2017 IEEE International Conference on Robotics and Automation (ICRA)
Singapore, May 29 - June 3, 2017

Drift-Correcting Self-Calibration for Visual-Inertial SLAM

Fernando Nobre, Michael Kasper and Christoffer Heckman*

***Abstract*— We present a solution for online simultaneous localization and mapping (SLAM) self-calibration in the presence of drift in calibration parameters in order to support accurate long-term operation. Calibration parameters such as the camera focal length or camera-to-IMU extrinsics are frequently subject to drift over long periods of operation, inducing cumulative error in the reconstruction. The key contributions are modeling calibration parameters as a *spatiotemporal* quantity: sensor-to-sensor spatial calibration and sensor intrinsic parameters are continuously time-varying, with statistical tests for change detection and regression. An analysis of the long term effects of inappropriately modeling time-varying sensor calibration is also provided. Constant-time operation is achieved by selecting only a fixed number of informative segments of the trajectory for calibration parameter estimation, giving the added benefit of avoiding early linearization errors by not rolling past measurements into a prior distribution. Our approach is validated with simulated and real-world data.**

I. Introduction

Autonomous platforms destined for long-term applications equipped with visual and inertial sensors have become increasingly ubiquitous. Generally these platforms must undergo sophisticated calibration routines to estimate extrinsic and intrinsic parameters to high degrees of certainty before sensor data may be interpreted and fused. Once fielded, calibration parameters are generally fixed for the lifetime of the platform, or are modeled as a piecewise constant function [1]. For many applications however, these platforms may experience gradual changes in calibration parameters due to e.g. temperature dilation, non-rigid mounting or accidental bumps that can change both sensor intrinsic and extrinsic parameters. Self-calibration addresses this by inferring intrinsic and extrinsic parameters pertaining to proprioceptive and exteroceptive sensors without using a known calibration target or a specific calibration routine. The motivation behind self-calibration is to remove the explicit, tedious, and sometimes nearly impossible calibration procedure from robotic applications and to enable robust long-term autonomous operation. Most approaches to online self-calibration that do not rely on marginalization (such as filtering, which is subject to linearization errors of past measurements) either assume calibration parameters remain constant or change in a piecewise constant fashion. These approximations work well for single digit percent drift over calibration parameters on relatively short (<200m) trajectories but induce considerable drift for longer periods of operation. Figure 1 (bottom) shows a simulation of time-varying calibration parameters and the traditional piecewise-constant approximations, demonstrating long periods of incorrectly-estimated parameters.

Fig. 1: Top: experimental robotic vehicle with a camera mounted on a pan-tilt unit for changing camera-to-IMU extrinsics (IMU rigidly mounted inside body of platform). Bottom: camera focal length ground truth (*solid green line*) over an 8 hour trajectory using our visual-inertial simulation pipeline. The piecewise-constant approximation (*dashed blue line*) lags behind the ground truth due to uncertainty in the measurements which make a small change in calibration parameters indistinguishable from noise. The constant assumption is the *dashed red line*. The ground truth shows the continuous time-varying nature of the parameter, which our method approximates.

We present a novel approach that approximately models calibration parameters as a continuous time-varying quantity, which we refer to as drift-correcting self-calibration (DCSC). By continuously estimating calibration parameters, no prior knowledge of calibration values or procedures is required. Furthermore, with the addition of statistical change detection and regression on drifting calibration parameters, long-term autonomy applications are greatly robustified against accidental changes where the calibration varies over time. This approach is based on probabilistically determining

This work was supported by the Toyota Motor Corporation

All authors are with the Autonomous Robotics and Perception Group at the University of Colorado, Boulder.

* Corresponding author. E-mail: christoffer.heckman at colorado.edu

segments where the motion provides enough excitation on the calibration parameters to permit observability [2], [3]. This permits seamless handling of degenerate motions and unknown calibration parameters, enabling the much sought after "power-on-and-go" operation. Our approach includes probabilistic change detection as well as change regression; the former system detects change events that require completely re-estimating calibration parameters, and the latter identifies the start of the change region so that past poses can be re-estimated with the correct calibration parameters. This approach is validated on camera intrinsic and camera-to-IMU extrinsic parameters, however is easily extensible to an arbitrary number of sensors [4]. To the authors' knowledge, DCSC is the first proposed solution to long-term drift due to time varying calibration parameters that does not rely on a prior distribution.

II. Related Work

The problem of self-calibration with varying camera intrinsics has received much attention in the literature in part due to the benefits outlined above. Both [5] and [6] considered a batch-solution self-calibration tailored to different intrinsic parameters. [7] presented a method to calibrate the varying intrinsics of a pinhole camera in a batch setting, given the rotation of the camera was known. A solution was also offered to align the rotation sensor and camera data in time.

Many current techniques for vision-aided inertial navigation use filtering approaches (e.g. [2], [3], [8]) or a smoothing formulation. In either case the estimation is made constant-time by rolling past information into a prior distribution. Filtering methods present the significant drawback of introducing inconsistencies due to linearization errors of past measurements which cannot be corrected post hoc, particularly troublesome for non-linear camera models. Some recent work has tackled these inconsistencies; see, e.g. [9], [10], [11], [12]. The state-of-the-art includes methods to estimate poses and landmarks along with calibration parameters, but these approaches do not output the marginals for the calibration parameters, which are desirable for long-term autonomy applications.

Building on these works, simultaneous solutions to the SLAM and self-calibration problem have been proposed but generally all online solutions assume constant calibration parameters. [13] proposed a method to recursively estimate camera and landmark 3D parameters as well as the intrinsic parameters of a nonlinear camera model in an online framework. [14] also developed a filtering solution to estimate both the camera pose and also intrinsics and extrinsics for a nonlinear camera model with rolling shutter and a commercial grade IMU in an online framework, but that approach does not output covariances in an MLE sense.

III. Methodology

The proposed method aims to continuously estimate both intrinsic and extrinsic calibration parameters [4], while also detecting change events due to sensor perturbation and regressing calibration parameters in an arbitrarily large change region. As such, DCSC has three distinct components: *Constant Time Self-Calibration* [15] is needed in order to continuously estimate the instantaneous belief over intrinsic and extrinsic calibration parameters at any point in the trajectory; *Change Detection* [1] signals a statistically significant difference in the means of the instantaneous belief over calibration parameters and the long term belief, indicating a high probability that the calibration parameters have been perturbed; and finally, *Change Regression* checks for the start of the change region and regresses the calibration parameters in that region, allowing for re-estimation of past poses with the correct calibration parameters, reducing the long-term drift. Each of these components are described in the following sections, with emphasis given to Change Regression which is a novel contribution on which this paper is focused. Alongside DCSC, a keyframe-based [16] pose-and-landmark non-linear maximum likelihood estimation is performed for real-time map updates. Thus, our implementation details will also describe a system for *adaptive SLAM estimation* [17], which is used to ensure the maximum likelihood poses and landmarks are estimated.

A. Calibration Parameter Modeling

Central to the proposed self-calibration methodology explained in the following sections is the modeling of how calibration parameters change over time. Most approaches to self-calibration estimate calibration parameters such as camera intrinsics, sensor-to-sensor extrinsics and time offset at the start of the trajectory and make the assumption that those parameters will remain fixed, i.e.: $\boldsymbol{x}_c(t) = \boldsymbol{x}_c(0)$ where $\boldsymbol{x}_c$ represents the calibration parameter vector, a function of time. This assumption is valid for short trajectories, however it breaks down when long term operation is desired, due to the inherent drift in sensor calibrations. Another option is modeling calibration parameters as a multivariable piecewise constant function, and probabilistically detecting change events, as in [1] and [4]:

$$\boldsymbol{x}_c(t) = \boldsymbol{x}_c(t_i) \quad \text{if } t_i \leq t \leq t_{i+1}, \tag{1}$$

where t_i, $i = \{1, \ldots, n\}$ defines n points of time in which the estimated calibration parameters are detected to have changed. Note in this case that the time-dependent parameter vector is approximated by a piecewise constant constant function, $\boldsymbol{x}_c(t_i)$. While this approach is well-suited to large sudden changes, it does not handle cases where the calibration parameters are slowly drifting over a long period of time due to the difficulty in distinguishing small changes from sensor noise. Figure 1 depicts this scenario.

Given the fact that sensor calibration parameters are often slowly-varying functions of time in long term autonomy applications, we propose the following model:

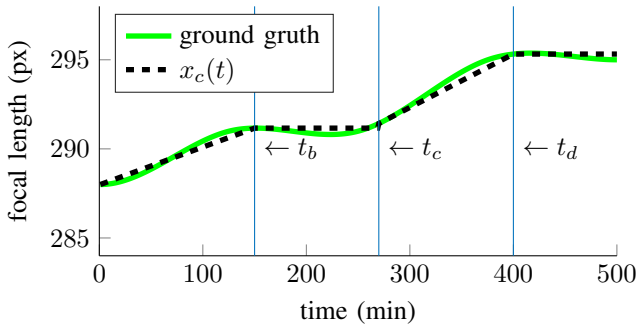

Fig. 2: Piecewise time-varying calibration parameters over simulated dataset.

$$\boldsymbol{x}_c(t) = \begin{cases} \boldsymbol{x}_c(t_i) & t_i \leq t \leq t_{i+1} \\ \boldsymbol{f}_i(t) & t_{i+1} \leq t \leq t_{i+2} \\ \boldsymbol{x}_c(t_{i+2}) & t_{i+2} \leq t \leq t_{i+3} \\ & \dots \end{cases} \tag{2}$$

where e.g. $t_{i+1} \leq t \leq t_{i+2}$ represents a change region that can be arbitrarily long. The key considerations for this model are 1) determining the start and end points for the change event, and 2) establishing basis functions for $\boldsymbol{f}_i(t)$; Figure 2 shows an example of this approach with a linear basis function. Section III-E goes into detail on both these aspects.

B. Constant Time Self-Calibration

The constant time self-calibrating framework [15], [4] is briefly summarized here in order to aid the exposition of the overall methodology. Due to the limited observability and high connectivity of calibration parameters in the SLAM pose graph, it is impractical to estimate these parameters in real time applications using conventional filtering or smoothing approaches (see [8], [18], [19], [12]). Instead every segment of M frames in the trajectory is analyzed, and the N most informative segments are added to a *priority queue*. Both M and N are tuning parameters dependent on the the calibration parameters being estimated. The information content of a segment is defined as the normalized entropy of its posterior. A score based on the uncertainty is associated to each segment, which is then compared to all the segments in the *priority queue*. If it has a better score, the worst scoring window in the queue is swapped out. Every time the priority queue is updated, a batch optimization over poses, landmarks and calibration parameters is run on all the segments in the queue to obtain a new set of calibration parameters. As such, the priority queue represents a rolling estimate of the N most informative segments in the trajectory, or in other words, it encodes the long-term belief over calibration parameters. The work in [4] extended the self-calibrating framework [15] to an arbitrary number of sensors and to calibrate both camera intrinsics and camera-to-IMU extrinsics. Figure 3 shows a graphical model representing the priority queue, candidate segment and the corresponding calibration parameters being estimated.

A maximum-likelihood estimation is performed on the state vector:

$$\boldsymbol{X} = \begin{bmatrix} \{ & \boldsymbol{x}_{wp_n} & \boldsymbol{v}_{w_n} & \boldsymbol{b}_{\boldsymbol{g}_n} & \boldsymbol{b}_{\boldsymbol{a}_n} & \} & \{\rho_k\} & \{\boldsymbol{x}_c\} \end{bmatrix}^T, \tag{3}$$

where $\boldsymbol{x}_{wp_n} \in SE(3)$ is the transformation from the coordinates of the n^{th} keyframe to world coordinates, $\boldsymbol{v}_{w_n} \in \mathbb{R}^3$ is the velocity vector of the n^{th} keyframe in world coordinates, and $\boldsymbol{b}_{\boldsymbol{g}_n} \in \mathbb{R}^3$ and $\boldsymbol{b}_{\boldsymbol{a}_n} \in \mathbb{R}^3$ are the gyroscope and accelerometer bias parameters for the n^{th} keyframe respectively. $\{\rho_k\}$ is the 1-D inverse-depth [20] parameter for the k^{th} landmark and $\boldsymbol{x}_c$ the calibration parameters. Note that $\boldsymbol{x}_{wp_n}$ has 6 degrees of freedom: 3 for translation, and 3 for rotation. To avoid singularities arising from a minimal representation (e.g. using Euler angles), the rotation component of the transformation $\boldsymbol{R} \in SO(3)$ is represented as a quaternion, with the optimization lifted to the tangent space (at the identity) of the $so(3)$ manifold.

The joint probability distribution of a given segment j with measurements Z_j can be factored using the conditional independence of individual measurements assumption as follows:

$$p\left(\boldsymbol{X}|Z_j\right) = p\left(Z_j|\boldsymbol{X}\right) p\left(\boldsymbol{X}\right) = \prod_{i=1}^{n} p(z_i|\boldsymbol{X}) \tag{4}$$

The prior term $P(\boldsymbol{X})$ is omitted since this approach explicitly avoids the use of a prior in favor of the priority queue. The practical effect of this is that the current estimation is conditioned on previous estimates rather than marginalizing them into a prior. The optimal estimate for the parameter vector can be obtained by maximizing the joint probability. Making the usual assumption of a Gaussian distribution over parameter noise, the probability distribution in Eq. (4) when considering only visual measurements can be written as:

$$p\left(z_i|\boldsymbol{X}\right) \propto \exp\left(-\frac{1}{2}\left\|z_i - h_i(\boldsymbol{X})\right\|_{\Sigma}^2\right), \tag{5}$$

where $||\cdot||_{\Sigma}^2$ is the squared Mahalanobis distance and $z_i \in \mathbb{R}^2$ is the 2D measured pixel location and $h(\mathbf{X})$ is the measurement model. Measurements are formed by the traditional sparse visual odometry geometry, wherein image keypoints are tracked across frames. We will now provide a very brief summary of this approach.

A landmark parameterized by inverse depth is projected onto an image resulting in a projected pixel coordinate $\boldsymbol{p}_{\text{proj}}$ which is formulated via a transfer function $\boldsymbol{T}$ as follows:

$$\begin{aligned} \boldsymbol{p}_{\text{proj}} &= T\left(\boldsymbol{p}_r, \boldsymbol{T}_{wp_m}, \boldsymbol{T}_{wp_r}\boldsymbol{T}_{pc}, \rho\right) \\ &= \mathcal{P}\left(\boldsymbol{T}_{pc}^{-1}\boldsymbol{T}_{wp_m}^{-1}\boldsymbol{T}_{wp_r}\boldsymbol{T}_{pc}\mathcal{P}^{-1}\left(\boldsymbol{p}_r, \rho\right)\right), \end{aligned} \tag{6}$$

where ρ is the inverse depth of the landmark, $\boldsymbol{T}_{wp_r}$ is the transformation from the coordinates of the reference keyframe (in which the landmark was first seen and initialized) to world coordinates, $\boldsymbol{T}_{wp_m}$ is the transformation from the measurement keyframe to world coordinates, $\boldsymbol{p}_r$ is the

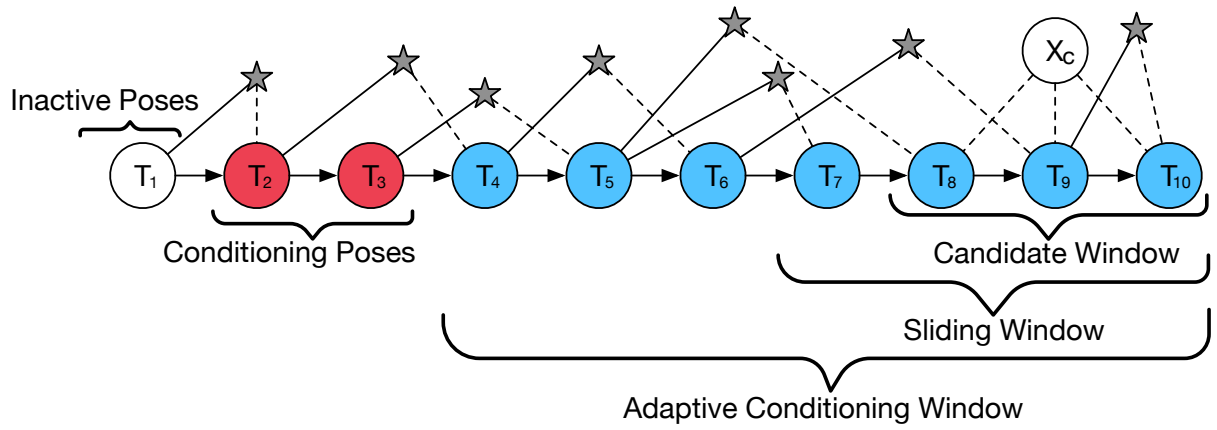

Fig. 3: Example pose graph. Poses being estimated (*blue*) are conditioned on past poses (*red*) and landmark positions (stars). Both the fixed sliding window and the adaptive window are conditioned on previous poses. The candidate window is not conditioned since it does not make the assumption that previous poses are correctly estimated.

2D image location where the original feature was initialized in the reference keyframe, $\boldsymbol{p}_m$ is the measured 2D image location in the measurement keyframe, $\boldsymbol{T}_{pc}$ is the transformation from the camera to the keyframe coordinates, $\mathcal{P}^{-1}$ is the 2D to 3D back-projection function and $\mathcal{P}$ is the 3D to 2D camera projection function which returns the predicted 2D image coordinates. The camera-to-keyframe transformation $\boldsymbol{T}_{pc}$ is non-identity as the keyframe is collocated on the inertial frame (the frame in which inertial measurements are made), to simplify the inertial integration. $\boldsymbol{T}_{pc}$ is the camera-to-IMU calibration parameter we have interest in estimating. The usual approach is to minimize a nonlinear least squares problem with the residual function

$$r_{\mathcal{V}_{m,k}} = \|\boldsymbol{e}_{\mathcal{V}_{m,k}}\|^2_{\Sigma_{\boldsymbol{p}_{m,k}}} = \|\boldsymbol{p}_{m,k} - \boldsymbol{p}_{\text{proj}}\|^2_{\Sigma_{\boldsymbol{p}_{m,k}}}, \quad (7)$$

where $\boldsymbol{p}_{m,k}$ is the measured 2D image location of the k^{th} landmark in the m^{th} keyframe with covariance $\Sigma_{\boldsymbol{p}_{m,k}}$.

Given this formulation, the estimates for the state vector in Eq. (3) can be obtained via maximum likelihood estimation [21]. The covariance matrix for the posterior distribution over the calibration parameters is obtained by inverting the problem's Fisher information matrix and extracting the appropriate submatrix. The effects of differing units in the calibration vector (such as rotation units and translation units, or focal length and central point) are removed by normalizing the covariance matrix as described in [15].

C. Initialization

As shown in [22], [23], having a good initial estimate can mean the difference between fast convergence and complete divergence. A good initial guess is needed on both intrinsic and extrinsic calibration parameters. We treat these cases separately as follows:

1) Camera Intrinsics Initialization: The camera intrinsic parameters are bootstrapped by running a batch optimization over the entire state vector (rig location, landmark inverse depths and all the camera intrinsic calibration parameters). Once the score of the batch estimation falls below a predetermined threshold, indicating that the uncertainty over calibration parameters is sufficiently small, estimation is handed over to the candidate segments and priority queue, as described in Section III-B.

2) Camera-to-IMU Initialization: We leverage the work from [22], [2], [3] which shows that with a minimum of three frames and five tracked features, it is possible to obtain the camera-to-IMU rotation. This initial rotation estimate can then be used to solve a linear system for an initial guess at the translation estimate. We consider the scenario where enough (five or more) features are observed across at least three frames. The tracked features can be used to obtain the relative rotation between two camera frames *i*, *j*: ${}^{C}\boldsymbol{R}_{ij}$ and integrating the IMU measurements to obtain the relative rotation: ${}^{B}\boldsymbol{R}_{ij}$, where *C* represents the camera frame and *B* the body frame, which is defined without loss of generality as the IMU frame. The following equation relates the camera rotation to the body rotation:

$${}^{C}\boldsymbol{R}_{ij} = {}^{C}_{B}\boldsymbol{R}\,{}^{B}\boldsymbol{R}_{ij}\,{}^{B}_{C}\boldsymbol{R} \Rightarrow {}^{C}\boldsymbol{R}_{ij}\,{}^{C}_{B}\boldsymbol{R} = {}^{C}_{B}\boldsymbol{R}\,{}^{B}\boldsymbol{R}_{ij}, \quad (8)$$

where ${}^{C}_{B}\boldsymbol{R}$ is the rotation of the body frame in the camera frame. In order to obtain ${}^{C}_{B}\boldsymbol{R}$ we employ an error-state formulation to minimize a robustified over-constrained least squares problem.

In our experience we find that collecting more than 3 frames yielded more reliable estimates; therefore, we use 20 frames for the initial rotation estimate. Once the estimate on ${}^{C}_{B}\boldsymbol{R}$ has converged, translation can be obtained by employing the method described in [22] by solving a linear system obtained from transferring the 3D position of a landmark from the camera to the body frame.

D. Change Detection

The priority queue posterior (with covariance Σ'_{PQ}) represents the uncertainty over the calibration parameters considering the top k segments in the trajectory. As these segments are usually not temporally consecutive, this distribution encodes the long term belief over the calibration parameters.

Conversely, the candidate segment posterior (with covariance Σ_s) is calculated based on the most recent measurements and represents an instantaneous belief over the calibration parameters. If there is a sudden change in calibration parameters, for example if the camera is rotated or moved to a different location on the platform, then this will manifest as a difference in the means of the two posterior distributions. The simple difference in means cannot be used as a change detecting mechanism however, since the uncertainty associated to the estimate needs to be taken into account. This procedure, comparing the means of two multivariate normal distributions with different covariances, is known as the Multivariate Behrens-Fisher problem. Using an F distribution as in [1], the null hypothesis that the means of the candidate segment and that of the priority queue are equal can be tested:

$$H_0 : \mu_{PQ} = \mu_s \tag{9}$$

By comparing the p-value corresponding to the F distribution to a significance parameter $\alpha = 0.1$ the null hypothesis can be rejected for $p \leq \alpha$. There are several events, such as feature-less environments, motion blur, loss of tracking and non-static features which may lead to an incorrectly estimated posterior for the candidate segment. In order to avoid these scenarios a simple test is used where ν_{cs} consecutive candidate segments must have $p \leq \alpha$ for a change event to be triggered, where $\nu_{\text{cs}} = 3$.

A failure case for this approach when dealing with slow-changing parameters is when the candidate window mean consistently differs from the priority queue mean, but not enough to clearly be distinguishable from noise, so the condition $p \leq \alpha$ for ν_{cs} consecutive segments will not be met and a change event will not be triggered. This will cause the priority queue to slowly drift towards the new calibration parameter as candidate segments are swapped in, however since a change was not detected, candidate segments which where estimated prior to the change event will remain in the queue, resulting in a sub-optimal global estimate.

This prompts a second criterion for detecting a change: if the mean of the priority queue, equal to $\boldsymbol{x}_c$, over the past 3 seconds fits a linear curve with slope $\lambda > \lambda_{\text{th}}$ a tuned threshold, a change event is triggered and the priority queue is re-estimated. This allows for past segments to be removed from the queue and re-estimation on the new parameters. Note that λ_{th} sets the DCSC algorithm's sensitivity to slow changes.

E. Change Regression

The change detection mechanism presented in Section III-D will not detect the exact onset of the change event. As shown in Figure 4, there can be a considerable time-delay between the start of a change event and the change detection. The critical failure case is a drifting calibration parameter, which is indistinguishable from noise until it differs significantly from the previous estimate. The option of simply making the change detection mechanism more sensitive to changes by adjusting the threshold parameter α is not viable since that will trigger change events on inaccurately estimated candidate segments, causing the priority queue to be routinely cleared and re-estimated, which can result in worse priority queue estimates and has a direct impact on both accuracy and real-time performance.

Instead of attempting to detect the exact onset of a change event as it occurs, the start point for the change event is regressed from the change detection point, which will be after the start of the change: when a change event is triggered as in Section III-D at keyframe n^* every previous keyframe $n < n^*$ is tested as the starting position for the change event. This is done by leveraging the novel probabilistic change detection introduced in [1].

1) Start Point Estimation: The intuition behind detecting the start point is that, according to the model described in Section III-A, the change region is preceded by a period in which the calibration parameters are relatively constant. Considering that the candidate window encodes the instantaneous belief over calibration parameters, then for every keyframe, the p-value for the null hypothesis test between the priority queue distribution and the latest candidate segment of which that frame was a part is stored. This allows for detecting the constant calibration regions, as depicted in Figure 2, by a segment in which the p-value is smaller than a threshold Φ_{th}. We employ the following heuristic: if ν_{sp} consecutive segments have a p-value divergence smaller than Φ_{th}, where $\nu_{\text{sp}} = 50$ is used. If the start point is unable to be estimated the change regression is aborted and the system falls back onto the default setting of re-estimating calibration parameters from the change detection point onward.

2) Parameter Regression: Once the start point for the change event has been determined, the calibration values for all poses during time (t_{i+1}, t_{i+2}) in Eq. (2) need to be re-estimated (see Figure 2 for an example; poses between times $(0, t_b)$ would require re-estimation). In our implementation this is accomplished by fitting a linear curve to the calibration value at the regressed start point of the change event and the converged priority queue values after the change event. The choice of basis function to fit is dependent on the sensor and how changes are expected. In our experiments camera focal length and camera-to-IMU extrinsics are estimated, which are considered to drift in an approximately linear form for handheld and vehicle-mounted applications. Special care is taken to guarantee that all poses in the change region are re-estimated with the updated calibration parameters.

F. Adaptive SLAM

Finally, an adaptive asynchronous conditioning [17] solution is employed to avoid the use of a prior distribution on the sliding window estimation. When conditioning is used instead of marginalization, current active parameters are conditioned on previous parameters, which are assumed to be correct. However since new information may alter the estimate for previous poses, a sliding window pose and landmark estimation is run on a separate thread. This sliding

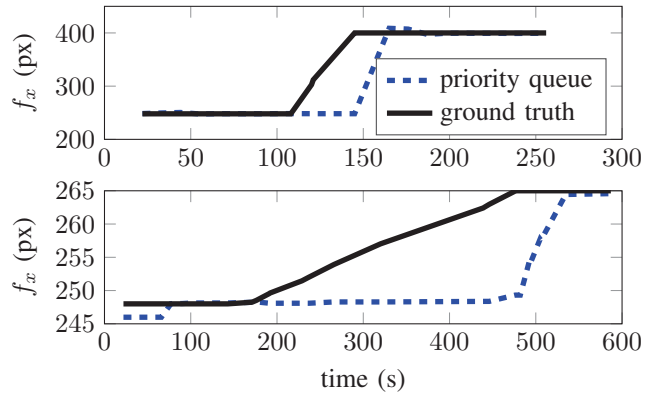

Fig. 4: Comparison of a fast change of the camera field of view from 130° to 45° (top figure) in camera focal length and a slow drift from 130° to 120° degrees (bottom figure). The drift over a long period of time greatly increases the area between the ground truth and the priority queue estimate, resulting in incorrect estimates for a large segment of the trajectory. Each are averaged over 10 simulations with noise added.

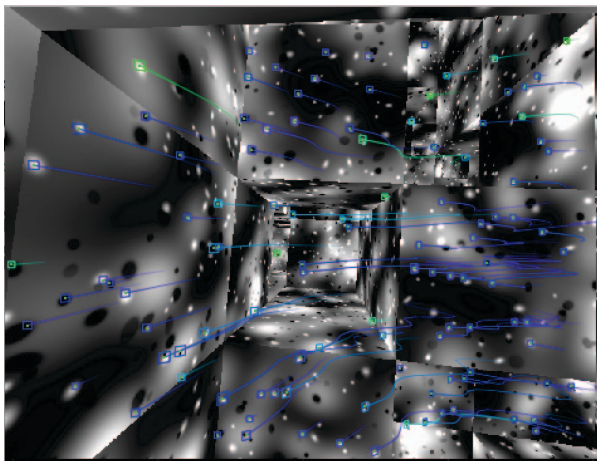

Fig. 5: Simulated visual trajectory with feature tracks.

window can adaptively increase its size to alter previous poses based on new measurements. The criteria to increase the window is based on the "tension" of the conditioning residuals, explained as follows. Conditioning residuals are the residual terms connecting an active and inactive pose. For example, a landmark that has a reference frame in an inactive pose, but is seen in an active pose will have a conditioning visual residual. The window is expanded when the the current estimate for a parameter falls outside of the expected estimate based on the conditioning residual. Since multiple sensor modalities are used, the Mahalanobis distance of each conditioning residual is thresholded in a χ^2 test to probabilistically determine when a residual is outside of its expected interval (inducing "tension" in that residual).

G. Visual Tracking

Visual tracking is inspired on the tracking component of [24], where the photometric error of a patch is directly miminized to find the new location of the feature. Harris corners are used for feature initialization, in image regions where there are a small number of active tracks. NCC scores for corresponding feature patches are thresholded at 0.875 to reject large changes in appearance. A keyframing approach [25] is used for improved performance and to deal with situations such as stationary camera.

IV. Simulations

Special attention was given to creating a pipeline for generating simulated visual and inertial data for evaluation of both the proposed algorithm and the effects of not properly accounting for changing calibration parameters over long trajectories. In order to be able to evaluate arbitrary motions, a random 6-DOF pose graph is generated using a system inspired by video game dynamics. This trajectory is then used to carve out a path in maze of cubes, ensuring that any simulated motion will be visually trackable. Figure 5 shows one such path, and the corresponding features and feature tracks. Simulated inertial accelerometer and gyroscope data corresponding to the trajectory is also generated for consistent scale and evaluating camera-to-IMU estimation. The use of simulated data allows for exact ground truth comparison, which is especially challenging when evaluating the response to slowly drifting to calibration parameters in real world settings.

V. Experiments and Results

A. Simulation

In order to evaluate the proposed method, the simulation pipeline described in Section IV was used to generate arbitrary trajectories with corresponding synchronized visual-inertial data. A series of Monte Carlo simulations of camera intrinsics and camera-to-IMU calibrations were performed. We chose a set of 5 trajectories, with varying degrees of excitation on each degree of freedom (so as to avoid known degenerate motions). For each trajectory we ran 30 simulations with varying calibration parameters: camera focal length and camera-to-IMU rotation. The camera's field of view was initialized at 130° and changed to 120° over a time period t_{change} drawn from a Gaussian distribution with a mean of 60s and standard deviation of 10s. The camera-to-IMU rotation was initialized at ${}^{C}_{B}\boldsymbol{\rho} = [0.53\ 0.53\ 0.53]^T$ and changed to ${}^{C}_{B}\boldsymbol{\rho} = [0.53\ 0.53\ 0.58]^T$. Camera focal length and camera-to-IMU rotation initial and final values were perturbed by a zero-mean Gaussian with standard deviation of 5°. Simulated camera data was captured at 15 frames per second, with IMU updates at 100Hz. For each projected feature point from the simulated images (640 × 480 resolution), independent zero-mean Gaussian noise with $\sigma = 0.5$ was added to the (u, v) pixel coordinates. Zero-mean Gaussian noise with $\sigma = 10^{-3}$ was also added to the IMU accelerometer and gyroscope measurements and biases. Each simulation was run on three different calibration schemes: the proposed DCSC algorithm, the piece-wise constant method described in [1] and [4] and the constant method where the calibration is obtained in the start of the trajectory and held constant throughout. The start time for the change event for

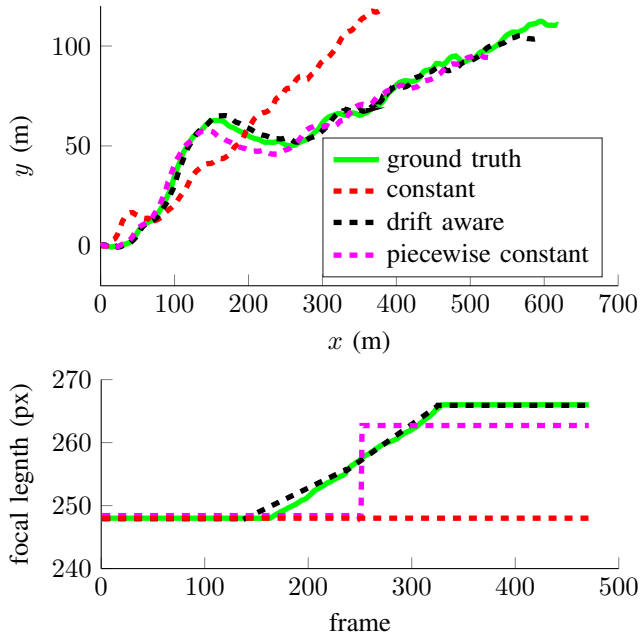

Fig. 6: 700m simulated trajectory. The camera focal length was changed linearly from 120° to 130° over a 60 second period. The proposed DCSC algorithm obtains the smallest translation error at 0.04% of the total distance traveled. The bottom figure shows the focal length (f_x) as the change occurs, and how each self-calibration scheme responds. DCSC also correctly finds the start and end point of the change event.

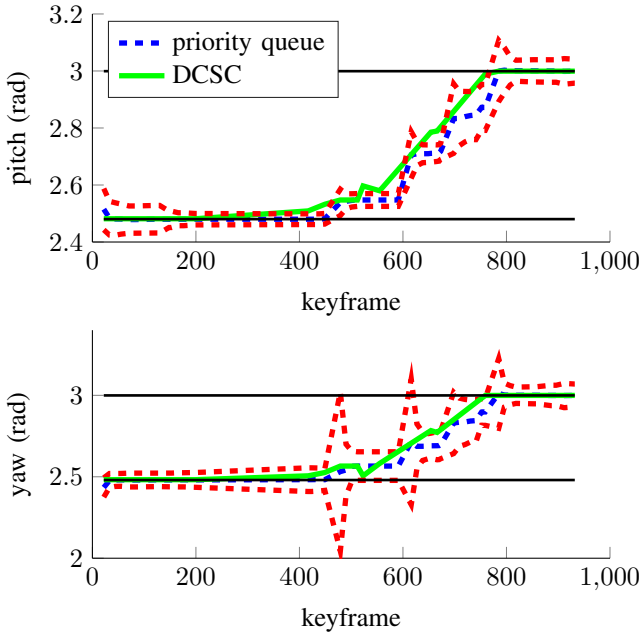

Fig. 7: Self-calibration and drift correction on experimental platform. Priority queue estimates (*blue dashed line*) and its 3σ bounds (*red dashed line*) compared to the DSCS drift correcting estimates (*green solid line*)

each simulation is drawn from a uniform distribution in the first half of the trajectory. The results of these simulations are shown in Table I for the DCSC algorithm, where ${}^{B}_{C}\rho$ is the total camera-to-IMU rotation error, % Drift is average translation drift and % Change Start is at which fraction of the change region the start point was detected.

One such trajectory is shown in Figure 6 where the DCSC method obtains final translation error of 28m which corresponds to 0.04% of the distance traveled and an average rotation error of 0.406 rad

B. Experiments with the Mobile Platform

In order to determine the performance of the proposed algorithm on data from real hardware, we used our experimental vehicle, depicted in Figure 1. The vehicle is equipped with a global shutter Ximea MQ022CG-CM camera with a wide field-of-view lens at 2040×1080 resolution downsampled to 640×480 mounted on a pan-tilt unit and an onboard Gladiator MEMS IMU capturing at 200Hz. Images were captured at 30Hz. The platform was driven over a 300m trajectory where the camera-to-IMU rotation was changed with the pan-tilt sensor over the course of 2 minutes. Figure 7 shows the comparison of the piecewise priority queue with it's 3σ bounds and the DCSC algorithm. The covariance on the priority queue spikes at change events when the queue is wiped but quickly tightens around the mean as segments are added to the queue. A ground truth was not available for the experiment so a comparison of start and end poses was used: The DCSC algorithm had a translation error of 1.13m, equivalent to 0.3% of the trajectory. Using the piecewise constant approximation the final translation error was 2.22m, or 0.7% of the trajectory.

VI. Discussion

This paper presents online, constant-time self-calibration and change detection with re-calibration for joint estimation of camera-to-IMU transform and camera intrinsic parameters, dealing explicitly with the case of drifting calibration parameters over long trajectories. The system is evaluated with experimental and simulated data and shown to converge to offline calibration estimates even in the presence of slowly drifting calibration parameters. The statistical change detection framework presented initially in [1] is used to detect change regions for drifting parameters and estimate the calibration parameters in the drift region.

The use of a drift correcting self-calibrating framework coupled with adaptive conditioning window for re-estimation of past poses allows this framework to operate in long-term applications where the accumulation of linearization errors in a prior distribution and the accumulation of incorrectly estimated calibration parameters over change periods would lead to significant drift. We present an analysis on the effects of inappropriate modeling of calibration parameters over long trajectories, and show how the use of a multivariate probabilistic change detection framework can greatly reduce the drift even in the presence of hard-to-detect incremental changes over time in calibration parameters. This method presents some failure cases that warrant further study, such as when the rate of drift is slow compared to the inherent noise in estimation errors, the boundary detection scheme presented may be inaccurate. An adaptive way of choosing all the tuning parameters is also necessary for this system to

TABLE I: DCSC Monte Carlo Simulation Results

	f_x Abs Error (px)		f_y Abs Error (px)		c_x Abs Error		c_y Abs Error		${}^B_C\rho$ Error		% Drift	% Change Start
Set	μ	3σ	μ	3σ	μ	3σ	μ	3σ	μ	3σ	-	-
1	1.01	16.62	4.34	25.21	0.95	10.42	1.14	11.02	0.09	0.05	0.012	0.195
2	2.32	20.24	3.58	27.85	1.32	14.98	2.04	17.90	0.12	0.04	0.069	0.153
3	1.76	14.22	2.91	18.82	0.52	11.52	0.87	9.88	0.16	0.09	0.093	0.107
4	3.42	23.13	4.02	30.14	1.80	12.92	1.91	11.71	0.07	0.03	0.095	0.206
5	1.89	13.87	1.94	15.02	0.89	9.78	0.99	10.53	0.15	0.11	0.094	0.124

be easily usable in practice. In future work we would like to investigate the use of different basis functions, such as higher order polynomials or Gaussian Processes. An observability analysis on the parameters for the basis functions is also left for future development.

REFERENCES

[1] N. Keivan and G. Sibley, "Online SLAM with any-time self-calibration and automatic change detection," in *International Conference on Robotics and Automation*. IEEE, 2015, pp. 5775–5782.

[2] E. S. Jones and S. Soatto, "Visual-inertial navigation, mapping and localization: A scalable real-time causal approach," *International Journal of Robotics Research*, vol. 30, no. 4, pp. 407–430, Apr. 2011.

[3] J. Kelly and G. S. Sukhatme, "Visual-Inertial Sensor Fusion: Localization, Mapping and Sensor-to-Sensor Self-calibration," *International Journal of Robotics Research*, vol. 30, no. 1, pp. 56–79, Jan. 2011.

[4] F. Nobre, C. Heckman, and G. Sibley, "Multi-sensor slam with online self-calibration and change detection," in *International Symposium on Experimental Robotics (ISER).*, 2016.

[5] A. Heyden and K. Astrom, "Euclidean reconstruction from image sequences with varying and unknown focal length and principal point," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Institute of Electrical Engineers, Inc (IEEE), 1997, pp. 438–443.

[6] M. Pollefeys, R. Koch, and L. V. Gool, "Self-calibration and metric reconstruction in spite of varying and unknown internal camera parameters," in *International Journal of Computer Vision*, 1999, pp. 7–25.

[7] J.-M. Frahm and R. Koch, "Camera calibration with known rotation," in *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*. IEEE, 2003, pp. 1418–1425.

[8] A. I. Mourikis and S. I. Roumeliotis, "A Multi-State Constraint Kalman Filter for Vision-aided Inertial Navigation," in *International Conference on Robotics and Automation*. IEEE, 2007, pp. 3565–3572.

[9] M. Li and A. I. Mourikis, "High-precision, consistent EKF-based visual–inertial odometry," *International Journal of Robotics Research*, vol. 32, no. 6, pp. 690–711, May 2013.

[10] J. A. Hesch, D. G. Kottas, S. L. Bowman, and S. I. Roumeliotis, "Towards Consistent Vision-Aided Inertial Navigation," *Algorithmic Foundations of Robotics*, vol. 86, no. Chapter 34, pp. 559–574, 2013.

[11] J. Civera, D. R. Bueno, A. J. Davison, and J. M. M. Montiel, "Camera self-calibration for sequential Bayesian structure from motion," in *International Conference on Robotics and Automation*. IEEE, 2009, pp. 403–408.

[12] M. Li, H. Yu, X. Zheng, and A. I. Mourikis, "High-fidelity sensor modeling and self-calibration in vision-aided inertial navigation." in *International Conference on Robotics and Automation*. IEEE, 2014, pp. 409–416.

[13] J. Civera, D. Bueno, A. Davison, and J. Montiel, "Camera self-calibration for sequential bayesian structure from motion." IEEE, 2009, pp. 403–408. [Online]. Available: http://dx.doi.org/10.1109/ROBOT.2009.5152719

[14] M. Li, H. Yu, X. Zheng, and A. I. Mourikis, "High-fidelity sensor modeling and self-calibration in vision-aided inertial navigation," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 409–416.

[15] N. Keivan and G. Sibley, "Constant-time monocular self-calibration," *Robotics and Biomimetics (ROBIO)*, pp. 1590–1595, 2014.

[16] G. Klein and D. Murray, "Parallel Tracking and Mapping for Small AR Workspaces," in *International Symposium on Mixed and Augmented Reality*. IEEE, 2007, pp. 225–234.

[17] N. Keivan and G. Sibley, "Asynchronous adaptive conditioning for visual-inertial SLAM." *International Journal of Robotics Research*, vol. 34, no. 13, pp. 1573–1589, 2015.

[18] M. Li and A. I. Mourikis, "3-D motion estimation and online temporal calibration for camera-IMU systems," in *International Conference on Robotics and Automation*. IEEE, 2013, pp. 5709–5716.

[19] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale, "Keyframe-based visual–inertial odometry using nonlinear optimization," *International Journal of Robotics Research*, vol. 34, no. 3, pp. 314–334, Mar. 2015.

[20] J. Civera, A. J. Davison, and J. M. M. Montiel, "Inverse Depth Parametrization for Monocular SLAM," *Transactions on Robotics*, vol. 24, no. 5, pp. 932–945, 2008.

[21] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon, "Bundle Adjustment - A Modern Synthesis." *Workshop on Vision Algorithms*, vol. 1883, no. Chapter 21, pp. 298–372, 1999.

[22] T.-C. Dong-Si and A. I. Mourikis, "Estimator initialization in vision-aided inertial navigation with unknown camera-IMU calibration," in *Intelligent Robots and Systems*. IEEE, 2012, pp. 1064–1071.

[23] L. Carlone, R. Tron, K. Daniilidis, and F. Dellaert, "Initialization techniques for 3D SLAM: A survey on rotation estimation and its use in pose graph optimization." in *International Conference on Robotics and Automation*. IEEE, 2015, pp. 4597–4604.

[24] C. Forster, M. Pizzoli, and D. Scaramuzza, "SVO: Fast semi-direct monocular visual odometry." in *International Conference on Robotics and Automation*. IEEE, 2014, pp. 15–22.

[25] C. Mei, G. Sibley, M. Cummins, P. Newman, and I. Reid, "RSLAM: A System for Large-Scale Mapping in Constant-Time Using Stereo," *International Journal of Computer Vision*, vol. 94, no. 2, pp. 198–214, June 2010.