center of the brightness should be fixed by a pixel around the feature point which location in the image is O, define the moment of the area around the feature point:

$$m_{pq} = \sum_{x,y} x^p y^q I(x, y) \tag{1}$$

$p, q \epsilon (0,1)$, with the center of the brightness:

$$C = (\frac{m_{10}}{m_{00}}, \frac{m_{01}}{m_{00}}) \tag{2}$$

Construct a vector with the direction from center of the feature point O to the center of the brightness C, denoted as **OC**, define the rotation angle in pixel area:

$$\theta = \arctan(m_{01}, m_{10}) \tag{3}$$

Based on the location of the feature point, we construct a binary feature string of ORB with the help of the improved BRISK feature vector. For instance, a pixel block p with smooth processing, define the binary test:

$$r(p; x, y) = \begin{cases} 1: p(x) < p(y) \\ 0: p(x) \geq p(y) \end{cases} \tag{4}$$

In which p(x) is the brightness of the pixel block at that point, then we get the binary feature string, the number of which is n bits.

$$f_n(p) = \sum_{1 \leq i \leq n} 2^{i-1} \tau(p; x, y) \tag{5}$$

The ORB feature point is training in several test positions around pixel block p, the number of the test position is n, where the variance of the ORB feature vector reach the highest. In other words, these vectors in the feature space is more differential than in the other position, make it more valuable to the matching and identification. What's more, consider the rotation between the pixel block and the main direction, a rotation of each test position $S = (x, y)^T$ is necessary:

$$S_\theta = R_\theta S \tag{6}$$

In which R_θ is the rotation matrix of the main direction, the rotation angle is θ, S_θ is the test position for the binary pixel after rotation.

The BRISK feature after rotation still have the character of binary string, the feature distance in the feature space can be represented by Hamming distance, in other words, the distance of the feature vector is the number of the different values of the position corresponding to the binary string. The calculation of the Hamming distance is as simple as an operation bit-by-bit and a sum operation, which increase the efficient of the matching for feature points.

B. *Matching and registration*

Based on the ORB feature distance between two images in the sequence of stitching image, we get the nearest neighbor feature points of each feature point, a 2nd nearest neighbor feature points of each feature point can be calculated at the same time. In order to achieve a better degree of differentiation in feature space, we just choose those point whose ratio of the 2nd nearest neighbor distance and the nearest neighbor distance is large as the reliable matching points. Then we filter the matching point in low reliability with the help of cross validation method, in other words, we calculate the matching feature of one image and calculate the inverse matching feature of another, the matching is not credible unless the two matching features are both exist.

As the reliable matching point is found, linear transformation [9] relationship between images will be calculated through the linear transformation. Let $x = (x, y, \omega)^T$ be the homogeneous coordinate of a feature point in one matching image, in which ω is a constant, the coordinate of the matching feature point corresponding to the one above is $x' = (x', y', \omega')^T$, with the homography matrix:

$$Hx = x' \tag{7}$$

where:

$$H = \begin{bmatrix} h_{11}, h_{12}, h_{13} \\ h_{21}, h_{22}, h_{23} \\ h_{31}, h_{32}, h_{33} \end{bmatrix} \tag{8}$$

Do the exterior products operation on both images:

$$x' \times Hx = x' \times x' = 0 \tag{9}$$

Then we have:

$$\begin{bmatrix} 0^T & -\omega' x^T & y' x'^T \\ \omega' x^T & 0^T & -x' x^T \\ -y' x^T & x' x^T & 0^T \end{bmatrix} \begin{bmatrix} h_1 \\ h_2 \\ h_3 \end{bmatrix} = 0 \tag{10}$$

Where $h_i = (h_{i1}, h_{i2}, h_{i3})^T (i \in (1,2,3))$ is the vector of homography matrix in row i, the degree of which is 8, formula (10) has two constraints (where line 3 is linear correlate with the first two lines), so at least 4 pairs of matching points would come to the unique solution of homography matrix H. Considerate that some mismatching is inevitable, least square method may cause a big bias. In this paper, we choose RANSAC to calculate the homography matrix H robustly.

After the homography transformation based on formula (7), the pixel from two different images at the same position in the overlap area in matching image would cause color deviation, so image fusion processing is necessary.

IV. PARALLEL POSITION-WEIGHTED IMAGE FUSION ALGORITHM

A. *Position-weighted image fusion algorithm*

As shown in Fig.2, image fusion based on two images which have already been matched. The color of the pixel in the overlap area is defined by the pixel from two images, let the scaling factor be α. As we all know, the color of the pixel in the overlap area should be close to the left image and the same to the other side on the right, makes the scaling factor α not a constant, which decided by the weighted-position. We should note that the edge of the overlap area is always the edge of one image but not the other, a constant scaling factor α would cause a trace like artifact.

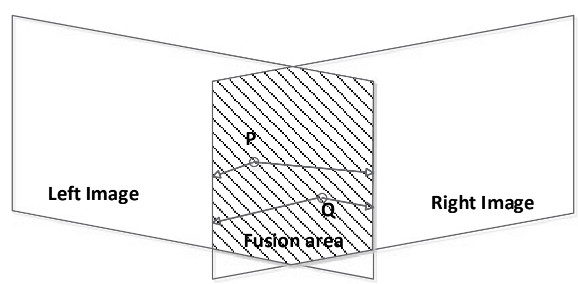


Fig.2 Position-weighted image fusion algorithm

Define the distance from a point $p(x, y)$ in the image to the left edge of the image, noted as $L_i(x, y)(i \in (1,2,3,4))$, the nearest distance from p to the left edge of the image is noted as:

$$DL(x, y) = \min(PL_i(x, y)) \tag{11}$$

Note that there is a Positive correlation between DL and the confidence of point p in the left image, in other words, the value of DL will increase as point p come closer to the center of the left image which means a high confidence degree from the left image and vice versa. Similarly, define the nearest distance from a point in the image to the right edge of the image, noted as $DR(x,y)$, with a similar confidence degree from the right image.

Considerate both DL and DR mentioned above, we define the factor of the fusion as follows:

$$\alpha(x,y) = \frac{DL}{DL+DR} \tag{12}$$

In which $0 \leq \alpha(x,y) \leq 1$, calculate the color-weighted of the point p in fusion image from left image.

Define the pixel value of the point p from the left image and the right image respectively as $IL_c(x,y)$ and $IR_c(x,y)$, and the pixel value of the fusion image at the same point is $I_c(x,y)$, in which $c \in (red, green, blue)$, the pixel value of point p in the image after position-weighted as follows:

$$I_c(x,y) = \alpha IL_c(x,y) + (1-\alpha) IR_c(x,y) \tag{13}$$

V. Parallel accelerate strategy

As the Position-weighted image fusion algorithm above is pixel-wise operation, this operation could be sent to multi-cores using OpenMP. Thus, this parallel boosting operation is also parallel accelerated. Then divide this rectangle into N parts in row equally (N is the number of core in CPU). Finally, we give the pseudocode of our proposed method in Algorithm 1.

Algorithm 1. Parallel accelerate algorithm

Input: Number of Core N, Core Number i, input image x, output image y, image height h;

Step 1: Parallel Feature Exaction

Divide image into N parts equally. According to the method in Section 3(A), do the feature extraction of each part in each core to calculate feature points simultaneously.

Step 2: Parallel Feature Matching

Divided both the left and right images into N parts equally. According to the method in Section 3(B), for instance, match each part in the left image to the whole right image, do the matching of each part in each core simultaneously.

Step 3: Parallel Image Fusion

Note that the shape of the overlap area is not a rectangle necessarily, we take the minimum circumscribed rectangle of the overlap as our input image. Then divide this input image into N parts equally, according to the method in Section 4, all of the nearest distance, the factor of the fusion and the pixel value can be calculated.

Output: The pixel value of each point in the overlap area of two images.

The specific strategy of dividing the image is as follows:

row_start = (h*i)/N, row_end = (h*(i+1))/N;

temp (row_start: row_end, :) = row_filter(I(row_start: row_end, :));

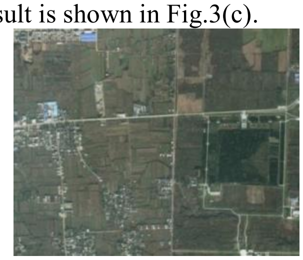
With the implement of the parallel design, computing speed is obviously improved and real-time image stitching for airborne embedded application is realizable. The specific results are shown and discussed in the next Section.

VI. Experimental Results

A. Figures and Tables

According to the implementation framework of parallel image detail enhancement on TMS320C6678, we can verify the proposed algorithm. This framework of video processing board based on TMS320C6678 is shown in Fig. 1. All methods are implemented with C++ and OpenMP programming languages.

Fig.3 shows the stitching result of aerial image sequence in different brightness, matching by the ORB feature method then optimized by several robust algorithm (such as best of 2nd nearest matcher, cross-validation and so on), the final matching result is shown in Fig.3(c).



a Left-side Image b Right-side Image

c Result of Image Matching

Fig. 3 Result for feather matching

After the registration process, the result of α-blending method and position-weighted method we propose for image fusion are as follows, we can obviously find the artifact on the edge of the overlap in α-blending method but not in ours. Thanks to the position-weighted algorithm, the image can be naturally stitched, the final result for image fusion as shown in Fig.4

a α-blending method

b Position-weighted method

Fig. 4 Result of image fusion

We apply our method in DSP platform using different image resolutions. The total time costs using 8 cores in 1024*768 and 720*576 resolution are 55.3ms and 28.9ms respectively. The real-time performance is proved in lower 720*576 resolution.

Table 1. shows the multi-core processing time and acceleration ratios. 1024*768 and 720*576 resolution images are listed in the table. The acceleration ratio is nearly linear to number of cores. Therefore, it's proved that all steps of Position-weighted image fusion algorithm are exactly parallel.

TABLE I. ACCELERATION RATIOS FOR DIFFERENT NUMBER OF CORES WITH DIFFERENT RESOLUTIONS

Number of Cores	1024*768		720*576	
	Processing Time (ms)	Acceleration Ratios	Processing Time (ms)	Acceleration Ratios
1	368.2	1.00	192.6	1.00
4	98.4	3.47	53.1	3.62
8	55.3	6.66	28.9	6.66

VII. CONCLUSION

This paper proposed an accurate and efficient algorithm for aerial image stitching based on the ORB feature point, to well eliminate the artifact of the stitching on the edge of the matching images. Then we gave out a parallel strategy realized on the TI's latest multicore DSP platform: TMS320C6678 to accelerate the algorithm above. Experimental results show that even for 720*576 resolution images, the processing time is below 30ms, which fulfilling the real-time application requirements, especially for airborne embedded application. Our method is insensitive to complex noise presented in input image data and proved to be a stable and efficient algorithm for image stitching and can be widely applied in airborne application field.

ACKNOWLEDGMENT

This work was supported by the AVIC Technology I novation Foundation under Grant No.2014D63130R and the Aviation Science Foundation under Grant No.2014ZC31004.

REFERENCES

[1] R.Szeliski, Image alignment and stitching: A tutorial[J]. Foundations and Trends in Computer Graphics and Vision, 2006, 2(1): 1-104.

[2] W.A.Forstner, feature based correspondence algorithm for image matching [J]. International Archives of Photogrammetry and Remote Sensing, 1986, 26(3): 150-166.

[3] C.Harris, M.A.Stephens, combined corner and edge detector [C]// Alvey vision conference, 1988: 15-50.

[4] D.G.Lower, Distinctive image features form scale-invariant keypoints[J]. International journal of computer vision, 2004, 60(2): 91-110.

[5] G.D.Hager, P.N.Belhumeur, Efficient region tracking with parametric models of geometry and illumination[J]. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 1998, 20(10): 1025-1039.

[6] S.Leutenegger, M.Chli, R.Y.Siegwart, BRISK Binary robust invariant scalable keypoints [C]// Computer Vision(ICCV), 2011 IEEE International Conference on. IEEE, 2011: 2548-2555.

[7] E.Rublee, V.Rabaud, K.Konolige K, et al. ORB: an efficient alternative to SIFT or SURF [C]// Computer Vision(ICCV), 2011 IEEE International Conference on. IEEE, 2011: 2564-2571.

[8] Xiangmin Zhao, Liming She. An Automatic and Robust Image Mosaic Algorithm[J]. Journal of Image and Graphics(Chinese), 2004, 9(4): 417-422.

[9] R.Hartley, A.Zisserman, Multiple view geometry in computer vision [M]. Cambridge: Cambridge University Press, 2000