2009 Third International Conference on Multimedia and Ubiquitous Engineering

Spherical Mapping based Motion Recovery for Panoramic Cameras

Yanli Li Zhong zhou Wei Wu
State Key Laboratory of Virtual Reality Technology and System, Beihang University
School of Computer Science and Engineering, Beihang University
Beijing, China
e-mail: liyl@vrlab.buaa.edu.cn

Abstract—**Although motion recovery for limited field-of-view (FOV) cameras has been studied for decades under the pin-hole projection, few attentions have been paid on omni-directional cameras. Omni-directional cameras are usually constituted complicatedly, especially some are made of several limited FOV cameras, and thus motion recovery for omni-directional cameras is more different. In this paper, an algorithm is presented to estimate motion of the omni-directional camera under the spherical projection. Unlike motion recovery algorithms for traditional limited FOV cameras, in which the intrinsic matrix plays an important part, this algorithm only estimates extrinsic matrices without the influence of intrinsic matrix. So it is more efficient and robust. To reduce computation, a method is also presented to select key frames using the translation offset of neighboring frames and remove some mismatched feature points. The experiments verify the efficiency of the algorithm and the visual result is demonstrated through a simulation experiment.**

Keywords- Motion recovery; Epipolar geometry; Spherical projection; Bundle adjustment

I. INTRODUCTION

Motion recovery, also called Structure and Motion recovery (SAM), is to estimate the postures and positions of cameras from sequence frames of the captured video. Based on the position and posture of each frame, 3-Dimension (3D) scene points can be obtained. It can be further applied to image reconstruction, object extraction, video compression, virtual navigation, augmented virtual and so on. Motion recovery and its applications for videos captured by limited FOV cameras have been studied for decades. Few attentions have been paid on panoramic videos. The panoramic video is composed of panoramas which cover the omni-directional scene. Therefore, the panoramic video has more advantages than the limited FOV video and its motion recovery can be used more widely.

Motion recovery for traditional limited FOV cameras is based on the pin-hole projection. The parameters need to be estimated involve internal parameters and external parameters. If the images are captured by cameras of short focal length, the distorted parameters, as a part of internal parameters need to be estimated too. Traditionally, internal parameters are acquired through the calibration, but the preprocessing is complex. To remove the influence of internal parameters, we propose an algorithm to recover motions based on the spherical projection. Unlike the pin-hole projection whose feature points are located on planar images, the spherical projection's feature points are located on the unit spherical surfaces. Because all features have been normalized under the unit spherical mapping, the influence of internal parameters is removed totally. With only estimation of the external parameter, the algorithm is more efficient and robust.

Since motion recovery bases on tracking points, the length and matching accuracy of tracking points between neighboring frames have a significant effect. Generally, the longer the baselines are, the more robust the estimation will be. Therefore, it is necessary to select key frames with a longer baseline and only estimate motions of key frames. Motions for other frames are obtained through the linear interpolation. In this paper, we analyze the motion trending of panoramic video and key points horizontal offset, with which key frames from video can be extracted and some mismatched tracking points can be removed.

II. RELATED WORK

As one of the important parts of computer vision, the theory of motion recovery has been studied for decades [1, 2].Traditional, it is started with features matching. The popular feature algorithms include Harris [3], KLT [4], SIFT [5] and SURF [6]. Then, inliers with consistent epipolar relationship are chosen through RANSAC [7] or other methods and external parameters are obtained through the epipolar relationship [8]. After adjustment for all motions in a unified frame, a bundle adjustment [9, 10] is adopted to refine all parameters finally. The whole process is totally automatic. Theory of motion recovery for limited FOV cameras is more mature, even in condition of cameras' zooming in or out constantly [11]. There is also some related commercial software [12]. While limited FOV cameras suffer from the problem that the field of view is limited. It's impossible to capture large field of scene. Ideally, one would like to use a panoramic camera to capture omni-directional videos which can be used in some special fields [13]. Some of those omni-directional devices are composed of multiple cameras and the recovered motion should be the motion of the whole device. Motoko,O. [14] and Iwane [15] only estimated motions of one camera and used them as the whole device's motions. Because the focal length of those cameras is usually short and the original images are distorted. It is necessary to calibrate the camera previously to acquire the distortion factors and the intrinsic matrix [16, 17]. This

978-0-7695-3658-3/09 $25.00 © 2009 IEEE
DOI 10.1109/MUE.2009.23

processing leads to a lot of complex computation and might imports errors. What's more, motions of one camera are not exactly motions of the whole device, especially when optical centers of cameras are not located at the same place. Kangni F., et al [18] presented a method to recover motions of such devices from cubic panoramas. They remapped each spherical panorama onto a cube to obtain the cubic panorama, extracted and matched features of each pair-wise cubic panoramas, used six sides of each cubic panorama to compute motions of the device. Their method does not need to calibrate camera and is easy to implement. But it needs to map each spherical panorama to the cubic panorama firstly. Concerning the panoramic videos we captured are under the spherical model and the spherical projection is more similar to real world scene rays' projection, we present an algorithm to recover postures and positions for spherical cameras with the theory of epipolar geometry for spherical cameras [19]. The proposed algorithm bases on the spherical projection geometry. Unlike the algorithms whose key points are located in the image planar [11, 12, 18], this algorithm bases on points of spherical surface. As those key points have been normalized under spherical mapping, the influence of the intrinsic matrix has been removed totally.

To select key frames from the video, Fitzgibbon [19] and Zhang [11] estimated the offset lengths of tracking points, with which they chose some suitable frames. As the moving trend of points on panoramic video is different with that on limited FOV video, their methods can not be applied to panoramic video directly. In this paper, we propose a method to select key frames from panoramic videos and remove some mismatched feature points meanwhile.

The rest of the paper is organized as follows: Section III describes the theory of spherical projection and gives some notations of spherical geometry, including the remapping from a spherical panorama to a sphere, epipolar geometry for spherical cameras and the elimination of mismatched key points. In section IV, the SAM algorithm of panoramic cameras will be presented. Section V provides experimental data to verify this algorithm. We discussed this method in section VI.

III. TWO-VIEW GEOMETRY FOR THE SPHERICAL PROJECTION

The projection model adopted here is the spherical model, so each panorama should be mapped on a unit sphere and the estimation procession is established on the spherical epipolar geometry. Two-view epipolar geometry is the basis for SAM, we will describe some notations and the two-view spherical epipolar theory in this section.

A. Mapping under the spherical model

The spherical coordinate is showed as Fig. 1. Suppose the width of panorama is w and the height is h. (x, y) is a pixel on the panorama and its corresponding angle under the spherical coordinate is (θ, δ). θ is the longitude angle and δ is the latitude angle. The spherical model is a unit sphere. The point $m(X,Y,Z)$ on the spherical surface is the corresponding point of (x, y).

Their conversion formulas are as follows:

$$\begin{cases} \theta = y * \pi / h \\ \delta = x * 2\pi / w \end{cases} \tag{1}$$

$$\begin{cases} X = \sin\theta\cos\delta \\ Y = \sin\theta\sin\delta \\ Z = \cos\theta \end{cases} \tag{2}$$

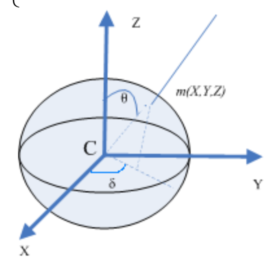


Figure 1. The spherical coordinate

Epipolar geometry for spherical projection is based on the key point pairs on spherical surface, so each key point of panoramas should be mapped on the unit spherical through (1) (2) to calculate its corresponding point firstly.

B. Epipolar geometry for spherical projection

Epipolar geometry is to study the motion relationship of two spherical cameras. Support there are two spherical cameras as showed in Fig. 2. The first camera is located on the origin of the world coordinate. Its rotation matrix is identity matrix and its translation vector is zero vector. So the projection matrix of camera 1 is $P_1 = [I \mid 0]$. R and t are the rotation matrix and the translation vector between camera 1 and camera 2. So the projection matrix of camera 2 is $P_2 = [R \mid t]$. If a 3D point M in world coordinate is mapped to m_1 on the unit spherical camera 1 and mapped to m_2 on the unit spherical camera 2, we have the equations:

$$\begin{cases} \lambda m_1 = P_1 M = M \\ \lambda' m_2 = P_2 M = RM + t \end{cases} \tag{3}$$

λ and λ' are the scale factors.

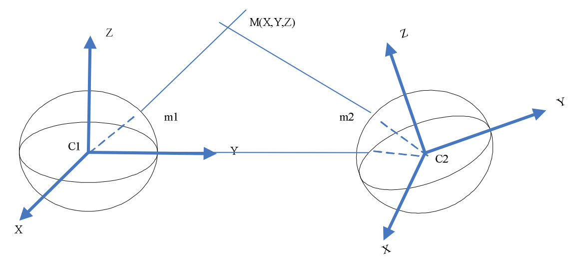


Figure 2. The epipolar geometry for two spherical cameras

Just as the theory of epipolar geometry for the pinhole cameras, the vector m_2, $R * m_1$ and t are coplanar. So point m_1 and point m_2 satisfies:

$$m_2^T E m_1 = 0 \tag{4}$$

E is the essential matrix, which can be recovered from all corresponding point pairs (m_1, m_2) using 8 point algorithm or 7 point algorithm or 5 point algorithm [20].

According to the epipolar geometry:

$$E = [t]_{\times} R \tag{5}$$

Introducing an anti-symmetric matrix S:

$$S = [t]_{\times} = \begin{bmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{bmatrix} \tag{6}$$

Then $E=SR$.

Here, R is an orthogonal matrix which satisfies: $RR' = R'R = I$, $\|R\|=1$.

Another two formats of R which are often used in computer vision are: format of quaternion $Q=(q_w, q_x, q_y, q_z)$ and format of Euler angle $U = (\alpha, \beta, \gamma)$.

Quaternion $Q=(q_w, q_x, q_y, q_z)$ can be transformed into R through equation:

$$R = \begin{bmatrix} 1-2q_yq_y-2q_zq_z & 2q_xq_y-2q_zq_w & 2q_xq_z+2q_yq_w \\ 2q_xq_y+2q_zq_w & 1-2q_xq_x-2q_zq_z & 2q_yq_z-2q_xq_w \\ 2q_xq_z-2q_yq_w & 2q_yq_z+2q_xq_w & 1-2q_xq_x-2q_yq_y \end{bmatrix} \tag{7}$$

Euler angle $U = (\alpha, \beta, \gamma)$ can be transformed into R through equation:

$$R = \begin{bmatrix} \cos\alpha\cos\beta & \cos\alpha\cos\beta\sin\gamma-\sin\alpha\cos\gamma & \cos\alpha\sin\beta\cos\gamma+\sin\alpha\sin\gamma \\ \sin\alpha\cos\beta & \sin\alpha\sin\beta\sin\gamma+\cos\alpha\cos\gamma & \sin\alpha\sin\beta\cos\gamma-\cos\alpha\sin\gamma \\ -\sin\beta & \cos\beta\sin\gamma & \cos\beta\cos\gamma \end{bmatrix} \tag{8}$$

The essential matrix E can be decomposed into a rotation matrix R and a translation vector t. Based on singular value decomposition (SVD), Hartlay [2] proposed a robust method to decompose E. E is decomposed by SVD:

$$E=UDV^T \tag{9}$$

The matrices R and t are acquired by:

$$t=U(0, 0, 1)^T \text{ or } t=-U(0, 0, 1)^T \tag{10}$$

$$R=UWV^T \text{ or } R=UW^TV^T \tag{11}$$

where $W = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

C. Elimination of mismatched key point pairs

The Essential Matrix E is essential to the whole algorithm. It only depends on the feature point pairs. If the number of mismatched key point pairs reaches to a certain extent, the Essential Matrix would be calculated incorrectly. Consequently, the accuracy of the rotation matrix R and the translation vector t would degrade. Therefore the selected key point pairs should be matched as precisely as possible.

We proposed a method to remove some mismatched key point pairs. The automatic feature tracking algorithm is based upon the iterative KLT algorithm [4]. Since the videos are usually captured along a road, if we suppose the camera is static and the scene are moving in front of the camera, the same tracking points will be located on lines. Those lines will map into curves on the panoramas. Those curves indicate the moving trend of key points. As Fig. 3 shows, all key points are extracted and matched between two frames, and the small lines indicate the orientation and offset length of the key point pairs.

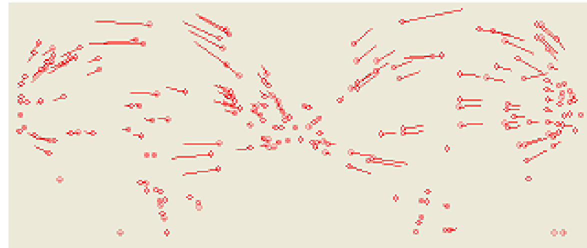


Figure 3. Orientation and offset length of key points in a frame

Two aspects are obtained after analyzing the moving trend:

- There are two moving trends for each panoramic video. One is that all key points will move towards center, while another is that all key points will move outsides.
- The horizontal offset length for each key point pair on two neighboring frames is in a certain range.

With the two aspects, some mismatch key points can be removed through the following algorithm:

Step 1: Make a judgment of the moving trend of the panoramic video. Whether a key point (x, y) is moving forwards or backwards is judged through $\hat{x} = \| x - w/2 \|$. x indicates the horizontal position of the key point and w is the width of the panorama frame. If most $\hat{x}$ become larger in the next frame, we support that the video is moving forwards.

Step 2: Calculate the maximum approximate horizontal offset length of key points. We only calculate from the key points whose corresponding angles on sphere satisfy: $\theta \in [\pi/4, 3\pi/4] \cup [5\pi/4, 7\pi/4]$, so the key points selected are located on two sides. Here, we only consider the situation when the vehicle is driving forwards. The horizontal offset length of each feature pair (x_1, y_1, x_2, y_2) on two neighboring frames is calculated through: $l(x_1, x_2) = x_1 - x_2$.

Establishing a histogram with $l(i)$ ($i=1\ldots N$) and we get the number of point pairs for each offset length. The number of pairs whose offset length is l is set as $hist(l)$. The estimated maximum offset L is calculated through:

$\sum_{l=0}^{l=L} hist(l) > r * N$, where N is the number of the total pairs, r is a ratio which is set as 0.9.

Step 3: Select key frames and remove some mismatched key points.

If $L<L_{max}$, this neighboring frame pair is considered being captured too close, so the latter frame of the frame pair will be removed from the key frames set, it would not be selected as a key frame. L_{max} is a constant which is set as $L_{max}=20$.

If this frame is selected as a key frame, it will be used to estimate its motion later. Otherwise we will consider the next frame. As for the neighboring key frame pairs, some mismatched key point are eliminated first. Key points will be removed if they do not satisfy:

$$\begin{cases} (x_1 - x_2) \in (0 \quad L] & x_1 < w/2 \\ (x_2 - x_1) \in (0 \quad L] & x_1 > w/2 \end{cases}$$

The elimination of the mismatched key point pairs is executed after key points are selected and tracked. As this process removes most of mismatched pairs, the essential matrix E will be calculated more accurately with RANSAC and the 8-point algorithm later.

To verify the robustness of the method, we chose 4 pairs of neighboring frames randomly and compared the following parameters before and after applying the elimination method:

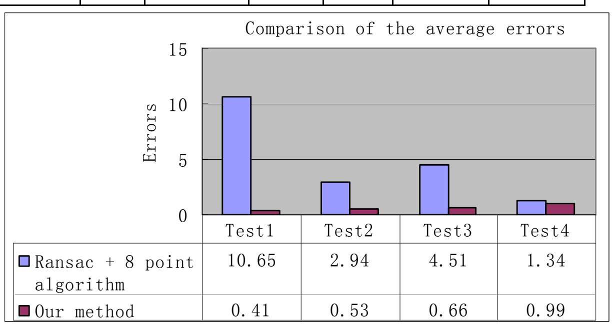
- N: The number of all KLT feature points of two neighboring frames.
- N_{inlier} : The number of inliers, inliers point pairs (m_1, m_2) are those point pairs which satisfy $m_2^{T}Em_1<T$, while the outliners are those whose error are larger than T, T is a constant.
- $\overline{e}$: The average errors.

$$\overline{e} = \sum_{i=N_{inlier}} \left\| m_2^{T} E m_1 \right\| \Big/ N_{inlier}$$

The parameters obtained are listed in Table 1:

TABLE I. THE COMPARISON OF ERRORS BEFORE AND AFTER APPLYING THE ELIMINATION METHOD OF OUTLINERS.

	Elimination + Ransac + 8 point algorithm			**Ransac + 8 point algorithm**		
	N	N_{inlier}	$\overline{e}$	***N***	N_{inlier}	$\overline{e}$
Test 1	115	103	10.6	115	69	0.41
Test 2	109	106	2.94	109	79	0.53
Test 3	108	102	4.51	108	86	0.66
Test 4	112	96	1.34	112	83	0.09



In table 1, it shows that if the key point pairs are selected with the elimination method before computing the essential matrix, some outliners will be removed previously and the essential matrix will be computed more accurately.

IV. SAM FOR A SEQUENCE SPHERICAL PANORAMA FRAMES

Based on the epipolar geometry for two neighboring frames, poses for all cameras can be adjusted in a unified frame and 3D scene points corresponding to feature points on the spherical surface can be obtained through the triangulation. A bundle adjustment is applied at last to refine all parameters. The whole process is described in the flowchart Chart 1.

A. *Adjustment of the positions and postures for all cameras*

After the relative position and posture between two cameras have been recovered through the epipolar geometry, the absolute positions and postures for all cameras can be iteratively computed if the coordinate of the first camera is set as the world coordinate. Suppose the relative position and posture are R_i and T_i for camera i-th and camera i+1-th, the corresponding relative projection matrices for all adjacent cameras are:

$$\begin{cases} P_1 = [I \mid 0] \\ P_i = [R_i \mid T_i] \quad i = 2 \ldots n \end{cases} \tag{12}$$

After adjustment, the absolute projection matrices for all cameras are:

$$\begin{cases} P_1' = [I \mid 0] \\ P_i' = [R_i R_{i-1}' \mid T_i * \prod_{j=1}^{i} s_j + R_i T_{i-1}'] \quad i = 2 \ldots n \end{cases} \tag{13}$$

where s_j is the scale factor, it is the average ratio of the distances of 3D points between the two neighboring models.

B. *Triangulation for spherical camera*

3D scene points are computed through triangulation. As Fig. 2 shows, we have the projection matrix P_1 of camera 1, the projection matrix P_2 of camera 2 and the point pair (m_1, m_2) on the spherical coordination. As $m_1 \times (P_1 M) = 0$, $m_2 \times (P_2 M) = 0$ and $m_1=(X_1, Y_1, Z_1)$, $m_2=(X_2, Y_2, Z_2)$, we have equations:

$$AM = \begin{bmatrix} X_1 P_1^{3T} - Z_1 P_1^{1T} \\ Y_1 P_1^{3T} - Z_1 P_1^{2T} \\ X_2 P_2^{3T} - Z_2 P_2^{1T} \\ Y_2 P_2^{3T} - Z_2 P_2^{2T} \end{bmatrix} M = 0 \tag{14}$$

where P_1^{i} is the i-th column of P_1, P_2^{i} is the i-th column of P_2 . A is decomposed based on the SVD, $A=UDV^{T}$ and $M=V\,(0, 0, 1)^{T}$

C. *Bundle adjustment*

Since errors are accumulated during the adjustment phase, the positions and postures of cameras and 3D points should be refined through a bundle adjustment. Bundle adjustment refines all parameters, including camera positions, postures and 3D points. After the bundle adjustment, the mean squared distances between the observed spherical surface points and the remapping points of 3D points on the spherical surface will be minimized.

Suppose the projection matrix of camera i-th is P_i (i=1…k), $m_{ij} (i \in \Gamma_j)$ is the map point of 3D points M_j (j=1…n) on camera i-th, where Γ_j is the set of image index indicating the remapping images of M_j. The cost function to be minimized is:

$$f = \min_{R_i, t_i} \sum_{j=1}^{n} \sum_{i \in \Gamma_j} d^2 (m_{ij}, \hat{m}_{ij}) \tag{15}$$

where $d^2 (m_{ij}, \hat{m}_{ij})$ donates the distance of the detected point and the projective point. This function is minimized through LM algorithm [21].

Algorithm: SAM for panoramic cameras
Input: *n* frames of a spherical panoramic video

I. Track KLT features along n panoramic frames

II. Select key frames and remove mismatched features through the method mentioned in section 3.3

III. For each pair of key panoramic frames

a) KLT features are remapped onto the united sphere to compute the corresponding points on the spherical surface. The KLT features are points whose spherical angle $\theta \in [\pi/4, 3\pi/4] \cup [5\pi/4, 7\pi/4]$.

b) The essential matrices E is computed with RANSAC and the 8-point algorithm, the rotation matrix R and the translation matrix t are decomposed through E

IV. Set the first frame's pose as R=I, t=0 and adjust rotation matrices and translation matrices for all key frames with the formula mentioned in section 4.1

V. Compute 3D points with the triangulation.

VI. Refine rotation matrices, translation matrices and 3D points through the bundle adjustment.

VII. Obtain motions for other frames with motions of key frames through linear interpolation

Output: Positions and postures for all frames, amount of 3D scene points.

Chart 1 The flowchart of SAM for a panoramic video

V. EXPERIMENTS

The omni-directional device used is Pointgrey Corp.'s Ladybug 2 which captures images at a rate of approximate 15FPS and the panorama size is 1280*720. The GPS of each panoramic frame was also captured and the rate of the GPS device is about 1 time/second. To reduce calculation, we first filtered frames with GPS, ensuring each filtered frame has a different GPS. Then we selected key frames from the filtered frames with the method mentioned in section 3.3. Fig. 4 shows a sequence of panoramic frames of Beihang campus.

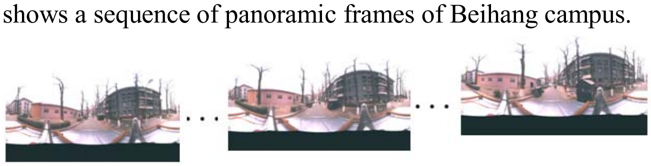


Figure 4. A sequence of panoramic frames

We implemented the algorithm discussed above in C++. To verify the algorithm, we tested three sequences of panoramic frames and compared the result with that of Boujou4. Boujou4 is commercial motion estimation software. As the videos that Boujou4 handles are ordinary limited FOV videos, we first converted each spherical panorama into the cube panorama to extract a cubic side as the inputted frame for Boujou4. Fig. 5 shows a sequence of frames composed of the cubic sides. As Florian Kangni [17] stated, if the width of the cubic side is L, the focal length can be set as L/2 in boujou4.

Figure 5. A sequence of frames composed of the cubic sides

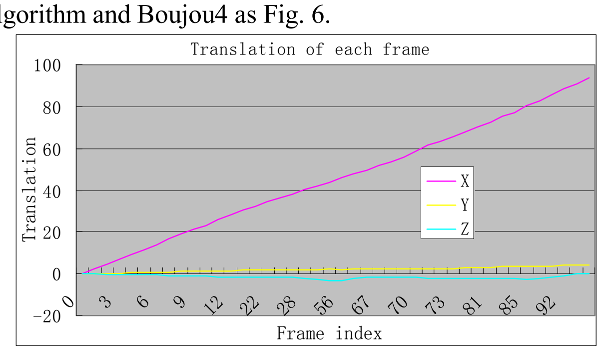
Both our algorithm and boujou4 are totally automatic. After the frames are inputted, feature points are first tracked along the frames; motions are then solved to get the translation parameters, the rotation parameters and amount of 3D points. The statistics are listed in Table 2.

TABLE II. STATISTICS OF OUR ALGORITHM AND BOUJOU 4

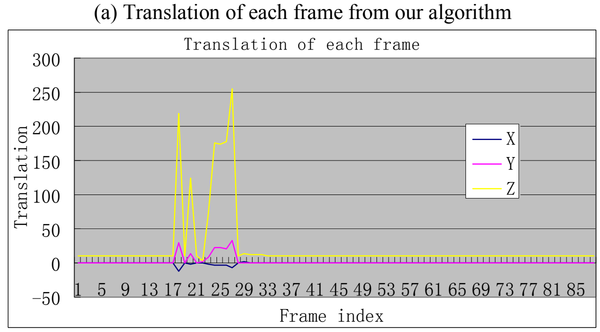
	Scene 1		Scene 2		Scene 3	
	Our algorithm	***Boujou4***	***Our algorithm***	***Boujou4***	***Our algorithm***	***Boujou4***
Frames	20	20	50	50	100	100
Key Frames	17	20	29	50	42	100
Average features.	226	375	237	405	236	418
3D points	1086	45	1693	155	2580	70
Residual Errors(pixels)	0.53	0.32	0.58	0.36	0.64	0.37
Tracking time (sec.)	14	16	31	30	62	63
Solving time (sec.)	1	16	10	173	18	436

As Table 2 shows, because our algorithm selected key frames from the inputted frames, the total frames handled in the solving process are reduced in our algorithm. Boujou4 extracted more feature points per image, but most were removed and 3D scene points were less. Through analysis of the residual errors, we can conclude that Boujou4 is more accurate, while the residual errors less than one pixel is enough for most cases. As for the performance, the tracking time of both is appropriately equivalent, but our algorithm took less time than that of Boujou4 in the solving process. As the frames are increased, our algorithm will take much less time than that of Boujou4.

We analyzed the translations of Scene 3 with our algorithm and Boujou4 as Fig. 6.



(a) Translation of each frame from our algorithm



(b) Translation of each frame from Boujou4

Figure 6. Translation of each frame from our algorithm and Boujou4

As Fig. 6(a) shows, the coordinate of the first frame is set as the world coordinate in our algorithm. The moving orientation of each frame pointed to X axis, so the X component of each frame was increased gradually. Because the vehicle moved evenly, the X component increased linearly. Fig. 6(b) shows the translations of boujou4. In the coordinate system of boujou4, it shows that the translations jittered greatly from the 15-th frame to the 30-th frame, which is due to short baseline between frames and we solved this problem through the process of key frame selection.

We took a simulated experiment with those parameters. The panoramas were mapped on the spheres. Then each sphere was rotated and translated in the world coordinate with its external parameters. 3D points were located in the same scene with their poses. We drew a set of lines between a 3D point and all spherical centers and found that the remapping points on spherical surfaces indicated the same scene point.

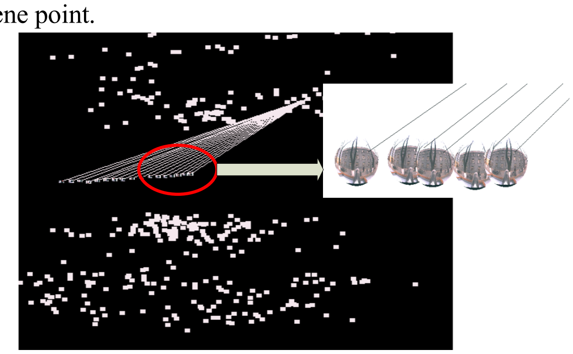


Figure 7. Simulative demonstrations of the panoramic spheres and the 3D points

VI. CONCLUSIONS

This paper presents an algorithm to estimate poses for spherical cameras. With a series of spherical panoramas, the motion parameters are computed automatically. There are still some works need to do in the future. Since the algorithm bases on feature matching and its can not handle well for the texture less scene. In the further, we will study ways to handle videos of texture less scene. What' more, some procedures can be executed synchronously in this algorithm, such as the computation of epipolar matrices for the pair-wise cameras. In fact, they can be accelerated with GPU, so GPU can be used when the algorithm needs to be executed real time. And further work will direct on applications based on the motions of the panoramic videos.

ACKNOWLEDGMENT

We would like to thank Shujun Zhang, Ke Chen and Shuang He for their insightful discussions. This work is supported by National Science & Technology Supporting Program (2008BAH37B08) and CEEUSRO project of Guangdong Province & Ministry of Education (2008A090400020) of China.

REFERENCES

[1] M. Pollefeys, L. V. Gool, M. Vergauwen, F. Verbiest, K. Cornelis, J. Tops, R. Koch, et al. "Visual Modeling with a Hand-Held Camera", Int. Journal of Computer Vision, 2004, vol.59, pp:207-232

[2] R.Hartley,Z.Andrew, "Multiple View Geometry in Computer Vision". Second edition, Cambridge University Press. UK, 2003,.

[3] C.J.Harris, M.Stephens. "A combined corner and edge detector". Proc. of the 4th Alvey Vision Conference, 1988, pp:147-151

[4] D.Bruce, Lucas AND T.Kanade, "An Iterative Image Registration Technique with an Application to Stereo Vision". Int. Joint Conference on Artificial Intelligence, 1981, pp:674-679

[5] D.G.Lowe,"Distinctive image features from scale invariant key points", Int. Journal of Computer Vision ,2004, vol.60, pp: 91-110

[6] H.Bay,T.Tuytelaars,van Gool L J. "SURF:Speeded Up Robust Features", European Conference on Computer Vision, 2006, pp:404-417

[7] M.A.Fischler, "Random sample consensus: A paradigm for modeling fitting with applications to image analysis and automated cartography". Comm. of the ACM, vol.24, 1981, pp:381-395

[8] Z. Zhang. "Determining the epipolar geometry and its uncertainty: A review". Int. Journal of Computer Vision, 1998, vol.27, pp: 161-195

[9] B.Triggs, "Bundle adjustment-a modern synthesis", Proc. of the Int. Workshop on Vision Algorithms, 1999, pp: 298-372

[10] Z.Zhang, "Incremental motion estimation through modified bundle adjustment". In ICIP, 2003, pp:343–346

[11] G.Zhang, X Qin., W.Hua, et al. "Robust Metric Reconstruction from Challenging Video Sequences", IEEE Conf. on Computer Vision and Pattern Recognition, 2007

[12] 2D3 Co. Ltd http://www.2d3.com/

[13] U.Matthew, A.Criminisi, S.Kang, S.Winder, R.Szeliski, , Hartley, R., "Image-based interactive exploration of real-world environments", IEEE Computer Graphics and Applications, 2004, vol.24, pp:52-63

[14] O.Motoko, S.Tomokazu, AND Y.Naokazu, "Estimating camera position and posture by using feature landmark database", Image Analysis: 14th Scandinavian Conference, 2005, pp:171-181

[15] Iwane Co. Ltd. http://www.iwane.com

[16] S.Ikeda, T.Sato, AND N.Yokoya, "A calibration method for an omnidirectional multi-camera system", Proc. SPIE Electronic Imaging 2003, pp: 499–507

[17] Z.Zhang, "A flexible new technique for camera calibration", IEEE Trans. on Pattern Analysis and Machine Intelligence, 1999, vol.22 , pp:1330-1334

[18] F.Kangni, R.Laganiere, "Orientation and Pose recovery from Spherical Panoramas", IEEE 11th Int. Conference on Computer Vision(ICCV2007), 2007.

[19] A. Fitzgibbon and A. Zisserman. "Automatic camera tracking". In, "Video Registration", M. Shah and R. Kumar, 2003, chapter 2, pp:18-35.

[20] N.David,"An Efficient Solution to the Five-Point Relative Pose Problem", IEEE Trans. on Pattern Analysis and Machine Intelligence, 2004, vol.26, pp:756-777

[21] J.Jorge, "The Levenberg-Marquardt algorithm implementation and theory", Numerical Analysis, Lecture Notes in Mathematics 630, 1978, pp:105-116