Online Calibration and Synchronization of Cellphone Camera and Gyroscope

Chao Jia and Brian L. Evans
Department of Electrical and Computer Engineering
Wireless Networking and Communications Group
The University of Texas at Austin, Austin, Texas, USA
Email: cjia@utexas.edu, bevans@ece.utexas.edu

***Abstract*—Gyroscope is playing a key role in helping estimate camera rotation during mobile video capture. The fusion of gyroscope and visual measurements needs the knowledge of camera projection parameters, the gyroscope bias and the relative orientation between gyroscope and camera. Moreover, the timestamps of gyroscope and video frames are usually not well synchronized. In this paper, we propose an online method that estimates all the necessary parameters while capturing videos. Our contributions are (1) simultaneous online camera self-calibration and camera-gyroscope calibration based on an implicit extended Kalman filter, and (2) generalization of coplanarity constraint of camera rotation in a rolling shutter camera model for cellphones. The proposed method is able to accurately estimate the needed parameters online with all kinds of camera motion, and can be embedded in gyro-aided applications such as video stabilization and feature tracking.**

I. INTRODUCTION

Mobile video capture is currently undergoing a huge growth with the fast development of smartphone industry. Besides video recording itself, the recorded videos also provide a great amount of opportunities for applications such as augmented reality and visual odometry. No matter what application mobile video capture is used for, camera motion estimation is an essential step to improve the video quality and better analyze the video content. Hand-held mobile devices like smartphones usually suffer from fast changing motion, which makes it difficult to track the camera motion accurately using only the captured videos. For this reason, inertial sensors on smartphones such as gyroscope and accelerometer have been used to help estimate camera motion due to their increasing accuracy, high sampling rate and robustness to light conditions. However, most existing works assume the inertial sensors have been calibrated and synchronized beforehand so that the relative pose of the inertial sensors to the camera, the measurement biases and the delay between the timestamps of different sensors are known. Moreover, camera self-calibration is often assumed to be done offline too. Some calibration methods can be only performed in laboratory environments with special devices, which further prevents amateur photographers from taking videos conveniently with the help of inertial sensors. In this paper, we focus on online calibration and synchronization of cellphone cameras and inertial sensors while capturing videos, without any prior knowledge about the devices.

The CMOS image sensors used in cellphone cameras capture different rows in a frame sequentially from top to bottom. When there is fast relative motion between the scene and the video camera, a frame can be distorted because each row was captured under different 3D-to-2D projections. This is known as rolling shutter effect [1] and has to be considered in calibration and fusion of visual and inertial sensors. The inertial sensor that we calibrate in this paper is gyroscope only. It has been shown that the rotation estimation from gyroscope has been used successfully in video stabilization [2] and feature tracking [3].

The proposed online calibration and synchronization is based on an extended Kalman filter (EKF). Although we care about camera rotation only, we do not assume any degeneration in the motion of the camera. By extending the recent proposed coplanarity constraint of camera rotation [4] to rolling shutter cameras, we come up with a novel implicit measurement that involves only camera rotation but works for any camera translation, including zero translation (pure rotation). The implicit measurements can be effectively used in the EKF to update the estimate of state vectors.

II. RELATED WORK

Camera self-calibration has been extensively studied [5], but previous work on online self-calibration is very rare. In [6] full-parameter online camera self-calibration is first proposed in the framework of sequential Bayesian structure from motion using a sum of Gaussian (SOG) filter. This work assumes a global shutter camera model and the motion of the camera has to contain large enough translation to make the structure from motion problem well-conditioned.

The inertial sensors (gyroscope and accelerometer) are widely used in navigation and simultaneous localization and mapping (SLAM) together with visual measurements [7]. The estimation of inertial sensor biases and relative pose between inertial sensors and camera has been recently implemented online during SLAM or navigation [8]. However, to the best of our knowledge all of the previous works assume that the camera itself has been calibrated, i.e., the camera projection parameters are known. Moreover, rolling shutter effect was not taken into account in fusion of inertial and visual sensors until very recently [9], [10].

In videos, the displacement of pixels between consecutive frames is mainly caused by camera rotation. Based on this fact, gyroscope was successfully applied to stabilize the video and remove rolling shutter effect [2], [11]. Similarly, gyroscope measurements were used to pre-warp the frames so that the search space of Kanade-Lucas-Tomasi (KLT) [12] feature tracker can be narrowed down to its convergence region [3]. In these works there is no need to use the accelerometer. Therefore, only the camera and the gyroscope need to be calibrated.

To calibrate the camera and gyroscope system, [2] proposed to quickly shake the camera while pointing at a far-away object (e.g., a building). Feature points between consecutive frames are matched and all parameters are estimated simultaneously by minimizing the homographic re-projection errors under pure rotation model. The calibration in [3] is also based on homography transformation of matched feature points assuming pure rotation, except that different parameters are estimated separately first and then refined through non-linear optimization. However, as shown in [3], when the camera translation is not negligible relative to the distance of the feature points to the camera, such pure rotation model becomes less accurate and the calibration results will deviate from the ground truth. Our

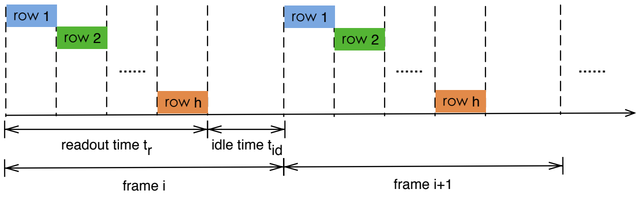

Fig. 1. Rows are captured sequentially in rolling shutter cameras.

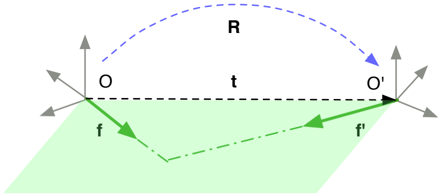

Fig. 2. The epipolar constraint on a pair of features in two viewpoints.

calibration method differs with [2], [3] not only in that it is online estimation, but also in that it does not assume zero translation at all. Therefore, the proposed calibration can be performed implicitly anytime and anywhere while the camera is recording video. This is especially convenient for amateur photographers who want to take stabilized videos with smartphone cameras.

III. Rolling Shutter Camera Model and Gyroscope

Points in the camera reference space are projected according to the pinhole camera model. Assuming the 3D point coordinates in the camera reference space is $[X_c, Y_c, Z_c]^{\mathrm{T}}$, its projection on image plane can be represented as

$$\begin{bmatrix} u_x \\ u_y \end{bmatrix} = \begin{bmatrix} c_x + f\frac{X_c}{Z_c} \\ c_y + f\frac{Y_c}{Z_c} \end{bmatrix}, \tag{1}$$

where f is the focal length and c_x, c_y are the principal point coordinates. Here we assume that the camera projection skew is zero and the pixel aspect ratio is 1 as in [6], which is a reasonable assumption for today's cellphone cameras.

In rolling shutter cameras, rows in each frame are exposed sequentially from top to bottom, as shown in Fig. 1 For an image pixel $\mathbf{u} = [u_x, u_y]^{\mathrm{T}}$ in frame i, the exposure time can be represented as $t(\mathbf{u}, i) = t_i + t_r \frac{u_y}{h}$, where t_i is the timestamp for frame i and h is the total number of rows in each frame. t_r is the readout time for each frame, which is usually about $60\% - 90\%$ of the time interval between frames.

Usually there is a constant delay t_d between the recorded timestamps of gyroscope and videos. Thus using the timestamps of gyroscopes as reference, the exposure time of pixel $\mathbf{u}$ in frame i should be modified as

$$t(\mathbf{u}, i) = t_i + t_d + t_r \frac{u_y}{h}. \tag{2}$$

To use the gyroscope readings we also need to know $\mathbf{q}_c$, the relative orientation of the camera in the gyroscope frame of reference (represented in quaternion). Finally, the bias of the gyroscope $\mathbf{b}_g$ needs to be considered. Therefore, in the online calibration we need to estimate the parameters f, c_x, c_y, t_r, t_d, $\mathbf{b}_g$ and $\mathbf{q}_c$.

IV. Coplanarity Constraint for Camera Rotation

First let us consider a global shutter camera in which all of the pixels in the same frame are captured at the same time. Assume the normalized 3D coordinate vectors of a certain feature in two viewpoints (frames) are $\mathbf{f}_i$ and $\mathbf{f}'_i$ (note that by inverting (1) we can not recover the absolute scale but only the direction of the 3D feature vector). The well-known epipolar constraint is $(\mathbf{f}_i \times \mathbf{R}\mathbf{f}'_i) \cdot \mathbf{t} = 0$, where $\mathbf{R}$ and $\mathbf{t}$ are the relative rotation and translation between the two viewpoints. The epipolar constraint means that the vectors $\mathbf{f}_i, \mathbf{R}\mathbf{f}'_i$ and $\mathbf{t}$ are coplanar, as shown in Fig. 2. By the epipolar constraint all vectors $\mathbf{f}_i \times \mathbf{R}\mathbf{f}'_i$ are perpendicular to the translation vector $\mathbf{t}$, and thus are coplanar ($\mathbf{t}$ is the normal vector of this plane). Such coplanarity can be expressed by the determinant of any three vectors being zero

$$|(\mathbf{f}_1 \times \mathbf{R}\mathbf{f}'_1) \ (\mathbf{f}_2 \times \mathbf{R}\mathbf{f}'_2) \ (\mathbf{f}_3 \times \mathbf{R}\mathbf{f}'_3)| = 0. \tag{3}$$

This coplanarity is introduced in [4] and does not depend on the translation at all. Another good property of (3) is that it is still valid in the extreme case of zero translation since all vectors $\mathbf{f}_i \times \mathbf{R}\mathbf{f}'_i$ will become zero.

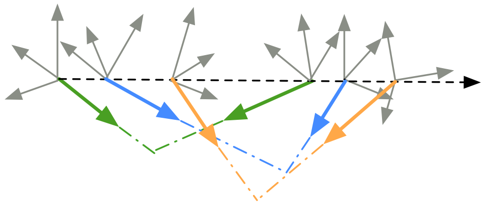

Fig. 3. The cross products of all matched features are perpendicular to the camera translation vector.

In rolling shutter cameras, however, the viewpoint is not unique for the features captured in the same frame. Note that both the traditional epipolar constraint and the coplanarity constraint (3) are expressed in the reference of one of the two viewpoints. In fact, this frame of reference can be chosen arbitrarily. Once the reference is fixed, we can represent the camera orientation corresponding to any feature (determined by its exposure moment for rolling shutter cameras) in this reference. For the matched features between any two consecutive frames in rolling shutter cameras, we propose the following constraint

$$|(\mathbf{R}_1\mathbf{f}_1 \times \mathbf{R}'_1\mathbf{f}'_1) \ (\mathbf{R}_2\mathbf{f}_2 \times \mathbf{R}'_2\mathbf{f}'_2) \ (\mathbf{R}_3\mathbf{f}_3 \times \mathbf{R}'_3\mathbf{f}'_3)| = 0. \tag{4}$$

Note that in (4) $\mathbf{R}'_1$ means the camera orientation correspond to feature 1 in the second frame, not the transpose of $\mathbf{R}_1$. Constraint (4) does not exactly hold in general cases but only under the assumption that the relative camera translations between the exposure moments for all pair of matched features are in the same direction. The readout time of two consecutive frames are at most 66ms (for 30 fps videos) and in such short period of time the camera translation can be well approximated by a constant direction. Note that such approximation is more general than the approximation used in [10] which assumes the linear velocity of the camera is constant. The constraint is illustrated by Fig. 3. We use the coplanarity constraint (4) as implicit measurement to estimate the all the parameters in an EKF. The way to represent the camera orientation corresponding to each feature using the parameters and gyroscope readings is shown in the next section.

V. EKF-based Online Calibration and Synchronization

The online calibration and synchronization is based on an EKF. Besides the parameters mentioned in Section III, we also estimate the

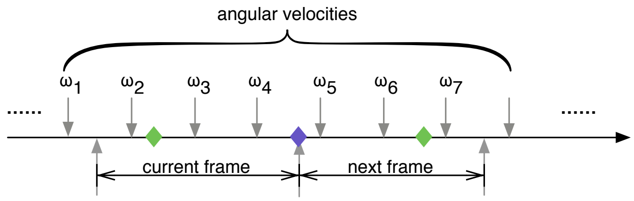

Fig. 4. Timing relationship between the gyroscope readings and the video frames.

true angular velocities corresponding to each gyroscope reading. The gyroscope in cellphones usually has a higher sampling rate than the video frame rate. Fig. 4 illustrates the timing relationship between the gyroscope readings and the video frames. Assume a pair of matched features $\mathbf{f}_i$ and $\mathbf{f}'_i$ are detected as green diamonds and the reference time is fixed as the timestamp of the next frame (shown as the purple diamond). The relative camera orientation can then be expressed by the angular velocities

$$\mathbf{R}_i = \prod_{n=1}^{M} \Theta(\boldsymbol{\omega}_n \Delta t_n^i), \tag{5}$$

where M is the total number of angular velocities involved in computing the relative orientation (M=7 in Fig. 4) and Δt_n^i is the time duration that the angular velocity $\boldsymbol{\omega}_n$ is used in the integration (assuming constant angular velocity between readings). Each sub-relative rotation matrix can be computed by exponentiating the skew symmetric matrix formed by the product of angular velocity and its duration:

$$\Theta(\boldsymbol{\omega}_n \Delta t_n^i) = \exp(\text{skew}(\boldsymbol{\omega}_n)\Delta t_n^i). \tag{6}$$

In this way, the relative camera orientation corresponding to any feature detected in the current and next frame can be expressed by the angular velocities. Note that the angular velocities (gyroscope readings) have to be transformed into the coordinate system of camera first using $\mathbf{q}_c$.

Our EKF evolves when every video frame is captured, as in [9]. The state vector is defined as

$$\mathbf{x} = [f \; c_x \; c_y \; t_r \; t_d \; \mathbf{b}_g^{\mathrm{T}} \; \mathbf{q}_c^{\mathrm{T}} \; \boldsymbol{\omega}_1^{\mathrm{T}} \ldots \boldsymbol{\omega}_M^{\mathrm{T}}]^{\mathrm{T}}. \tag{7}$$

A. State Prediction

All the parameters appeared in Section III except $\mathbf{b}_g$ are constant so they are just copied in state prediction. We model the dynamics of $\mathbf{b}_g$ by random-walk process. Since the EKF evolves from frame to frame, the angular velocities in the state vector are propagated simultaneously. As shown in Fig. 4, there will be an overlap of several angular velocities between consecutive state vectors. The angular velocities are propagated as following:

$$\begin{cases} \boldsymbol{\omega}_{n_{k|k-1}} = \boldsymbol{\omega}_{n_{k-1|k-1}}, & \text{if } \boldsymbol{\omega}_n \text{ appears in the previous state} \\ \boldsymbol{\omega}_{n_{k|k-1}} = \hat{\boldsymbol{\omega}}_n + \mathbf{b}_g + \mathbf{n}_g, & \text{otherwise} \end{cases} \tag{8}$$

where $\hat{\boldsymbol{\omega}}_n$ is the gyroscope reading and $\mathbf{n}_g$ is the gyroscope measurement noise.

B. State Update

After features are matched between the current frame and the next frame, we randomly picked N groups of features with 3 features in each group. In this way we can get N measurements from the coplanarity constraint shown in Section IV. For instance, the measurement formed by features 1,2 and 3 is

$$0 = \;|(\mathbf{R}_1(\mathbf{f}_1 + \mathbf{v}_1) \times \mathbf{R}'_1(\mathbf{f}'_1 + \mathbf{v}'_1)) \; (\mathbf{R}_2(\mathbf{f}_2 + \mathbf{v}_2) \tag{9}$$

$$\times \mathbf{R}'_2(\mathbf{f}'_2 + \mathbf{v}'_2)) \; (\mathbf{R}_3(\mathbf{f}_3 + \mathbf{v}_3) \times \mathbf{R}'_3(\mathbf{f}'_3 + \mathbf{v}'_3))|, \tag{10}$$

where $\mathbf{v}_i$ and $\mathbf{v}'_i$ are feature detection errors. Note that the 3D feature vectors $\{\mathbf{f}_i, \mathbf{f}'_i\}$ are obtained by inverting the camera projection (1) and then normalizing. All of the N coplanarity constraints generates N implicit measurements. The state update is performed right after state prediction is done. Only one round of state prediction and update is needed once a new frame is read and all features are tracked.

C. State Initialization

The state vector needs to be initialized carefully to make the EKF work properly. We initialize the principle point coordinates c_x, c_y to be the center of the frame. The focal length is initialized using the horizontal view angle provided by Android camera API. If the operation system of the smartphone does not provide the value of horizontal view angle, SOG filters can be used with several initial guesses as in [6]. The readout time t_r is initialized as 0.0275 ms which is about 82.5% of the entire interval between frames. The coordinate-system of the gyroscope is defined relative to the screen of the phone in its default orientation in all Android phones. Thus we can get the initial guess of $\mathbf{q}_c$ depending on whether we are using front or rear camera. This initial guess is usually accurate enough, but our calibration is necessary since the camera is sometimes not perfectly aligned with the screen of the phone. The initial values of all other parameters (t_d and $\mathbf{b}_g$) are just set as 0.

To make sure that the true value lies in the 3σ intervals of the initial Gaussian distributions, we initialize the standard deviation of c_x, c_y, f, t_r, t_d as 6.67 pixels, 6.67 pixels, 20 pixels, 0.00167 s and 0.01 s, respectively. The standard deviation of each element in $\mathbf{b}_g$ and $\mathbf{q}_c$ is initialized as 0.0067. We set the standard deviation of gyroscope measurement noise and feature detection error as 0.003 rads/s and 1 pixels, respectively.

VI. Experimental results

In our experiments, we use a Google Nexus S Android smartphone that is equipped with a three-axis gyroscope. We capture the videos and the gyroscope readings from the smartphone and perform the proposed online calibration and synchronization in MATLAB. The feature points are tracked using KLT tracker. We divide the frame into 4 equally sized bins and perform outlier rejection locally within each bin by computing a homography transformation using RANSAC [13], as in [14]. The ground truth of camera projection parameters are obtained using the offline camera calibration method in [15]. The ground truth of all other parameters are estimated using the batch optimization method in [2]. Note that to compute the ground truth the video need to be carefully captured since [2] assumes pure camera rotation. We test the performance of the proposed method on various video sequences and show the results on two typical sequences: one shot while running forward and the other shot while panning the camera in front of a building.

The running sequence is used to test the performance of the algorithm under arbitrary camera motion, including very high frequency shake and non-zero translation. The estimation errors of the online calibration and synchronization are shown in Fig. 5, with blue lines representing the estimation error and red lines representing the 99.7%(3σ) uncertainty bounds. For the relative orientation $\mathbf{q}_c$ we only show the Euclidean error between the estimated quaternion vector and the ground truth. Unlike other parameters, the gyroscope

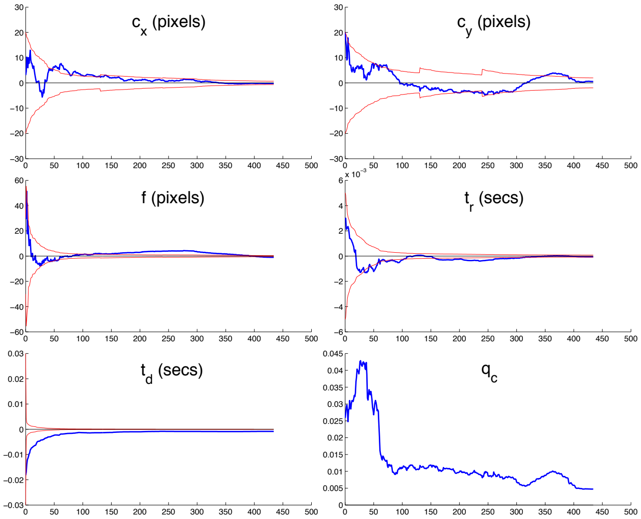

Fig. 5. Estimation error over the running sequence.

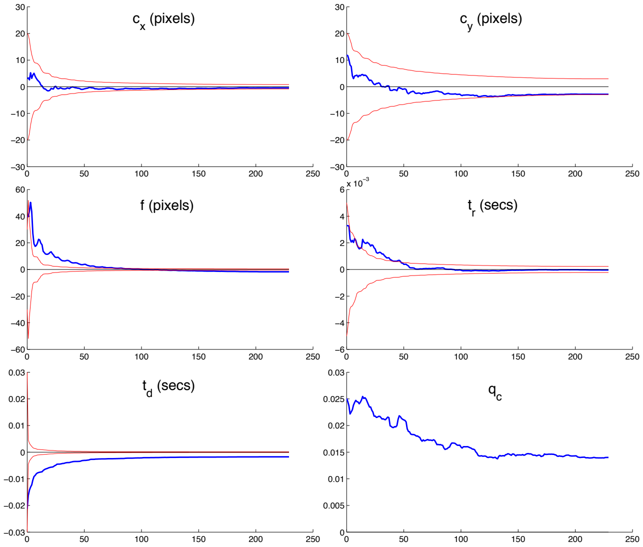

Fig. 6. Estimation error over the panning sequence

bias $\mathbf{b}_g$ keeps changing so there is no unique ground truth value and we do not show it in the results. From Fig. 5 we can observe that the proposed method is able to accurately estimate the parameters in hundreds of frames. The estimation of the focal length f and the sensor delay t_d appears to be over-confident due to the highly nonlinearity of the measurement equations, but the estimates themselves still converge to the the ground truth very well.

In the second test video sequence we simply pan the camera in front of a building. This video is used to test the algorithm under (almost) zero camera translation (pure rotation). The estimation errors are shown in Fig. 6. The proposed algorithm works equally well and converge even faster than the running sequence, because the panning motion guarantees large pixel displacement between consecutive frames and makes the parameters more observable. The estimate of the relative orientation $\mathbf{q}_c$ is not as accurate as in the running sequence, which we believe is caused by the single form of motion (panning).

The current running speed of the proposed algorithm with MATLAB implementation (feature detection and tracking is implemented using mex functions of OpenCV implementation [16]) is 7 fps on a laptop with 2.3GHz Intel i5 processor. Note that the we do not have to run the calibration using every pair of adjacent frames, so the proposed algorithm is possible to run in real-time.

VII. Conclusions

In this paper we propose an online calibration and synchronization algorithm for cellphones that is able to estimate not only the camera projection parameters, but also the gyroscope bias, the relative orientation between the camera and gyroscope, and the delay between the timestamps of the two sensors. The proposed algorithm is based on the generalization of the coplanarity constraint of the cross products of matched features in a rolling shutter camera model. Experiments run on real data collected from cellphones show that the proposed algorithm can successfully estimate all of the needed parameters with different kinds of motion of the cellphones. This online calibration and synchronization of rolling shutter camera and gyroscope make it more convenient for high quality video recording, gyro-aided feature tracking, and visual-inertial navigation.

References

[1] C. Geyer, M. Meingast, and S. Sastry, "Geometric models of rolling-shutter cameras," in *Proc. Workshop Omnidirectional Vision*, 2005.

[2] A. Karpenko, D. Jacobs, J. Baek, and M. Levoy, "Digital video stabilization and rolling shutter correction using gyroscopes," Stanford University, Tech. Rep., Mar. 2011.

[3] M. Hwangbo, J.-S. Kim, and T. Kanade, "Gyro-aided feature tracking for a moving camera: fusion, auto-calibration and GPU implementation," *Intl. Journal Robotics Research*, vol. 30, no. 14, pp. 1755–1774, 2011.

[4] L. Kneip, R. Siegwart, and M. Pollefeys, "Finding the exact rotation between two images independently of the translation," in *Proc. European Conf. Computer Vision*, Oct. 2012.

[5] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2004.

[6] J. Civera, D. Bueno, A. Davison, and J. Montiel, "Camera self-calibration for sequetial Bayesian structure from motion," in *Proc. IEEE Intl. Conf. Robotics and Automation*, 2009.

[7] P. Corke, J. Lobo, and J. Dias, "An introduction to inertial and visual sensing," *Intl. Journal Robotics Research*, vol. 26, no. 6, pp. 519–535, 2007.

[8] J. Kelly and G. Sukhatme, "Visual-inertial sensor fusion: localization, mapping and sensor-to-sensor self-calibration," *Intl. Journal Robotics Research*, vol. 30, no. 1, pp. 56–79, 2011.

[9] C. Jia and B. L. Evans, "Probabilistic 3-D motion estimation for rolling shutter video rectification from visual and inertial measurements," in *Proc. IEEE Intl. Workshop Multimedia Signal Processing*, Sep. 2012.

[10] M. Li, B. Kim, and A. Mourikis, "Real-time motion tracking on a cellphone using inertial sensing and a rolling shutter camera," in *Proc. IEEE Intl. Robotics and Automation*, May 2013.

[11] G. Hanning, N. Forslöw, P.-E. Forssén, E. Ringaby, D. Törnqvist, and J. Callmer, "Stabilizing cell phone video using inertial measurement sensors," in *Proc. IEEE Intl. Workshop Mobile Vision*, Nov. 2011.

[12] B. Lucas and T. Kanade, "An iterative image registration technique with application to stereo vision," in *Proc. Intl. Joint Conf. Artificial Intelligence*, 1981, pp. 674–679.

[13] M. Fischler and R. Bolles, "Random sample consensus, a paradigm for model fitting with applications to image analysis and automated cartography," *Comms. of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[14] M. Grundmann, V. Kwatra, D. Castro, and I. Essa, "Calibration-free rolling shutter removal," in *Proc. IEEE Intl. Conf. Computational Photography*, Apr. 2012.

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

[16] K. Yamaguchi, "mexopencv," http://www.cs.stonybrook.edu/~kyamagu/mexopencv/.