2008 IEEE International Conference on
Robotics and Automation
Pasadena, CA, USA, May 19-23, 2008

Motion Estimation using Multiple Non-Overlapping Cameras for Small Unmanned Aerial Vehicles

Jun-Sik Kim, Myung Hwangbo and Takeo Kanade

Abstract— **An imaging sensor made of multiple light-weight non-overlapping cameras is an effective sensor for a small unmanned aerial vehicle that has strong payload limitation. This paper presents a method for motion estimation by assuming that such a multi-camera system is a spherical imaging system (that is, the cameras share a single optical center). We derive analytically and empirically a condition for a multi-camera system to be modeled as a spherical camera. Interestingly, not only does the spherical assumption simplify the algorithms and calibration procedure, but also motion estimation based on that assumption becomes more accurate.**

I. INTRODUCTION

Vision-based motion estimation without the benefit of a global positioning system (GPS) data for small unmanned aerial vehicles (UAVs) is an important and challenging problem. For a payload-limited, small sized UAV, it is not practical to carry precise, and usually heavy sensors. Motion estimation using cameras has been widely investigated, known as *structure-from-motion* (SFM) in computer vision (or in robotics as *visual simultaneous localization and mapping* (V-SLAM)). For SFM, one can use either a single camera or stereo cameras. While stereo cameras can reconstruct absolute distances of motions and the scene structure, stereo on a small UAV cannot afford a sufficiently large baseline for accurate reconstruction. Thus, a single camera based SFM is generally used for an aerial vehicle, although its motions are determined only up to scale.

The fundamental difficulty of real-time SFM is the *translation-rotation ambiguity*. The apparent motion or the *optical flow* between two frames is hard to distinguish between, for example, small sideway translational motion and small panning rotational motion. Similarly, it is very difficult to distinguish these two motion components from images. Baker et al. [1] showed that this difficulty of decoupling the two is mainly caused by an insufficient field of views (FOV) of the imaging system. By expanding the FOV, the ambiguity can be reduced. Between two ways to widen FOV, using an optical system having a wider FOV or using multiple cameras, we choose to use multiple cameras with non-overlapping views to expand FOVs with the minimal payload because a wider-FOV optical system including a high-resolution camera is generally much heavier than several low-weight cameras.

The problem of estimating motions of multiple non-overlapping cameras is most conveniently cast in the form of a generalized camera model [2], and several algorithms have been proposed. Chen and Chang, and Frahm et al. proposed pose estimation methods for a generalized camera [3], [4], [5]. To use these methods, some known 3D points are required in advance. In contrast, one can use initial motions obtained with an accurate odometry rather than known 3D points [6]. For UAVs, all of these methods are not applicable because there are neither known 3D points nor an accurate odometry. Ego-motion estimation of multiple cameras without prior knowledge of scene points or motion has been also investigated. Chen et al. [7] presented an algorithm based on non-linear optimization. Another algorithm by Baker et al. [1] decomposed rotation and translation estimations with their "Argus eye" system. Pless [8] showed that there is a linear constraint of correspondences in a generalized camera, just like in a projective camera, and motion parameters can be extracted from the constraint. These algorithms explicitly use the distance between camera centers, so the case of our UAV's multi-camera system, for which we will use a spherical camera model, will become a degenerate one.

We assume that a set of multiple non-overlapping cameras on a small UAV can be modeled as a spherical camera because their camera centers are sufficiently close compared to the distance to the scene. One of the contributions of this paper is to find a condition for this spherical assumption both analytically and empirically. Based on the spherical assumption, we also present a method to estimate a rotation between non-overlapping cameras without any artificial markers. Comparing to conventional motion estimation using a single camera, we show that the spherical assumption for multiple cameras is effective for motion estimation for small UAVs. Interestingly, not only does the spherical assumption simplify the algorithms and calibration procedure, but the motion estimation based on that assumption becomes more accurate.

II. MULTI-CAMERA SYSTEM AS A SPHERICAL CAMERA

Our UAV camera system, shown in Fig. 1, consists of three low-weight cameras placed in three orthogonal directions with non-overlapping field of views. The overall weight is just 300 g including batteries. The distance between camera centers is about 100 mm.

This work has been supported by DARPA under Grant W31P4Q-06-C-0488 and Air Force Research Laboratory under Grant MNK-BAA-06-2002.The authors would like to thank Dr. Arseneau for his proofreading and valuable discussions.

J.-S. Kim, M. Hwangbo and T. Kanade are with the Robotics Institute, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213, USA. {kimjs,myung,tk}@cs.cmu.edu

978-1-4244-1647-9/08/$25.00 ©2008 IEEE.

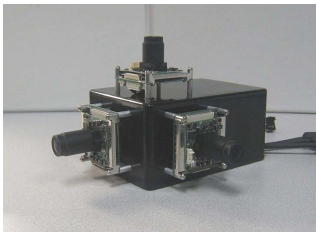

Fig. 1. Multiple non-overlapping view camera system

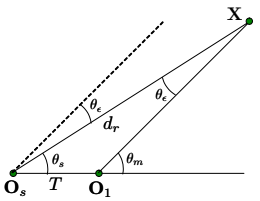

Fig. 2. Error caused by the spherical assumption

We will treat this set of cameras as a single spherical camera, or equivalently we will assume that the three cameras share a common projection center. Applying a spherical assumption to non-spherical cameras induces positional errors when mapping the feature positions located in the individual camera image to the spherical image. If the induced errors are less than image noise, the spherical assumption can be applied safely.

Fig. 2 depicts how the errors are induced by the spherical assumption. There are a physical camera center $\mathbf{O}_1$ and a spherical camera center $\mathbf{O}_s$ with a distance T. Suppose that this camera system sees a point $\mathbf{X}$ whose distance from the camera $\mathbf{O}_s$ is d_r. The angle measured with the camera $\mathbf{O}_1$ is θ_m which would be θ_s if $\mathbf{O}_1$ coincided with $\mathbf{O}_s$. We can say that the error by the assumption is $\theta_\varepsilon \triangleq \theta_m - \theta_s$.

Using a triangle $\mathbf{X}\mathbf{O}_1\mathbf{O}_s$,

$$\sin\theta_\varepsilon = -\frac{T}{d_r}\sin\theta_m$$

applying the law of sine. Because the image noise $|\theta'_\varepsilon|$ should be very small and larger than $|\theta_\varepsilon|$, we can approximate the condition as

$$|\theta'_\varepsilon| > |\theta_\varepsilon| = \frac{T}{d_r}\sin|\theta_m|$$

which gives

$$T < \frac{|\theta'_\varepsilon|}{\sin|\theta_m|}d_r.$$

To have a larger range of T given the image noise $|\theta'_\varepsilon|$, $\max(\sin|\theta_m|)$ should be as small as possible. Thus, the spherical assumption would be applicable when 1) the viewing direction is parallel to the line through the camera centers, 2) the FOV is narrow, and 3) scene points are far from the camera centers relative to the distance between centers.

For example, assume that image noise is given as one pixel when the cameras have 80° FOV with 300 pixels seeing in radial directions. In this case, the one-pixel image error is 0.4° and $\max|\theta_m| = 40^\circ$. The condition for the spherical camera assumption is $T < 0.0109 d_r$. If $d_r > 10$ m, the distance T should be less than 109 mm to assume a spherical camera safely.

Note that this is a very strict and sufficient condition for images to be spherical, not for motion estimation. We empirically determine the condition of T and d_r for motion estimation in Sec. V.

III. CALIBRATION OF MULTI-CAMERA SYSTEM

To estimate motions of a multi-camera system, intrinsic parameters and relative poses of cameras in a rig coordinate system are required. One can achieve this calibration using a fixed pattern in cases of a single camera and stereo cameras. However, it is not good for multiple non-overlapping cameras, because the FOV of the camera system is so large that the fixed pattern is hardly large enough.

We develop a calibration method without any fixed large pattern using a motion constraint of the multi-camera system.

A. A Motion Constraint of Multi-Camera System

The cameras in the multi-camera system are fixed in a camera rig, thus motions of each camera in its local camera coordinate system are related with the motion of the rig and the pose of the camera in the rig coordinate system.

Without loss of generality, we can set the rig coordinate system as the coordinate system of the first camera. Suppose that the second camera coordinate is expressed with a rotation R_{12} and a translation $\mathbf{t}_{12}$ in the rig coordinate system. If the camera rig moves by a rotation R_1 and a translation $\mathbf{t}_1$, the rotation R_2 and the translation $\mathbf{t}_2$ in the coordinate of the second camera is expressed as

$$\begin{bmatrix}\mathsf{R}_2 & \mathbf{t}_2\\ \mathbf{0}_3^\top & 1\end{bmatrix} = \begin{bmatrix}\mathsf{R}_{12} & \mathbf{t}_{12}\\ \mathbf{0}_3^\top & 1\end{bmatrix}\begin{bmatrix}\mathsf{R}_1 & \mathbf{t}_1\\ \mathbf{0}_3^\top & 1\end{bmatrix}\begin{bmatrix}\mathsf{R}_{12} & \mathbf{t}_{12}\\ \mathbf{0}_3^\top & 1\end{bmatrix}^{-1} \quad (1)$$

and this equation is a form of $\mathsf{AX} = \mathsf{XB}$ on the Euclidean group.

Using plane based methods [9], the intrinsic parameters and the camera motions of each camera can be estimated individually. With the estimated camera motions, one can estimate the poses R_{12} and $\mathbf{t}_{12}$ of the second camera in the rig coordinate system by solving $\mathsf{AX} = \mathsf{XB}$ on the Euclidean group [10]. This problem is known as *hand-eye calibration* for a robotic manipulator [11]. This method still uses a pattern for each camera, but it is not required to know the poses of the patterns in a common coordinate system. This gives more flexibility to calibrate multiple non-overlapping cameras.

B. Calibration of Rotations Using Motion Estimation

By assuming a spherical camera, only the rotations between cameras are required, not the discrepancy between camera centers. In this case, it is possible to estimate the rotations between cameras without any pattern, by estimating motions of each camera directly from image sequences.

Eq. (1) consists of two parts: rotation and translation. The rotation part can be written as

$$\mathsf{R}_2\mathsf{R}_{12} = \mathsf{R}_{12}\mathsf{R}_1$$

which is also a form of $\mathsf{AX} = \mathsf{XB}$ on the rotation group. We can measure the rotations R_1 and R_2 of the intrinsically calibrated cameras using a conventional structure from motion algorithm. We use the five-point algorithm [12] for motion estimation and Park and Martin's method [11] for solving the $\mathsf{R}_2\mathsf{R}_{12} = \mathsf{R}_{12}\mathsf{R}_1$ problem in a least square manner. Note that the absolute translation can not be estimated, because the estimated translations of each camera are all defined up to scale.

This proposed method does not require any pattern to calibrate a set of cameras, although it needs intrinsic parameters of each camera. One can notice that this method is a variant of Caspi and Irani's image alignment algorithm[13] for intrinsically calibrated cameras.

IV. STRUCTURE FROM MOTION OF SPHERICAL CAMERAS

To estimate a trajectory of a set of cameras, one can apply a series of algorithms: motion estimation between frames, integration of the motions, and optional bundle adjustment [14]. We apply these three steps to image sequences from the multiple non-overlapping cameras. Assuming a spherical camera, one can use any existing algorithm for a pin-hole camera such as a visual odometry [15].

A. Motion Estimation Between Frames of Spherical Cameras

Among the motion estimation methods for a single focal point camera, we use the five-point algorithm [12], which is one of the most stable motion estimators.

To estimate a motion of a spherical camera, we follow the three steps: mapping points on a unit sphere, estimating motions using random sample consensus (RANSAC), and optimizing motion parameters.

1) Mapping points on a unit sphere: At first, image features from an image from each camera are normalized using the intrinsic and distortion parameters. The normalized points are mapped on a unit sphere using the rotation of the camera in the rig coordinate system. To represent a point on a unit sphere, we use a nonhomogeneous 3-vector which is a directional unit vector from the sphere center. A point on a unit sphere $\mathbf{x}'_{kn}$ is calculated as

$$\mathbf{x}'_{kn} = \mathsf{R}_{k1}\mathbf{x}_{kn} \quad \text{s.t.} \quad |\mathbf{x}_{kn}| = 1$$

where $\mathbf{x}_{kn}$ is a normalized point observed by camera k, and R_{k1} is a rotation matrix from the local coordinate of the camera k to the rig coordinate system.

2) Estimating an essential matrix with RANSAC: It is not expected that all the correspondences are tracked and matched correctly. To eliminate false matches, we apply a RANSAC algorithm in estimating an essential matrix. Considerations in implementing RANSAC include an hypothesis generator with a small number of points and a verification function for a hypothesis. We use the five-point algorithm [12] to generate a hypothesis of an essential matrix, and a geometric error function as a verification function.

The geometric error function for correspondences $\mathbf{x}_1$ and $\mathbf{x}_2$ given a hypothesis E is defined as

$$d(\mathsf{E}, \mathbf{x}_1, \mathbf{x}_2) = \frac{|\mathbf{x}_2^\top \mathsf{E}\mathbf{x}_1|}{\sqrt{a^2+b^2}} + \frac{|\mathbf{x}_1^\top \mathsf{E}^\top \mathbf{x}_2|}{\sqrt{d^2+e^2}}$$

where a, b, d and e are the epipolar line coefficients defined as $(a,b,c)^\top = \mathsf{E}\mathbf{x}_1$ and $(d,e,f)^\top = \mathsf{E}^\top\mathbf{x}_2$. One can notice that the defined geometric error function is the same with the geometric epipolar error for a projective camera. The epipolar error for a projective camera can be used, because the spherical camera also has a single focal point.

3) Optimizing motion parameters: After getting an initial estimate of an essential matrix and a set of inliers, the essential matrix can be optimized by minimizing

$$C(\mathsf{E}; \mathbf{x}_1, \mathbf{x}_2) = \sum_{\{\mathbf{x}_1, \mathbf{x}_2 \in P\}} d(\mathsf{E}, \mathbf{x}_1, \mathbf{x}_2)$$

where P is a set of inlier pairs. We use the Levenberg-Marquardt algorithm implemented by Lourakis [16].

The motion parameters are retrieved from the estimated essential matrix using cheirality [14].

B. Trajectory Estimation by Integrating Motions Between Two Views

An estimated motion between frames is a velocity of the moving camera rig, because it represents a motion in a *unit time difference*. One can estimate a trajectory of the camera rig by incrementally integrating the estimated motions between frames as

$$\begin{bmatrix} \mathsf{R}_j & \mathbf{t}_j \\ \mathbf{0}^\top & 1 \end{bmatrix} = \begin{bmatrix} \mathsf{R}_{ij} & \mathbf{t}_{ij} \\ \mathbf{0}^\top & 1 \end{bmatrix} \begin{bmatrix} \mathsf{R}_i & \mathbf{t}_i \\ \mathbf{0}^\top & 1 \end{bmatrix} \tag{2}$$

which describes an update from the frame i to the frame j. R_i and $\mathbf{t}_i$ express a pose of the camera rig at frame i and R_{ij} and $\mathbf{t}_{ij}$ are a rotation matrix and a translation vector between frame i and j, respectively.

Note that it is impossible to get a traveling distance in Euclidean space, because there is no absolute distance available in this system. We can expect to estimate a trajectory defined up to scale. Therefore, one problem is to match scales of the translational motions, i.e. $\mathbf{t}_i$ and $\mathbf{t}_{ij}$ in Eq. (2). To match scales of the translational motions, we utilize reconstructed structures calculated by simple triangulation. By comparing the scales of the reconstructed structures, one can estimate a scale ratio between the structures, which is a ratio of the traveling distances.

The problem of estimating scales is formulated as

$$\arg\min \sum_{\text{all } n} \left(\mathbf{X}_n - s \begin{bmatrix} \mathsf{R}_i & \mathbf{t}_i \\ \mathbf{0}^\top & 1 \end{bmatrix}^{-1} \mathbf{X}_n^{ij} \right)^2$$

where $\mathbf{X}_n$ and $\mathbf{X}_n^{ij}$ are the corresponding 3D points reconstructed previously and between frame i and j, respectively. We use the Levenberg-Marquardt algorithm [16] to solve this optimization problem. After scale matching, the 3D points $\mathbf{X}_n$ should be updated using correspondences of all the frames to get more precise results.

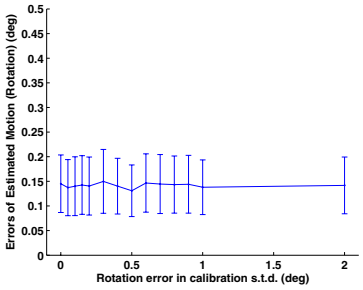

Fig. 3. Accuracy analysis of the proposed motion estimation algorithm under inaccurate calibration.

C. Optional Bundle Adjustment

The trajectory estimated by integration is obtained by minimizing the cost functions defined between two views. As time goes by, estimation errors are accumulating gradually. To reduce the accumulated errors, a bundle adjustment (BA) can be used [14].

Because the BA tries to optimize structures of scene points and motions of cameras simultaneously, the problem becomes very large. We use an efficient implementation by Lourakis [17] using the sparsity of the problem.

The cost function used is a distance c between a measured feature $\mathbf{x}$ and a predicted one from a given motion $\{\mathsf{R},\mathbf{t}\}$ and a scene point $\mathbf{X}$ in camera i, defined as

$$c(\mathbf{x},\{\mathsf{R},\mathbf{t}\},\mathbf{X}) = \|\mathbf{x} - \mathsf{R}_i(\mathsf{R}\mathbf{X}+\mathbf{t})\|_2^2$$

where R_i is a rotation matrix of the camera i in the rig coordinate system. This cost function is defined on an image plane of each camera, although the motion estimation between frames is formulated on a unit sphere. We use this because evaluating a Jacobian of this function takes less time than those defined on a unit sphere.

V. EXPERIMENTS

To analyze the performance of the proposed algorithms, we made a series of experiments using simulated and real data.

In generating simulated data, we assumed a small UAV which flies in a hallway. The width, height, and depth of the presumed hallway are 10 m, 20 m and 20 m, respectively. We randomly generated 300 points on each wall. The distances from the points to the wall were selected randomly with a Gaussian distribution whose standard deviation (s.t.d.) is 1 m.

The camera system was modeled from the real camera setup in Fig. 1. We used real calibration data obtained using patterns [10] for more realistic data generation.

A. Calibration of Multi-Camera System

At first, we tested the performance of the proposed calibration algorithm in Sec. III-B.

1) Required accuracy of multi-camera calibration: Before analyzing the proposed algorithm, we checked the required accuracy of the rotation between cameras in the rig for the proposed motion estimation. Fig. 3 shows the effects of errors in rotations between cameras on the accuracy of motion estimation. For this experiment, image features were disturbed with Gaussian noise whose standard deviation is 1 pixel. We made 100 trials in each experiment.

The proposed method works robustly even under 2° of rotation errors, because the optical flows between frames do not change much when the viewing direction changes a little. From the analysis in Fig. 3, we can conclude that less than 2° of errors in rotations between cameras is sufficient to get a good result with the proposed motion estimation algorithm.

2) Simulation aspects: Because the proposed calibration algorithm is using the estimated motions of single cameras, the rotation measurements are more erroneous than those of the pattern based method [10]. To find conditions under which the calibration can be achieved, we tested the algorithm in four aspects: the number of measurements, the variation of rotations, the distance of translation, and the variation of translation distances. In all cases, image features were disturbed with Gaussian noise of $\sigma = 1$ pixel.

3) Number of measurements: In testing the effect of the number of measurements, we generated motions of the camera rig randomly. Each motion gives one measurement for calibration. As in the first figure in Fig. 4, the estimated rotation between cameras becomes stable using more than 40 motions. In analyzing the other aspects, we used 100 motions.

4) Variation of rotation of the rig: The second figure in Fig. 4 shows the effect of the rotation variation of the rig motions. For this experiment, we randomly rotated the rig with random translations whose deviation is 50 mm. If standard deviation of the rig rotations is larger than 10°, the proposed method works well.

5) Direction of translations: The third experiment addresses the effect of the amount of translation. In this case, the rotations of the rig were randomly generated in the range of [-10,10]$^\circ$. The standard deviation of their translation is 50 mm with the given translation in a fixed direction. We can see that the estimation errors are increased, when the translational directions of the measurements are biased. This means that if a small UAV flies rotating in a fixed direction, the estimated motions of cameras have more errors than those from a complexly moving UAV.

6) Variation of translation of the rig: The last figure in Fig. 4 shows the effect varying the translations of the rig with no bias. This shows that a larger translation makes more errors in estimating rotation between cameras. This is because motion estimation of a single camera suffers from the translation-rotation ambiguity. The rotation in smaller translation appears more dominant than that in larger translation in analyzing optical flows.

Note that the estimation error is always less than the required accuracy of calibration given in Sec. V-A.1.

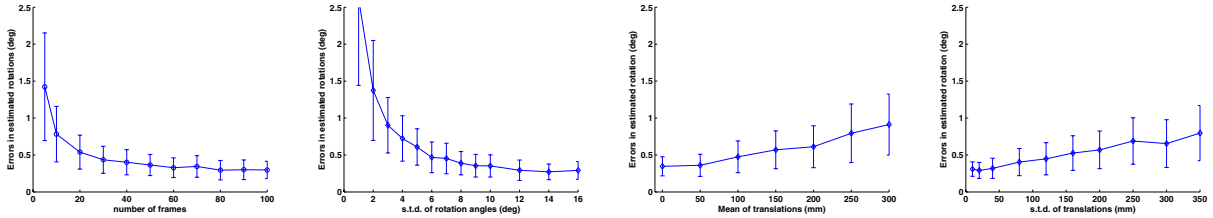

Fig. 4. Accuracy analysis of the proposed calibration algorithm

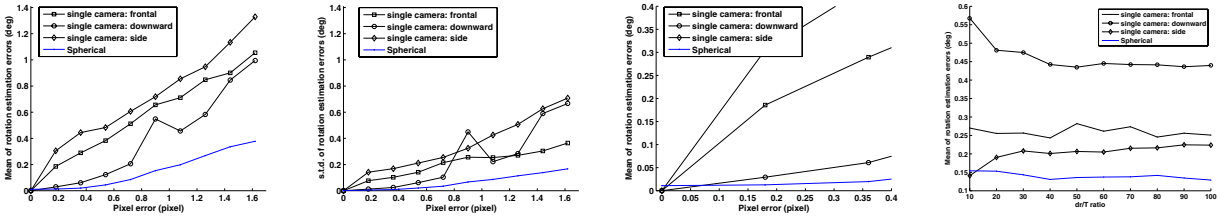

Fig. 5. Accuracy analysis of rotation estimation using the proposed method

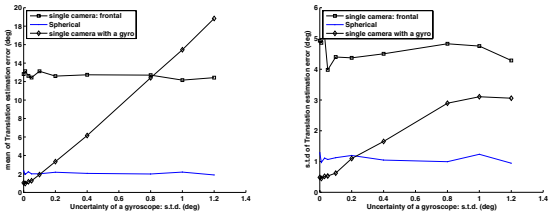

Fig. 6. Comparison of translation direction estimation algorithms

Fig. 7. Experimental setup for data acquisition

B. Motion Estimation using Spherical Cameras

The next series of experiments is about motion estimation using the spherical assumption. We generated motions of the camera rig randomly. 500 trials were performed for each experiment.

1) Estimation of rotation components: We compare the performances of rotation estimation using a single camera with the five-point algorithm [12] and a spherical model used in this paper. The first and the second figures in Fig. 5 show the mean and the standard deviation of the absolute errors of the estimated rotations, respectively. We can see that using multiple cameras with a spherical assumption improves both the accuracy and stability in estimating rotations from image sequences.

The third figure in Fig. 5 is an enlarged version of the first figure. One can notice the larger error occurs in motion estimates with the spherical assumption, when the image noise is small. This is the error due to the spherical camera assumption. In Sec. II, we derive a strict and sufficient condition for multiple cameras to be a spherical camera. The rightmost figure in Fig. 5 shows the performance on variations of d_r to T ratios defined in Sec. II when the image noise is 1 pixel. This experiment is to find the condition in which multiple cameras are to be a spherical camera in estimating motions. The spherical assumption gives better results when the distance ratio of d_r and T is larger than 20, although the theoretical ratio is about 100 as shown in Sec. II.

2) Estimation of translation components: As a second experiment, we compare the accuracy of the estimated translation directions. Because the translation-rotation ambiguity is caused by difficulties in decomposing translation and rotation from image flows, the more accurate rotation estimation makes the more accurate estimation of translation directions. Thus, we compared our algorithm to another method using an external gyroscope sensor [18].

For this experiment, we contaminated the measurements of the gyroscope sensor with Gaussian noise as well as image features. Because the features are disturbed with the same Gaussian noise in this experiment, the error in estimating motions without a gyroscope should be consistent. Fig. 6 shows the mean and the standard deviation of the estimated translational directions. Clearly, the spherical model works much better than the single camera. In addition, it can compute the translational directions more accurately than the method using a gyroscope when the gyroscope error is larger than 0.2°. Even when the gyroscope works very well, the spherical camera model works comparably. The stability of the spherical camera assumption is also better than those of the other methods.

Fig. 8. Examples of input images

Fig. 9. Estimated camera motions using the spherical model.

C. Experiment in the real world

In experiments using real image sequences, we used the camera set in Fig. 7. Because it is very hard to fly a small UAV at a low altitude, we attached the set of cameras on the top of a van. Fig. 8 shows examples of synchronized input images of the three cameras. One can notice that there is little overlap between images. We used the KLT feature tracker [19] to find correspondences between frames.

In the first experiments, we used three cameras which see in front, side and rear directions, respectively. The driving distance is about 300 m in 45 seconds. The recovered trajectory is shown in the left figure of Fig. 9, which is overlaid on a satellite image to verify the result. In the second experiments, a different configuration of cameras is used; front, left and right directions. It traveled about 300 m in 60 seconds, and the right figure in Fig. 9 shows the estimated trajectory. Without bundle adjustment, estimating motions from tracked features runs at about 10 frames/second although it highly depends on the number of RANSAC iterations. Feature tracking from image sequences is still the most time-consuming process, which can be reduced by using a graphical processing unit [20].

VI. CONCLUSIONS

In this paper, we develop a motion estimation system using multiple non-overlapping cameras for a small UAV. Because of payload limitation of a small UAV, light-weight multiple cameras are used to resolve the translation-rotation ambiguity of motion estimation using images. We make a spherical camera assumption that the multiple non-overlapping cameras have a common projection center and identify when this assumption holds both analytically and empirically. The spherical assumption simplifies motion estimation of multiple non-overlapping cameras to a conventional problem for a single projective camera. Based on the spherical assumption, we also propose a method to estimate rotations between non-overlapping cameras using a motion constraint between cameras.

The experiments show that the accuracy of rotations between cameras does not greatly affect the accuracy of motion estimation too much, and the proposed calibration method based on the spherical camera model can fulfill the requirements of the calibration accuracy with a sufficient number of motions. We also show that the spherical assumption makes motion estimation more accurate for our UAV application. Interestingly, the spherical assumption for the non-overlapping cameras make the motion estimation algorithm both simpler and more accurate for our UAV.

REFERENCES

[1] P. Baker, C. Fermuller, Y. Aloimonos, and R. Pless, "A spherical eye from multiple cameras (makes better models of the world)," in *IEEE Computer Vision and Pattern Recognition or CVPR*, 2001.

[2] M. D. Grossberg and S. K. Nayar, "A general imaging model and a method for finding its parameters." in *International Conference on Computer Vision*, 2001, pp. 108–115.

[3] W. Chang and C. Chen, "Pose estimation for multiple camera systems," in *International Conference on Pattern Recognition*, 2004, pp. III: 262–265.

[4] C. Chen and W. Chang, "On pose recovery for generalized visual sensors," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 26, no. 7, pp. 848–861, July 2004.

[5] J. Frahm, K. Koser, and R. Koch, "Pose estimation for multi-camera systems," in *26th Symposium of the German Association for Pattern Recognition (DAGM)*, 2004, pp. 286–293.

[6] M. Kaess and F. Dellaert, "Visual SLAM with a multi-camera rig," Georgia Institute of Technology, Tech. Rep. GIT-GVU-06-06, Feb 2006.

[7] Y. Chen, L. Liou, Y. Hung, and C. Fuh, "Three-dimensional ego-motion estimation from motion fields observed with multiple cameras," *Pattern Recognition*, vol. 34, no. 8, pp. 1573–1583, August 2001.

[8] R. Pless, "Using many cameras as one," in *IEEE Computer Vision and Pattern Recognition or CVPR*, 2003, pp. II: 587–593.

[9] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, pp. 1330–1334, November 2000.

[10] H. Kim, J.-S. Kim, and I. S. Kweon, "Motion estimation using centers of non-overlapping cameras," in *Proceedings of the 13th Japan-Korea Joint Workshop on Frontiers of Computer Vision*, 2007, pp. 387–393.

[11] F. Park and B. Martin, "Robot sensor calibration: Solving $\mathsf{AX} = \mathsf{XB}$ on the euclidean group," *IEEE Transactions on Robotics and Automation*, vol. 10, no. 5, pp. 717–721, 1994.

[12] D. Nister, "An efficient solution to the five-point relative pose problem," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 26, no. 6, pp. 756–777, June 2004.

[13] Y. Caspi and M. Irani, "Aligning non-overlapping sequences." *International Journal of Computer Vision*, vol. 48, no. 1, pp. 39–51, 2002.

[14] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge, 2003.

[15] D. Nister, O. Naroditsky, and J. Bergen, "Visual odometry for ground vehicle applications," *Journal of Field Robotics*, vol. 23, pp. 3–20, 2006.

[16] M. Lourakis, "levmar: Levenberg-marquardt nonlinear least squares algorithms in C/C++," [web page] `http://www.ics.forth.gr/~lourakis/levmar/`, Jul. 2004.

[17] M. Lourakis and A. Argyros, "The design and implementation of a generic sparse bundle adjustment software package based on the levenberg-marquardt algorithm," Institute of Computer Science - FORTH, Heraklion, Crete, Greece, Tech. Rep. 340, Aug. 2004.

[18] T. Mukai and N. Ohnishi, "The recovery of object shape and camera motion using a sensing system with a video camera and a gyro sensor," in *International Conference on Computer Vision*, 1999, pp. 411–417.

[19] Y. Ma, S. Soatto, J. Kosecka, and S. Sastry, *An Invitation to 3-D Vision: From Images to Geometric Models*. SpringerVerlag, 2003.

[20] S. Sinha, J.-M. Frahm, and M. Pollefeys, "GPU-based. video feature tracking and matching," University of North Carolina at Chapel Hill, Technical Report TR06-012, May 2006.