LETTER

Real Time Aerial Video Stitching via Sensor Refinement and Priority Scan

Chao LIAO^{†a)}, Student Member, Guijin WANG^{†b)}, Member, Bei HE[†], Chenbo SHI[†], Student Members, Yongling SHEN^{††}, and Xinggang LIN^{†c)}, Nonmembers

SUMMARY The time efficiency of aerial video stitching is still an open problem due to the huge amount of input frames, which usually results in prohibitive complexities in both image registration and blending. In this paper, we propose an efficient framework aiming to stitch aerial videos in real time. Reasonable distortions are allowed as a tradeoff for acceleration. Instead of searching for globally optimized solutions, we directly refine frame positions with sensor data to compensate for the accumulative error in alignment. A priority scan method is proposed to select pixels within overlapping area into the final panorama for blending, which avoids complicated operations like weighting or averaging on pixels. Experiments show that our method can generate satisfying results at very competitive speed.

key words: aerial video stitching, image registration, sensor refinement, priority scan

1. Introduction

In recent years, low-cost unmanned aerial vehicles have been developed and widely used in various applications, such as agricultural information gathering, rural rescue and surveillance. Terrain videos captured on these vehicles from low altitude have better resolution than satellite images. But generally speaking, building panorama views from aerial video is computationally expensive task. It would be much more convenient if they could be stitched in real time with some distortion allowed.

There are several difficulties in stitching aerial video. Firstly, video usually contains huge amount of frames which results in prohibitive registration and blending time. Secondly, local alignment error propagates when a new frame is concatenated to the previous one. But the availability of exterior sensor data from gyroscope or positioning device can help to improve precision. Thirdly, the flight route may be curvy which cannot guarantee the scene moves in an invariant direction in the video. The stitching method should be robust enough to handle the change in moving direction.

Many research works have been reported in building panorama. Brown [1] matched unordered images with invariant features and blend with multi-band approach. But

detecting feature points is time consuming and prohibitive for real time applications. For images captured from a common optical center, Szeliski [2] proposed a direct method to align them with a 3D rotational model and blend with feathering method. This model is not applicable for aerial videos with moving cameras. Steedly [3] proposed a key frame selection scheme with better speed, yet still far from real time. Peleg [4] proposed a strip-based manifold projection method which can stitch videos in real time. But it is designed for panning camera without considering direction change in the scene, hence cannot be applied directly in aerial videos.

Many recent efforts on aerial video stitching aim to improve the accuracy without giving much consideration to efficiency. Heiner [5] applied expensive bundle adjustment technique to search globally optimized solution. Caballero [6] maintained an online mosaic to play the role of global satellite map. They all rely upon expensive local image features. Zhu [7] proposed an efficient method to stitch aerial video with GPS data. The geo-track from sensor is combined with frame alignment results to improve accuracy. This method collect pixels with scan lines, but still cannot handle direction changes in the scene.

In this paper, we propose an efficient framework aiming to build panorama views from aerial videos with only low-precision sensor data. Hence some tradeoff is made between accuracy and speed. As feature point detection and bundle adjustment techniques are far too expensive for large frame numbers, we choose to align consecutive frames with fast direct registration method and refine the positions with exterior sensor data. A pixel selection strategy, namely priority scan, is proposed to blend overlapping images efficiently. Experiments show that our method can achieve real time performance and produce satisfying panorama. It is robust for handling frequent direction change in the scene.

2. Framework

The schematic of our aerial video stitching framework is illustrated in Fig. 1. The framework consists of two major parts, frame registration and blending. And the modules are designed to achieve a fast overall stitching speed, while preserving a reasonable accuracy. Registration is achieved by first performing a pairwise alignment between neighboring frames and then concatenating new frames to the previous ones. As accumulative error is inevitable in this step, we incorporate sensor data like GPS to refine the frame positions

Manuscript received December 22, 2011.

Manuscript revised March 28, 2012.

[†]The authors are with the Department of Electronic Engineering, Tsinghua University, Beijing, 100084, China.

^{††}The author is with 54th Institute of China Electronics Technology Group Corporation, Shijiazhuang, 050081, China.

a) E-mail: liaoc02@mails.tsinghua.edu.cn

b) E-mail: wangguijin@tsinghua.edu.cn

c) E-mail: xglin@tsinghua.edu.cn

DOI: 10.1587/transinf.E95.D.2146

Fig. 1 Aerial video stitching framework.

and improve the accuracy. With compensated positions, we propose a pixel selection strategy, namely priority scan, to blend all frames into the final panorama efficiently.

3. Frame Alignment

For two frames I_i, I_j taken from a common planar scene, their relationship can be described with a homography [8], as shown in Eq. (1). This equation holds for aerial video if the aircraft is flying at high altitude or over a flat area. In these cases the terrain can be treated as planar. However, high degrees of freedom cannot guarantee high precision. In fast coarse-to-fine alignment, homography is found to be not as stable as affine model [10]. By setting $p_7 = p_8 = 0$, we actually use affine inter-frame motion for most aerial videos and achieve real time alignment.

$$H_{ij} = K(R_c - \frac{t_c}{h}n_d)K^{-1} = \begin{bmatrix} p_1 & p_2 & p_3 \\ p_4 & p_5 & p_6 \\ p_7 & p_8 & 1 \end{bmatrix} \quad (1)$$

3.1 Pairwise Registration

We prefer the Lucas-Kanade method [9] and a hierarchical acceleration mechanism [10] for efficient alignment. This method iteratively searches for the model parameters p by minimizing the difference term E between two frames I and T , as shown in Eq. (2). The iteration stops when the update vector Δp is small enough and indicates the convergency.

$$E = \sum_i [I(\text{warp}(x_i, p + \Delta p)) - T(x_i)]^2 \quad (2)$$

This alignment module is very fast and important to the overall stitching speed. But it may fail in difficult situations such as illumination change or large textureless areas. e.g. video frames dominated by water, grass-land, sand or house roof are possible causes. If the convergency is not achieved in limited iterations, we turn to feature-based alignment [1] for back up. This way we can enhance the average speed while preserving robustness in worst cases.

3.2 Sensor Refinement

Exterior sensor data can be used to refine the frame posi-

tions since their precision only depends on hardware which is immune to accumulative error. Compared with searching for globally optimized solutions via bundle adjustment, this method works in real time and provides comparable result.

For a certain frame I_i , its registration matrix in the panorama H_i is evaluated by concatenating all homography $H_{j(j+1)}$ on the path from the reference frame I_0 to I_i , as shown in Eq. (3). With sensor data available, the refined position $\bar{H}_i$ is acquired by shifting the frame center towards the location where the sensor indicates.

$$H_i = \prod_{j=i-1}^0 H_{j(j+1)}, H_{i+1} = H_{i(i+1)}\bar{H}_i \quad (3)$$

Let S_i denotes the scene point at the center of frame I_i . We can predict its coordinate in the panorama in two ways. By concatenating I_i to the previous frame, we obtain a prediction X_i from alignment. By establishing the mapping from GPS to the panorama coordinate, we obtain another prediction X_g from sensor. The problem is how to generate a refined result by combining two sources. As the precision of our sensor is insufficient to reflect the pixel-level inter-frame motion, we prefer to place more weights on frame alignment. The refinement formula is listed in Eq. (4). When the difference between two predictions exceeds a threshold, we shift X_i towards X_g with a fixed length δ . δ is set to a small value like 1-2 pixels to compensate for the accumulative error. In this way, large registration error can be distributed and compensated in several frames to avoid visible seams or sudden gaps in the panorama. This limited update length also provides robustness against occasionally false sensor data. The threshold value depends on the resolution that the sensor data can provide in the panorama coordinates. Compared with Zhu's [7] method, they place more weights on sensor prediction, which requires higher-precision device and Kalman smoothing for reliable geo-track.

$$X_r = X_i + \delta \cdot \frac{X_g - X_i}{\|X_g - X_i\|}, \text{ if } \|X_g - X_i\| > thr \quad (4)$$

4. Priority Scan

To efficiently blend many video frames into a panorama is also a problem. Many composition methods like multi-band [1] or feathering [2] essentially assign different weights to the pixels within overlapping area and average them. Gradient-based composition methods are even more complicated which need to solve equations on image boundaries. They are too slow for real time requirement because the overlapping area is very large in video.

Instead of considering all overlapping pixels, Peleg [4] preferred to assign full weight to those pixels near frame center and discard other ones. This method simulates a pushbroom camera. Narrow strips near frame center are extracted and stitched into panorama. The strip content is equivalent to the pixels collected on the pushbroom scan line. The overlapping area would separate into several strips

depending on when the scan line reaches the scene. As any scene content is swept across by the camera only once, no two strips should contain overlapping pixels. Hence it is not necessary to assign weights to pixels which results in an extraordinary blending speed. However, this method cannot handle direction change in the scene as the strips are designed for panning camera. Its vertical strip works well if the scene moves horizontally, but fails to collect any pixel if the scene moves vertically. Besides, the compulsive rectification from tilted strip to rectangular shape also introduces artificial distortion if the camera is not really panning.

To overcome these difficulties while preserving the blending speed, we propose a pixel selection strategy, namely priority scan, to simulate the pushbroom camera. Instead of extracting strips explicitly, we define a priority template to represent the scan area and collect pixels via priority comparison. The template is a numerical image with the same size as the input frames. Its value indicates the rendering priority of certain pixel in the final panorama. We define two perpendicular scan lines in the template by setting pixels near them with high priority q_1 , other pixels with low priority q_0 , as shown in Fig. 2 (a). The scan area can also be defined in other shapes. e.g. If we want to discard distorted pixels near the border of the frames, we can set their priority values to zero in the template.

The procedure of priority scan is listed in Table 1. With refined register matrix, we warp both the video frame and the priority template to the panorama coordinate, see Fig. 2 (b). From the warped vertex, we can identify the outline rectangle R . Only pixels within R could be updated. For each location y_i in R , we update it with new pixel value in the warped frame, only if its priority in the warped template is higher than its current priority in the panorama. Since $q_1 > q_0 > 0$, new pixels within scan area will replace existing pixels of low priority. New pixels outside scan area can still fill empty area of the panorama which can remove potential seams. This priority rendering mechanism is equivalent to a virtual pushbroom scan, but requires no explicit-form strips. As the scan area warps with each frame, it is adaptive to the direction change of the moving scene. The distortion introduced in strip rectification also

Fig. 2 Pixels within scan area will be rendered with high priority.

Table 1 The procedure of priority scan.

1. Set higher value to the scan area in the priority template T
2. warp frame I to I_w , template T to T_w in the panorama coordinate
3. find the outline rectangle R from the warped vertex of I_w
4. for each pixel y_i in R , find its current priority $P_M(y_i)$ in panorama M
5. render the pixel $M(y_i) = I_w(y_i)$ if $T_w(y_i) > P_M(y_i)$

disappear with strips. Zhu [7] use scan lines instead of strips to collect pixels. But their scan lines can only translate with directions unchanged. Hence they still require an invariant moving direction of the scene.

5. Experiments

We test the performance of our method with both simulated and real aerial videos. The experiments are carried out on a PC platform with 3.0 GHz Core2 CPU and 2 GB memory. The result is compared with strip-based method [4] and Image Composite Editor (ICE). ICE is a software published by Microsoft, which integrates mature panorama-building techniques [2], [3] and is a reference in comparison.

Figure 4 is the stitching results of a simulated video containing 1000 frames of size 320×240 . The coarse-to-fine Lucas-Kanade method can register almost all frames in this video except 2 difficult pairs with slow convergency. Figure 3 shows one such example around the river bank. The textureless water areas cannot provide correct predictions on the gradient direction in which the alignment error would decrease. Hence the update vector Δp to the model parameters fails to converge in 40 iterations in Fig. 3 (b). These worst cases can be handled by combining feature-based method as a fall back with Lucas-Kanade alignment, as shown in Fig. 3 (a). 53 out of the 1000 frames (5.3%) in the simulated video are adjusted by the refinement formula in Eq. (4). The parameters are specified to $\delta = 1$, $thr = 5$ in this case. thr is set small because the simulated sensor data are quite accurate and predict the frame locations very well.

The flight route in Fig. 4 is set to curve which reflects the direction change of the scene. There are two turning phases in the route where the aircraft flies in northeast and in southeast separately. In Fig. 4 (b), the result of strip-based method becomes very narrow at the turning phase. The vertical strip (the blue rectangle) cannot collect as much pixels as in the horizontal phase. In contrast, our priority scan method handles the turning smoothly, as the scan area warps with the rotating frames. The effective scan range (the dotted red line) at the turning phase is significantly larger in our method, as shown in Fig. 4 (c).

The quantitative evaluation for this video is listed in Table 2. Since ICE discards many non-key frames, it covers less pixels than our method. In fact, our method achieves the highest coverage rate and is most similar to the ground truth. As many pixels need to be compared before rendering, the blending speed of priority scan is slower than the strip-based method. But with efficient pyramidal registration, our

Fig. 3 Back up Lucas-Kanade registration with feature-based method.

Fig. 4 Comparison on simulated video.

Table 2 Quantitative evaluation on simulated video.

Method	Valid Pixels	Coverage	Speed
ICE	60560	84.40%	< 5 fps
Strip-based	46110	64.26%	34.5 fps
This	71735	99.97%	29.8 fps

Fig. 5 Comparison on real aerial video.

method also achieves a real time stitching speed.

Figure 5 is the stitching results of a real video containing about 500 frames of size 512×344 . The camera keeps rotating when the aircraft moves forward. Hence the scene

changes its moving direction in this video. Because the texture is rich in this scene, it is sufficient to register all frames in this video with Lucas-Kanade method. The sensor refinement formula adjusts 87 out of 500 frames (17.4%) in the real video, with the parameters set to $\delta = 1.5$, $thr = 10$.

The result of ICE in Fig. 5 (a) is near the ground truth. However, the broken road inside the circle reveals the registration error, which is caused by unfitting inter-frame motion models [2]. In Fig. 5 (b), the strip-based method forcibly rectifies the tilted panorama into a straight one. The left and right edges of the scene are parallel, but obviously this is not right as the camera is not really panning. Figure 5 (c-d) are our stitching results, and Fig. 5 (c) is the one without sensor refinement. Inside the rectangles, the cut on the road in Fig. 5 (c) disappears in Fig. 5 (d), which is a benefit from sensor refinement. The road in Fig. 5 (d) is the most continuous one compared with other results. This shows that our method can handle camera rotations smoothly and proves the effectiveness of priority scan.

6. Conclusion

In this paper, we present an efficient framework to stitch aerial videos in real time with low-precision sensor data. Fast modules of frame alignment and sensor refinement are presented for acceleration. A blending scheme, namely priority scan, is also proposed which can deal with frequent direction change in the scene. Our method can stitch aerial video in real time without much observable distortion. In the future, we will consider integrate depth data for 3D map.

References

- [1] M. Brown and D. Lowe, "Recognizing panoramas," Proc. 9th Int. Conf. on Computer Vision, pp.1218–1225, Nice, France, Oct. 2003.
- [2] R. Szeliski and H.Y. Shum, "Creating full view panoramic image mosaics and texture-mapped models," Proc. ACM SIGGRAPH, pp.251–258, Los Angeles, 1997.
- [3] D. Steedly, C. Pal, and R. Szeliski, "Efficiently registering video into panoramic mosaics," Proc. 10th ICCV, vol.1-2, pp.1300–1307, 2005.
- [4] S. Peleg, B. Rousso, A. Rav-Acha, and A. Zomet, "Mosaicing on adaptive manifolds," IEEE Trans. Pattern Anal. Mach. Intell., vol.22, no.10, pp.1144–1154, 2000.
- [5] B. Heiner and C.N. Taylor, "Creation of Geo-Referenced Mosaics from MAV Video and Telemetry Using Constrained Optimization and Bundle Adjustment," ICIRS, pp.5173–5178, 2009.
- [6] F. Caballero, L. Merino, J. Ferruz, and A. Ollero, "Improving vision-based planar motion estimation for unmanned aerial vehicles through online mosaicing," IEEE ICRA, vol.1-10, pp.2860–2865, 2006.
- [7] Z.G. Zhu, E.M. Riseman, A.R. Hanson, and H. Schultz, "An efficient method for geo-referenced video mosaicing for environmental monitoring," Machine Vision and Applications, vol.16, no.4, pp.203–216, 2005.
- [8] R. Hartley and A. Zisserman ed., Multiple View Geometry in Computer Vision, Cambridge University Press, 2000.
- [9] S. Baker and I. Matthews, "Lucas-kanade 20 years on: A unifying framework," IJCV, vol.56, no.3, pp.221–255, 2004.
- [10] J.R. Bergen, P. Anandan, K. Hanna, and R. Hingorani, "Hierarchical model-based motion estimation," Proc. 2nd ECCV, pp.237–252, 1992.