

Outward-Looking Circular Motion Analysis of Large Image Sequences

Guang Jiang, Yichen Wei,
Long Quan, *Sr. Member, IEEE*,
Hung-tat Tsui, *Member, IEEE*, and
Heung Yeung Shum,
Sr. Member, IEEE

Abstract—This paper presents a novel and simple method of analyzing the motion of a large image sequence captured by a calibrated outward-looking video camera moving on a circular trajectory for large-scale environment applications. Previous circular motion algorithms mainly focus on inward-looking turntable-like setups. They are not suitable for outward-looking motion where the conic trajectory of corresponding points degenerates to straight lines. The circular motion of a calibrated camera essentially has only one unknown rotation angle for each frame. The motion recovery for the entire sequence computes only one fundamental matrix of a pair of frames to extract the angular motion of the pair using Laguerre's formula and then propagates the computation of the unknown rotation angles to the other frames by tracking one point over at least three frames. Finally, a maximum-likelihood estimation is developed for the optimization of the whole sequence. Extensive experiments demonstrate the validity of the method and the feasibility of the application in image-based rendering.

Index Terms—Structure from motion, circular motion, single axis motion, concentric mosaic.

1 INTRODUCTION

CIRCULAR motion, or single axis motion, is a practical setup for image-based modeling and rendering. It arises from traditional 3D modeling using an inward-looking turntable [6] and rendering with outward-looking concentric mosaics [22]. The motion is a particular case of the more general planar motion [1] as all rotations are restricted to be around the same axis.

The outward-looking circular motion is of high interest due to its widely involved applications, such as image-based rendering [22] or stereo panorama [18]. A typical outward-looking setup uses a CCD camera rotating on a horizontal plane around a vertical axis located away from the camera optical center. The internal parameters of the camera are assumed to be fixed. A motor-controlled device is usually used to ensure the uniform angular motion. This may become inconvenient sometimes, for example, in large-scale outdoor environments where a power supply is not readily available. In such cases, using a hand-controlled device together with a structure from motion algorithm would make the applications more practical. However, this problem has received very little attention up to now. To the best of our knowledge, all previous circular motion algorithms only work well in small-scale inward looking cases with a moderate number of images. They all encounter difficulties in large-scale outward looking applications with a large number of images.

- G. Jiang is with the Department of Computer Science, Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong and the National Key Laboratory of ISN, Xidian University, Xi'an 710071, P.R. China. E-mail: gjiang@cs.ust.hk.
- Y. Wei and L. Quan are with the Department of Computer Science, Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong. E-mail: {yichenw, quan}@cs.ust.hk.
- H.-t. Tsui is with the Department of Electronic Engineering, Chinese University of Hong Kong, Shatin, N.T., Hong Kong. E-mail: htsui@ee.cuhk.edu.hk.
- H.Y. Shum is with Microsoft Research Asia, Number 49, ZhiChun Road, Beijing, P.R. China. E-mail: hshum@microsoft.com.

Manuscript received 19 Apr. 2004; revised 11 Aug. 2004; accepted 17 Aug. 2004; published online 14 Jan. 2005.

Recommended for acceptance by S. Sclaroff.

For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-0183-0404.

The above considerations motivated our development of a simple and efficient method capable of computing circular motion geometry of a large image sequence captured under outward-looking setup. The new method is particularly interesting in that the rotation angle computation is propagated efficiently between different tracked feature points using Laguerre's formula. The propagation is initialized using only one fundamental matrix of a pair of images between which feature points have been reliably tracked. Neither any more fundamental matrices nor trifocal tensors are computed. One important application of the new method is image-based rendering using concentric mosaics [22]. Concentric mosaic is a good trade-off between the ease of image acquisition and viewing space, among various plenoptic function-based approaches [15], [12], [7]. The geometry of a concentric mosaic capturing system is formulated as outward-looking circular motion with a calibrated camera and unknown rotation angles. We demonstrated the composition of concentric mosaics and rendering results in the experiments.

The paper is organized as follows: Section 2 reviews the circular motion geometry. Section 3 analyzes the insufficiency of previous methods and motivates our method. The method is elaborated in Sections 4, 5, and 6. Experimental results are reported in Section 7. Section 8 gives a conclusion of the paper. A preliminary short version of this paper has been reported at the ACCV 2004 [11].

2 REVIEW OF CIRCULAR MOTION GEOMETRY

The relative motion between the scene and the camera under circular motion can be described as a rotation around a fixed axis. The most common case is a static camera viewing an object on a rotating turntable [6]. Without loss of generality, we assume that the rotation axis is the vertical z -axis in world coordinate frame and the camera is moving on a horizontal plane.

As similar to planar motion [1], [4], there are some image fixed entities which are geometrically invariant, as illustrated in Fig. 1. They include two lines [6]. One is the image of the rotation axis, l_s , which is a line of fixed points in all images since the rotation axis is fixed with respect to different views. The second is the image of the vanishing line of the horizontal plane, l_h , called the horizon line. It is a fixed line but not a line of fixed points. Since the image of the absolute conic (IAC) is fixed under rigid motion, the intersection of the horizon line l_h and IAC, the two points i, j , remain fixed in all the images. They are actually the image of the two circular points, I, J , on the horizontal planes. Since the horizon line l_h can be determined by the images of circular points, i, j , there are in total six degrees of freedom for those image fixed entities, with two for each image of the two circular points and two for the line l_s .

3 MOTIVATION

The key of a circular motion algorithm is to first determine the image fixed entities, and then the motion parameters, the rotation angles. We analyze previous circular motion methods and show that they are infeasible for our problem, the outward-looking circular motion analysis of a large image sequence.

Fitzgibbon et al. [6] use multiple view tensors [8], [5]. Fundamental matrices are computed to obtain the horizon line, l_h , and the image of rotation axis, l_s . The rotation angles are then extracted from trifocal tensors for each image triplet, which is too computationally expensive for a large image sequence. Mendonca et al. [16], [17] show that the circular motion geometry can be obtained only from the object profiles without using point correspondences. Although their method is simple and efficient, it essentially only applies to single object modeling using inward-looking turntable and requires that object silhouettes can be obtained reliably.

The methods proposed in [10], [9] are closer to our approach. The main observation is that, under circular motion, the trajectory of one space point is a circle and this circle projects to a conic in the image. The image fixed entities can be obtained by fitting conics from point correspondences. Their method works well for an inward-looking

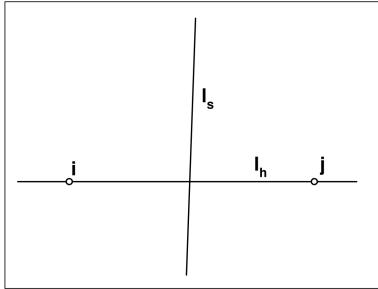


Fig. 1. The image fixed entities under circular motion: I_s , the image of fixed rotation axis; I_h , the image of the intersection line of all parallel horizontal planes; i, j , the image of the circular points on these planes.

turntable as the conic trajectory is evident and the point correspondences are well scattered. However, for the outward-looking case, the observed conic trajectory becomes barely curved, almost degenerating to a straight line due to the limited viewing capability of the camera under such a setup. The conic fitting problem becomes ill-conditioned and unstable. Fig. 2 illustrates the different conic trajectories for both cases. Our method avoids the conic fitting problem but rather only computes the conic centers. We essentially simplify the ill-conditioned problem by calibrating the camera in advance, which is analyzed later.

4 ROTATION ANGLE RECOVERY

As illustrated in Fig. 3, let us consider the equivalent case where the camera is fixed and the scene is rotating around the rotation axis. The trajectory of the space point A is a circle with center O_A . Image points a_1 and a_2 are the projection of the same space point A in different positions A_1 and A_2 . They are on a conic with center o_a . In order to compute the rotation angle θ , the key of our approach is to apply Laguerre's formula [20] in image plane instead of in space plane. If the image of the circular points on the space plane, i, j , are known, we have

$$\theta = \frac{1}{2i} \log(\{\mathbf{o}_a \times \mathbf{a}_1, \mathbf{o}_a \times \mathbf{a}_2; \mathbf{o}_a \times \mathbf{i}, \mathbf{o}_a \times \mathbf{j}\}), \quad (1)$$

and the rotation angle recovery reduces to the computation of the conic center o_a . Note that we do not fit the whole conic. On one hand, this is numerically unstable. On the other hand, it is also unnecessary since only the conic center is required, which allows a simpler geometry approach, as explained later. In essence, we reduce the difficult conic fitting problem to an easier task by calibrating the camera in advance. Although the calibration problem itself is also quite difficult, it is relatively easier in our targeted applications where the constant internal parameter assumption usually holds.

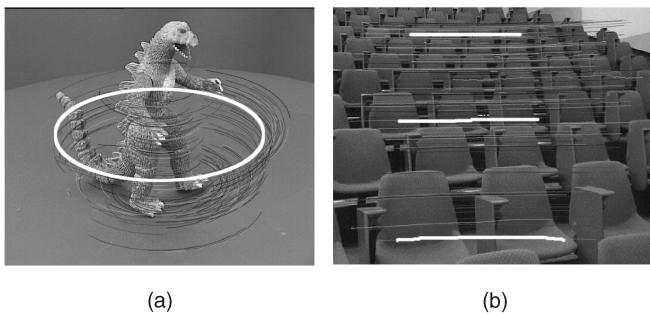


Fig. 2. (a) The famous dinosaur sequence taken by an inward-looking turntable. The conic loci of tracked points are apparent. A fitted conic is shown in highlight. (b) The classroom sequence taken using an outward-looking camera. The point loci almost degenerate to straight lines, as shown in highlight. The conic fitting is infeasible.

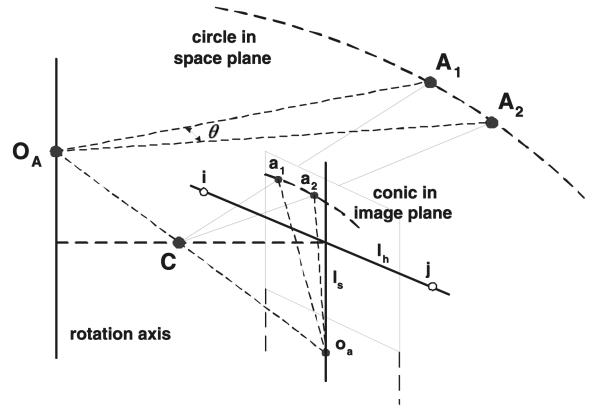


Fig. 3. Computation of the rotation angle in the image plane instead of in the space plane. Assume that camera C is fixed and space point A is rotating. The rotating angle is computed from Laguerre's formula as $\theta = \frac{1}{2i} \log(\{l_{O_A A_1}, l_{O_A A_2}; l_{O_A i}, l_{O_A j}\}) = \frac{1}{2i} \log(\{l_{o_a a_1}, l_{o_a a_2}; l_{o_a i}, l_{o_a j}\})$.

The images of the circular points i and j can be obtained from an offline calibration method [26], [23] or a self-calibration method using 2D trifocal tensor [1] or 1D trifocal tensor [4]. The image of the absolute conic ω can be obtained from the internal parameters K of the calibrated camera, $\omega = (KK^T)^{-1}$. Therefore, the image of circular points i and j are the intersection of ω and the horizon line I_h . The method of computing I_h is described later.

5 PROPAGATION OF ANGULAR MOTION

5.1 Initialization from a Fundamental Matrix

A pair of images with a large baseline is first chosen to compute the fundamental matrix F reliably [14], [25] using point correspondences. We call this pair *the reference pair* of the sequence. The fixed line I_s and I_h are decomposed from the rank-2 symmetric part of the fundamental matrix, $F_s = F + F^T$ [6]. The rotation angle between the reference pair is obtained using Laguerre's formula as:

$$\theta = \frac{1}{2i} \log(\{e_1, e_2; i, j\}), \quad (2)$$

where e_1 and e_2 are the two epipoles extracted from the fundamental matrix F .

The proof of (2) is as follows: As the camera moves on the horizontal plane, e_1 and e_2 must be on the horizon line, I_h . Let us select a point o on the line I_s and connect it to e_1 and e_2 . Since o is a fixed point in the two images, the two lines l_{oe_1} and l_{oe_2} are the corresponding epipolar lines of o in the two images, respectively. Select a point b_1 from the line l_{oe_1} , its corresponding point b_2 must be on the line l_{oe_2} due to the constraints that b_2 must be on the line passing through the point e_2 . Based on the similar deduction derived earlier, the rotation angle between the two images is computed from the Laguerre's formula:

$$\theta = \frac{1}{2i} \log(\{l_{ob_1}, l_{ob_2}; l_{oi}, l_{oj}\}) = \frac{1}{2i} \log(\{l_{oe_1}, l_{oe_2}; l_{oi}, l_{oj}\}),$$

which is equivalent to (2).

5.2 Computation of the Space-Image Homography

For a space point A moving on a space plane (refer to Fig. 3), the transformation between the space points A_i and the image points a_i , $i = 1, 2, \dots$, is a planar homography H_a [8], such that $a_i \simeq H_a A_i$, where \simeq denotes equivalence up to scale.

When a rotation angle θ between the two images containing the corresponding image points a_1 and a_2 is known, the space points can simply be assumed as $A_1 = (r, 0, 1)^T$ and $A_2 = (r \cos(\theta), r \sin(\theta), 1)^T$, where r is the radius of the space circle. Now, we have four point correspondences,

$$\mathbf{A}_1 \leftrightarrow \mathbf{a}_1, \mathbf{A}_2 \leftrightarrow \mathbf{a}_2, \mathbf{I} = (1, i, 0)^T \leftrightarrow \mathbf{i}, \mathbf{J} = (1, -i, 0)^T \leftrightarrow \mathbf{j},$$

between the space plane and the image plane. Here, \mathbf{I}, \mathbf{J} are the circular points on the space plane. Let $(\mathbf{h}_1 \mathbf{h}_2 \mathbf{h}_3)$ be the homography computed uniquely for $r = 1$ from the linear system

$$\mathbf{h}_1 + \mathbf{h}_3 \simeq \mathbf{a}_1, \cos(\theta)\mathbf{h}_1 + \sin(\theta)\mathbf{h}_2 + \mathbf{h}_3 \simeq \mathbf{a}_2, \mathbf{h}_1 + i\mathbf{h}_2 \simeq \mathbf{i}, \mathbf{h}_1 - i\mathbf{h}_2 \simeq \mathbf{j}.$$

It could easily be derived that $\mathbf{H}_a = (\frac{1}{r}\mathbf{h}_1 \frac{1}{r}\mathbf{h}_2 \mathbf{h}_3)$ for arbitrary r , that is, the third column in \mathbf{H}_a is determined uniquely and the first two columns are determined up to the unknown r . Therefore, the conic center, \mathbf{o}_a , is obtained uniquely as

$$\mathbf{o}_a \simeq \mathbf{H}_a(0, 0, 1)^T = \mathbf{h}_3. \quad (3)$$

Once \mathbf{o}_a is known, it becomes straightforward to use Laguerre's formula (1) to compute the rotation angle for any third view which has a visible corresponding point, say \mathbf{a}_3 . This is the key point of angular motion propagation, as explained in the next section. Note that the problem of only computing the conic center is much easier than fitting the whole conic. It is also better geometrically constrained, that is, the conic center must lie on the image of the rotation axis, \mathbf{l}_s . We actually use the projection of computed conic center from (3) onto the line \mathbf{l}_s as a first estimation. This constraint is also exploited in the final optimization.

5.3 Propagation of Angular Motion

Since the image of space point \mathbf{A} can possibly be tracked in other images other than the reference pair, the rotation angle of any image containing the tracked point of \mathbf{a} can be computed using Laguerre's formula (1) once the conic center \mathbf{o}_a is obtained from (3). For example, \mathbf{a}_3 is a corresponding point in a third view, the angular motion between views 2 and 3 is

$$\theta_{23} = \frac{1}{2i} \log(\{\mathbf{o}_a \times \mathbf{a}_2, \mathbf{o}_a \times \mathbf{a}_3; \mathbf{o}_a \times \mathbf{i}, \mathbf{o}_a \times \mathbf{j}\}).$$

This procedure can be repeated for all tracked corresponding points in the reference pair. In other words, the computation of the rotation angles can simply be propagated to any view in which at least one point is in correspondence with one point in the reference pair.

There will still be many frames which do not have any "visible" tracked point from the reference pair. However, notice that the above angle propagation procedure can be extended to any view which has one "visible" tracked point from any pair of views whose rotation angle is already computed. For example, points \mathbf{b}_2 and \mathbf{b}_3 are "visible" corresponding points related the known rotation angle θ_{23} , the image of the circle center \mathbf{o}_b can be obtained by calculating a homography \mathbf{H}_b according to (3). Therefore, all angular motions involving the views related to point \mathbf{b} can be obtained similarly, that is, for any view in which point \mathbf{b} is tracked.

The above propagation procedure can be performed along the whole sequence unless there are few features in the scene. Note that the method does not need to compute any other quantities such as the fundamental matrix. Only tracked points are sufficient. In the extreme case, feature points tracked over only three frames suffice to compute all the angles.

5.4 Implementation Considerations

The above method uses minimal data and is very efficient. In real data, noise and outliers are inevitable and we exploit the high redundancy in the data to enhance the robustness of the algorithm at different stages, by taking the following considerations into account, typically for a large image sequence of thousands of frames.

- The very small baseline between consecutive frames makes the feature point tracking practical over a long range of frames, almost up to the viewing capability of the camera used. To improve the computation accuracy, the fundamental matrix in the initialization and homography in the propagation are computed using robust techniques [8] from all point correspondences tracked over a relatively long

distance. Furthermore, since we have the freedom of choosing the reference pair in the initialization, multiple candidate pairs are randomly selected from the whole sequence at first and the image fixed entities and initial angle are computed for each candidate pair. Then, we choose the reference pair that gives rise to the average minimal distance to the median values of all the computed quantities.

- In the angle computation and propagation, one tracked feature point suffices for computing many angles. In practice, there are always multiple tracked feature points whose frame ranges overlap and they will give inconsistent angle values. Whenever multiple values are possible, we take the median as the initial estimated value.

6 OPTIMIZATION

The above method gives reasonable initial estimation for geometry and motion parameters and a bundle adjustment is applied at last to refine all the quantities. We cannot use the general bundle adjustment that minimizes reprojection errors [6] for efficiency consideration since there would be too many parameters involved. We also cannot minimize the sum of squared distances of all points to their corresponding conics [10] since we do not fit conics. Instead, a simple cost function is derived and efficient optimization technique is developed to exploit the sparseness of the cost function.

Computation of metric rectifying homography. The cost function should be defined in metric space. A natural choice is the family of space planes where the space point moves. As discussed previously, there is an image-space transformation, a planar homography \mathbf{H} , which performs metric rectification for the points in the image plane to the points in the space plane. Such a homography can be decomposed into a concatenation of two matrices \mathbf{A} and \mathbf{P} , representing affine and "pure projective" transformations, respectively, [3], [13], $\mathbf{H} = \mathbf{AP}$, where

$$\mathbf{P} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ l_1 & l_2 & l_3 \end{pmatrix}$$

can be determined by the horizon line $\mathbf{l}_h = (l_1, l_2, l_3)^T$ and

$$\mathbf{A} = \begin{pmatrix} -\frac{1}{\beta} & \frac{\alpha}{\beta} & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

can be determined from the points, $(\alpha \pm i\beta, 1, 0)^T$, which are obtained by applying \mathbf{P} to the image of circular points \mathbf{i} and \mathbf{j} . Let \mathbf{i} and \mathbf{j} be $(a \pm ib, c \pm id, 1)^T$, we have $\mathbf{l}_h = \mathbf{i} \times \mathbf{j} = (d, -b, bc - ad)^T$, $\alpha = \frac{ac + bd}{c^2 + d^2}$, $\beta = \frac{bc - ad}{c^2 + d^2}$ and, finally,

$$\mathbf{H} = \begin{pmatrix} c^2 + d^2 & -ac - bd & 0 \\ 0 & ad - bc & 0 \\ d(ad - bc) & -b(ad - bc) & -(ad - bc)^2 \end{pmatrix}. \quad (4)$$

Definition of the cost function. Ideally, two corresponding image points $\mathbf{a}_1, \mathbf{a}_2$, the corresponding conic center \mathbf{o}_a , and rotation angle θ should satisfy

$$\mathbf{R}(\theta)(\mathbf{H}\mathbf{a}_1 - \mathbf{H}\mathbf{o}_a) = (\mathbf{H}\mathbf{a}_2 - \mathbf{H}\mathbf{o}_a), \leftrightarrow \mathbf{R}(\theta)\overrightarrow{\mathbf{O}_A \mathbf{A}_1} = \overrightarrow{\mathbf{O}_A \mathbf{A}_2}, \quad (5)$$

where $\mathbf{R}(\theta)$ is a rotation matrix. Note the equivalence here should be satisfied strictly, not up to scale.

For real data, (5) does not hold due to noise and we minimize the residual error over all tracked feature points. Let n be the number of feature points. For i th feature point, it is tracked over m_i continuous frames and denoted as $\mathbf{x}_1^i, \mathbf{x}_2^i, \dots, \mathbf{x}_{m_i}^i$. Let $frame(i, j)$ return the number of frames in which the i th feature point is tracked until the j th frame, where $1 \leq I \leq n, 1 \leq j \leq m_i$. Let \mathbf{o}^i be the conic center corresponding to the i th feature point and θ_m be the rotation angle between the m th frame and the $(m+1)$ th frame. The sum of residual error is

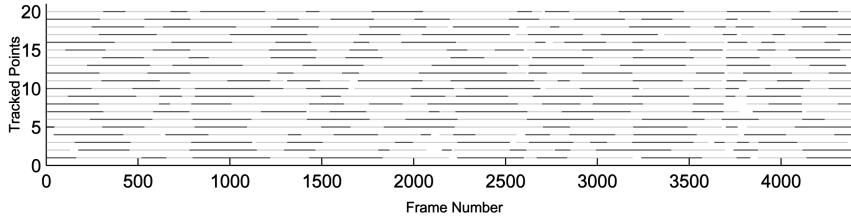


Fig. 4. Tracking of 20 feature points simultaneously in each frame over the whole Classroom sequence. Each line segment represents the tracking of one point. Points tracked in less than 100 frames are discarded for computational stability and not shown in the figure.

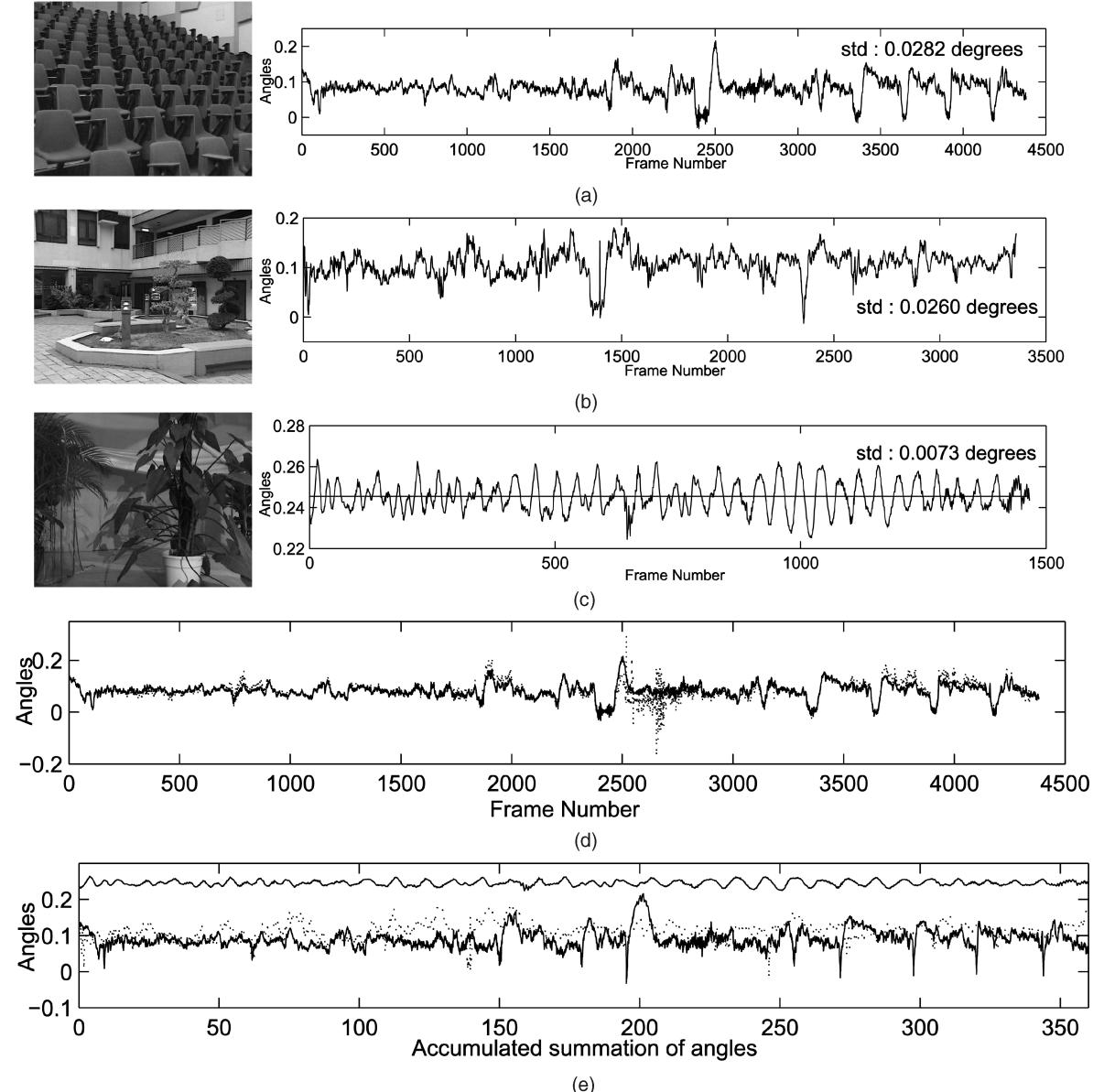


Fig. 5. (a) Classroom. (b) Garden. (c) Flower. (d) Classroom (before and after optimization). Recovered rotation angles. (a), (b), and (c) show the angles and standard deviation after optimization of the three image sequences, respectively. Horizontal straight line in (c) is the ground truth, which is quite close to our result. (d) Shows the angles before (dotted) and after (solid) optimization of the Classroom sequence. Noise and variations are reduced significantly by optimization step. (e) Plots the three optimized results together to show that the standard deviation in third experiment (with uniform ground truth) is much smaller than that of the first two. It also demonstrates that the closure constraint is almost perfectly satisfied after optimization.

$$\sum_{i=1}^n \sum_{j=1}^{m_i-1} \left\| \mathbf{R}(\theta_{frame(i,j)}) \frac{(\mathbf{Hx}_j^i - \mathbf{Ho}^i)}{\|\mathbf{Hx}_j^i - \mathbf{Ho}^i\|} - \frac{(\mathbf{Hx}_{j+1}^i - \mathbf{Ho}^i)}{\|\mathbf{Hx}_{j+1}^i - \mathbf{Ho}^i\|} \right\|. \quad (6)$$

The normalization in (6) is necessary. Since different feature points are transformed by \mathbf{H} onto space planes with different and unknown scales, the space vectors are normalized to have the same

scale to avoid any bias induced by space points which happens to have very large scale.

If an image sequence possesses a rotation larger than 360 degrees, we can search for a frame which is most similar as the first frame and does not reach the 360 degree rotation, and then insert the first frame after it as the ending frame to get a sequence perfectly covering

TABLE 1
Running Performance (on a Pentium IV 1.9G PC)

experiment	#angles	#feature points	error by propagation	error after optimization	#iter	$\sum_i \theta_i$	time(sec)
classroom	4381	1362(100)	0.0456	0.0254	3	359.79	2653
classroom	4381	726(50)	0.0341	0.0141	4	359.89	710
garden	3360	1235(100)	0.0852	0.0303	5	359.78	1647
garden	3360	626(50)	0.0416	0.0139	4	359.90	351
flower	1466	1112(100)	0.0033	0.0018	2	359.98	373
flower	1466	566(50)	0.0016	0.00078	2	359.99	34

The third column shows the total number of all feature points and the number of feature points tracked simultaneously.

360 degrees view of field. This closure constraint could be explicitly incorporated by introducing the Lagrange term $\lambda |\sum_m \theta_m - 2\pi|$. Therefore, the final optimization problem is given as

$$\min \left(\sum_{i=1}^n \sum_{j=1}^{m_i-1} \left\| \mathbf{R}(\theta_{frame(i,j)}) \frac{(\mathbf{Hx}_j^i - \mathbf{Ho}^i)}{\|\mathbf{Hx}_j^i - \mathbf{Ho}^i\|} - \frac{(\mathbf{Hx}_{j+1}^i - \mathbf{Ho}^i)}{\|\mathbf{Hx}_{j+1}^i - \mathbf{Ho}^i\|} \right\| + \lambda |\sum_m \theta_m - 2\pi| \right), \text{ with } \mathbf{l}_s \mathbf{o} = 0 \quad (7)$$

with respect to a total of $6 + m + n$ parameters, 6 for the image fixed entities, m for all rotation angles and n for all conic centers.

Optimization. The optimization can be performed using standard nonlinear algorithms such as Lagrange multiplier method or Levenberg-Marquart (LM) method, based on the initial estimation. In practice, since there are typically thousands of images ($m = 1,000 - 4,000$) in our targeted applications, and hundreds of feature points ($n = 500 - 700$) appearing in the whole sequence, the large number of unknowns makes the standard methods intractable due to their height time complexity $O((m+n)^3)$.

Notice that each term in the first part of (7) only depends on eight unknowns, $\theta_{frame(i,j)}$, \mathbf{o}^i , i, j , and \mathbf{l}_s . This gives rise to a regular sparse block structure in the Jacobian matrix of the cost function. As similar as in [8], an efficient LM algorithm is developed to exploit the sparseness and reduces the time complexity to $O(m^2 + n^3)$, making the optimization problem tractable even for a very large image

sequence. Note that although the last term imposing the closure constraint in (7) does depend on all unknown rotation angles and injures the “sparseness” of the problem to some extent, it turns out that this only results in a LM algorithm that is a little bit more complicated than the one in [8], but does not increase the time complexity.

Although we do not use the closure constraint as a hard constraint, it is almost perfectly satisfied by using a relatively large constant λ ($\lambda = 1$ in all the experiments).

7 EXPERIMENTS AND APPLICATIONS

In this section, we first report the experimental results to demonstrate the validity of the method and then show some preliminary results for the application of concentric mosaics.

Classroom. We demonstrate the complete process of our method in this experiment. A digital video camera is mounted on a horizontal bar that is fixed on a tripod and can be rotated by hand. We extracted 4,382 frames with the resolution 720×576 from a video taken in an indoor classroom. The camera’s internal parameters, including the radial distortion, are calibrated using the method in [26]. Radial distortion is corrected for each frame.

The feature point tracking is efficiently accomplished by the standard methods reported in [24], [21]. Simultaneously tracking 50 feature points over 4,382 frames takes about 20 minutes on a P4 1.9G PC. The right image in Fig. 2 shows the loci of a group of feature



Fig. 6. Composed concentric mosaics using the recovered rotation angles.

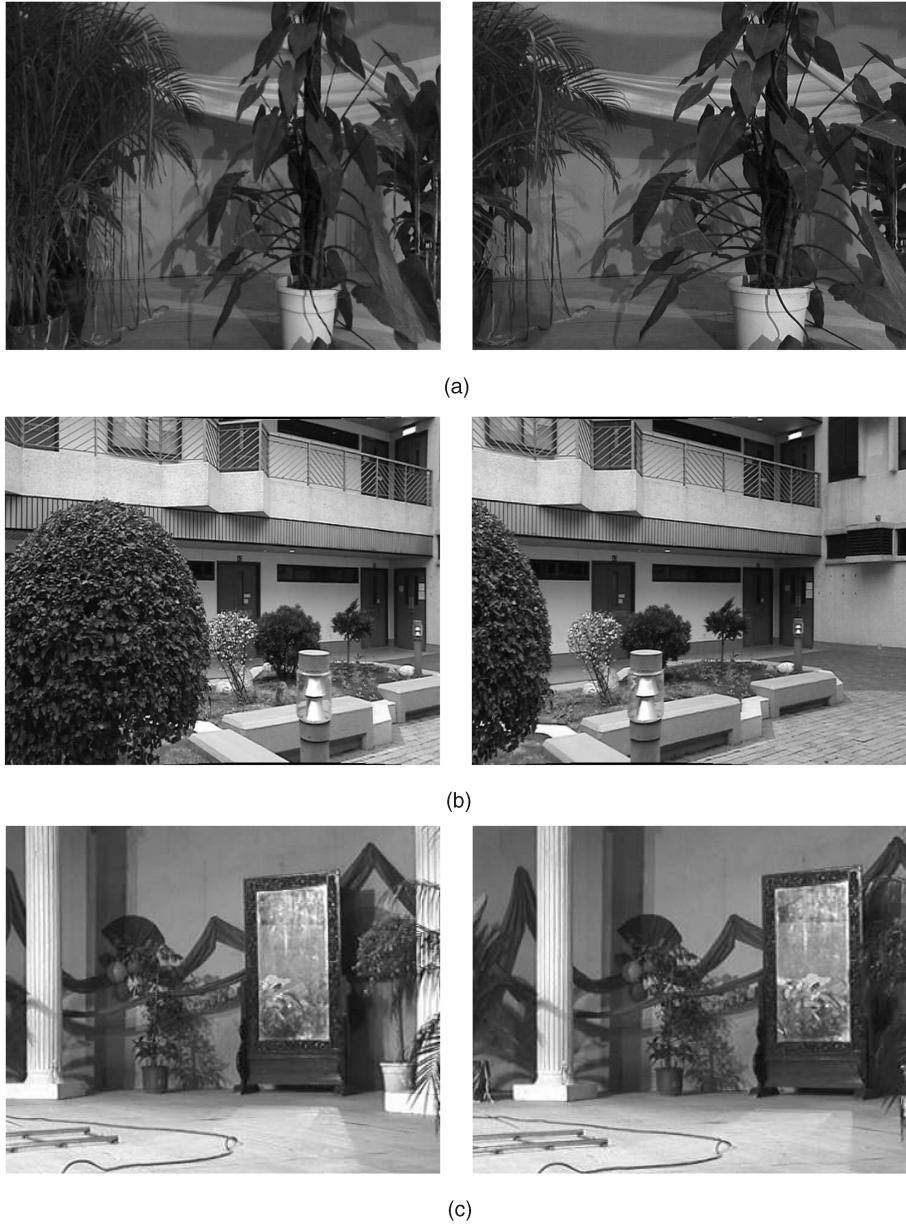


Fig. 7. Rendered images from concentric mosaics under novel viewpoints and different camera motions. (a) A forward camera motion. (b) A lateral camera motion. (c) A lateral and rotating camera motion. Notice that zooming, occlusion, and reflection effects are well rendered and no obvious artifact is observed.

points tracked over 600 frames. The two frames corresponding to the endpoints of these trajectory curves are selected as the reference pair in the initialization.

Fig. 4 illustrates the tracks of 20 feature points simultaneously along the whole Classroom sequence. Whenever a feature point is lost in tracking, a new one is added in the current frame to keep the number of tracked points constant. Although 20 points are shown here, we used a larger number (50 or 100) in the implementation to make the computation more robust. If a point cannot be tracked in relatively long frames, it is discarded.

The recovered rotation angles are shown in Fig. 5. Fig. 5a shows the optimized results for Classroom sequence. Fig. 5d plots the results before and after optimization for comparison. We notice that the results, even before optimization, are already satisfactory (summation of all angles is 352 degrees). The optimization step reduces the variation, keeps the same curve shape, and makes the result satisfying the closure constraint. Some negative values are observed in the middle of the sequence. This is due to hand trembling during the video taking process.

Garden. In the second experiment, we took a video in an outdoor garden with the hand-rotating camera and extracted 3,361 frames. The recovered rotation angles are shown in Fig. 5b. We are more careful to avoid hand trembling while taking the video, and consequently the recovered rotation angles have smaller variation than that of the Classroom sequence. No negative angle is observed.

Flower. In the third experiment, we took an indoor video using a motor-controlled device to obtain the uniform angles as ground truth. This is used to validate our approach. In all, 1,467 frames are extracted which is much less than those of the first two experiments and this makes the point tracking more difficult. The recovered rotation angles as well as the true values (the horizontal straight line) are shown in Fig. 5c for comparison. Our result is very close to the ground truth and the standard deviation is much smaller than that of the first two experiments, as observed in Fig. 5e. This verifies the effectiveness of our approach.

Running performance. The whole algorithm is quite efficient, taking into account the large number of images involved. Most time is consumed in the precalibration, removal of radial distortion, feature tracking, and the final optimization. The propagation step is very fast. It took only several hours to process the video sequence and obtain the final result. The low working load makes our approach suitable for practical applications.

Table 1 summarizes the performance in the optimization step. For each image sequence, we obtained the results by using either 100 or 50 feature points tracked simultaneously. We found the final results almost the same, while using 100 points is significantly more time-consuming. The results shown in Fig. 5 are obtained using 50 points.

Concentric mosaics composition and rendering. After the rotation angles are recovered, panoramic images can be constructed from the original images [22], [19], [18]. Fig. 6 shows the composed concentric mosaics. They can be used for efficient rendering [22], [2]. Fig. 7 illustrates several images rendered from novel viewpoints under different camera motions. It can be seen that the zooming, occlusion, and reflection effects are well rendered and no obvious artifact is observed.

8 CONCLUSION

This paper presented a new method capable of efficiently analyzing a large sequence of images captured by a hand-controlled outward-looking camera under circular motion. The method is remarkably simple as it needs only one fundamental matrix to initialize an efficient procedure that automatically propagates angle computation to the whole sequence. The simplicity is achieved by pre-calibrating the camera. This method provides a practical way of capturing large-scale outdoor environments with a hand-controlled camera and makes relevant applications more practical. Experiments on real video sequences demonstrate the validity and efficiency of the algorithm. Applications on concentric mosaics are also shown.

ACKNOWLEDGMENTS

This work is supported by the Hong Kong RGC Grant HKUST6188/02E, HKUST6182/04E and CUHK4378/02E.

REFERENCES

- [1] M. Armstrong, A. Zisserman, and R. Hartley, "Self-Calibration from Image Triplets," *Proc. Fourth European Conf. Computer Vision*, vol. 1064, pp. 3-16, Apr. 1996.
- [2] J.X. Chai, S.B. Kang, and H.Y. Shum, "Rendering with Non-Uniform Concentric Mosaics," *Proc. SMILE Workshop Structure from Multiple Images in Large Scale Environments*, 2000.
- [3] O. Faugeras, "Stratification of Three-Dimensional Vision: Projective, Affine and Metric Representations," *J. Optical Soc. Am.*, vol. 12, pp. 465-484, 1995.
- [4] O. Faugeras, L. Quan, and P. Sturm, "Self-Calibration of a 1D Projective Camera and Its Application to the Self-Calibration of a 2D Projective Camera," *Proc. Fifth European Conf. Computer Vision*, pp. 36-52, June 1998.
- [5] O.D. Faugeras and Q.T. Luong, *Geometry of Multiple Images*. MIT Press, 2001.
- [6] A.W. Fitzgibbon, G. Cross, and A. Zisserman, "Automatic 3D Model Construction for Turn-Table Sequences," *Proc. SMILE Workshop Structure from Multiple Images in Large Scale Environments*, pp. 154-169, 1998.
- [7] S.J. Gortler, R. Grzeszczuk, R. Szeliski, and M. Cohen, "The Lumigraph," *Proc. SIGGRAPH*, pp. 43-54, 1996.
- [8] R.I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge Univ. Press, June 2000.
- [9] G. Jiang, L. Quan, and H. Tsui, "Circular Motion Geometry by Minimal 2 Points in 4 Images," *Proc. Ninth Int'l Conf. Computer Vision*, pp. 221-227, 2003.
- [10] G. Jiang, H. Tsui, L. Quan, and A. Zisserman, "Single Axis Geometry by Fitting Conics," *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2003.
- [11] G. Jiang, Y. Wei, H. Tsui, and L. Quan, "Construction and Rendering of Concentric Mosaics from a Handheld Camera," *Proc. Asian Conf. Computer Vision*, 2004.
- [12] M. Levoy and P. Hanrahan, "Light Field Rendering," *Proc. SIGGRAPH*, pp. 31-42, 1996.
- [13] D. Liebowitz and A. Zisserman, "Metric Rectification for Perspective Images of Planes," *Proc. Conf. Computer Vision and Pattern Recognition*, 1998.
- [14] Q.T. Luong and O. Faugeras, "The Fundamental Matrix: Theory, Algorithms and Stability Analysis," *Int'l J. Computer Vision*, vol. 17, no. 1, pp. 43-76, 1996.
- [15] L. McMillan and G. Bishop, "Plenoptic Modeling: An Image-Based Rendering System," *Proc. SIGGRAPH*, pp. 39-46, 1995.
- [16] P.R.S. Mendonca, K.Y.K. Wong, and R. Cipolla, "Camera Pose Estimation and Reconstruction from Image Profiles under Circular Motion," *Proc. Sixth European Conf. Computer Vision*, vol. II, pp. 864-877, 2000.
- [17] P.R.S. Mendonca, K.Y.K. Wong, and R. Cipolla, "Epipolar Geometry from Profiles under Circular Motion," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 23, no. 6, pp. 604-616, June 2001.
- [18] S. Peleg and M.B. Ezra, "Stereo Panorama with a Single Camera," *Proc. Conf. Computer Vision and Pattern Recognition*, 1999.
- [19] S. Peleg and J. Herman, "Panoramic Mosaics by Manifold Projection," *Proc. Conf. Computer Vision and Pattern Recognition*, pp. 338-343, 1997.
- [20] J.G. Semple and G.T. Kneebone, *Algebraic Projective Geometry*. 1952.
- [21] J. Shi and C. Tomasi, "Good Features to Track," *Proc. Conf. Computer Vision and Pattern Recognition*, pp. 593-600, 1994.
- [22] H.Y. Shum and L.W. He, "Rendering with Concentric Mosaics," *Proc. SIGGRAPH*, pp. 299-306, 1999.
- [23] P. Sturm and S.J. Maybank, "On Plane-Based Camera Calibration," *Proc. Conf. on Computer Vision and Pattern Recognition*, 1999.
- [24] C. Tomasi and T. Kanade, "Detection and Tracking of Point Features," Technical report CMU-CS-91-132, Carnegie Mellon Univ., 1991.
- [25] Z. Zhang, "Determining the Epipolar Geometry and Its Uncertainty: A Review," *Int'l J. Computer Vision*, vol. 27, no. 2, pp. 161-195, Mar. 1998.
- [26] Z. Zhang, "Flexible Camera Calibration by Viewing a Plane from Unknown Orientations," *Proc. Seventh Int'l Conf. Computer Vision*, Sept. 1999.

▷ For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/publications/dlib.