Lost in Translation (and Rotation): Rapid Extrinsic Calibration for 2D and 3D LIDARs

Will Maddern, Alastair Harrison and Paul Newman

***Abstract*—This paper describes a novel method for determining the extrinsic calibration parameters between 2D and 3D LIDAR sensors with respect to a vehicle base frame. To recover the calibration parameters we attempt to optimize the quality of a 3D point cloud produced by the vehicle as it traverses an unknown, unmodified environment. The point cloud quality metric is derived from Rényi Quadratic Entropy and quantifies the compactness of the point distribution using only a single tuning parameter. We also present a fast approximate method to reduce the computational requirements of the entropy evaluation, allowing unsupervised calibration in vast environments with millions of points. The algorithm is analyzed using real world data gathered in many locations, showing robust calibration performance and substantial speed improvements from the approximations.**

I. Introduction

Light Detection and Ranging (LIDAR) sensors have emerged as a dominant sensor in mobile robotics over the past decade. A LIDAR is often the primary source of sensor data for robots performing SLAM [1], [2], 3D reconstruction [3], [4], obstacle detection and avoidance [5], feature detection and scene classification [6] as well as many other applications in the field of mobile robotics. The active nature of LIDARs make them an attractive sensor for outdoor mobile robots; LIDARs can detect scene structure and appearance regardless of background illumination.

The utility of a LIDAR as a sensor for outdoor mobile robots was demonstrated in the recent DARPA Grand Challenge [7] and Urban Challenge [8], with the majority of vehicles using a combination of 2D and 3D LIDARs [9]. For data consistency, each of these sensors required an extrinsic calibration to recover the offset between the sensor frame and the vehicle base frame [10]. The majority of current LIDAR calibration techniques involve one or more of the following: a supervised calibration procedure, known calibration targets, an artificial calibration environment, assumptions about the structure of the environment, arbitrary tuning parameters or lengthy computation time.

In this paper, we propose an unsupervised algorithm for determining the extrinsic calibration between at least one 3D laser and multiple 2D lasers and a vehicle base frame. Our proposed method has a number of advantages over existing methods; primarily, we make minimal assumptions about the test environment and do not require any environment modification (such as placement of calibration targets). We use an entropy-based point cloud quality metric which is well grounded in information theory, provides a wide basin of convergence and requires only one tuning parameter. Finally, we exploit properties of the entropy formulation to create an efficient tree-based implementation; this allows calibration in large test environments with millions of laser points. This adds only a single tuning parameter, which gracefully trades accuracy for speed. We evaluate the performance of our algorithm by determining calibration parameters over 20 different test locations, demonstrating excellent repeatability. We also show substantial computational savings over existing entropy-based approaches.

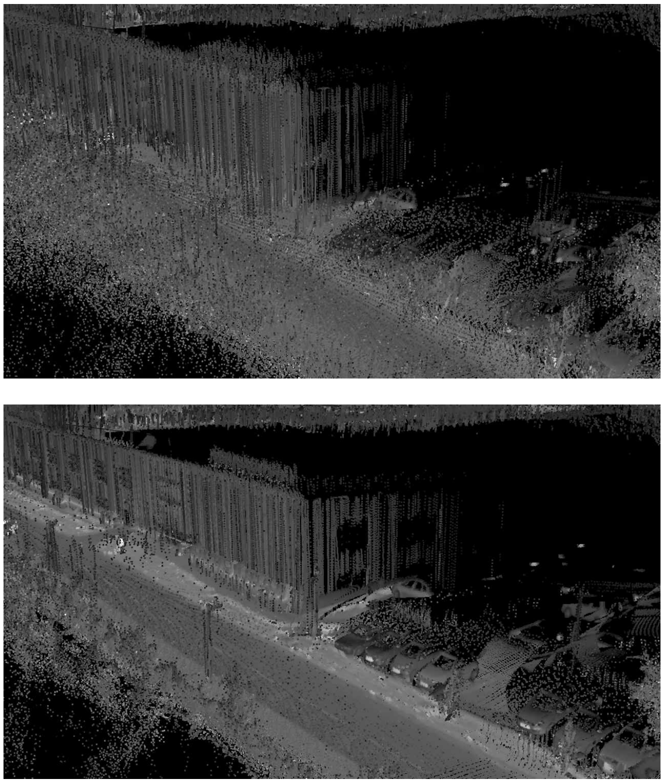

Figure 1: 3D point cloud visualization of a section of Begbroke Science Park before and after calibration. The 2D LIDAR is angled downwards by 8°, and therefore covers a volume up to 2m above the ground plane. In the lower image, good calibration is evident between the two point clouds, particularly on sections of the building, lamp posts and automobiles.

This work was supported by ARC Special Research Initiative on Thinking Systems TS0669699, an Australian Postgraduate Award and EPSRC Leadership Fellowship Grant EP/I005021/1.

Will Maddern is with the School of Electrical Engineering and Computer Science, Queensland University of Technology, Australia. w.maddern@qut.edu.au

Alastair Harrison and Paul Newman are with the Mobile Robotics Group, Dept. Engineering Science, University of Oxford, UK. {arh, pnewman}@robots.ox.ac.uk

II. Related Work

A number of authors have addressed the problem of determining the location of one or more LIDARs relative to a vehicle base frame. It is typically not feasible to measure the rigid transformation directly (and alignment errors of only 1° can result in measurement errors of over 0.25m at a distance of 15m), so the most common approach is to perform a calibration routine with all sensors rigidly attached to the vehicle as it moves through an environment.

One approach to recover the extrinsic calibration parameters is to include them as additional states in a metric SLAM system. The method presented in [11] constructs a hypergraph, which when solved using graph-based optimization provides an estimate of the path taken by a robot through an environment along with the local offsets of a 2D LIDAR in the x-y plane and the diameter of the robot's wheels. A similar approach is presented in [12], where manifold-based optimization is applied to solve a number of extrinsic calibration scenarios. An observability analysis of required vehicle motion for successful calibration is provided in [13], which calibrates the extrinsics of multiple 2D LIDAR sensors on a moving platform. However, these methods have so far been restricted to 2D LIDARs experiencing only planar motion.

Extrinsic calibration of 2D LIDAR sensors on a vehicle platform experiencing 3D motion presents a greater challenge. The approach taken in [14] uses a pole with retro-reflective tape applied in known locations as a calibration target in an environment with a flat ground plane. After driving the vehicle past the pole multiple times with a 2D LIDAR fixed rigidly to the vehicle, an optimization procedure determines the calibration parameters which provide the most accurate 3D reconstruction of the pole. More recently, [15] demonstrated a calibration technique using multiple retro-reflective markers in an unknown environment and an efficient second-order cone-based optimization. Both of these methods rely on known calibration targets to aid in the construction of a 3D point cloud from 2D scans.

A 2D LIDAR does not generally provide enough coverage to construct a 3D point cloud for calibration purposes. Fixed 2D LIDARs are typically configured as either forward facing or push-broom. Forward-facing 2D LIDARs observe a planar 'slice' of the environment from many different locations, but typically struggle with vehicle pitch and roll and related effects such as ground-strike in general 3D vehicle motion. The availability of full-field 3D LIDAR systems such as the Velodyne [16] permits extrinsic calibration with no prior knowledge of the environment and no requirement for calibration targets; instead, extrinsic calibration parameters are chosen to maximize a point cloud quality metric.

The approach presented in [17] finds both extrinsic and intrinsic calibration parameters for a Velodyne HDL-64E on a moving vehicle platform. The optimization function seeks to minimize ICP-like point-to-plane distances using surface normals generated from local neighborhoods. A grid search strategy is used to derive the optimal calibration parameters. While this method does not require any modification of the environment, the normal computation step will perform best in environments which are locally planar (generally valid in urban areas).

An alternative metric for 3D point cloud quality based on Rényi's Quadratic Entropy (RQE) [18] is presented in [19], where it is used to calibrate the intrinsic parameters of a multi-beam rotating 3D LIDAR. The scanned workspace is represented using a Gaussian Mixture Model (GMM). By evaluating the entropy of this model, the quality of the point cloud may be quantified. Similar methods have seen use in deformable image and shape registration [20], [21], as a probabilistic alternative to ICP for point registration and scan matching [22], [23] and clustering and machine learning [24] amongst others. Since there is no explicit data association step required, entropy-based methods provide a large basin of convergence. However, they require an exhaustive pairwise correlation which is computationally expensive for large datasets. Section VI introduces a method to reduce the expense by exploiting the approximate sparsity of the cost function.

III. Problem Formulation

While our approach is not limited to a particular configuration of sensors, we are specifically interested in the case of a mobile platform with at least one 3D LIDAR and any number of 2D LIDARs, all of which are rigidly attached to the vehicle. The platform must also provide a mechanism for determining the vehicle trajectory, in order that LIDAR measurements may be projected relative to it. Note that the trajectory needs only to be *locally* metrically accurate; global metric accuracy is not necessary. Thus it could come from a GPS+Inertial system, relative visual odometry, or a LIDAR SLAM system for example.

Our algorithm is able to make use of the relative covariance between points, so a system providing that is preferable. In this work, we use an accurate Inertial Navigation System (INS) which provides pose error estimates (the state covariance matrix from a Kalman filter used to combine inertial data with GPS). We assume the intrinsics of each sensor have been correctly calibrated; the ranges reported by each LIDAR relative to the sensor base frame must be locally consistent. For single-beam 2D LIDARs the factory calibration is generally sufficient; in the case of a Velodyne or other multi-beam 3D LIDAR, additional intrinsic calibration stages [19], [17] may be necessary to achieve the desired accuracy.

A. Kinematic Chain

Here we derive the kinematic chain for the platform and sensors based on a global trajectory estimate. We reiterate that this is not a specific requirement of the method, and that it would work equally well on relative trajectory segments using a sliding-window approach to bring the trajectory into a local metric frame. Consider a 3D laser range finder L which takes measurements $\mathbf{Z} = \{\mathbf{z}_1 \dots \mathbf{z}_N\}$, where each measurement consists of a point $\mathbf{z}_i = [x_i, y_i, z_i]^T$in the sensor base frame and corresponds to an object in the world

$\mathbf{X} = \{\mathbf{x}_1 \dots \mathbf{x}_N\}$. The history of vehicle poses is denoted by $\mathbf{Y} = \{\mathbf{y}_1 \dots \mathbf{y}_N\}$, where each pose consists of a global location and orientation $\mathbf{y}_i = [x_{Gi}, y_{Gi}, z_{Gi}, \alpha_{Gi}, \beta_{Gi}, \phi_{Gi}]^T$ and covariance matrix $\mathbf{Q}_i$. The sensor model h is defined as $\mathbf{z}_i = h(\mathbf{x}_i|\mathbf{y}_i, \Theta)$, where Θ is the set of six extrinsic calibration parameters which define the local offset between the vehicle origin and the sensor base frame:

$$\Theta = [x_L, y_L, z_L, \alpha_L, \beta_L, \phi_L]^T \tag{1}$$

To estimate the position of the measured points $\hat{\mathbf{x}}_i$ given the measurements and global poses we apply the inverse sensor model $\hat{\mathbf{x}}_i = h^{-1}(\mathbf{z}_i|\mathbf{y}_i, \Theta)$, which corresponds to the kinematic chain,

$$\begin{aligned} h^{-1}(\mathbf{z}_i|\mathbf{y}_i, \Theta) &= \mathbf{T}(x_{Gi}, y_{Gi}, z_{Gi})\,\mathbf{R}(\alpha_{Gi}, \beta_{Gi}, \phi_{Gi}) \\ &\quad \mathbf{T}(x_L, y_L, z_L)\,\mathbf{R}(\alpha_L, \beta_L, \phi_L)\,\mathbf{z}_i \end{aligned} \tag{2}$$

Here $\mathbf{T}(x, y, z)$ represents a translation along each respective axis and $\mathbf{R}(\alpha, \beta, \phi)$ represents a rotation of roll angle α, pitch angle β and yaw angle ϕ. The covariance of each estimated point due to global position error is calculated as follows:

$$\Sigma_i = J_i \mathbf{Q}_i J_i^T, \text{ where } J_i = \frac{\delta h^{-1}(\mathbf{z}_i|\mathbf{y}_i, \Theta)}{\delta \mathbf{y}_i} \tag{3}$$

Combining the estimated position of objects over a series of measurements produces an estimated point cloud $\hat{\mathbf{X}} = \{\hat{\mathbf{x}}_1 \dots \hat{\mathbf{x}}_N\}$, with point covariances $\mathbf{\Sigma} = \{\Sigma_1 \dots \Sigma_N\}$.

B. Clock Synchronization

The quality of the measured point cloud $\hat{\mathbf{X}}$ is not only dependent on the extrinsic calibration parameters Θ but also the accuracy of the timestamps for each measurement. As we are calibrating multiple heterogeneous sources of measurements, each with its own internal clock, we require a method of mapping timestamps from measurement devices to a central clock. We use the TICSync library [25] to determine the frequency and offset of each measurement device clock relative to the central clock. This is performed on line using a variant of the efficient convex hull algorithm outlined in [26].

IV. Point Cloud Quality Evaluation

Using only the history of measurements $\mathbf{Z}$ and vehicle poses $\mathbf{Y}$, we wish to maximize the value of a point cloud quality function $E(\Theta|\mathbf{Z}, \mathbf{Y})$ providing the most likely estimate for the extrinsic calibration parameters Θ. We wish to define this quality function as a measure of point cloud 'crispness', without making assumptions about the underlying structure of the calibration environment (such as requiring the existence of large planar surfaces). Our cost function is based on Rényi Quadratic Entropy (RQE) because, as we will see, it offers excellent performance, a large convergence basin, makes no explicit assumptions about the environment and requires only a single tuning parameter.

We assume that our point cloud measurements $\hat{\mathbf{X}} = \{\hat{\mathbf{x}}_1 \dots \hat{\mathbf{x}}_N\}$ are drawn from an underlying distribution $p(\mathbf{x})$, which represents the probability of drawing a measurement from a location $\mathbf{x}$. We apply Parzen Window density estimation [27] to represent $p(\mathbf{x})$ as a Gaussian Mixture Model (GMM),

$$p(\mathbf{x}) = \frac{1}{N}\sum_{i=1}^{N} G\left(\mathbf{x} - \hat{\mathbf{x}}_i, \Sigma_i + \sigma^2\mathbf{I}\right) \tag{4}$$

where $G(\mathbf{x} - \mu, \Sigma)$ is a Gaussian with mean μ and covariance Σ. We use an isotropic kernel with variance σ^2 convolved with the vehicle localization uncertainty, Σ_i for each measurement. The parameter σ is the only tuning parameter of the system, being related to the expected measurement noise of the sensor, and the sampling density of the dataset. The 'crispness' of the point cloud can now be expressed in terms of the entropy of $p(\mathbf{x})$.

An entropy measure proposed by Rényi [18] offers an efficient way to quantify the compactness of a GMM distribution [28]. Specifically, the Rényi Quadratic Entropy of a stochastic variable X with probability density $p(\mathbf{x})$ is defined as

$$H_{RQE}[X] = -\log \int p(\mathbf{x})^2 d\mathbf{x}. \tag{5}$$

Substituting the Gaussian Mixture Model of Equation 4 into equation 5 gives

$$\begin{aligned} H_{RQE}[\hat{X}] &= -\log \int \left(\frac{1}{N}\sum_{i=1}^{N} G\left(\mathbf{x} - \hat{\mathbf{x}}_i, \Sigma_i + \sigma^2\mathbf{I}\right)\right)^2 d\mathbf{x} \\ &= -\log\Bigg(\frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N}\int G\left(\mathbf{x} - \hat{\mathbf{x}}_i, \Sigma_i + \sigma^2\mathbf{I}\right) \\ &\qquad G\left(\mathbf{x} - \hat{\mathbf{x}}_j, \Sigma_j + \sigma^2\mathbf{I}\right) d\mathbf{x}\Bigg) \end{aligned} \tag{6}$$

By noting the following Gaussian integral identity,

$$\begin{aligned} &\int G(\mathbf{x} - \hat{\mathbf{x}}_1, \Sigma_1)\, G(\mathbf{x} - \hat{\mathbf{x}}_2, \Sigma_2)\, d\mathbf{x} = \\ &\qquad G(\hat{\mathbf{x}}_1 - \hat{\mathbf{x}}_2, \Sigma_1 + \Sigma_2), \end{aligned} \tag{7}$$

simplification of Equation 6 yields a closed-form representation for the Rényi Quadratic Entropy of the GMM:

$$\begin{aligned} H_{RQE}[\hat{\mathbf{X}}] = -\log\Bigg(&\frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N} \\ &G\left(\hat{\mathbf{x}}_i - \hat{\mathbf{x}}_j, \Sigma_i + \Sigma_j + 2\sigma^2\mathbf{I}\right)\Bigg) \end{aligned} \tag{8}$$

This expression can be interpreted as an information-theoretic measure of the compactness of the points in $\hat{\mathbf{X}}$. For the purposes of optimization we note that log is a monotonic operator and that the scale factor is unnecessary, so these terms are removed to produce our point cloud quality function:

$$E[\hat{\mathbf{X}}] = \sum_{i=1}^{N}\sum_{j=1}^{N} G\left(\hat{\mathbf{x}}_i - \hat{\mathbf{x}}_j, \Sigma_i + \Sigma_j + 2\sigma^2\mathbf{I}\right) \tag{9}$$

For a given value of σ, the cost function depends only on pairwise distances between points in $\hat{\mathbf{X}}$ and the associated global position uncertainty Σ of each point. As it stands, this has an $O(N^2)$ evaluation cost, but an efficient approximation is presented in Section VI.

V. Extrinsic Calibration

Given a vehicle trajectory and recordings of multiple 2D and 3D LIDARs $L_1 \dots L_n$, we wish to estimate the relative locations $\Theta_1 \dots \Theta_n$ of each LIDAR. This requires movement of the platform as objects in the environment must be observed from at least two locations. To effectively constrain the calibration axes, the vehicle trajectory must include periods of acceleration in all 6 degrees of freedom. We assume that an initial guess Θ_0 is available from manual measurements of each sensor's location, though these need not be highly accurate. As in [17], we make no assumptions about the environment other than that it is generally static.

A. 3D LIDAR to Vehicle Frame Calibration

As a 3D LIDAR provides good coverage of the local environment, the calibration of offsets between a 3D LIDAR and the vehicle INS system is straightforward since the vast majority of measurements to objects $\mathbf{X}$ are made from at least two locations. The point cloud quality function is formed by substituting the inverse sensor model into the cost function of Equation 9:

$$E\left(\Theta|\mathbf{Z},\mathbf{Y}\right) = \sum_{i=1}^{N}\sum_{j=1}^{N} G\Big(h^{-1}\left(\mathbf{z}_i|\mathbf{y}_i,\Theta\right) - h^{-1}\left(\mathbf{z}_j|\mathbf{y}_j,\Theta\right), J_i\mathbf{Q}_iJ_i^T + J_j\mathbf{Q}_jJ_j^T + 2\sigma^2\mathbf{I}\Big) \quad (10)$$

The optimal offset parameters $\hat{\Theta}$ are recovered by non-linear maximization of the cost function, so that

$$\hat{\Theta} = \underset{\Theta}{\operatorname{argmax}} E\left(\Theta|\mathbf{Z},\mathbf{Y}\right) \quad (11)$$

As the cost function makes use of convolved Gaussians which have continuous analytical derivatives, the optimization can be performed using Newton's method or other gradient-based optimization algorithms.

B. 2D LIDAR to INS and 3D LIDAR

Once the vehicle has a calibrated 3D LIDAR, we can use the structure of the global 3D point cloud (or locally metrically accurate sections rendered from a relative trajectory) to calibrate the 2D LIDAR. To achieve this, we use the Jensen-Rényi Divergence [20] between the two point clouds, which boils down to finding the entropy contribution of only the cross-terms between the proposed 2D LIDAR point cloud and the calibrated 3D LIDAR point cloud. Intuitively, the optimization should not be dominated by the intrinsic crispness of either point cloud; it is their similarity that we wish to maximize. Given a set $\mathbf{Z}$ of N observations from a 2D LIDAR, a corresponding set of global poses $\mathbf{Y}$ and a set $\hat{\mathbf{X}}$ of M calibrated points from a 3D LIDAR (with corresponding covariances $\mathbf{\Sigma}$), taking the Jensen-Rényi Divergence between the two point clouds leads to a cross-term entropy function,

$$E(\Theta|\mathbf{Z},\mathbf{Y},\hat{\mathbf{X}}) = \sum_{i=1}^{N}\sum_{j=1}^{M} G\Big(h^{-1}(\mathbf{z}_i|\mathbf{y}_i,\Theta) - \hat{\mathbf{x}}_j, \quad (12) \\ J_i\mathbf{Q}_iJ_i^T + \Sigma_j + 2\sigma^2\mathbf{I}\Big)$$

The offset between the 2D LIDAR and the vehicle frame is recovered by maximization of this cost function, so that

$$\hat{\Theta} = \underset{\Theta}{\operatorname{argmax}} E\left(\Theta|\mathbf{Z},\mathbf{Y},\hat{\mathbf{X}}\right) \quad (13)$$

Once again we apply a Newton method to perform the optimization, as for the 3D LIDAR in Section V-A.

VI. Computational Enhancements

The cost function of Equation 9 is $O(N^2)$ (where N is the size of the point cloud) as it requires the computation of all pairwise entropy contributions. For larger environments this becomes prohibitively expensive, so we now present a series of computational enhancements to keep the optimization tractable. A trivial improvement can be found by seeing that Equation 9 evaluates contributions for each pair of points twice - once in each direction. Computation time can be halved by evaluating in one direction only, so that

$$E[\hat{\mathbf{X}}] = 2\sum_{i=1}^{N}\sum_{j=i}^{N} G\left(\hat{\mathbf{x}}_i - \hat{\mathbf{x}}_j, \Sigma_i + \Sigma_j + 2\sigma^2\mathbf{I}\right). \quad (14)$$

Notice that the inner sum now only covers the range $[i, n]$. Given that the typical values chosen for σ are small relative to the absolute size of the global point cloud, it is likely that the vast majority of point pairs produce negligible entropy contributions. As we have knowledge of the location and covariances of each point pair we can set an upper bound on the pairwise distance above which all entropy contributions are negligible:

$$G\left(\mathbf{x} - \mu, \Sigma\right) \approx 0, \quad \forall \mathbf{x} : \|\mathbf{x} - \mu\| \geq k\lambda_1(\Sigma) \quad (15)$$

Here we adopt the spectrum notation for eigenvalues, where the eigenvalues of an $n \times n$ Hermitian matrix A are represented as the ordered set $\{\lambda_1(A) \geq \lambda_2(A) \geq \dots \lambda_n(A)\}$. The parameter k is used to adjust the threshold as a proportion of the maximum eigenvalue $\lambda_1(\Sigma)$. Use of this approximation greatly sparsifies the structure of the cost function since it is not necessary to evaluate spatially distant point pairs. Notice that using this approximation in the RQE cost function requires that for every pair of points $(\hat{\mathbf{x}}_i, \hat{\mathbf{x}}_j)$ we evaluate a threshold, $\|\hat{\mathbf{x}}_i - \hat{\mathbf{x}}_j\| \geq k\lambda_1\left(\Sigma_i + \Sigma_j + 2\sigma^2\mathbf{I}\right)$. Since this depends on the covariance of both points, it still requires that we visit all possible pairs. Recall that covariance matrices are symmetric positive-definite and therefore Hermitian. For Hermitian matrices the following inequality holds [29]: $\lambda_1(\Sigma_1) + \lambda_1(\Sigma_2) \geq \lambda_1(\Sigma_1 + \Sigma_2)$, which leads us to a more conservative threshold,

$$\begin{aligned} \lambda_1(\Sigma_i + \Sigma_j + 2\sigma^2\mathbf{I}) &\leq \lambda_1(\Sigma_i) + \lambda_1(\Sigma_j) + 2\sigma^2 \quad (16) \\ &\leq 2\max\left(\lambda_1(\Sigma_i), \lambda_1(\Sigma_j)\right) + 2\sigma^2 \end{aligned}$$

This new form allows us to obtain an efficient algorithm for approximate RQE evaluation. First we compute the maximum eigenvalues for all point covariances, $\Sigma_i \in \mathbf{\Sigma}$ in $O(N)$ time. We then sort the points in descending order by their maximum eigenvalue, and iterate through them. For each point we perform a search for all other points within

Algorithm 1 Fixed-radius approximate RQE

```
procedure RQEAPPROX($\hat{\mathbf{X}}, \mathbf{\Sigma}, \sigma, k$)
    $E \leftarrow 0$
    $\lambda_{1\dots N} \leftarrow [\text{MAXEIGENVAL}(\Sigma_i) \mid i \in [1 \dots N]]$
    ▷ Rank $\lambda_{1\dots N}$ so that $\lambda_{r_i} \leq \lambda_{r_{i-1}} \quad \forall i \in [2 \dots N]$
    $r_{1\dots N} \leftarrow \text{RANKORDERDESCENDING}(\lambda_{1\dots N})$
    for $i \leftarrow 1, N$ do
        $R \leftarrow 2k\left(\lambda_{r_i} + \sigma^2\right)$ ▷ Search radius
        for all $j \in \text{NEIGHBORHOOD}(\hat{\mathbf{x}}_{\mathbf{r_i}}, \hat{\mathbf{X}}, R)$ do
            if $\lambda_{r_i} > \lambda_j$ then
                $E \leftarrow E + G\left(\hat{\mathbf{x}}_{\mathbf{r_i}} - \hat{\mathbf{x}}_{\mathbf{j}}, \Sigma_{r_i} + \Sigma_j + 2\sigma^2\mathbf{I}\right)$
            end if
        end for
    end for
    return $2E$          ▷ The RQE result
end procedure
```

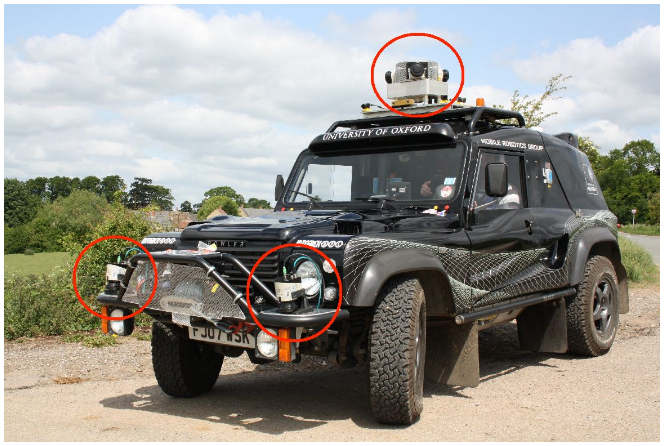

Figure 2: The Wildcat vehicle has a spinning 3D LIDAR on the roof, and two 2D LIDARs on the front bumper at different elevation angles (sensors circled in red). The INS system is mounted over the rear axle.

a radius $R_i = 2k\left(\lambda_1\left(\Sigma_i\right) + \sigma^2\right)$. Finally, we evaluate the entropy contribution between $\hat{\mathbf{x}}_{\mathbf{i}}$ and all neighbors that appear later in the list (thus having smaller maximum eigenvalue). Crucially, this ensures that entropy contributions are computed in one direction only, and that the search neighborhood is monotonically decreasing as evaluation of the cost function progresses. Algorithm 1 shows the key steps.

Note that two components of this algorithm are particularly suited for the application of tree-based data structures; the eigenvalue ranking function $r(i)$ can be efficiently generated using an $O(N \log N)$ sort, and the point cloud $\hat{\mathbf{X}}$ can be stored in a *k*-d tree which allows fixed radius neighborhood searches in worst-case $O(N^{2/3} + p)$ time, where p is the number of neighbors retrieved. For large datasets, these tree-based representations provide significant computational savings over exhaustive searching of the point cloud, delivering a cost function that may be evaluated in $O(N \log N + N^{5/3} + N\bar{p})$ time, where $\bar{p}$ is the mean neighborhood size retrieved by the k-d tree.

VII. Experimental Results

A. Experimental Setup

The mobile platform used for our experiments is the Wildcat, pictured in Figure 2. Vehicle pose is provided by an Oxford Technical Solutions (OxTS) RT-3042, which combines inertial measurements with differential GPS updates using a Kalman filter. 6-DoF pose estimates and pose uncertainty estimates are provided at 100Hz with a stated position accuracy of 0.4m CEP and orientation accuracy of 0.03° (1σ) in pitch and roll and 0.1° (1σ) in yaw. The Wildcat is equipped with a roof-mounted 3D LIDAR comprising of 3 SICK LMS-151 sensors fixed to a rotating plate [19], and a pair of front bumper mounted SICK LMS-151 sensors. Each LMS-151 has a field of view of 270 degrees and a measurement range of 50m with a stated accuracy of 12mm (1σ).

The calibration experiments were conducted at the Begbroke Science Park near Oxford, UK. The Wildcat was driven multiple times around the 0.7km perimeter road, passing natural features such as trees and hedges along with man-made features such as buildings, car parks and road signs. No reflective markers or other calibration aids were placed in the environment, and the only environment-specific aspect to data collection was the choice of collection time (to minimize the number of pedestrians, cars and other dynamic objects in the data). The trajectory was spliced into 20 distinct 'locations' and an independent calibration performed for each one, in order to quantify the precision of our method.

Data from the INS, 3D LIDAR and 2D LIDARs was recorded using the on-board PC and the calibration was performed as an off-line process for convenience, though an on-line implementation is quite possible. The core algorithm to evaluate the cost function was implemented in C++, taking advantage of the 'eigen'[1] library for matrix manipulation and eigendecomposition and a *k*-d tree based approximate nearest neighbor library[2] for the fixed-radius search.

B. Calibration Results

Table 1 lists the extrinsic calibration values determined for the 3D LIDAR and the downward-facing 2D LIDAR for σ = 5mm aggregated over 20 different locations. The mean 3D LIDAR calibration was used to generate the global point cloud for calibration of the 2D LIDAR.

Note the large difference between the manually measured yaw angle and the calibrated yaw angle of the 3D LIDAR. This is not due to a physical 9.6° mounting error of the base plate; the intrinsic calibration of the 3D LIDAR performed in [19] does not derive the angular offset between the base plate and the encoder origin, so it manifests itself as a rotation of the sensor frame.

Figure 3 illustrates the distribution of extrinsic parameters across all 20 different calibration locations. To facilitate comparison between axes, the mean value for each axis has been subtracted.

[1] http://eigen.tuxfamily.org/

[2] http://www.cs.umd.edu/~mount/ANN/

	3D LIDAR			2D LIDAR		
	Seed Value	Mean	Std Dev	Seed Value	Mean	Std Dev
x_L(m)	1.13	1.153	0.004	3.3	3.269	0.020
y_L(m)	0	0.000	0.001	0.67	0.761	0.013
z_L(m)	-1.13	-1.145	0.005	0.4	0.390	0.011
α_L(°)	0	0.006	0.001	-8	-6.904	0.299
β_L(°)	0	0.002	0.001	180	179.03	0.466
ϕ_L(°)	90	99.639	0.257	270	270.18	0.210

Table I: Extrinsic calibration parameters obtained by the optimization procedure for σ = 5mm. Values are an ensemble average over 20 different locations.

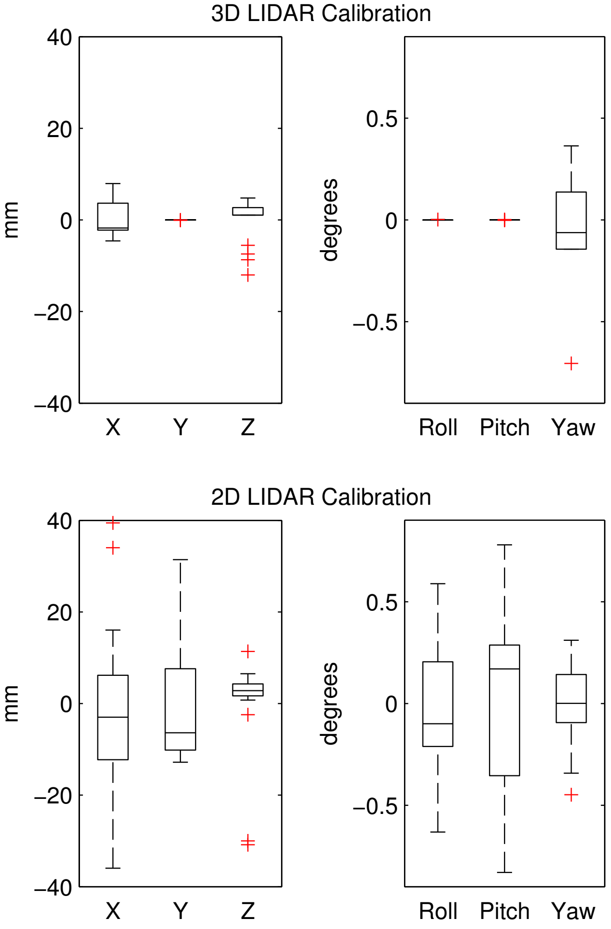

Figure 3: Calibration parameter box-plots about mean for 3D LIDAR and 2D LIDAR for 20 different locations. The 3D LIDAR problem is better conditioned, so the spread of calibration results is tighter than for the 2D LIDAR.

Figure 1 presents qualitative results in the form of a visualization of the point clouds produced by the 2D and 3D LIDAR before and after calibration. The calibrated point cloud appears crisp and the correspondence between the 2D and 3D LIDAR is apparent.

C. Performance

Figure 4 demonstrates how, for an 800,000 point section of the dataset, the efficiency and accuracy of our approximate RQE method vary with the k parameter. All experiments were performed using a single core of a 2.2GHz Intel Core i5 processor. The exhaustive pairwise approach requires over 15,000 seconds to evaluate the RQE cost function. The approximate method requires only 7 seconds to calculate a cost value within 5% of the exhaustive method and only 20 seconds to evaluate a cost value within 0.1% of the exhaustive method.

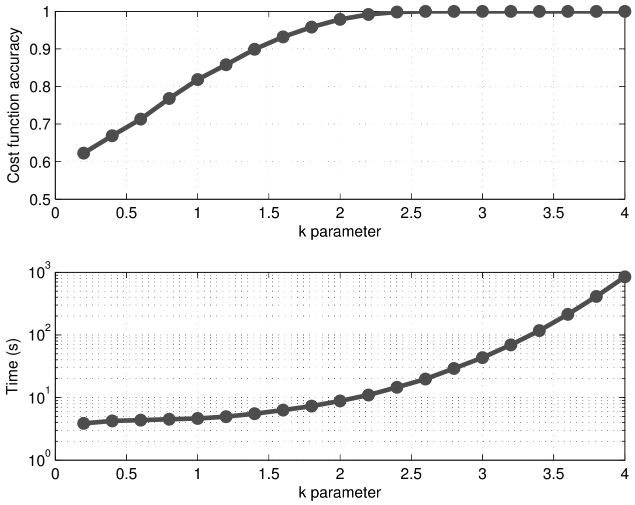

Figure 4: Accuracy of the approximate RQE algorithm relative to the full exhaustive search, for a set of 800,000 points. The k parameter allows us to make a smooth trade-off between efficiency and accuracy. Evaluating the full $O(N^2)$ cost function takes over 15000 seconds, and is essentially intractable for our purposes.

D. Discussion

In certain sections of the driven trajectory, the vehicle lost GPS lock when it passed under trees or close to buildings which explains some of the calibration outliers for certain locations. Nonetheless, the results demonstrate good precision for the 3D LIDAR extrinsic calibration algorithm, with lateral and vertical displacement tightly pinned down. The greater variation in x (forwards/backwards) is likely down to the manner in which the data was gathered; most of the time the vehicle was driving straight on open road with free space in front and behind and relatively featureless buildings or bushes to the sides. Thus a calibration error in x simply shifts the points in a way which does not significantly affect the ‘crispness’ of the point cloud. Conversely an error in lateral calibration is quickly identified by a pitching or rolling motion, which would cause the observations of the walls of the building to distort. The 2D LIDAR calibration is unsurprisingly not as precise as the 3D calibration, yet it certainly has greater precision than would be achieved by hand. Deliberate attempts to exercise all 6 degrees of freedom of the vehicle in a suitably complex environment may yield greater precision.

Empirically we find that the basin of convergence for the full RQE cost function with wide kernels is impressively large, with successful convergence from initial guesses of a meter away. As the σ value (and therefore the radius of influence of each point) is reduced, the algorithm becomes faster and the convergence basin shrinks gracefully. See [19] for a discussion on the effect of the kernel parameter, σ.

We performed a second set of experiments where we made no use of the measurement covariances from the INS. In most cases precision was around 5 times lower, which clearly shows the benefit of using non-isotropic kernels.

VIII. Conclusions

Though we obtain the vehicle trajectory using an INS, there is no reason why it should not come from a different source such as visual odometry or LIDAR SLAM, even if the trajectory is only locally metrically accurate. In future work we intend to investigate the use of other sources of trajectory information for sensor calibration. There is also further work to be done on intelligently choosing GMM kernel size in the RQE cost function, to take into account the varying laser sample density on different surfaces in the environment.

We have presented a method for fully automatic extrinsic calibration between 2D and 3D LIDAR sensors and the vehicle frame. Our algorithm has a principled information theoretic basis and has only two tuning parameters. It provides significant computational savings over an exhaustive RQE evaluation, in exchange for a graceful degradation of accuracy and convergence basin. This rapid and repeatable calibration in unstructured natural environments (without the need for special calibration targets) results in a procedure which can tractably be performed on-line. What we have proposed and demonstrated here is in stark contrast to the traditional approach of carrying out infrequent and expensive manual calibrations whenever the sensors are adjusted. This is no-hassle calibration.

References

[1] D. Cole and P. M. Newman, “Using laser range data for 3D SLAM in outdoor environments,” in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Orlando Florida USA, May 2006.

[2] S. Thrun, “Simultaneous localization and mapping,” *Robotics and cognitive approaches to spatial mapping*, pp. 13–41, 2008.

[3] P. Newman, D. Cole, and K. Ho, “Outdoor SLAM using visual appearance and laser ranging,” *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, pp. 1180 – 1187, 2006.

[4] D. Borrmann, J. Elseberg, K. Lingemann, A. Nüchter, and J. Hertzberg, “Globally consistent 3D mapping with scan matching,” *Robotics and Autonomous Systems*, vol. 56, no. 2, pp. 130–142, 2008.

[5] C. Urmson, J. Anhalt, H. Bae, J. A. D. Bagnell, and E. Al, “Autonomous driving in urban environments: Boss and the urban challenge,” *Journal of Field Robotics Special Issue on the 2007 DARPA Urban Challenge, Part I*, vol. 25, no. 1, pp. 425–466, June 2008.

[6] P. Newman, G. Sibley, M. Smith, M. Cummins, A. Harrison, C. Mei, I. Posner, R. Shade, D. Schroeter, L. Murphy *et al.*, “Navigating, recognizing and describing urban spaces with vision and lasers,” *The International Journal of Robotics Research*, vol. 28, no. 11-12, p. 1406, 2009.

[7] S. Thrun, “Winning the DARPA grand challenge,” *Machine Learning: ECML 2006*, pp. 4–4, 2006.

[8] K. Iagnemma, M. Buehler, and S. Singh, “Special issue on the 2007 DARPA urban challenge,” *Journal of Field Robotics*, vol. 25, pp. 423–860, 2008.

[9] M. Buehler, K. Iagnemma, and S. Singh, “Special issue on the 2007 DARPA urban challenge, part II,” *Journal of Field Robotics*, vol. 25, no. 9, pp. 567–724, 2008.

[10] J. Levinson, J. Askeland, J. Becker, J. Dolson, D. Held, S. Kammel, J. Kolter, D. Langer, O. Pink, V. Pratt *et al.*, “Towards fully autonomous driving: Systems and algorithms,” in *Intelligent Vehicles Symposium (IV), 2011 IEEE*. IEEE, 2011, pp. 163–168.

[11] R. Kümmerle, G. Grisetti, and W. Burgard, “Simultaneous Calibration, Localization, and Mapping.” in *in Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2011.

[12] R. Wagner, O. Birbach, and U. Frese, “Rapid development of manifold-based graph optimization systems for multi-sensor calibration and slam,” in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*. IEEE, 2011, pp. 3305–3312.

[13] J. Brookshire and S. Teller, “Automatic calibration of multiple coplanar sensors,” in *Proceedings of Robotics: Science and Systems*, Los Angeles, CA, USA, June 2011.

[14] J. Underwood, A. Hill, and S. Scheding, “Calibration of range sensor pose on mobile platforms,” in *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*. IEEE, 2007, pp. 3866–3871.

[15] C. Gao and J. Spletzer, “On-line calibration of multiple LIDARs on a mobile vehicle platform,” in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*. IEEE, 2010, pp. 279–284.

[16] VelodyneHDL, *Velodyne’s HDL-64E: A High Definition Lidar Sensor for 3-D Applications*, 2007.

[17] J. Levinson and S. Thrun, “Unsupervised calibration for multi-beam lasers,” in *In Proceedings of the International Symposium on Experimental Robotics (ISER)*, 2010.

[18] A. Rényi, “On measures of entropy and information,” in *Fourth Berkeley Symposium on Mathematical Statistics and Probability*, 1961, pp. 547–561.

[19] M. Sheehan, A. Harrison, and P. Newman, “Automatic self-calibration of a full field-of-view 3D n-laser scanner,” in *In Proceedings of the International Symposium on Experimental Robotics (ISER2010)*, New Delhi and Agra, India, December 2010.

[20] F. Wang, T. F. Syeda-Mahmood, B. C. Vemuri, D. Beymer, and A. Rangarajan, “Closed-form jensen-renyi divergence for mixture of gaussians and applications to group-wise shape registration.” in *MICCAI (1)*, ser. Lecture Notes in Computer Science, G.-Z. Yang, D. J. Hawkes, D. Rueckert, J. A. Noble, and C. J. T. 0002, Eds., vol. 5761. Springer, 2009, pp. 648–655.

[21] Y. He, A. B. Hamza, and H. Krim, “A generalized divergence measure for robust image registration,” *IEEE Transactions on Signal Processing*, vol. 51, pp. 1211–1220, 2003.

[22] Y. Tsin and T. Kanade, “A correlation-based approach to robust point set registration,” in *Computer Vision-ECCV 2004*. Springer, 2004, pp. 558–569.

[23] B. Jian and B. Vemuri, “Robust point set registration using gaussian mixture models,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 33, no. 8, pp. 1633 –1645, aug. 2011.

[24] R. Jenssen, K. Hild, D. Erdogmus, J. Principe, and T. Eltoft, “Clustering using renyi’s entropy,” *Neural Networks, 2003. Proceedings of the International Joint Conference on DOI - UR -*, vol. 1, pp. 523– 528 vol.1, 2003.

[25] A. Harrison and P. Newman, “TICSync: Knowing when things happened,” in *Proc. IEEE International Conference on Robotics and Automation (ICRA2011)*, Shanghai, China, May 2011.

[26] L. Zhang, Z. Liu, and C. Honghui Xia, “Clock synchronization algorithms for network measurements,” in *INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 1. IEEE, 2002, pp. 160–169.

[27] E. Parzen, “On estimation of a probability density function and mode,” *The annals of mathematical statistics*, vol. 33, no. 3, pp. 1065–1076, Jan 1962. [Online]. Available: http://www.jstor.org/stable/2237880

[28] J. Principe and D. Xu, “Information-theoretic learning using Renyi’s quadratic entropy,” in *Proceedings of the 1st International Workshop on Independent Component Analysis and Signal Separation, Aussois, France*. Citeseer, 1999, pp. 407–412.

[29] A. Knutson and T. Tao, “Honeycombs and sums of hermitian matrices,” *Notices of the American Mathematical Society (AMS)*, vol. 48, no. 2, pp. 175–186, 2011.