

Motion-Based Calibration of Multimodal Sensor Arrays

Zachary Taylor and Juan Nieto
University of Sydney, Australia
{z.taylor, j.nieto}@acfr.usyd.edu.au

Abstract—This paper formulates a new pipeline for automated extrinsic calibration of multi-sensor mobile platforms. The new method can operate on any combination of cameras, navigation sensors and 3D lidars. Current methods for extrinsic calibration are either based on special markers and/or checkerboards, or they require a precise parameters initialisation for the calibration to converge. These two limitations prevent them from being fully automatic. The method presented in this paper removes these restrictions. By combining information extracted from both, platform’s motion estimates and external observations, our approach eliminates the need for special markers and also removes the need for manual initialisation. A third advantage is that the motion-based automatic initialisation does not require overlapping field of view between sensors. The paper also provides a method to estimate the accuracy of the resulting calibration. We illustrate the generalisation of our approach and validate its performance by showing results with two contrasting datasets. The first dataset was collected in a city with a car platform, and the second one was collected in a tree-crop farm with a Segway platform.

I. INTRODUCTION

Autonomous and remote control platforms that utilise a large range of sensors are steadily becoming more common. Sensing redundancy is a pre-requisite for their robust operation. For the sensor observations to be combined, each sensors position, orientation and coordinate system must be known. Automated calibration of multi-modal sensors is a non trivial problem due to the different type of information collected by the sensors. This is aggravated by the fact that in many applications different sensors will have minimum or zero overlapping field of view. Physically measuring sensor positions is difficult and impractical due to the sensors’ housing and the high accuracy required to give a precise alignment, especially in long-range sensing outdoor applications. For mobile robots, working on topologically variable environments, as found in agriculture or mining, this can result in significantly degraded calibration after as little as a few hours of operation. Therefore, to achieve long-term autonomy at the service of non-expert users, calibration parameters have to be determined efficiently and dynamically updated over the lifetime of the robot.

Traditionally, sensors were manually calibrated by either placing markers in the scene or by hand labelling control points in the sensor outputs [1], [2]. Approaches based on these techniques are impractical as they are slow, labour intensive and usually require a user with some level of technical knowledge.

In more recent years, automated methods for calibrating pairs of sensors have appeared. These include configurations

such as 3D lidar to camera calibration [3], [4], [5], [6], camera to IMU calibration [7], [8] and odometry based calibration [9]. While these systems can in theory be effectively used for calibration, they present drawbacks that limit their application in many practical scenarios. First, they are generally restricted to a single type of sensor pair, second their success will depend on the quality of the initialisation provided by the user, and third, approaches relying exclusively on external sensed data require overlapping fields of view [10]. In addition, the majority of the methods do not provide an indication of the accuracy of the resulting calibration, a key for consistent data fusion and to prevent failures in the calibration.

In this paper we propose an approach for automated extrinsic calibration of any number and combination of cameras, 3D lidar and IMU sensors present on a mobile platform. Without initialisation, we obtain estimates for the extrinsics of all the available sensors by combining individual motions estimates provided by each of them. This part of the process is based on recent ideas for markerless hand-eye calibration using structure-from-motion techniques [11], [12]. We will show that our motion-based approach is complementary to previous approaches, providing results that can be used to initialise method relying on external data and therefore requiring initialisation [3], [4], [5], [6]. We also show how to obtain an estimate of uncertainty for the calibrations parameters, so the system can make an informed decision about which sensors may require further refinement. All of the source code used to generate these results is publicly available at [13].

The key contributions of our paper are:

- A motion-based calibration method that utilises all of the sensors motion and their associated uncertainty.
- The use of a motion-based initialisation to constrain existing lidar-camera techniques.
- The introduction of a new motion based lidar-camera metric.
- A calibration method that gives an estimate of the uncertainty of the resulting system.
- Evaluation in two different environments with two different platforms.

II. RELATED WORK

As mentioned above, previous works have focused on specific pairs of sensor modalities. Therefore, we present and analyse the related work according to the sensor modalities being calibrated.

1) *Camera-Camera Calibration*: In a recent work presented in [9], four cameras with non-overlapping fields of view are calibrated on a vehicle. The method operates by first using visual odometry in combination with the cars egomotion provided by odometry to give a course 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.

2) *Lidar-Camera Calibration*: One of the first markerless calibration approaches was the work presented in [3]. 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. An edge image is produced from the camera. The two outputs are then combined and a grid search is used to find the parameters that maximise a cost metric.

Three very similar methods have been independently developed and presented in [4], [14] and [6]. 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 MI value is then taken between the lidar’s intensity of return and the intensity of the corresponding points in the camera’s image. When the MI value is maximised, the system is assumed to be perfectly calibrated. The main differences among these methods are on the optimisations strategies.

In our most recent work [15], [10] we presented a calibration method that is based on the alignment of the orientation of gradients formed from the lidar and camera. The gradient for each point in the camera and lidar is first found. The lidar points are then projected onto the cameras image and the dot product of the gradient of the overlapping points are taken. This result is then summed and normalised by the strength of the gradients present in the image. A particle swarm global optimiser is used to find the parameters that maximise this metric.

The main drawback of these cost functions is that they are non-convex and therefore will require initialisation.

A. Motion-based Sensor-Camera Calibration

In [7] the authors presented a method that utilises structure from motion in combination with a Kalman filter to calibrate an INS system with a stereo camera rig. More related to our approach are recent contributions to the hand-eye calibration for calibrating a camera mounted to a robotic arm. These techniques have made use of structure from motion approaches to allow them to operate without requiring markers or checkerboards [11], [12]. The main difference between these approaches and our own is that they are limited to calibrating a single pair of sensors and do not make use of the sensors uncertainty or the overlapping field of view some sensors have.

III. OVERVIEW OF METHOD

Throughout this paper the following notation will be used

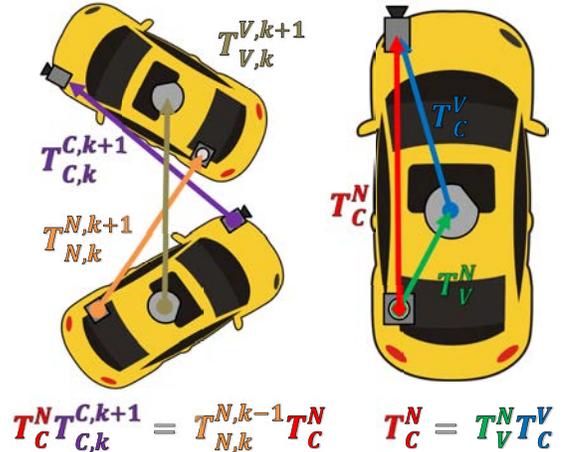


Fig. 1. A diagram of a car with a camera (C), velodyne lidar (V) and nav sensor (N). The car on the right shows the three sensors positions on the vehicle. The image on the left shows the transformation these sensors undergo at timestep k

- T_B^A : The transformation from sensor A to sensor B.
- $T_{B,j}^{A,i}$: The transformation from sensor A at timestep i to sensor B at timestep j
- R : A Rotation matrix
- t : A translation vector

Figure 1 shows depiction of how each sensors motion and relative position are related. As the vehicle moves the transformation between any two sensors A and B can be recovered by using Equation 1.

$$T_B^A T_{B,k}^{B,k-1} = T_{A,k}^{A,k-1} T_B^A \quad (1)$$

This is the basic equation used in most hand-eye calibration techniques. Note that it only considers a single pair of sensors. Our approach takes the fundamental ideas used by hand eye calibration and place them into an algorithm that takes into account the relative motion of all sensors present in a system, is robust to noise and outliers, and provides estimates for the variance of the resulting transformations. Then when sensor overlap exists we exploit sensor specific techniques to further refine the system taking advantage of all the information the sensors provide.

The steps followed to give the overall sensor transformation and variance estimates are summarised in Algorithm 1.

IV. METHOD

A. Finding the sensor motion

The approach utilised to calculate the transformations between consecutive frames depends on the sensor type. In our pipeline, three different approaches were included to work with 3D lidar sensors, navigation sensors and image sensors.

ALGORITHM 1

1. Given n sensor readings from m sensors
 2. Convert sensor readings into sensor motion $T_{i,k}^{i,k-1}$. Each transformation has the associated variance $(\sigma_{i,k}^{i,k-1})^2$
 3. Set one of the lidar or nav sensors to be the base sensor B
 4. Convert $R_{i,k}^{i,k-1}$ to angle axis form $A_{i,k}^{i,k-1}$
- For** $i = 1, \dots, m$
5. Find a coarse estimate for R_i^B by using the Kabsch algorithm to solve $A_{B,k}^{B,k-1} = R_B^i A_{i,k}^{i,k-1}$ weighting the elements by $W_k = (\max((\sigma_{i,k}^{i,k-1})^2) + \max((\sigma_{B,k}^{B,k-1})^2))^{-0.5}$
 6. Label all sensor readings over 3σ from the solution outliers
- end**
7. Find estimate for R_i^B by minimising
$$\sum_{i=1}^m \sum_{j=i}^m \sum_{k=2}^n \sqrt{Rerr_{ijk}^T \sigma^2 Rerr_{ijk}}$$

$$Rerr_{ijk} = (A_{j,k}^{j,k-1} - R_j^i A_{i,k}^{i,k-1})$$

$$\sigma^2 = (\sigma_{j,k}^{j,k-1})^2 + R_j^i (\sigma_{i,k}^{i,k-1})^2 (R_j^i)^T$$
 using the coarse estimate as an initial guess in the optimisation.
- For** $i = 1, \dots, m$
8. Find a coarse estimate for t_i^B by solving $t_i^B = (R_{i,k}^{i,k-1} - I)^{-1} (R_{i,k}^B t_{B,k}^{B,k-1} - t_{i,k}^{i,k-1})$ and weighting the elements by $W_k = (\max((\sigma_{i,k}^{i,k-1})^2) + \max((\sigma_{B,k}^{B,k-1})^2))^{-0.5}$
- end**
9. Find estimate for t_i^B by minimising
$$\sum_{i=1}^m \sum_{j=i}^m \sum_{k=2}^n \sqrt{Terr_{ijk}^T \sigma^2 Terr_{ijk}}$$

$$Terr_{ijk} = (R_{i,k}^{i,k-1} - I)t_i^j + t_{i,k}^{i,k-1} - R_{i,k}^j t_{j,k}^{j,k-1}$$
 using the coarse estimate as an initial guess in the optimisation.
 10. Bootstrap sensor readings and re-estimate R_i^B and t_i^B to give variance estimate $(\sigma_i^B)^2$
 11. Find all sensors that have overlapping field of view
 12. Use sensor specific metrics to estimate the transformation between sensors with overlapping field of view
 13. Combine results to give final R_i^B , t_i^B and $(\sigma_i^B)^2$.

1) *3D lidar sensors*: 3D lidar sensors use one or more laser range finders to generate a 3D map of the surrounding environment. To calculate the transform from one sensor scan to the next the *iterative closest point* (ICP) [16] algorithm is used. A point to plane variant of ICP is used and the identity matrix is used for the initial guess as to the transformation. To estimate the covariance the data is bootstrapped and the ICP matching is run 100 times. In our implementation to allow for shorter run times the bootstrapped data was also subsampled to 5000 points.

2) *Navigation sensors*: This category includes sensors that provide motion estimates, such as IMU's and wheel odometry. These require little processing to convert their standard outputs into a transformation between two positions and the covariance is usually provided as an output in the data stream.

3) *Imaging sensors*: This group covers sensors that produce a 2D image of the world such as RGB and IR cameras. The set of transforms that describe the movement of the sensors are calculated, up to scale ambiguity using a standard structure from motion approach. To get an estimate of the covariance of this method we take the points the method uses to estimate the fundamental matrix and bootstrap them 100 times. The resulting sets of points are used to re-estimate this matrix and the subsequent transformation estimate.

4) *Accounting for timing offset*: As a typical sensor array can be formed by a range of sensors running asynchronously, the times at which readings occur can differ significantly [17]. To account for this each of the sensor's transforms are interpolated at the times when the slowest updating sensor obtained readings. For the translation linear interpolation is used and for the rotation **Slerp** (*spherical linear interpolation*) is used.

B. Estimating the inter-sensor rotations

Given that all of the sensors are rigidly mounted the rotation of any two sensors here labelled A and B can be given by the following Equation

$$R_B^A R_{B,k}^{B,k-1} = R_{A,k}^{A,k-1} R_B^A \quad (2)$$

In our implementation an angle axis representation of the sensor rotations is used, this simplifies Equation 2 as if $A_{A,k}^{A,k-1}$ is our rotation vector then the sensor rotations are related by

$$A_{B,k}^{B,k-1} = R_B^A A_{A,k}^{A,k-1} \quad (3)$$

We utilise this equation to generate a coarse estimate for the system and reject outlier points. This is done as follows.

First, one of the sensors is arbitrarily designated the base sensor. Next a slightly modified version of the Kabsch algorithm is used to find the transformation between this sensor and each other sensor. The Kabsch algorithm is an approach that calculates the rotation matrix between two vectors providing the smallest least squared error. The

Kabsch algorithm we used had been modified to give a non-equal weighting to each sensor reading. The weight assigned to the readings at each timestep is given by Equation 4

$$W_k = (\max((\sigma_{A,k}^{A,k-1})^2) + \max((\sigma_{B,k}^{B,k-1})^2))^{-0.5} \quad (4)$$

where σ^2 is the variance. The taking of the maximum values makes the variance independent of the rotation and allows it to be reduced to a single number.

Next an outlier rejection step is applied where all points that are over three standard deviations away from where the rotation solution obtained would place them are rejected. With an initial guess to the solution calculated and the outliers removed we move onto the next step of the process where all of the sensor rotations are optimised simultaneously.

This stage of the estimation was inspired by how SLAM approaches combine pose information. However, it differs slightly as in SLAM each pose is generally only connected to a few other poses in a sparse graph structure, where in our application, every scan of every sensor contributes to the estimation of the systems extrinsic calibration and therefore the graph is fully connected. This dense structure of the problem prevents many of the sparse SLAM solutions from being used. In order to keep our problem tractable we only directly compare sensor transforms made at the same timestep.

The initial estimate is used to form a rotation matrix that gives the rotation between any two of the sensors. Using this rotation matrix the variance for the sensors is found by 5. This is used to find the error for the pair of sensors as is shown in Equation 6.

$$\sigma^2 = (\sigma_{B,k}^{B,k-1})^2 + R_B^A (\sigma_{A,k}^{A,k-1})^2 (R_B^A)^T \quad (5)$$

$$\begin{aligned} RotError &= \sqrt{Rerr_{ijk}^T \sigma^2 Rerr_{ijk}} \\ Rerr_{ijk} &= (A_{j,k}^{j,k-1} - R_j^i A_{i,k}^{i,k-1}) \end{aligned} \quad (6)$$

The error for all of the sensor pairs is combined to form a single error metric for the system. This error is minimised using a gradient decent optimiser (in our implementation the Nelder-Mead Simplex) to find the optimal rotation angles.

C. Estimating the inter-sensor translations

If the system contains only monocular cameras no measure of absolute scale is present and this prevents the translation from being calculated. However in any other system, if the base sensor's transforms do not have scale ambiguity, once the rotation matrix is known the translation of the sensors can be calculated. It is found using a method similar to that of the rotation. By starting with

$$t_B^A = (R_{B,k}^{B,k-1} - I)^{-1} (R_B^A t_{A,k}^{A,k-1} - t_{B,k}^{B,k-1}) \quad (7)$$

The terms can be rearranged and combined with information from other timesteps to give the following system of Equations

$$\begin{bmatrix} R_{B,k}^{B,k-1} - I \\ R_{B,k}^{B,k} - I \\ R_{B,k+1}^{B,k+1} - I \\ R_{B,k+2}^{B,k+2} - I \\ \dots \end{bmatrix} t_B^A = \begin{bmatrix} R_B^A t_{A,k}^{A,k-1} - t_{B,k}^{B,k-1} \\ R_B^A t_{A,k}^{A,k} - t_{B,k}^{B,k} \\ R_B^A t_{A,k+1}^{A,k+1} - t_{B,k+1}^{B,k+1} \\ R_B^A t_{A,k+2}^{A,k+2} - t_{B,k+2}^{B,k+2} \\ \dots \end{bmatrix} \quad (8)$$

The only unknown here is t_B^A . The translation estimate depends significantly on the accuracy of the rotation matrix estimations and is the most sensitive parameter to noise. This sensitivity comes from the $R_{B,k}^{B,k-1} - I$ terms that make up the first matrix. As for our application (ground platforms) in most instances $R_{B,k}^{B,k-1} \approx I$ leading to a matrix that ≈ 0 . Dividing by this matrix reduces the SNR and therefore degrades the estimation. Consequently the translation estimates are generally less accurate than the rotation estimates.

This formulation must be slightly modified to be used with cameras, as they only provide an estimate of their position up to a scale ambiguity. To correct for this when a camera is used the scale at each frame is simultaneously estimated.

A rough estimate for the translation is found by utilising Equation 8, which provides translation estimates of each sensor with respect to the base sensor. The elements of this equation are first weighted by their variance using Equation 4.

Once this has been performed the overall translation is found by minimising the following error functions

$$\begin{aligned} TransError &= \sqrt{Terr_{ijk}^T \sigma^2 Terr_{ijk}} \\ Terr_{ijk} &= (R_{i,k}^{i,k-1} - I) t_i^B + t_{i,k}^{i,k-1} - R_i^B t_{B,k}^{B,k-1} \end{aligned} \quad (9)$$

The final cost function to be optimised is the sum of Equation 9 over all sensor combinations, as was also done for the rotations.

D. Calculating the overall Variance

To find the overall variance of the system the sensor transformations are bootstrapped and the coarse optimisation re-run 100 times. Bootstrapping is used to estimate the variance as it naturally takes into account several non-linear relationships such as the estimated translations dependence on the rotations accuracy. This method of variance estimation has the drawback of significantly increasing the runtime of the inter-sensor estimation stage. However as this time is significantly less than that used by the motion estimation step it has negligible effect on the overall runtime.

E. Refining the inter-sensor transformations

One of the key benefits introduced with our approach is removing the restriction of the initialisation needed for calibration methods based on external observations. Our motion-based calibration described above is complementary to these methods. The estimates provided by our approach can be utilised to initialise non-convex calibration approaches such as [3], [15], rendering a fully automated calibration pipeline. We will call this second stage the refinement process.

For cameras and 3D lidar scanners methods such as Levinson’s [3] or GOM [15] can be utilised for the calibration refinement. The calculation of the gradients for GOM is slightly modified in this implementation (calculated by projecting the lidar onto an image plane) to better handle the sparse nature of the velodyne lidar point cloud. Given that the transformation of the velodyne sensors position between timesteps is known from the motion-base calibration, we introduce a third method for refinement. This method works by assuming that the points in the world will appear the same colour to a camera in two successive frames. It operates by first projecting the velodyne points onto an image at timestep k and finding the colour of the points at these positions. The same scan is then projected into the image taken at timestep $k+1$ and the mean difference between the colour of these points and the previous points is taken. It is assumed that this value is minimised when the sensors are perfectly calibrated.

All three of the velodyne-camera alignment methods described above give highly non-convex functions that are challenging to optimise. Given the large search space, these techniques are always constrained to search a small area around an initial guess. However in our case we have an estimated variance, rather than a strict region in the search space. Because of this in our implementation we make use of the *Covariance Matrix Adaptation Evolution Strategy* **CMA-ES** optimisation technique. This technique randomly samples the search space using a multivariate normal distribution that is constantly updated. This optimisation strategy works well with our approach as the initial normal distribution required can be set using the variance from the estimation. This means that the optimiser only has to search a small portion of the search space and can rapidly converge to the correct solution. To give an indication of the variance of this method a local optimiser is used to optimise the values of each scan used individually starting at the solution the CMA-ES technique provides.

F. Combining the refined results

The pairwise transformations calculated in the refinement step will not represent a consistent solution to the transformation of the sensors. That is, if there are three sensors A, B and C then $T_B^A T_C^B \neq T_C^A$. 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 the base sensor 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 optimised using a Nelder-Mead Simplex optimisation. As the camera-camera transforms contain scale ambiguity only the rotation portion of these transforms is considered.

A. Experimental Setup

The method was evaluated using data collected with two different platforms, in two different type of environments: (i) the KITTI dataset which consists of a car moving in an urban environment and (ii) the *Australian Centre for Field Robotics* (ACFR’s) sensor vehicle known as “Shrimp” moving in a farm.

1) *KITTI Dataset Car*: The KITTI dataset is a well known publicly available dataset obtained from a sensor vehicle driving in the city of Karlsruhe in Germany [18]. The sensor vehicle is equipped with two sets of stereo cameras, a Velodyne HDL-64E and a GPS/IMU system. This system was chosen for calibration due to the ease of availability and the excellent ground truth available due to the recalibration of its sensors before every drive.

All of the results presented here were tested using drive 27 of the dataset. In this dataset the car drives through a residential neighbourhood. In this test while the KITTI Vehicle has an RTK GPS it frequently losses its connection resulting in jumps in the GPS location. 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 to test the calibration method on.

2) *ACFR’s Shrimp*: Shrimp is a general purpose sensor vehicle used by the ACFR to gather data for a wide range of application. Due to this it is equipped with an exceptionally large array of sensors. For our evaluation we used the ladybug panospheric camera and velodyne HDL-64E lidar. The dataset used is a two minute loop driven around a shed in an almond field. This proves a very challenging dataset for calibration as both the ICP used in matching lidar scans and the SURF matching used on the cameras frequently mismatch points on the almond trees leading to sensor transform estimates that are significantly noisier than those obtained with the KITTI dataset.

B. Aligning Two Sensors

The first experiment run was the calibration of the KITTI car’s lidar and IMU. The results were also compared with a least squares approach that does not make use of the readings variance estimates. In the experiment a set of continuous sensor readings is selected from the dataset at random and the extrinsic calibration between them found. This is compared to the known ground truth provided with the dataset. The number of scans used was varied from 50 to 1000 in increments of 50 and for each number of scans the experiment was repeated 100 times with the mean reported.

Figure 2 shows the absolute error of the calibration as well as one standard deviation of the predicted error. For all readings the calibration significantly improves as more readings are used for the first few hundred scans before slowly tapering off. Our method outperforms the least squared method in the vast majority of cases. In rotation, yaw and pitch angles were the most accurately estimated. This was to be expected as the motion of the vehicle is roughly

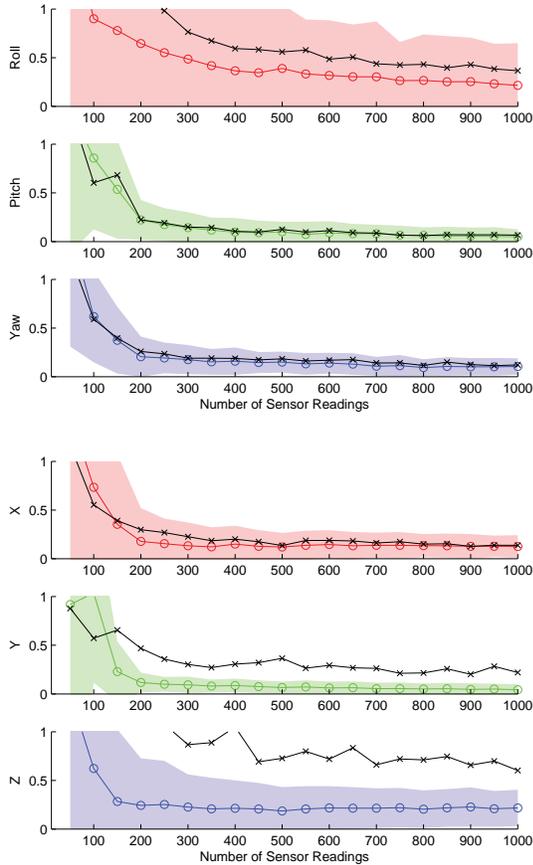


Fig. 2. The error in rotation in degrees and translation in metres for varying numbers of sensor readings. The black line with x’s shows the least squares result, while the coloured line with o’s gives the result of our approach. The shaded region gives one standard deviation of our approaches estimated variance

planner giving very little motion from which roll can be estimated. For large numbers of scans the method estimated the roll, pitch and yaw between the sensors to within 0.5 degree of error.

The calibration for the translation was poorer than the rotation. This is due to the reliance on the rotation calculation and the sensors tending to give noisier translation estimates. For translation our method significantly outperformed the least squared method, this is most likely due to the RTK GPS frequently losing signal, producing significant variation in the sensors translation accuracy jumping position during the dataset.

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. This demonstrates that the predicted variance provides a consistent indication of the estimates accuracy.

C. Refining Camera-Velodyne Calibration

The second experiment evaluates the refinement stage by using the KITTI vehicles velodyne and camera to refine

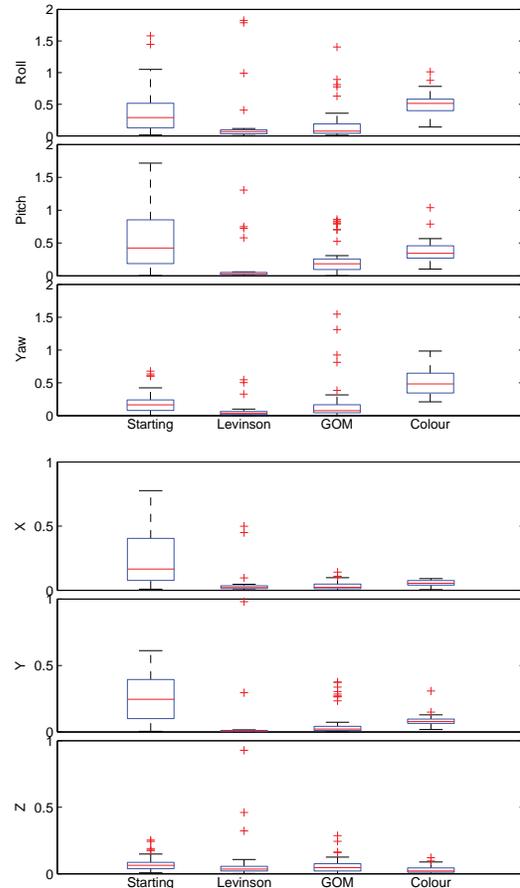


Fig. 3. Box plot of the error in rotation in degrees and translation in metres for the three refinement approaches.

the alignment. In this experiment 500 scans were randomly selected and used to find an estimate of the calibration between the velodyne and leftmost camera on the KITTI vehicle. This result was then refined using a subset of 50 scans and one of the three methods covered in section IV-E.

To demonstrate the importance of an estimation of accuracy, an optimisation that does not consider the estimated variance was undertaken. The search space for this experiment was set as the entire feasible range of values (360° range for angles and $|X|, |Y|, |Z| < 3m$). The experiment was repeated 50 times and the results are shown in Figures 3 and Table I. Note that to allow the results to be shown at a reasonable scale, outliers were excluded (7 from the Levinson method results, 2 from the GOM method results and none from the colour methods).

From the results shown in Figure 3 it can be seen that all of the methods generally significantly improve translation error. The results for the rotation error were more mixed, with the colour based method giving less accurate yaw estimates than the initial hand-eye calibration. Overall, the Levinson method tended to either converge to a highly accurate solution or a position with a large error. The GOM method was slightly more reliable though still error prone and the colour based method while generally the least accurate, always converged

Metric	Roll	Pitch	Yaw	X	Y	Z
Lev	58.2	18.0	14.3	1.97	1.42	3.28
GOM	86.3	43.9	49.5	1.26	0.66	2.93
Colour	3.3	3.6	5.7	1.09	1.41	0.51

TABLE I

MEDIAN ERROR IN ROTATION IN DEGREES AND TRANSLATION IN METRES FOR THE REFINEMENT WHEN THE OPTIMISER DOES NOT UTILISE THE STARTING VARIANCE ESTIMATE.

to reasonable solutions. It was due to this reliability that the colour based solution was used in our full system calibration test.

Table I shows the results obtained without using the variance estimate to constrain the search space. In this experiment all three metrics perform poorly, giving large errors for all values. This experiment was included to show the limitations of the metrics and to demonstrate the need to constrain the search space in order to obtain reliable results.

D. Simultaneous Calibration of Multiple Sensors

To evaluate the effects that simultaneously calibrating all sensors has on the results an experiment was performed using Shrimp and aligning its velodyne with the five horizontal cameras of its ladybug camera. The experiment was first performed using 200 readings to calibrate the sensor combining all of the readings. In a second experiment the velodyne was again aligned with the cameras however this time only the velodyne to camera transform for each camera was optimised. The mean error in rotation and translation between every pair of sensors was found and is shown in Figure 4. As can be seen in the Figure, utilising the camera-camera transformations substantially improves the calibration results. This was to be expected as performing a simultaneous calibration of the sensors provides significantly more information on the sensor relationships. In translation the results slightly improve however due to the low number of scans used and noisy sensor outputs the error in the translation estimation means it would be of limited use in calibrating a real system.

E. Full alignment of the KITTI dataset sensors

Finally, results with the full calibration process are shown for the KITTI dataset. We used 500 scans for the motion estimation step, and the Colour matching method was used for refining the velodyne-camera alignment. The experiment was repeated 50 times and the results are shown in Figure 5.

For these results all of the sensors had a maximum rotation error of 1.5 degrees with mean errors below 0.5 degrees. For translation the IMU had the largest errors in position due to relying purely on the motion stage of the estimation. For the cameras the error in X and Y was generally around 0.1 m. The errors in Z tended to be slightly larger due to motion in this direction being more difficult for the refinement metrics to observe.

Overall, while the method provides accurate rotation estimates, the estimated translation generally has an error that

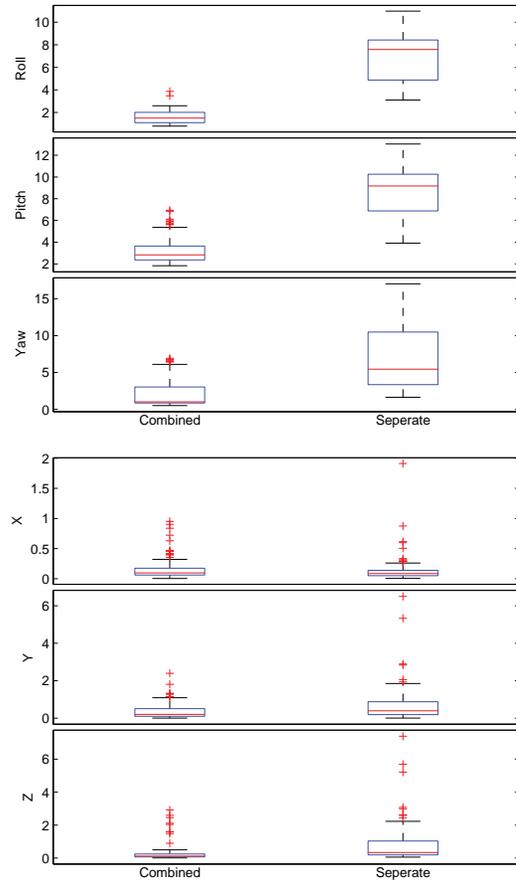


Fig. 4. Box plot of the error in rotation in degrees and translation in metres for combining sensor readings and performing separate optimisations.

can only be reduced by collecting data in more topologically varied environments, as opposed to typical urban roads. However, given the asynchronous nature of the sensors and the distance to most targets, these errors in translation will generally have little impact on high level perception tasks, or visualisation such as colouring a lidar point cloud as shown in Figure 6. Nevertheless, the variance estimated by our approach permits the user to decide whether a particular sensor's calibration needs further refinement.

VI. CONCLUSION

This paper has presented a new approach for calibrating multi-modal sensor arrays mounted on mobile platforms. Unlike previous automated methods, the new calibration approach does not require initial knowledge of the sensor positions and requires no markers or registration aids to be placed in the scene. These properties make our approach suitable for long-term autonomy and non-expert users. In urban environments, the motion-based module is able to provide accurate rotation and coarse translation estimates even when there is no overlap between the sensors. The method is complementary to previous techniques, so when sensors overlap exists, the calibration can be further refined utilising previous calibration methods. In addition, the ap-

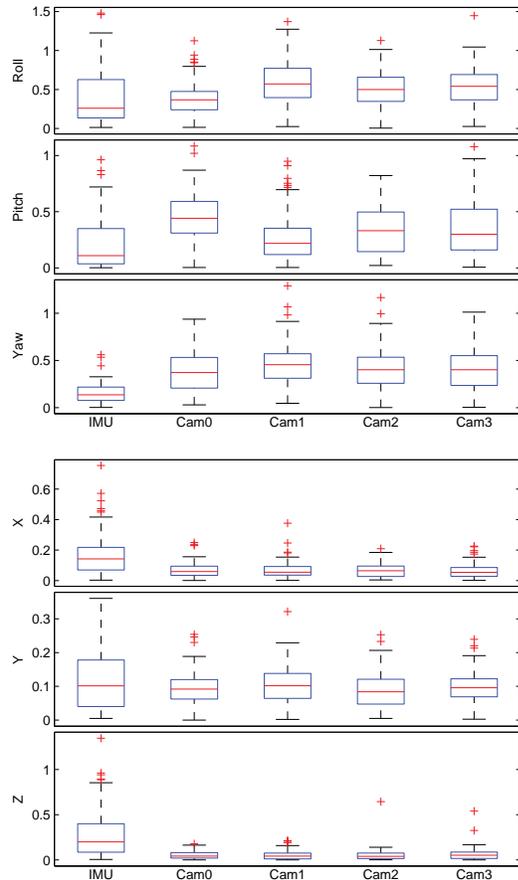


Fig. 5. Box plot of the error in rotation in degrees and translation in metres for performing the full calibration process.



Fig. 6. A velodyne scan coloured by one of the KITTI cameras.

proach provides an estimate of the variance of the calibration. The method was validated using scans from two separate mobile platforms.

ACKNOWLEDGEMENTS

This work has been supported by the Rio Tinto Centre for Mine Automation and the Australian Centre for Field Robotics, University of Sydney.

REFERENCES

[1] A. Geiger, F. Moosmann, O. Car, and B. Schuster, "Automatic camera and range sensor calibration using a single shot," *2012 IEEE International Conference on Robotics and Automation*, pp. 3936–3943, May 2012. [Online]. Available: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6224570>

[2] R. Unnikrishnan and M. Hebert, "Fast extrinsic calibration of a laser rangefinder to a camera," 2005. [Online]. Available: <http://repository.cmu.edu/robotics/339/>

[3] J. Levinson and S. Thrun, "Automatic Calibration of Cameras and Lasers in Arbitrary Scenes," in *International Symposium on Experimental Robotics*, 2012.

[4] G. Pandey, J. R. Mcbride, S. Savarese, and R. M. Eustice, "Automatic Targetless Extrinsic Calibration of a 3D Lidar and Camera by Maximizing Mutual Information," *Twenty-Sixth AAAI Conference on Artificial Intelligence*, vol. 26, pp. 2053–2059, 2012. [Online]. Available: <http://www.aaai.org/ocs/index.php/AAAI/AAAI12/paper/viewPDFInterstitial/5029/5371>

[5] A. Napier, P. Corke, and P. Newman, "Cross-Calibration of Push-Broom 2D LIDARs and Cameras In Natural Scenes," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2013.

[6] Z. Taylor and J. Nieto, "A Mutual Information Approach to Automatic Calibration of Camera and Lidar in Natural Environments," in *the Australian Conference on Robotics and Automation (ACRA)*, 2012, pp. 3–5. [Online]. Available: <http://www.araa.asn.au/acra/acra2012/papers/pap113.pdf>

[7] S. Schneider, T. Luettel, and H.-J. Wuensche, "Odometry-based online extrinsic sensor calibration," *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, no. 2, pp. 1287–1292, Nov. 2013. [Online]. Available: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6696515>

[8] M. Li, H. Yu, X. Zheng, and A. Mourikis, "High-fidelity Sensor Modeling and Self-Calibration in Vision-aided Inertial Navigation," *ee.ucr.edu*. [Online]. Available: <http://www.ee.ucr.edu/~mli/ICRA2014.pdf>

[9] L. Heng, B. Li, and M. Pollefeys, "CamOdoCal: Automatic intrinsic and extrinsic calibration of a rig with multiple generic cameras and odometry," ... *and Systems (IROS), 2013 IEEE/RSJ ...*, pp. 1793–1800, Nov. 2013. [Online]. Available: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6696592>

[10] Z. Taylor, J. Nieto, and D. Johnson, "MultiModal Sensor Calibration Using a Gradient Orientation Measure," *Journal of Field Robotics*, vol. 00, no. 0, pp. 1–21, 2014. [Online]. Available: <http://onlinelibrary.wiley.com/doi/10.1002/rob.21523/full>

[11] N. Andreff, "Robot Hand-Eye Calibration Using Structure-from-Motion," *The International Journal of Robotics Research*, vol. 20, no. 3, pp. 228–248, Mar. 2001. [Online]. Available: <http://ijr.sagepub.com/cgi/doi/10.1177/02783640122067372>

[12] J. Heller, M. Havlena, A. Sugimoto, and T. Pajdla, "Structure-from-motion based hand-eye calibration using L_{\inf} minimization," *Cvpr 2011*, pp. 3497–3503, Jun. 2011. [Online]. Available: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5995629>

[13] Z. Taylor, "Calibration Source Code." [Online]. Available: <http://www.zjtaylor.com/code>

[14] R. Wang, F. P. Ferrie, and J. Macfarlane, "Automatic registration of mobile LiDAR and spherical panoramas," *2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pp. 33–40, Jun. 2012. [Online]. Available: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6238912>

[15] Z. Taylor, J. Nieto, and D. Johnson, "Automatic calibration of multi-modal sensor systems using a gradient orientation measure," *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1293–1300, Nov. 2013. [Online]. Available: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6696516>

[16] P. Besl and N. McKay, "A Method for Registration of 3-D Shapes," *IEEE transactions on pattern analysis and machine intelligence*, vol. 14, no. 2, pp. 239–256, 1992.

[17] A. Harrison and P. Newman, "TICSynC: Knowing when things happened," *2011 IEEE International Conference on Robotics and Automation*, pp. 356–363, May 2011. [Online]. Available: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5980112>

[18] A. Geiger and P. Lenz, "Are we ready for autonomous driving? the kitti vision benchmark suite," *IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 3354–3361, Jun. 2012. [Online]. Available: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6248074>