Motion-Based Calibration of Multimodal Sensor Extrinsics and Timing Offset Estimation

Zachary Taylor and Juan Nieto

***Abstract*—This paper presents a system for calibrating the extrinsic parameters and timing offsets of an array of cameras, 3-D lidars, and global positioning system/inertial navigation system sensors, without the requirement of any markers or other calibration aids. The aim of the approach is to achieve calibration accuracies comparable with state-of-the-art methods, while requiring less initial information about the system being calibrated and thus being more suitable for use by end users. The method operates by utilizing the motion of the system being calibrated. By estimating the motion each individual sensor observes, an estimate of the extrinsic calibration of the sensors is obtained. Our approach extends standard techniques for motion-based calibration by incorporating estimates of the accuracy of each sensor's readings. This yields a probabilistic approach that calibrates all sensors simultaneously and facilitates the estimation of the uncertainty in the final calibration. In addition, we combine this motion-based approach with appearance information. This gives an approach that requires no initial calibration estimate and takes advantage of all available alignment information to provide an accurate and robust calibration for the system. The new framework is validated with datasets collected with different platforms and different sensors' configurations, and compared with state-of-the-art approaches.**

***Index Terms*—Calibration and identification, extrinsics, field robots, timing offset.**

I. Introduction

DUE to advances in technology, the price, size, and power requirements of a large range of sensors have begun to rapidly decrease. When this is combined with the steady improvements in the capability of the sensors, it is now becoming common for multiple sensors of different modalities to be used for intelligent perception tasks. If the outputs of these sensors are combined, they can provide a far richer view of the world than could be given by any one sensor alone. An example of this is shown in Fig. 1, where the outputs of a lidar, camera, and inertial navigation system/global positioning system (INS/GPS) are fused to create a colored 3-D representation of the world.

Before these different sensor modalities can be combined in a meaningful way, the extrinsic transformation between the sensors must be found. Traditionally, these sensors were manually calibrated by either placing markers in the scene or by hand labeling control points in the sensor outputs. These types of methods have significant limitations. They suffer from being time consuming, are labor intensive, and, in some circumstances, give results that contain significant error [1].

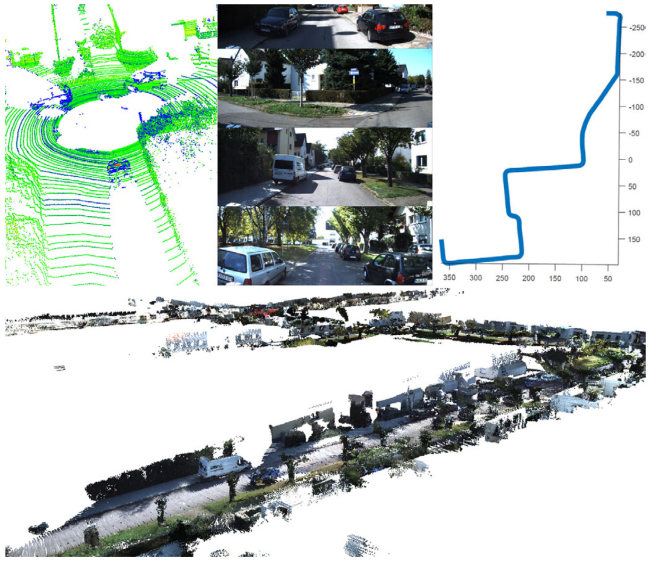

Fig. 1. Data taken from a section of the KITTI dataset. (Top left) One of the point clouds generated by the vehicles 3-D lidar. (Top Middle) View from one of the vehicles mounted cameras. (Top Right) INS/GPS plot of the vehicles movements. (Bottom) Three-dimensional map generated by fusing the sensors using calibration values provided by our approach.

To address these issues, the last few years have seen a range of markerless automatic calibration methods for finding the extrinsic parameters between 3-D lidars and cameras mounted to mobile vehicles [1]–[4]. These methods significantly improve upon the hand-labeling methods. However, to calibrate the systems in a reliable manner, the metrics require an initial estimate to the solution and a constrained search space [3]. These constraints are required, as repeated and similar structures in the environment, in combination with the limited field of view of the sensors, result in the metrics cost function being nonconvex with a large number of local optima. These metrics are also limited to only operating between 3-D lidar and cameras, as well as requiring overlapping fields of view and the exact timing between the sensors to be known.

While the constraint of requiring an accurate initial guess for the system poses little challenge for most robotics experts, systems utilizing these sensors are beginning to see significant use by end users in fields such as agriculture and mining. If these systems then have additional sensors mounted, require significant repairs, or undergo modification, detailed knowledge of the sensors' operating principles, transformation systems, and

Manuscript received March 7, 2016; accepted June 12, 2016. Date of publication August 29, 2016; date of current version September 30, 2016. This paper was recommended for publication by Associate Editor F. Kanehiro and Editor C. Torras upon evaluation of the reviewers' comments. This work was supported by the Rio Tinto Centre for Mine Automation, Australian Centre for Field Robotics, University of Sydney.

The authors were with the Rio Tinto Centre for Mine Automation, Australian Centre for Field Robotics, University of Sydney, Sydney, NSW 2006, Australia. They are now with the Autonomous Systems Laboratory, ETH Zurich, 8092 Zürich, Switzerland (e-mail: ztaylor@ethz.ch; jnieto@ethz.ch).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TRO.2016.2596771

timing delays is required to perform the calibration the system will require before it can begin performing tasks. This limits the versatility and ease with which an end user can utilize their robotic platforms.

This paper aims to remove the need for an initial guess or parameter tuning and, therefore, reduce the prerequisite knowledge needed to calibrate such a system, while providing calibrations of comparable quality to existing more laborious techniques. It accomplishes this by looking at additional cues that can be used to calibrate these systems, specifically their motion. Motion-based calibration in the form of "hand–eye" calibration is a well-established technique for calibrating robotic arm mounted cameras [5]. This paper extends existing hand–eye calibration approaches and incorporates them into a new probabilistic framework that can be used to calibrate mobile system's sensors.

The system operates by examining the motion of individual sensors as well as the uncertainty in their readings. Unlike methods that rely on intersensor similarities in the scene's appearance, hereafter referred to as appearance-based methods, the system requires no initial estimate of the calibration parameters. Also unlike standard hand–eye calibration methods, the system is able to simultaneously calibrate any number of sensors, is robust to outliers, utilizes uncertainty information, and requires no form of markers or other calibration aids. Importantly, due to its consideration of the uncertainty of the inputs, the system can also provide an estimate of the accuracy of its resulting calibration. Finally, where overlap exists in the sensor's fields of view, the system utilizes the motion-based calibration to constrain an appearance-based refinement step to further increase its accuracy. This allows our approach to take full advantage of all motion and appearance cues present in the scene.

This paper builds upon and expands our previous work in [6] providing significant improvements in our method of variance estimation, outlier handling, probability estimation, and extending the method to consider timing offset. The code used to generate the results found in this paper is publicly available at http://www.zjtaylor.com/code.

Specifically, this paper presents the following contributions.

1) A pipeline for extrinsic sensor calibration and timing offset estimation that requires no vehicle or situation specific parameters. This calibration can operate on data a system records as it operates in an arbitrary environment.
2) A probabilistic formulation of the hand–eye calibration problem that incorporates the uncertainty of each sensor's readings can be applied to multiple sensors simultaneously and allows the uncertainty in the final calibration to be evaluated.
3) The use of motion-based calibration to constrain and guide appearance-based calibration techniques.
4) The development of a new multimodal metric, the intensity-motion (IM) metric for aligning lidar with cameras. This metric is designed for use with mobile vehicle-based systems and utilizes both appearance and motion characteristics.
5) Extensive evaluations of all methods presented and comparisons with state-of-the-art algorithms.

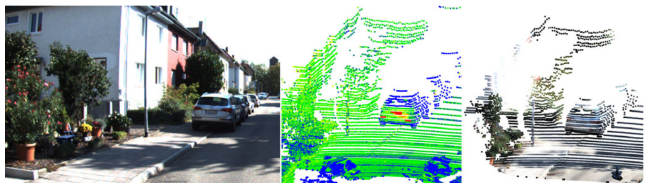

Fig. 2. Example of the type of data the markerless lidar camera methods typically operate on. (Left) Image taken by the KITTI sensor vehicle. (Middle) Velodyne scan of the same area. (Right) Image projected onto the scan after alignment.

II. Related Work

The work related to our method can be divided into two separate areas: the markerless appearance-based calibration of 3-D lidar with cameras and the motion-based calibration of sensors.

A. Appearance-Based Metrics

These metrics calibrate mobile 3-D lidar scanners to operate with cameras based on the appearance of the surroundings. An example of the data these methods typically operate on is shown in Fig. 2.

One of the first approaches that did not rely on markers was presented in [1]. Their method operates on the principle that depth discontinuities detected by the lidar will tend to lie on edges in the image. Depth discontinuities are isolated by measuring the difference between successive lidar points and removing points with a depth change of less than 30 cm. An edge image is produced from the camera that is then blurred to increase the capture region of the optimizer. The two outputs are combined by projecting the isolated lidar points onto the edge image and multiplying the magnitude of each depth discontinuity by the intensity of the edge image at that point. The sum of the result is taken, and a grid search is used to find the parameters that maximize the resulting metric.

Two very similar methods have been independently developed in [2] and [4]. These methods use the known intrinsic values of the camera and estimated extrinsic parameters to project the lidar's scan onto the camera's image. The mutual information value is then taken between the lidar's intensity of return and the intensity of the corresponding points in the camera's image. When this value is maximized, the system is assumed to be perfectly calibrated. Both implementations utilize local optimizers and require aggregation of a large set of scans to converge to the global maximum.

An approach by Tamas and Kato [7] that finds the correspondence between planar regions has been used to obtain accurate calibration between 3-D lidar and camera images.

An approach has been developed in [8] for registering a push broom 2-D lidar with a camera. To form an image from the 2-D scanner, its scans are first combined with an accurate navigation solution for the mobile system to generate a 3-D scan. A 2-D image is then produced from this 3-D scan using a camera model. The two images have the magnitude of gradients present in them calculated and normalized over a small patch around them. The camera and lidar are assumed to be aligned when the

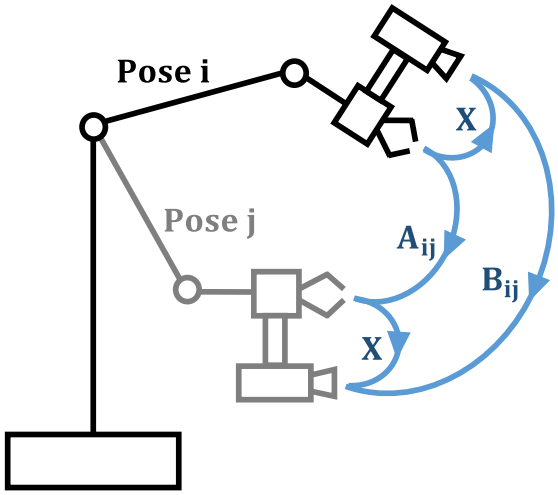

Fig. 3. Standard hand–eye calibration problem. A robotic arm is moved through a series of poses. By examining the transformation experienced by the robotic hand A and the Camera (aka the "Eye") B, the transformation between the camera and hand X can be recovered using the relationship $AX = XB$.

sum of the differences in these gradient magnitude images is minimized.

B. Motion-Based Metrics

The hand–eye calibration problem is a well-known problem in robotics [5], [9], [10]. It is usually expressed as follows: If two points are rigidly connected and both points undergo a series of known transformations, how can the transformation between the two points be recovered? The name is derived from one of the first robotic applications of the problem, where a camera was mounted onto the "hand" of a robotic arm and the transformation between the "hand" and the camera or "eye" needed to be calculated. The problem is also sometimes referred to as $AX = XB$, as if A and B are the transformations the two sensors undergo, and X is the transformation between them. A depiction of the calibration process for a robotic arm is shown in Fig. 3.

In its most basic form, the problem is well understood, with techniques developed in the late 1980 s giving efficient methods that are optimal in the least-squared sense [5]. These methods operate by first finding the rotation axis for each transform. All points on a rigid body share the same rotation axis; therefore, this can be found before the translation is known. Once these axes are found, they can be optimally aligned using an approach known as the Kabsch algorithm [11]. After the rotation has been found, the translation can be obtained using simple linear algebra.

These techniques have the disadvantage of requiring calibrated markers to be placed in the scene to allow the camera's transformations to be calculated. More recently, several methods have made use of visual odometry methods to remove this limitation [12], [13]. As visual odometry utilizing a monocular camera has no scale associated with it, this additional parameter must be estimated for every transformation pair.

Another recent modification of the problem was given in [14], in which a solution is calculated for two sensors that operate asynchronously and provide data at undefined intervals. In this approach, the fact that the magnitude of a rotation is constant for all points on a rigid body is exploited. This magnitude allows the authors to find the most likely correspondences of the data.

The problem of timing calibration was also examined by Kelly and Sukhatme [15] who calibrated timing offsets at the same time as the six-degree-of-freedom extrinsics for the sensors of a PR2 robot. A checkerboard was used to allow observation of the motion of the monocular camera sensor.

This technique has also been adapted for real-time applications, for example, a method that makes use of a Kalman filter to calibrate an INS system with a stereo camera rig was presented in [16].

Several authors have looked into using motion-based techniques to solve the problem of IMU–camera calibration, with many focusing on the use of low-cost smart phone sensors. Ovren and Forssen [17] used motion to find the time offset between a camera and IMU. Fleps *et al.* [18] used a batch optimization system that found the timing offset and extrinsic transformation. While in theory their approach could work with natural features, in all their real-world tests, they used chessboards and corner detection. Keivan and Sibley [19] used an IMU to calibrate the focal length and center point of a camera. The process also detected if the calibration parameters undergo significant change and in this event recalibrate. A phone IMU–camera calibration approach was presented in [20]. In this method, a large number of parameters are estimated in addition to the extrinsics to account for the imperfect nature of the phone's low-cost sensors.

Underwood [21] utilized the motion of a vehicle to calibrate a GPS/INS to operate with 2-D lidar. To overcome the inherent observability issues of using a 2-D lidar, they constrained the environment to a field with a single vertical pole that was used as a target. The authors also considered the variance of the resulting calibration.

In [22], four cameras with nonoverlapping fields of view are calibrated on a vehicle. The method operates by first using visual odometry in combination with the car's motion provided by odometry, to give a coarse estimate of the cameras' position. This is refined by matching points observed by multiple cameras as the vehicle makes a series of tight turns. Bundle adjustment is then performed to refine the camera position estimates. The main limitation of this method is that it was specifically designed for vision sensors and makes use of feature matching between multiple sensors to refine the calibration.

III. Relating Sensor Motion to Offset

Before we can utilize sensor motion to calibrate a system, we must first explore how this motion is related to the sensors' parameters. For a mobile vehicle with rigidly mounted sensors, Fig. 4 shows a depiction of how each sensor's motion and relative position are related. As the vehicle moves, the transformation between any two sensors x and y at timestep k can be recovered by using

$$T_y^x T_{y,k}^{y,k-1} = T_{x,k}^{x,k-1} T_y^x. \quad (1)$$

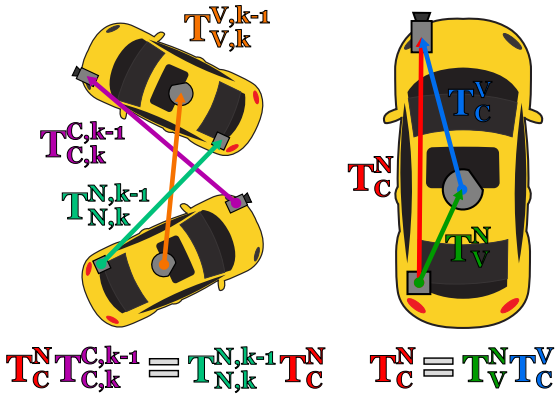

Fig. 4. Diagram of a car with a camera (C), Velodyne lidar (V), and navigation sensor (N). The image on the right shows the three sensors positions on the vehicle. The image on the left shows the transformation these sensors undergo at timestep k.

Our objective in performing the motion-based calibration of a system is to find the values of the intersensor rotation R^x_y, translation t^x_y, and timing offset τ^x_y for which (1) holds. In order to achieve this, we begin by first examining (1) in terms of its rotational and translational components. This yields the following two equations:

$$R^x_y R^{y,k-1}_{y,k} = R^{x,k-1}_{x,k} R^x_y \tag{2}$$

$$R^x_y t^{y,k-1}_{y,k} + t^x_y = R^{x,k-1}_{x,k} t^x_y + t^{x,k-1}_{x,k}. \tag{3}$$

Importantly, if the terms of the rotational component [see (2)] are examined, we see that this equation is not dependent on the translational offset. This allows the rotational component R^x_y to be found before the translational component t^x_y is known, decoupling these two terms.

While the above figure and equations assume that no time offset is present in the sensor readings, this assumption is not required. If we take two sensors that were started independently and assign variables a and b to represent their individual timesteps, then (2) becomes

$$R^x_y R^{y,b-1}_{y,b} = R^{x,a-1}_{x,a} R^x_y \tag{4}$$

which for nonconstant rotational velocities only holds for

$$\tau^x_y = b - a. \tag{5}$$

Furthermore, the magnitude of the angle through which a rotation matrix rotates a point is independent of the frame of reference. Thus, if the angular magnitude of the rotation is given by θ, the relationship can be expressed as

$$\tau^x_y = b - a \Rightarrow \theta^{y,b-1}_{y,b} = \theta^{x,a-1}_{x,a}. \tag{6}$$

This equation is now completely independent of the extrinsic transformations between the sensors, only depending on the timing offset τ^x_y. This allows the timing offset between the sensors to be found before any extrinsic transformation information is known.

Equations (2), (3), and (6) are the standard equations that many authors have made use of in aligning sensor data from motion. If the assumption of a rigid mounting holds and the sensors undergo sufficient nondegenerate motion, these equations will give the unknown offset between the sensors.

However, due to several factors, by far the most significant being noise in the sensor motion estimates, the observed values will not perfectly conform to these equations. Typically, to account for these differences in readings, an approach that finds the least-squares error for the system of equations is used. While this can be an effective approximation method, it implies that every sensor reading was made with equal accuracy. Furthermore, if the approach was extended to consider more than two sensors, it would imply that each sensor's readings were of equal worth.

In a constrained environment that has been designed to calibrate a specific sensor set, this assumption of equal accuracy in each sensor reading may be approximately true. However, if we wish to calibrate a system composed of a wide range of sensors, as it moves through an unconstrained environment, these assumptions will detrimentally affect our system. Issues such as an RTK GPS losing connection with its base-station, or a visual-odometry system encountering an area with insufficient near-field objects to accurately estimate its translation, must be accounted for. In order for a system to handle a wide range of sensors, as well as the range of possible situations an unconstrained environment will provide, it must be able to reason about the value of each reading it receives.

To account for this, we reformulate the calibration problem to one that not only considers the value of the sensor readings, but also the confidence that each sensor has in its values. This allows the formation of an approach that utilizes the evidence given by each sensor reading and reasons as to the most probable sensor configuration. Importantly, as this system takes into consideration the accuracy of each sensor reading and the effect it has on the resulting system, it is also able to give an accurate assessment of its confidence in the final configuration. We see this as one of the key advantages of our approach, as it allows a user or robotic system to know the confidence it can place in the obtained configuration.

IV. Overview of the Approach

Our approach can be divided into the following stages

1) *Estimation of individual sensor motion:* Each sensor output is used to generate an individual estimate of the motion the sensor has undergone as well as an estimate of the uncertainty in this transformation.
2) *Estimation of offsets from motion cues:* The timing offset between the sensors is found first, followed by the rotational offset, then finally the translational offset. Each of these offset calculations can be further divided into four distinct stages.
    a) *Initialization:* An efficient globally convex process is applied to a simplified system to provide an initial estimate for the offset parameters.
    b) *Likelihood estimation:* An equation describing the likelihood of the offset parameters is formed.
    c) *Optimization:* The given likelihood equation is used to find the optimal offset parameters for the system.

d) *Uncertainty estimation:* The uncertainty in the parameter estimates is found and applied to the system.

3) *Utilizing appearance to refine the calibration:* When sensors have overlapping fields of view, the appearance of the environment in these overlapping regions may be utilized to further enhance the estimated sensor calibration. In this step, we make use of camera–camera matching and a newly proposed camera–lidar metric to refine the final calibration.

V. Estimation of Individual Sensor Motion

To represent the sensor motion, we utilize a six-element vector $[x, y, z, rx, ry, rz]$. The first three elements give the translational component of the transform. The next three elements are an angle–axis representation of the rotation. To represent uncertainty, we assume that the six elements under consideration are approximately independent, giving a second six-element vector containing the associated uncertainty.

To facilitate the motion-based calibration process described above, individual sensor motions, as well as an estimate of their uncertainty, is required. As different sensor modalities provide different types of information, the approach used for motion estimation is sensor dependent. The following subsections explain the approaches we used to obtain sensor motion for the different sensor modalities utilized in our experiments.

A. Three-Dimensional Lidar

To calculate the transform from one scan to the next, a point-to-plane iterative closest point algorithm is used. In this implementation, we also ignore points with large errors. In our implementation, we used 0.2 m as the threshold. The movement of the lidar during the scan is compensated for by using the previously estimated motion of the sensor to find the position of the points relative to the scanner head when 50% of the scan had been completed.

To estimate the variance of the calculated transformation vector, we make use of the approximate delta method that will be examined in Section VI-D. For this method, while Velodyne specifies that the lasers have a standard deviation in their range of approximately 2 cm, we use a σ of 10 cm to account for any imperfections in the intrinsic parameters of the setup.

B. Cameras

The camera transforms are found using a fairly standard visual odometry approach. First, Harris corners are detected in an image before Lucas–Kanade optical flow is used to find the location of these points in the next image. A MAPSAC [23] implementation is used to reject outlier points. The inliers and intrinsic parameters are used to find the essential matrix using the eight-point normalized algorithm [24]. The essential matrix is finally used to find the transformation matrix between the positions. As the only sensor employed is a monocular camera and its position on the vehicle is unknown, no sense of absolute scale of the movements can be obtained. An estimate of the transformations variance is computed utilizing the same modified delta approximation method utilized in computing the lidars variance.

C. Inertial Navigation System/Global Positioning System

GPS/INS sensors give the transformations directly, and the variance of the sensor is given either by the manufacturer as a fixed value or, in the case of the translation, is given as one of the outputs of the sensor. In order to convert the variance from the given roll, pitch, and yaw values into the form used by our approach, a simple Monte Carlo sampling approach was used.

VI. Estimation of Offset From Motion

A. Initialization

Our decision to consider all of the sensors in the system in the calculation of the likelihood estimation results in a measure that, while generally convex in a large region around the optimal solution, is not guaranteed to be globally convex. This means if we wish to optimize the system using an efficient local optimization strategy, we must provide the system with an initial estimate of the calibration parameters that is within the locally convex region of the global optima. Asking the user to provide this estimate would detract from our goal of a system that can be calibrated by a nonexpert user.

Instead to overcome this issue, we initialize the system by using a simplified approach that does not suffer these same convexity issues. The solution this approach provides, while less accurate, will generally be well within the locally convex region of the full metric, solving the initialization issue without requiring user input. The process of initialization for each set of offset parameters is as follows.

1) Timing: For most practical setups, only small timing offsets will be present in the system. Because of this, the system can simply be initialized assuming zero offset between the sensors. However, for timing offset, a second issue exists.

The presented process for finding the timing offset between the sensors assumes that the sensors obtain readings simultaneously, and the unknown offset is given as a discrete number of timesteps. In practice, however, sensor readings are generally obtained in an asynchronous manner.

To compensate for this, we apply the current estimate for the timing offsets to the data before interpolating it at n equally spaced intervals to create synchronous data. In our implementation, $n = 10\,000$ steps were chosen, as this was found to give a suitable balance between accuracy and runtime.

Once the timing offsets have been found, synchronized sensor-motion readings are obtained by interpolating the motion at the times when the slowest updating sensor obtained readings, removing this issue from the following stages. At this stage, some simple outlier rejection was also performed by removing timesteps where the sensor angular magnitude readings differed by over ten standard deviations.

2) Rotational: As stated above, in the case of the rotational offset, the parameters used to initialize the optimization framework are found by simplifying our treatment of the problem. This allows an efficient analytical approximate solution to the system to be obtained.

In this approximation, one of the sensors is first arbitrarily designated to be the base sensor B. We then proceed to calculate the approximate rotation between this sensor B and each other sensor i, R_i^B. This is accomplished through the use of a slightly modified version of the Kabsch algorithm [11]. The Kabsch algorithm is an approach that calculates the rotation matrix between two sets of vectors providing the least-squared error. In our approach, we modify the algorithm to give nonequal weighting to each sensor reading. The weighting used is the sum of the diagonal of the covariance matrix of the rotation axis.

3) Translational: As in the case of the rotation, the translational initialization operates by simplifying the variance estimation and only considering a single pair of sensors to allow the formation of an efficient analytical approximation. If we take (3) and combine it with information from n other timesteps, this gives the following system of equations:

$$t_y^x = \begin{bmatrix} R_{y,2}^{y,1} - I \\ R_{y,3}^{y,2} - I \\ \vdots \\ R_{y,n}^{y,n-1} - I \end{bmatrix}^{-1} \begin{bmatrix} R_y^x t_{x,2}^{x,1} - t_{y,2}^{y,1} \\ R_y^x t_{x,3}^{x,2} - t_{y,3}^{y,2} \\ \vdots \\ R_y^x t_{x,n}^{x,n-1} - t_{y,n}^{y,n-1} \end{bmatrix}. \quad (7)$$

The only unknown here is t_y^x. Solving this system of equations will yield a least-squares estimate for the translation and is the approach taken by Tsai and Lenz [5]. Just as in the case of rotation, we can improve the estimate of t_y^x given by this equation by weighting each of the terms with a value derived from taking the inverse of a simplified sum of its variance.

In cases where one of the sensors is a monocular camera, the motion is only given up to scale ambiguity. Because of this, the scale of the motion at each timestep must also be estimated. One possible approach to solving this problem is simultaneously solving for the scale terms s_k. However, if n is the number of sensor readings, the calculation of this approach requires the inversion of a $3n$ by the $n+3$ matrix. This inversion presents a significant computational load in situations where several thousand estimates are made. As this calculation is only used to initialize a second refinement stage, exact accuracy of the solution is not required, and therefore, to ease the difficulty of computation, a rough approximation is made. We assume for the sake of this initial calculation that the offset between the sensors is approximately 0. With this approximation in place, the scale can be estimated as

$$s_k \approx R_y^x t_{x,k}^{x,k-1} (t_{y,k}^{y,k-1})^{-1}. \quad (8)$$

We wish to clearly note here that this scale approximation is only used for this one equation, with all other stages considering the offset when estimating the scale.

B. Likelihood Estimation

Thanks to the previous stages of the approach, we now have estimated sensor transforms with their associated uncertainty and an initial estimate of the calibration parameters. Before we can use these to find the optimal calibration, we must develop a measure that indicates the likelihood of a set of parameters being correct for a given set of measurements. This is accomplished by reformulating the basic hand–eye equations to consider the uncertainty in the measurements in a probabilistic manner.

1) Timing: To find the likelihood of a given timing offset, we find the angle θ through which each sensor rotates along with its associated uncertainty. The rotation of two sensors x and y at timestep k is linked via (6). Using this and modeling the sensor observations by their first and second moments, the relative likelihood of these sensor observations for this system is given via

$$\mathcal{L}_\theta(x,y,k) = \frac{\exp\left(-\frac{(\theta_{x,k}^{x,k-1} - \theta_{y,k}^{y,k-1})^2}{2(\mathrm{cov}(\theta_{x,k}^{x,k-1}) + \mathrm{cov}(\theta_{y,k}^{y,k-1}))}\right)}{\sqrt{2\pi(\mathrm{cov}(\theta_{x,k}^{x,k-1}) + \mathrm{cov}(\theta_{y,k}^{y,k-1}))}}. \quad (9)$$

This formulation is equivalent to the evaluation of the probability distribution function of a Gaussian distribution. The distribution is formed using the difference between the two sensor readings for the angle and is evaluated at 0.

2) Rotational: Equation (2) gave the relationship between sensor movement and their rotational offset. However, due to our use of an angle–axis representation of the sensor rotations, this equation can be simplified. If A is our rotation vector, then the sensor rotations are related by

$$A_{y,k}^{y,k-1} = R_y^x A_{x,k}^{x,k-1}. \quad (10)$$

From this starting point, we perform a process similar to that done for the timing offset. We first utilize the above equation to find the error present for the readings of two sensors x and y at timestep k. This is expressed as

$$R_{\mathrm{err}}(x,y,k) = A_{y,k}^{y,k-1} - R_y^x A_{x,k}^{x,k-1} \quad (11)$$

with associated variance

$$\mathrm{cov}(R_{\mathrm{err}}(x,y,k)) = \mathrm{cov}(R_{y,k}^{y,k-1}) + R_y^x \,\mathrm{cov}(R_{x,k}^{x,k-1})\,(R_y^x)^T. \quad (12)$$

These equations are then combined to give the likelihood of the readings for the given system:

$$\mathcal{L}_R(x,y,k) = \frac{\exp\left(-\frac{R_{\mathrm{err}}(x,y,k)^2}{2\,\mathrm{cov}(R_{\mathrm{err}}(x,y,k))}\right)}{\sqrt{2\pi\,\mathrm{cov}(R_{\mathrm{err}}(x,y,k))}}. \quad (13)$$

This is equivalent to evaluating the likelihood of a Gaussian distribution with mean R_{err} and covariance $\mathrm{cov}(R_{\mathrm{err}}(x,y,k))$ at 0.

3) Translational: The translational offset is related to the sensor motion and rotational offset via (3). Again, taking the error present for two sensors at timestep k, we can calculate the error in the system as

$$t_{\mathrm{err}}(x,y,k) = (R_{y,k}^{y,k-1} - I)t_y^x + t_{y,k}^{y,k-1} - R_y^x t_{x,k}^{x,k-1}. \quad (14)$$

However, due to the interaction of multiple variables containing uncertainty, the variance of the above system cannot be given using a simple analytical combination of the components. We overcome this issue by making use of the delta approximation [25] to estimate $\mathrm{cov}(t_{\mathrm{err}}(x,y,k))$.

Once this approximation has been performed, this information is again used to give the likelihood of the readings via

$$\mathcal{L}_t(x,y,k) = \frac{\exp\left(-\frac{t_{\text{err}}(x,y,k)^2}{2\ \text{cov}(t_{\text{err}}(x,y,k))}\right)}{\sqrt{2\pi\ \text{cov}(t_{\text{err}}(x,y,k))}}. \quad (15)$$

In the case of monocular cameras, unless further assumptions about the system are made, the translational motion estimates will be normalized with no sense of scale. In the case of a system solely comprised of cameras, this prevents the calculation of the translational offset. In any other setup, however, we are able to utilize the camera's motion to give the offset. This is done through the introduction of a scale term s_k for the translation that is estimated at the same time as the translational offset t_y^x. If in this instance sensor y is the camera, (14) would become

$$t_{\text{err}}(x,y,k) = (R_{y,k}^{y,k-1} - I)t_y^x + s_k t_{y,k}^{y,k-1} - R_y^x t_{x,k}^{x,k-1}. \quad (16)$$

With this new term for estimating the error in the modeled translation in place, all other steps in the translation offset likelihood estimation proceed in the same manner.

While the approach given allows for the estimation of the translational offset between two cameras through two scaling parameters (assuming that at least one sensor with absolute scale is also present in the system), in practice, these estimates are not utilized. This is done as the two unknown parameters present in each timestep result in the translation of two cameras relative to each other conveying minimal additional information about the system. As in most systems, cameras are the most numerous sensor; considering these correspondences also greatly increases the computation time of the system.

C. Optimization

The above processes give the likelihood of a pair of sensor readings fitting our rigid sensor model, for a given set of offsets. We now combine all this information into a measure of the overall system likelihood and optimize to find the maximum-likelihood estimate for the calibration.

1) Combining Likelihood Estimates: Assuming that each sensor reading is independent of the others, the joint likelihood of scans can be found by multiplying together the likelihood functions. In practice, for numerical convenience, this is done through the equivalent operation of summing log-likelihoods.

Finding the parameters that maximize each of these pairwise log-likelihood sums would result in a series of pairwise offsets between the sensors. Unfortunately, due to sensor noise, it is unlikely that these estimates would form a consistent system. That is, if there are three sensors x, y, and z, then for this process, $T_y^x T_z^y \neq T_z^x$. To prevent this issue from occurring, we consider the likelihood of all possible pairwise offsets simultaneously, summing their pairwise log-likelihoods to give a single measure of the likelihood for the entire system. This likelihood should be maximized when the correct timing, rotation, and translation parameters are utilized.

More formally, if the system is comprised of n sensors each making m readings, our objective is to find the parameters that maximize the following expression:

$$\sum_{x=1}^{n}\sum_{y=1}^{n}\sum_{k=1}^{m} \log \mathcal{L}(x,y,k) \quad (17)$$

for each of the likelihood expressions given by (9), (13), and (15).

The above approach is an optimal strategy when the covariance is Gaussian and known exactly. However, in some circumstances, the estimated covariance can form a poor representation of the true error present in an estimate. Cases where the variance has been overestimated have little impact on the system other than to undervalue some information. However, points where the variance has been greatly underestimated will have a large impact on a method's accuracy and reliability. These points are deemed to be outliers, and their impact must be mitigated to allow the system to converge to a correct solution.

In our implementation, we made used of the trimmed means outlier rejection strategy. Trimmed means (also known as truncated means) is a process via which the data are sorted and a set ratio of the worst performing data is labeled as outliers and discarded before the mean of the remaining data is taken. This trimmed value is used when log-likelihood readings are summed to generate the overall likelihood in the optimization process. In our implementation, 25% of the data were rejected. Note that the framework is not restricted to any particular outlier rejection approach, and other outlier insensitive methods can also be utilized.

2) Finding the Maximum Likelihood Solution: The overall system is optimized using the Nelder–Mead simplex optimization strategy. This optimization was used as it was found to reliably converge to the maximum in an efficient manner.

When monocular cameras are utilized, a scale term must be estimated for every timestep. To accomplish this at each function evaluation, (16) is rearranged to solve for the scale parameter. This estimate and its variance, in combination with the other parameters, are then used to assess the likelihood of the system's configuration.

D. Uncertainty Estimation

1) Calculating the Uncertainty of the Final Calibration: After the optimization has located the most likely parameters, the uncertainty in this estimate is found. This is calculated by taking the maximum values obtained from two strategies. We first make use of the Cramer–Rao lower bound. This estimate is a lower bound, however, and in many situations was found to be optimistic about the actual accuracy of our readings. To overcome this issue, we combine it with a second method, an approximation to the delta method [26].

The delta method is a simple and powerful method for approximating the variance of a function. It operates by forming a first-order approximation to the function at the point of interest and finding the variance of this approximation. Let $z = f(x)$, where f is an arbitrary function relating two variables x and z; the delta method gives an approximation to the covariance of

this equation by

$$\operatorname{cov}(z) \approx \frac{df}{\partial x} \operatorname{cov}(x) \frac{df}{\partial x}^{T}. \tag{18}$$

As this method is only used when no analytical or simple approximation to the variance exists, the analytical calculation of $\frac{df}{\partial x}$ is typically also intractable. Because of this, we make use of simple finite-difference methods in its calculation. This provides accurate uncertainty estimates in many situations; however, in calculating the uncertainty of the optimization process, the method can be slow or even intractable to calculate. To overcome this, we make use of an approximate form presented in [26]. This method of uncertainty calculation has also been used in several other stages of our approach, such as the estimation of lidar points and the uncertainty in the translational offset likelihood.

2) Propagating Uncertainty: In this calibration process, each stage utilizes the previous stage and thus will be influenced by the accuracy of its solutions. The dependence of the translation on the rotational offset estimation's accuracy is captured in the calculation of $\mathcal{L}_t(x, y, k)$. However, the same is not true of the timing offset. While both the rotational and translational offset equations depend on it, their equations implicitly assume exact correspondence between the scans. To overcome this limitation, we transfer the uncertainty from the time parameter to the sensor motion estimates.

This is accomplished by assuming that the timing offset estimation is sufficiently accurate that the rate of change of the sensor motion between the actual time and the estimated time is roughly constant. That is, if τ represents the timing offset, $\hat{\tau}$ the true value of the timing offset, T the transformation the sensor undergoes, and $\Delta\tau$ the error in our timing offset estimation, then

$$\left.\frac{dT}{d\tau}\right|_{\tau=\hat{\tau}} \approx \left.\frac{dT}{d\tau}\right|_{\tau=\hat{\tau}+\Delta\tau}. \tag{19}$$

With this assumption, the angular and linear rotational velocities can be taken to be approximately constant for this time range. This means that the relationship between timing error and position error can be linearly approximated as

$$\Delta T \approx \Delta\tau \left.\frac{dT}{d\tau}\right|_{\tau=\hat{\tau}}. \tag{20}$$

The variance given by this equation is added to the variance already present in the transformation estimates.

VII. Utilizing Appearance to Refine the Calibration

While the accuracy of the motion-based methods depends on the dataset, motion, and sensors used, through experimentation, several common trends appear. Under all but the most extreme cases, accurate rotation with low uncertainty is generally obtained. The translation estimates, however, show more uncertainty. Generally, for a nonholonomic ground vehicle, the translation will be most accurately estimated in the direction tangential to the ground plane and perpendicular to the vehicle's main direction of motion. This accuracy stems from the large differences in translation the sensors will undergo during a turning maneuver and, for most systems, was in the range of 10–100 mm. The least accurately estimated parameter was the translation in the direction of the normal of the ground plane. This is to be expected as with only planar movement, this parameter would be completely unobservable. This means that this parameter has only been observed through the subtle movements caused by defects in the road surface and the rocking of the system on its suspension. This parameter's accuracy was typically in the range of 100–1000 mm.

With this level of error possible in the translation estimation, the output of the motion calibration provided by near-planar motion is of limited direct practical value. However, as we will show, the motion calibration result is still an important part of our fully automated framework, since it can provide the initialization required by appearance-based calibration methods.

Initially, the true calibration between the sensors could take any parameter in a large 6-D search space. For appearance-based metrics, this search space is typically highly nonconvex and contains a large number of local optima. Therefore, searching the entire search space is error prone and often intractable. Our motion-based calibration estimates and associated variance vastly reduce the feasible regions the solution could lie within. Generally, only one or two parameters still contain large uncertainty, with the remainder constrained to small regions. Because of this, our motion-based calibration is complementary to these appearance-based methods through the initialization and constraining of their search space.

For cameras and 3-D lidar scanners, methods such as those examined in Section II-A can be utilized for the calibration refinement. These metrics operate by examining points at a single timestep; however, we can exploit the already obtained motion information in combination with the appearance information to form a new lidar–camera alignment metric.

A. Lidar–Camera Intensity-Motion Metric

Markerless metrics used to match between lidar scans and camera images either rely on correlating the intensity of the two modalities (MI, NMI) [2] or aligning edges (Levinson's method [1], GOM [3]). The problem with these metrics is that as the two sensors perceive the world in fundamentally different ways, there will be many cases in which the features of interest are not present in both modalities.

The addition of motion information greatly simplifies our problem as it now becomes possible to align lidar and cameras using only monomodal matching. As the metric used to achieve this alignment makes use of both intensity cues and motion cues in its development, we will henceforth refer to it as the IM metric.

In this metric, a camera image and its corresponding lidar scan are found. The motion information estimated by the lidar is then used to transform the lidar's point cloud to the position it was in relative to the camera at the time when its image was taken. Once this has been done, the camera image is projected onto the lidar scan giving each of the lidar points an associated color. The same lidar scan is then matched to the next camera image with the same process of using the motion information to compensate

for the timing difference. This image is again projected onto the lidar scan. This process has resulted in two different images being used to give color information to the same lidar scan. If we assume constant lighting conditions, camera settings, and a static environment, it would then be expected that, if the camera offset was correct, both possible colorings would be identical. Because of this, we can take the mean-squared difference in their intensity as an indication of the sensor's alignment. Minimizing this error should result in the correct lidar–camera offset.

While in this simple form, the metric will operate well in many scenarios to increase its reliability, several changes are made. First, a slight Gaussian blur is applied to the image, as this was found to increase the robustness of the method to noise and provide a smoother search space increasing an optimizer's likelihood of correctly converging to the true offset.

The second change made is to help minimize the impact that occlusions will have on the image. As the two images used are taken from two different perspectives, occlusions will impact the alignment. These occlusions will also occur due to the difference in the location of the lidar and camera. While a ray-tracing process could be implemented to calculate which points are occluded at each estimation step, this is a computationally expensive process. Instead, we opt for a simple strategy that preemptively removes points that will have a high chance of becoming occluded during the sensor calibration. In this process, we first project the points onto a sphere and find the 50 closest neighbors for each point. The neighboring point closest to the camera is found, and the distance between these points is found. This distance is then divided by the distance from the sensor to the points. If the final value is greater than a threshold (in our case set to 0.1), the point is rejected as having a high probability of being occluded.

The final and most significant change is to operate on a gradient image. By using a gradient image, the importance of a metric correctly aligning the boundaries between objects is significantly increased. It also allows us to remove regions of low texture by removing points with a gradient of zero. Removing these points prevents the metric from attempting to align the entire scan with a low-texture region of the image, such as the sky.

The metric also allows for an estimation of its accuracy. This is done by once again making use of the delta method on the optimization of the metric, with the variance of the motion used to find the variance in the output parameters.

B. Lidar–Camera Optimization

When optimizing these methods, we wish to make use of both the estimated solution and its associated variance provided by the motion estimation process. As we are operating with an appearance metric, we also wish to make use of a global optimizer, as while significantly constrained, the metric may still have multiple optima within the feasible search space. To meet these requirements, we make use of the CMA-ES optimization technique [27]. This technique randomly samples the search space using a multivariate normal distribution that is constantly updated. This optimization strategy works well with our approach, as the initial normal distribution required can be set using the variance from the motion-based estimation. This means that the optimizer only has to search a small portion of the search space and can rapidly converge to the correct solution.

C. Camera–Camera Optimization

If both of the sensors are cameras and they have overlapping fields of view, the calibration may be refined using simple monomodal matching. This is done by detecting and matching features present in the two cameras' field of view. MAPSAC is then used to reject outliers and the normalized transformation between the two cameras is found. The process of forming a transformation estimate from the inliers is bootstrapped to give an estimate of the transformational variance.

D. Combining the Refined Results

The pairwise transformations calculated in the appearance-based refinement step will not represent a consistent set of solutions to the transformation of the sensors. This is due to each pair of sensors obtaining their appearance-based calibration independently. This means, we again face the issue that if there are three sensors x, y, and z, then for the current estimates, $T_y^x T_z^y \neq T_z^x$. The camera-to-camera transformations also contain scale ambiguity.

To correct for this and find a consistent solution, the transformations are combined. This is done by using the calculated parameters to find a consistent transformation that has the highest probability of occurring. We do this by first using the transformations to one of the sensors to generate an initial guess. The probability of this solution occurring, given the transformation and variance of all the pairwise transforms, is calculated and used as a cost function that is optimized using Nelder–Mead simplex optimization.

VIII. Experimental Platforms

Two different platforms were used to evaluate the performance of the approach: the KITTI dataset and the Australian Centre for Field Robotics (ACFR) Shrimp platform.

A. KITTI Dataset Car

The KITTI dataset is a well-known publicly available dataset obtained from a sensor vehicle driving in the city of Karlsruhe, Germany [28]. The sensor vehicle is equipped with two sets of stereo cameras: a Velodyne HDL-64E and a GPS/INS system. This system was chosen for calibration due to the ease of availability and the provided ground truth in the sensor extrinsic parameters.

All of the results presented here were tested using drive 27 of the dataset. In this dataset, the car drives through a residential neighborhood. Drive 27 was selected, as it is one of the longest of all the drives provided in the KITTI dataset, giving 4000 consecutive frames of information on which to test the calibration method. The calibration provided with the dataset was taken as ground truth for our experiments. While this provided calibration will contain some unknown error, it was performed by

Fig. 5. ACFR's shrimp robot.

a team of experts using a series of sensor specific automated and manual techniques [28]. This means that it can be expected to be as accurate a calibration as current techniques allow, and comparing our results to it should offer a strong indication our methods performance.

B. Australian Centre for Field Robotics's Shrimp

To demonstrate that our method is not platform specific and can be applied to different situations without any parameter tuning, a second significantly different experimental platform, i.e., ACFR's Shrimp, was used. Shrimp is a general-purpose sensor vehicle used by the ACFR to gather data for a wide range of applications. Due to this, it is equipped with an exceptionally large array of sensors. For our experiments, we made use of its ladybug panospheric camera, Velodyne HDL-64E lidar, and Novetal GPS/INS system. The setup is shown in Fig. 5. For our experiments, the Shrimp vehicle was driven around a quadrangle outside the ACFR at the University of Sydney for approximately 6 min.

The extrinsic calibration of this platform further motivated the development of our approach. This is because, while many calibrations had previously been performed on the system (lidar–GPS/INS [21], Velodyne–camera [1], camera–camera via checkerboard-based methods, rough timing offsets via manual observation of data, etc.), these had taken significant time to obtain, been performed by experts, and utilized accurately measured initial guesses. Despite this, most were of unknown quality with no indication of their accuracy provided. Due to this limitation, we only made use of the factory calibration provided with the ladybug camera in generating ground-truth data for our comparisons on this system.

C. Finding Timing Offset

We began the testing of our approach by looking at its ability to compensate for timing offsets between the sensors. In this experiment, the approach was used to correct for timing offsets in the cameras of the KITTI and Shrimp datasets. Only the cameras were made use of due to the accurate ground truth available between these sensors. This is due to all cameras being triggered by the same signal giving near-simultaneous images. If any other sensors were to be incorporated, we would be forced to rely on the time stamping of the system's, which is of unknown quality in the KITTI dataset, and known to have a maximum lag of 60 ms for the Shrimp dataset.

To test the system, random contiguous sections of the drives were taken ranging from 10 to 200 s in length in 10-s increments; the cameras then had a random timing offset applied to them. The calibration was run, and the mean absolute error in the timing was found. Two experiments were run; in the first experiment, a random timing offset between -0.1 and 0.1 s was applied to each of the cameras. In the second experiment, up to 1 s of random offset was applied. Each experiment was performed 100 times. The results are shown in Fig. 6.

For both datasets, several common trends appear. For the shortest time period tested (10 s), the results are generally quite poor. This is to be expected due to the small number of motion cues the method has available to align the timing offset. The accuracy of the method rapidly improves over the next few timesteps. After this initial period of improvement, while the robustness of the method continues to improve (seen through the reduction in outlier points), the mean accuracy of the method does not change significantly.

For 200 s of data and 1 s of the initial offset, the KITTI dataset gives a final median error of roughly 6 ms with a worst-case error of 40 ms. The Shrimp dataset gives a median error of 2 ms with a worst-case error of 9 ms. In these datasets, the cameras used give readings every 100–120 ms.

In this experiment, the Shrimp platform's timing calibration tended to be significantly more accurate than that given by the KITTI dataset. This difference most likely comes from Shrimp traveling over unpaved ground. This results in most timesteps experiencing a significant rotation and allowing more motion cues from which to estimate the timing offset.

D. Aligning Two Sensors

To test our approaches' ability to align two sensors with no overlap in their field of view, two experiments were performed. In these experiments, the calibration between a GPS/INS system and a Velodyne lidar was found. In the first experiment, the calibration was performed on the KITTI dataset. In the second experiment, the Shrimp dataset was used. The results were also compared with a least-squares approach that does not make use of the reading's variance estimates. In the experiment, a set of continuous sensor readings is selected at random from the dataset and the extrinsic calibration between them found. This is compared with the known ground truth provided with the dataset. The length of information was set between 10 and 300 s in 10-s increments. For each time period, the experiment was repeated 500 times with the mean reported. While our method calculates the rotation in the angle–axis format to allow for intuitive understanding, we convert this to Euler angles when displaying the results. The associated estimated standard deviation is also converted using a simple Monte Carlo method.

Fig. 7 shows the absolute error of the calibration. For all readings, our calibration significantly improves, as more readings are used for the first few hundred scans, before slowly tapering off.

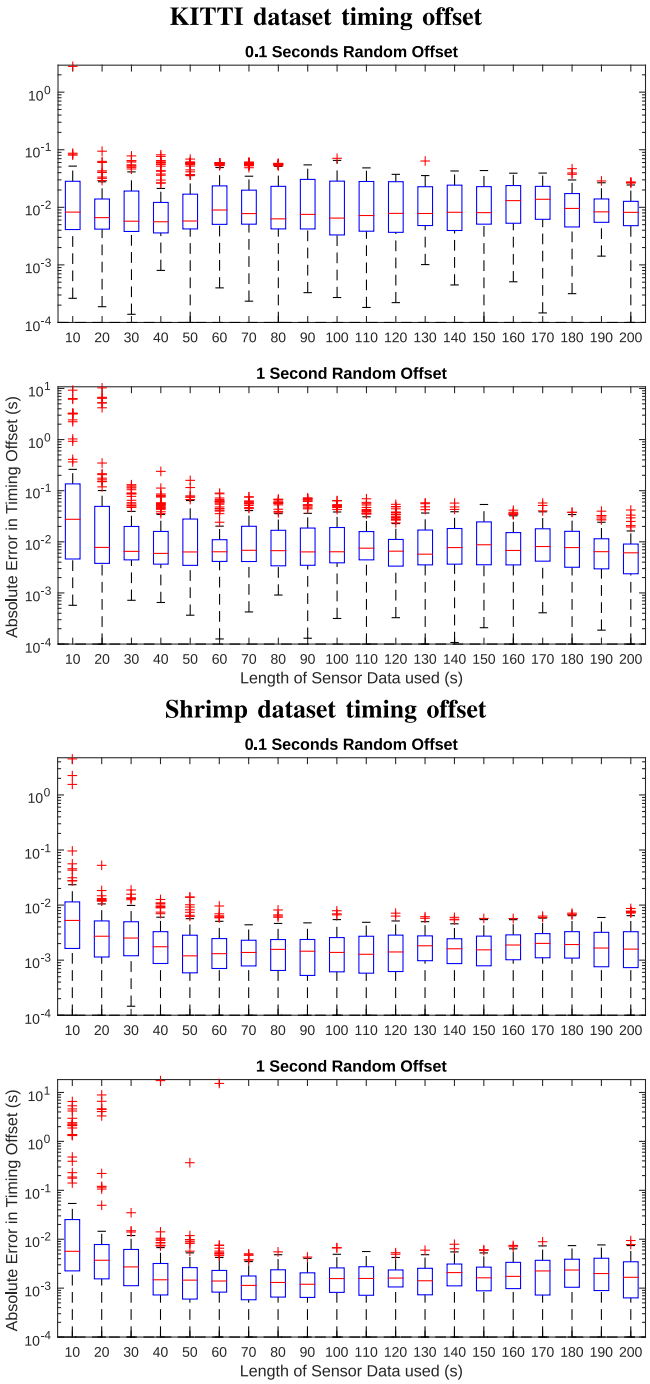

Fig. 6. Box plots of the error in the estimated sensor timing offset. Between 10 and 200 s of continuous sensor data were used to find the offset in the sensor timing. Each experiment was repeated 500 times.

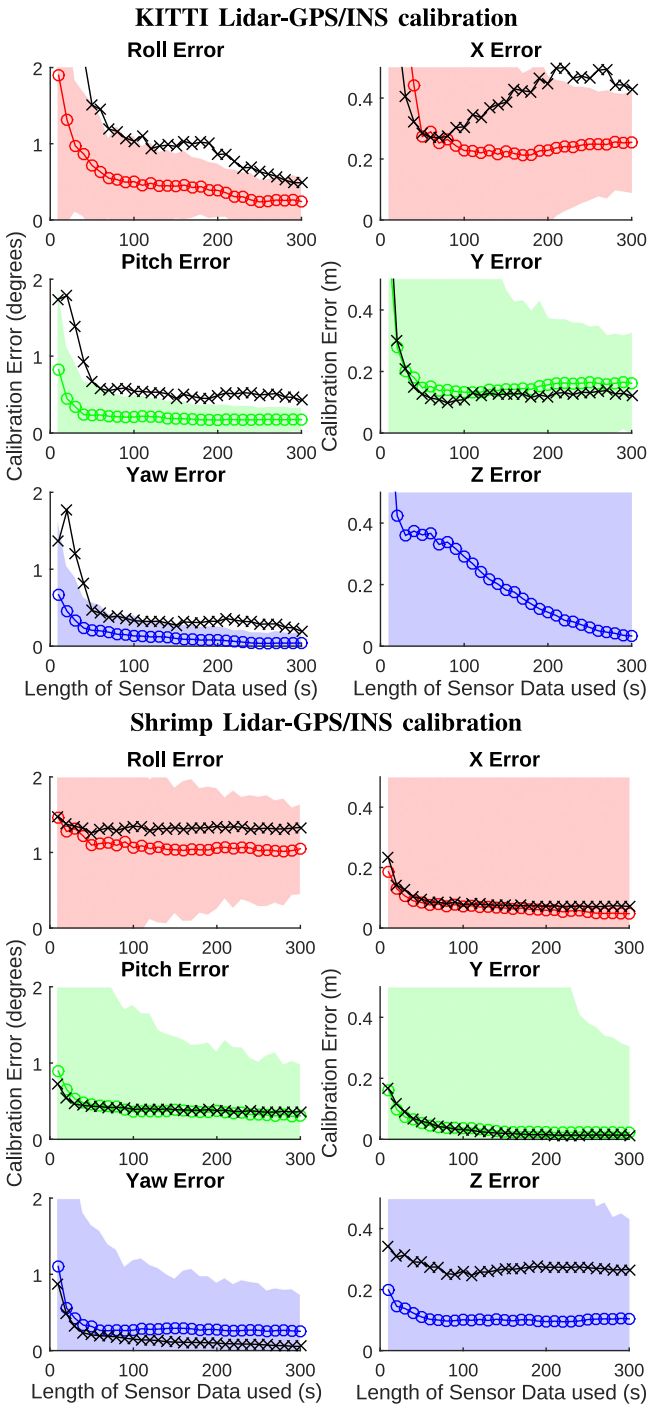

Fig. 7. Error in rotation in degrees and translation in meters for optimizations using 10–200 s of sensor data. The black x's show the least-squares result, while the colored o's gives the result of our approach. The shaded region gives one standard deviation of the estimated uncertainty provided by our approach.

In rotation, yaw was the most accurately estimated. This was to be expected as the motion of the vehicle is roughly planar giving less motion, from which roll and pitch can be estimated. For a large numbers of scans, the method estimated the rotation in the KITTI dataset to within 0.5° of error and in the Shrimp dataset to within 2° of error. Our method outperformed the least-squares method on the KITTI dataset, while the least-squares method gave similar results on the Shrimp dataset. We believe the reason for the similar performance on the Shrimp platform to be due to all the driving in this dataset occurring within a single small courtyard. This meant that there was less variation in the accuracy of the lidar and GPS readings during the calibration than in the KITTI dataset. Because of this, considering the sensors' uncertainty had less of an impact on the results.

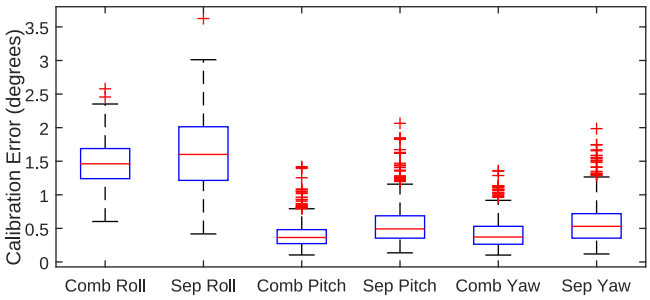

Fig. 8. Error in rotation in degrees when calibrating five of Shrimps cameras. Two results are shown, the calibration obtained by combining all sensor motion (labeled comb) and the results of separately calibrating each camera with respect to the first one (labeled sep).

Fig. 9. Spherical panoramic image created from the five individual cameras. The top image was created using the manufacturers values (taken as ground truth in our experiments) and the bottom with the mean results from the experiment.

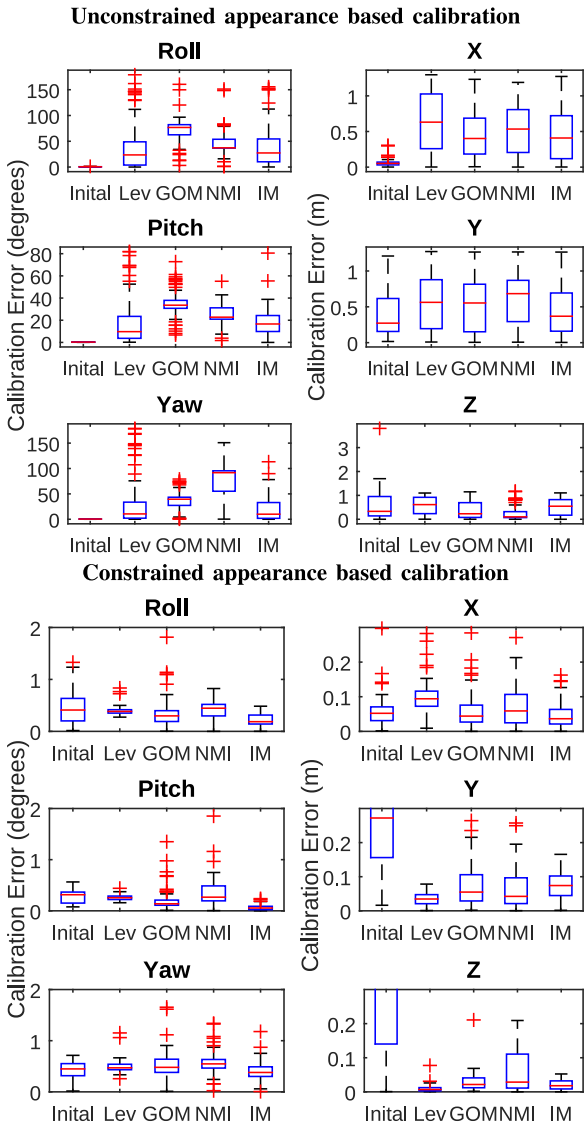

Fig. 10. Box plot of error in alignment for appearance-based metrics. The unconstrained results search the entire viable search space, whereas the constrained results made use of the variance given by the motion-based calibration. Note the axis of the constrained optimization excludes several outliers; the number of these outliers can be found in Table I. All rotations are in degrees and translations in meters.

However, even in cases where our method does not outperform the least-squares approach, it offers the advantage of providing an estimate of the uncertainty in its values.

For translation, our method significantly outperformed the least-squares method. The least-squares method, in many instances, also began to perform more poorly as the data used increased. The most likely explanation for this decrease in performance is that the more data are used for the calculation, the higher the probability an outlier will be present in the data. The largest difference in performance can be seen when estimating the Z offset. This difference is due to the roughly planar motion that makes the Z-axis the most sensitive parameter to noise and outliers, which the simple least-squares method fails to account for.

The accuracy of the predicted variance of the result is more difficult to assess than that of the calibration. However, all of the errors were within a range of around 1.5 standard deviations from the actual error. Viewing the data suggests that the estimated translation variance may be slightly conservative in its estimates. Overall, the method gives a consistent indication of the estimation's accuracy.

E. Calibrating a Panoramic Camera Rig

An experiment was run calibrating the five horizontally facing cameras of the Shrimp platforms ladybug camera system. As only monocular camera matching is performed, no sense of scale is present, and therefore, only the rotational offsets can be estimated. This limitation does not significantly impede the use of multiple cameras in forming a panoramic image; however, as in most practical systems, the cameras are placed in close proximity, where translation is negligible. For the calibration process, 30 s of continuous data were used, and the process was repeated 500 times. Two variations of the experiment were run. In the first, we utilized the offset likelihood between all the cameras, and in the second, only the likelihood between the forward facing camera and each of the other cameras was considered. The error in this calibration was found using the ground truth provided by the manufacturer to find the mean error between any combination of two cameras. The results of these experiments are shown in Fig. 8.

Fig. 11. Lidar camera alignment before (top) and after (bottom) appearance-based refinement using IM. The initial calibration has accurate rotation values but significant uncertainty in the translation. After the refinement, most of this uncertainty has been removed.

TABLE I
PERCENTAGE OF SOLUTIONS WHOSE PARAMETERS ARE WITHIN 0.3 M AND 2° OF THE CORRECT CALIBRATION

	Initial	NMI	GOM	Levinson	IM
Full Search Space	43%	9%	9%	13%	23%
Constrained	43%	84%	89%	95%	100%

From the results, it can be seen that the consideration of all the sensor information yields slightly more accurate results than the more simplistic pairwise calibration, where each sensor is only calibrated against one other. As in previous experiments, roll was the least accurately estimated due to the vehicle giving few motion cues from which to estimate it. Once the calibration has been performed, the results can be combined with the sensor intrinsics to create a spherical panoramic image. To give a qualitative measure of these results, a panorama was created using the mean value of the experimental results and compared to the manufacturers calibration; this is shown in Fig. 9. In this instance, the results of the process are visually comparable with the image generated from the ground truth.

F. Constraining Appearance-Based Metrics

To evaluate the use of motion-based calibration for constraining the search space of appearance metrics, an experiment was performed. Initially, 100 s of data from the KITTI dataset were used to align the system's Velodyne with its leftmost camera. From this initial estimate, appearance-based alignment of the sensors is then performed. The search space for the problem is defined so that all rotations are considered, and the magnitude of the X, Y, and Z offsets of the two sensors is limited to be under 1 m.

From this starting point, a CMA-ES optimizer is first run considering the entire search space. This was done by setting the initial multivariate Gaussian to have a σ that is 50% of the extent of our search space. Once this optimization has been performed, the approach is rerun, this time making use of the variance estimate provided by the motion stage and using this information to initialize CMA-ES's Gaussian distribution. In the appearance stage, 25 scan-image pairs were used; these were randomly selected from the available data. The experiment was repeated 50 times, and the GOM [3], NMI, Levinson [1], and IM metric were evaluated. The results of this experiment are presented in Fig. 10 and Table I.

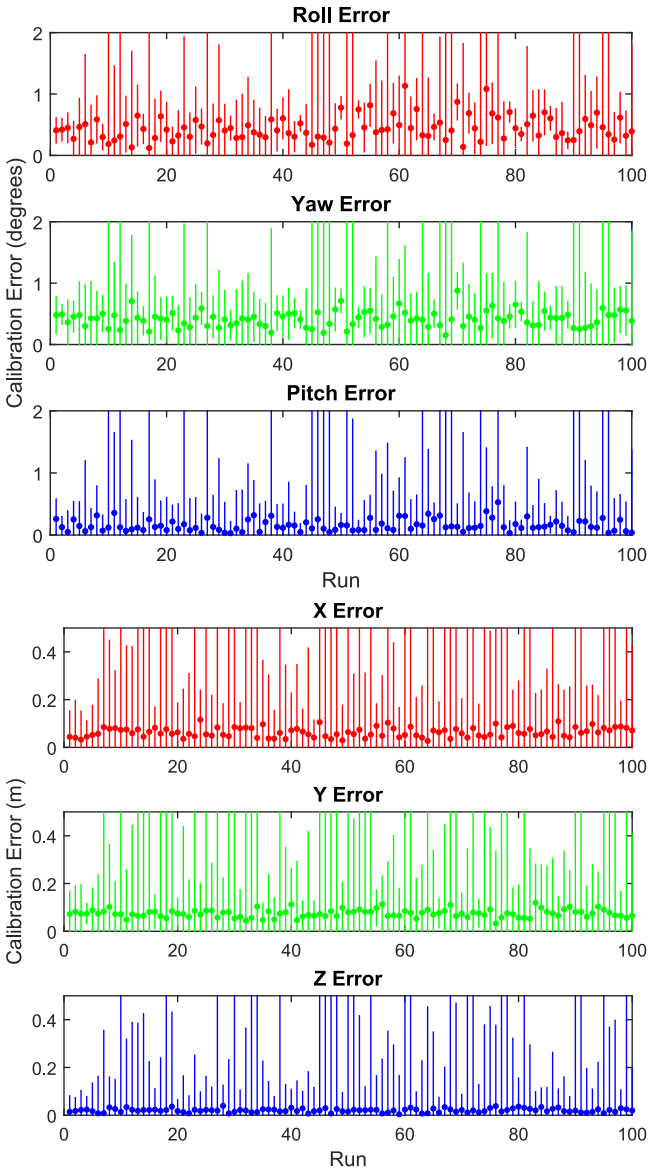

Fig. 12. Plot of the resulting error when the full calibration method is used to align the cameras and Velodyne lidar in the KITTI dataset. The mean error between the sensors is shown for all 100 runs with error bars giving the estimated standard deviation. Each run used a randomly selected section of 100 s of data.

TABLE II
MEAN ERROR WHEN THE FULL METHOD IS USED TO CALIBRATE THE VELODYNE AND CAMERAS ON THE KITTI DATASET

X	Y	Z	roll	pitch	yaw
0.063 m	0.075 m	0.020 m	0.438°	0.415°	0.154°

For the case of the constrained optimization, on average, all metrics typically gave either improved or similar calibration values. For many of the metrics, in some instances, the optimization failed, and the system converged to a result with significant error. The rate at which this occurred for the different metrics is shown in Table I. In this experiment, the constrained IM metric was the only method that did not suffer from these occasional outliers. An example of the improvement the constrained appearance-based refinement can give is shown in Fig. 11.

For the case where the appearance metrics were optimized over the entire search space, the results were exceptionally poor. In many cases, metrics gave results similar or in some cases worse than random guessing. These findings show the limitations of appearance metrics in regard to initialization, and the issues encountered when the metrics are applied without regard for them. There are two issues that combine to give these poor results. The first issue is that the appearance metrics are often slightly bias to sensor overlap, and this can cause a global optima in a location that does not correspond to the correct calibration parameters. The second issue is when the system is simply unable to locate the correct optima. This is caused by the search space being too large and complex for the optimizer to efficiently explore. These difficulties are alleviated by constraining the optimization, as only solutions that are likely under the motion estimation step are evaluated.

G. Full Alignment of Multiple Sensors

To test how the entire process performs when applied to the calibration of a system, all stages of the process—the timing estimation, motion estimation, and refinement, were used to align four cameras and the Velodyne scanner in the KITTI dataset. In each test run, a section of 100 s of contiguous data was randomly selected, and the IM metric was used for refining the alignment. A Gaussian with a σ $\frac{1}{500}$th of the image width was used for blurring the images in IM's image preprocessing stage. The experiment was repeated 100 times. The results of those runs are presented in Fig. 12, and the mean error is given in Table II.

This calibration gave an accurate calibration in almost all cases. The estimated standard deviation was generally slightly conservative, but generally gave a reasonable and believable indication of the reliability of the calculation.

IX. Conclusion

This paper has presented an approach for calibrating the extrinsics and timing offset of any number of cameras, 3-D lidars, and GPS/INS systems mounted on a mobile platform. Unlike most existing solutions, the system does not require any initial estimate of the sensor configurations to perform the calibration. It also requires no markers or other calibration aids. The process operates in two stages. First, the motion of the system is used to estimate the sensor timing offset and extrinsics. The extrinsic calibration is then further refined by exploiting appearance information in overlapping sensor views. The system also considers sensor uncertainty to increase its robustness and allow it to provide an estimate of the reliability of the final calibration. The performance of the system was demonstrated through a series of experiments. The experimental evaluations validated our approach, demonstrating its ability to achieve accurate calibrations without manual initialization and without the use of any external calibration aids

References

[1] J. Levinson and S. Thrun, "Automatic online calibration of cameras and lasers," in *Proc. Robot.: Sci. Syst. Conf.*, 2013, pp. 24–28.

[2] G. Pandey, J. R. Mcbride, S. Savarese, and R. M. Eustice, "Automatic extrinsic calibration of vision and Lidar by maximizing mutual information," *J. Field Robot.*, vol. 32, pp. 696–722, 2015.

[3] Z. Taylor, J. Nieto, and D. Johnson, "Multi-modal sensor calibration using a gradient orientation measure," *J. Field Robot.*, vol. 32, pp. 675–695, 2015.

[4] R. Wang, F. P. Ferrie, and J. Macfarlane, "Automatic registration of mobile LiDAR and spherical panoramas," in *Proc. IEEE Comput. Soc. Conf. Comput. Vision Pattern Recog. Workshops*, Jun 2012, pp. 33–40.

[5] R. Tsai and R. Lenz, "A new technique for fully autonomous and efficient 3D robotics hand/eye calibration," *IEEE Trans. Robot. Autom.*, vol. 5, no. 3, pp. 345–358, Jun. 1989.

[6] Z. Taylor and J. Nieto, "Motion-based calibration of multimodal sensor arrays," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2015, pp. 4843–4850.

[7] L. Tamas and Z. Kato, "Targetless calibration of a lidar—Perspective camera pair," in *Proc. IEEE Int. Conf. Comput. Vision*, 2013, pp. 668–675.

[8] A. Napier, P. Corke, and P. Newman, "Cross-calibration of push-broom 2D LIDARs and cameras in natural scenes," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2013, pp. 3679–3684.

[9] Y. C. Shiu and S. Ahmad, "Calibration of wrist-mounted robotic sensors by solving homogeneous transform equations of the form AX = XB," *IEEE Trans. Robot. Autom.*, vol. 5, no. 1, pp. 16–29, Feb. 1989.

[10] C. C. Wang, "Extrinsic calibration of a vision sensor mounted on a robot," *IEEE Trans. Robot. Autom.*, vol. 8, no. 2, pp. 161–175, Apr. 1992.

[11] W. Kabsch, "A discussion of the solution for the best rotation to relate two sets of vectors," *Acta Crystallographica Sec. A*, vol. 34, no. 5, pp. 827–828, 1978.

[12] N. Andreff, R. Horaud, and B. Espiau, "Robot hand-eye calibration using structure-from-motion," *Int. J. Robot. Res.*, vol. 20, no. 3, pp. 228–248, Mar. 2001.

[13] J. Heller, M. Havlena, A. Sugimoto, and T. Pajdla, "Structure-from-motion based hand-eye calibration using L infinity minimization," in *Proc. IEEE Conf. Comput. Vision Pattern Recog.*, Jun. 2011, pp. 3497–3503.

[14] M. K. Ackerman, A. Cheng, B. Shiffman, E. Boctor, and G. Chirikjian, "Sensor calibration with unknown correspondence: Solving AX=XB using Euclidean-group invariants," in *Proc. IEEE Int. Conf. Intell. Robots Syst.*, Nov. 2013, pp. 1308–1313.

[15] J. Kelly and G. S. Sukhatme, *Experimental Robotics*, vol. 79. Berlin, Germany: Springer, 2014, pp. 195–209. [Online]. Available: http://link.springer.com/10.1007/978-3-642-28572-1

[16] S. Schneider, T. Luettel, and H.-J. Wuensche, "Odometry-based online extrinsic sensor calibration," in *Proc. IEEE Int. Conf. Intell. Robots Syst.*, Nov. 2013, pp. 1287–1292.

[17] H. Ovren and P.-E. Forssen, "Gyroscope-based video stabilisation with auto-calibration," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2015, pp. 2090–2097.

[18] M. Fleps, E. Mair, O. Ruepp, M. Suppa, and D. Burschka, "Optimization based IMU camera calibration," in *Proc. IEEE Int. Conf. Intell. Robots Syst.*, 2011, pp. 3297–3304.

[19] N. Keivan and G. Sibley, "Online SLAM with any-time self-calibration and automatic change detection," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2015, pp. 5775–5782.

[20] M. Li, H. Yu, X. Zheng, and A. Mourikis, "High-fidelity sensor modeling and self-calibration in vision-aided inertial navigation," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2014, pp. 409–416.

[21] J. P. Underwood, "Reliable and safe autonomy for ground vehicles in unstructured environments," Ph.D. dissertation, Univ. Sydney, Sydney, Australia, 2009.

[22] L. Heng, B. Li, and M. Pollefeys, "CamOdoCal: Automatic intrinsic and extrinsic calibration of a rig with multiple generic cameras and odometry," in *Proc. IEEE Int. Conf. Intell. Robots Syst.*, Nov. 2013, pp. 1793–1800.

[23] P. H. S. Torr, "Bayesian model estimation and selection for epipolar geometry and generic manifold fitting," *Int. J. Comput. Vision*, vol. 50, no. 1, pp. 35–61, 2002.
[24] R. Hartley, "In defense of the eight-point algorithm," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 6, pp. 580–593, Jun. 1997. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=601246
[25] G. W. Oehlert, "A note on the delta method," *Amer. Statist.*, vol. 46, no. 1, pp. 27–29, 1992.
[26] A. Censi and R. La, "An accurate closed-form estimate of ICPs covariance," in *Proc. IEEE Int. Conf. Robot. Autom.*, Apr. 2007, pp. 10–14.
[27] N. Hansen and A. Ostermeier, "Adapting arbitrary normal mutation distributions in evolution strategies: The covariance matrix adaptation," in *Proc. IEEE Int. Conf. Evol. Comput.*, 1996, pp. 312–317.
[28] A. Geiger and P. Lenz, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *Proc. IEEE Conf. Comput. Vision Pattern Recog.*, Jun. 2012, pp. 3354–3361.

Zachary Taylor received the Ph.D. degree in robotics from University of Sydney, Sydney, Australia, in 2015.

He joined the Autonomous Systems Laboratory, ETH Zurich, Zürich, Switzerland, as a Postdoctoral Researcher in 2016. His research interests include calibration, multimodal sensors, and persistent autonomy.

Juan Nieto received the Ph.D. degree in robotics from University of Sydney, Sydney, Australia, in 2005.

He was a Research Associate with the Australian Centre for Field Robotics until 2007. From 2007 to 2015, he was a Senior Research Fellow with the Rio Tinto Centre for Mine Automation, University of Sydney. In 2015, he joined the Autonomous Systems Lab, ETH Zurich, Zürich, Switzerland, where he is currently a Deputy Director. He has more than 80 publications in international journals and conferences. His main research interests include navigation systems and perception.