Exploiting Known Unknowns: Scene Induced Cross-Calibration of Lidar-Stereo Systems*

Terry Scott, Akshay A. Morye, Pedro Piniés, Lina M. Paz, Ingmar Posner, Paul Newman

Abstract— We propose an automatic, targetless, data-driven, extrinsic calibration method to calibrate push-broom 2D lidars with a multi-camera system. The calibration problem is decoupled into alternating optimisers over two hierarchical levels, where both levels are linked with a penalty term. The lower-level optimises the six degrees-of-freedom (DoF) rigid-body transforms between the lidar and each camera of the multi-camera unit by minimising the Normalised Information Distance between intensity measurements obtained from both sensor modalities. The upper-level minimises a nonlinear least squares error between the lower-level solutions. We describe the theory, implement the method, and provide a detailed performance analysis with experiments on real-world data.

I. INTRODUCTION

Light Detection and Ranging sensors (Lidars) are abundant in mobile robotics applications such as surveying environments to build maps, autonomous navigation, and a wide range of other perception and motion planning tasks. Cameras are often used with lidars for associating photo-realistic information with lidar range and reflectance measurements. Optimal calibration enables us to accurately register data from these sensor modalities to a common coordinate reference frame. This is helpful for creating a visually rich interpretation of the environment within which the robot traverses (see Fig. 1).

Using cameras with lidars has facilitated, and gradually improved, autonomous navigation capabilities [1]–[3]. Obtaining calibration parameters by manually measuring the position and orientation of sensors is prone to considerable errors, since the true position of the sensing element is often occluded by the sensor's protective casing. Within autonomous navigation systems, calibration of sensors mounted on a robot platform is fundamental for robust and efficient autonomy. Several state-of-the-art methods for lidar-camera calibration exist.

A. Literature Review

A majority of the extrinsic calibration techniques use known calibration targets [4]–[6]. These comprise of fiducial markers or checkerboards. An early implementation of this technique is described in [4], where the authors calibrate a 2D lidar by minimising a reprojection error computed from co-observing a checkerboard from a camera and the lidar.

Also using a target, Naroditsky et al. [5] calibrate a 2D lidar to a camera by first computing the relative transform
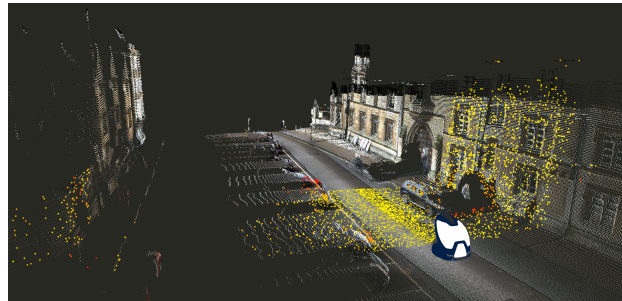
Fig. 1: Push-broom 2D Lidar: An autonomous vehicle with a push-broom 2D lidar traversing through an environment represented by a coloured 3D point cloud. Using vehicle motion, a lidar swathe is generated (shown as yellow points) to simulate a 3D point cloud. The lidar swathe helps us simulate sensor fields-of-view overlap as a function of vehicle motion.

using six measurements of a checkerboard, and then performing a RANSAC-based least-squares refinement.

The authors of [6] calibrate a multi-planar 2D lidar system to a camera by solving for the geometric constraints of a planar checkerboard viewed by both sensors. This is followed by a maximum likelihood-based refinement step.

Pandey et al. [7] extend the method in [4] for application to 3D lidars. The plane of the checkerboard is extracted in both the laser and camera data, and with multiple scans, the plane normals can be aligned and thus the relative position of each sensor can be determined. The authors in [8] provide a 3D lidar-camera calibration toolbox which iteratively minimises a geometric planar constraint-based nonlinear least-squares cost function. Reducing the data processing burden, [9] et al. provide a method that uses multiple checkerboards to calibrate a multi-beam lidar to multiple cameras, using only a single camera image. An alternative method to calibrate 3D lidars is implemented in [10]. Therein, the authors decouple the problem of intrinsic and extrinsic calibration into sub-problems, and iteratively increase the accuracy of the initial estimates for each sub-problem by minimising a batch nonlinear least-squares cost function.

The methods listed above are only applicable in the presence of known calibration targets and require these targets to be observed by all sensors simultaneously.

Recently, a few methods that perform calibration in the absence of fiducial targets have been proposed. Scaramuzza et al. [11] propose a targetless calibration technique, whereby the parameters are calculated by applying the perspective-from-n-points (PnP) algorithm [12] on manually selected point correspondences between camera pixels and lidar points. The method of [13] alleviates the need for user input
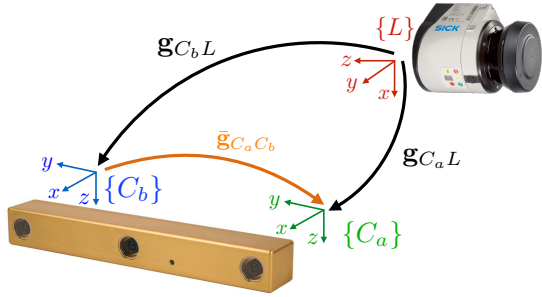
Fig. 2: Alternating Calibration: A setup with three local sensor frames of reference; the lidar frame $\{L\}$ (red), and the camera frames $\{C_a\}$ (green) and $\{C_b\}$ (blue), associated with each camera of a stereo camera unit. The solid curved arrows denote the 6-DoF transforms between the individual frames, where the black arrows denote the transforms to be optimised and the orange arrow denotes a known fixed transform.

by discovering edges in an omnidirectional camera image and correlating these to discontinuities in the 3D lidar range measurements.

In [14], the authors maximise the Mutual Information computed by registering camera pixel intensities to reflectance values obtained from a Velodyne 64-beam lidar. This approach to data association is most similar to ours as we attempt to find the parameters that describe the maximum similarity between the image intensity and laser reflectance.

B. Motivation and Contribution

For the methods above, overlapping fields-of-view (FoV) between both sensor modalities is prerequisite. This leads to design restrictions. In addition, 2D lidars are smaller and cheaper than 3D lidars, thus making them favourable for commercial robotics applications.

For mobile robotics applications, calibration can change due to some system characteristics. For example, vibration due to motion could lead to a requirement for periodic sensor recalibration. Target-based calibration approaches limit the ability to perform seamless recalibration. Thus, we propose an automatic, targetless, extrinsic calibration approach, where calibration is performed from the data collected in the robot's workspace.

There are a number of assumptions we make that allow us to solve this calibration problem. Principally, to deal with the non-overlapping fields of view, we need an accurate estimate of motion. The method used for generating a 3D map with a 2D lidar is described in the next section.

We assume that all sensors have known intrinsic parameters. To leverage the constraints afforded to us by a stereo- or multi-camera setup, we presume that the extrinsic calibration is provided by the manufacturer or otherwise easily obtainable.

For the method herein, we decouple a global optimisation problem into multiple sub-problems spread over two levels. This hierarchical, alternating optimisation enables the solvers to be implemented in parallel, over multiple CPU nodes. We expect such parallelisation to scale better with an increase in the number of sensors to be calibrated.

II. Problem and Solution Overview

We design the problem as a data-dependent, closed-loop, hierarchical relationship between alternating optimisers. The global problem is decoupled into sub-problems distributed over a lower-level and an upper-level. The lower-level solves $N_C N_L$ problems, with N_C being the number of cameras and N_L being the number of lidars. The upper-level implements a nonlinear least-squares refinement step to minimise the error between the solutions computed by the $N_C N_L$ lower-level optimisers. The upper and lower-level optimisers are linked with a quadratic penalty term [15]. Optimisation stops when a user-defined threshold is satisfied, or when time allotted for optimisation elapses.

A. Notation

A 6-DoF rigid-body transform that registers entities defined in source frame $\{A\}$ to destination frame $\{B\}$ is described by matrix $\mathbf{G}_{BA} \in \text{SE}(3)$. Matrix $\mathbf{G}_{BA}$ is parameterised by a tuple $\mathbf{g}_{BA} \in \Re^6$, where $\mathbf{g}_{BA} = (t_x, t_y, t_z, \theta, \rho, \psi)$, with t_x, t_y, and t_z being the relative translation components in metres, and θ, ρ, and ψ being the relative rotational components in radians, i.e., roll, pitch, and yaw angles, respectively.

Fig. 2 illustrates an example where $N_C = 2$ and $N_L = 1$. The tuples $\mathbf{g}_{C_a L}$ and $\mathbf{g}_{C_b L}$ define transforms that register an entity defined in lidar frame $\{L\}$, to a frame associated with each camera of the stereo camera unit. The tuple $\bar{\mathbf{g}}_{C_a C_b}$ defines a fixed and known rigid-body transform which registers entities defined in $\{C_b\}$ to $\{C_a\}$. The transforms $\mathbf{g}_{C_a L}$ and $\mathbf{g}_{C_b L}$ are denoted by solid black curved arrows, while the known transform $\bar{\mathbf{g}}_{C_a C_b}$ is denoted by a solid orange curved arrow.

B. Lidar Swathe Generation

Our approach is related to the calibration technique proposed in [16]. The authors in [16] calibrate a push-broom 2D lidar to a camera by minimising an edge-based, weighted sum of squares distance *(SSD)* cost function. Therein, the *SSD* cost is a function of the alignment of edges found in both camera images and dense lidar reflectance images obtained by interpolating lidar points projected into the camera image plane. Similar to [16], we exploit vehicle motion to generate a swathe of lidar points for simulating a 3D point cloud [17].

As described in [17], and as illustrated in Fig. 3, given an estimate of sensor motion, a 3D point cloud can be simulated by generating a recurring and metrically correct swathe of lidar points from a push-broom 2D lidar. Assuming that vehicle motion causes eventual overlap between the FoV of both sensor modalities, we can then project this generated swathe into the relevant camera's image plane and compute a measure of similarity between observations from the sensors.

Vehicle motion is prerequisite for swathe generation. Fig. 3 shows an example scenario, where a point $\mathbf{p} = [x, y, z]^\top$, is observed in the lidar frame $\{L^j\}$ at time j, and in the camera frame $\{C^k\}$ at time k, to provide observations $^{L^j}\mathbf{p}$
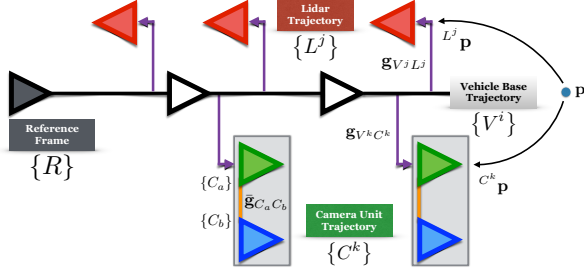
Fig. 3: Lidar Swathe and Sensor Calibration: The vehicle base trajectory is denoted as $\{V^i\}$. The lidar and camera trajectories $\{L^j\}$ and $\{C^k\}$, respectively, are defined relative to the appropriate pose of the vehicle trajectory. The projection of a point $\mathbf{p} \in \Re^3$ (solid blue dot) into the lidar and camera frames is indicated by black solid curved arrows. The transforms to be optimised are denoted by $\mathbf{g}_{V^j L^j}$ and $\mathbf{g}_{V^k C^k}$ (solid purple arrows). The transform $\bar{\mathbf{g}}_{C_a C_b}$ between the frames $\{C_a\}$ and $\{C_b\}$ of the multi-camera unit is fixed and known (solid orange line). The reference frame $\{R\}$ is shown as a pose on the base trajectory.

and $^{C^k}\mathbf{p}$, respectively.[1] The observations $^{L^j}\mathbf{p}$ and $^{C^k}\mathbf{p}$ can be registered to a common frame of reference $\{R\}$ by using the rigid-body transforms $\mathbf{g}_{V^j L^j}$ and $\mathbf{g}_{V^k C^k}$ as follows:

$$^R\mathbf{p} = \mathbf{g}_{RV^j} \oplus \mathbf{g}_{V^j L^j} \oplus {}^{L^j}\mathbf{p}, \quad (1)$$

$$^R\mathbf{p} = \mathbf{g}_{RV^k} \oplus \mathbf{g}_{V^k C^k} \oplus {}^{C^k}\mathbf{p}, \quad (2)$$

where $\oplus$ is a composition operator. In Eqns. (1) and (2), the time indices k and j on the camera and laser observations indicate that they may observe the same object at different times. The time index i on the vehicle frame indicates that vehicle motion provides a vehicle base trajectory $\{V^i\}$. The intermediate transforms $\mathbf{g}_{V^j L^j}$ and $\mathbf{g}_{V^k C^k}$ are fixed rigid-body transforms, where the time indices j and k merely indicate the pose along the vehicle base trajectory to which the observations are referenced to. Frame $\{R\}$ may be some global reference frame or the vehicle pose at a previous time.

The vehicle base trajectory $\{V^i\}$ can be obtained from several approaches, e.g. inertial navigation systems (INS), visual odometry (VO), etc. For the work presented herein, we use a stereo camera unit to obtain the vehicle base trajectory via VO. We assign a base frame $\{C\}$ to the stereo unit, where $\{C\}$ is aligned with frame $\{V\}$. Thus, in Eqn. (2), $\mathbf{g}_{VC} = \mathbf{0}^\top \Longleftrightarrow \mathbf{G}_{VC} = \mathbf{I}$.

Hence, for the case where the vehicle base trajectory is obtained using VO, we can rewrite Eqns. (1) and (2) as

$$^R\mathbf{p} = \mathbf{g}_{RC^j} \oplus \mathbf{g}_{C^j L^j} \oplus {}^{L^j}\mathbf{p}, \quad (3)$$

$$^R\mathbf{p} = \mathbf{g}_{RC^k} \oplus {}^{C^k}\mathbf{p}. \quad (4)$$

Eqns. (3) and (4) show that any $\mathbf{p} \in \Re^3$ observed in both sensor modalities at different times, can be projected to a common frame if accurate estimates for camera pose at times j and k exist (obtained from VO), and if an optimised rigid-body transform $\mathbf{g}_{C^j L^j}$ is available. Thus, swathe generation

[1]A scan at time j observes a set of points P^j. We approximate that all points in P^j are observed at the same time, while in practice, each scan takes a specified amount of time.

is a function of the extrinsic calibration parameters to be optimised. This paper focuses on optimising $\mathbf{g}_{C^j L^j}$.[2]

III. CALIBRATION METHODOLOGY

The method described here can be applied to any multi-camera system with accurate inter-camera transforms.

In this section we first design a lidar-camera calibration cost function and discuss its desirable properties. Subsequently, we formulate the calibration optimisation problem and then describe how it is decoupled into sub-problems that are distributed over hierarchical optimisation levels.

A. Normalised Information Distance (NID)

Let us begin by considering the objective function proposed in [14]. Mutual Information $MI(X, Y)$ is a measure of the strength of the statistical correlation between two discrete random variables X and Y. Intuitively, it indicates a measure of similarity between two distributions, and is defined as

$$MI(X;Y) = H(X) + H(Y) - H(X,Y), \quad (5)$$

$$\text{with,} \quad H(X) = -\sum_{x \in \mathcal{X}} p_x \log(p_x), \quad (6)$$

$$H(X,Y) = -\sum_{\substack{x \in \mathcal{X} \\ y \in \mathcal{Y}}} p_{xy} \log(p_{xy}). \quad (7)$$

In Eqns. (6) and (7), p_x is the marginal pdf of X, p_{xy} is the joint pdf of $\{X, Y\}$, and $H(X)$ and $H(X, Y)$ denote the entropy of X, and the joint entropy of $\{X, Y\}$, respectively. The symbols $\mathcal{X}$ and $\mathcal{Y}$ are the alphabets of X and Y, where X is the reflectance value for a point $\mathbf{p} \in \Re^3$ observed by lidar L, and Y is the image intensity value of the pixel to which $\mathbf{p}$ is projected to, in camera C. For X and Y, the Normalised Information Distance (NID) is

$$NID(X,Y) = \frac{H(X,Y) - MI(X;Y)}{H(X,Y)}. \quad (8)$$

Like MI, NID is a measure of the similarity between X and Y, but unlike MI, NID is a true metric [18], [19]. Thus, NID is symmetric, non-negative, and bounded, i.e. $0 \leq NID(X, Y) \leq 1$. NID satisfies the triangle inequality, i.e. $NID(X, Y) + NID(Y, Z) \geq NID(X, Z)$, and $NID(X, Y) = 0 \Longleftrightarrow X = Y$. Note that larger values of NID indicate greater dissimilarity between the distributions.

B. Calibration Problem Formulation

Given a scene observed by camera C_p and lidar L, we seek to minimise the dissimilarity between the pixel intensity values associated with the observed scene in camera C_p, and the reflectance values associated with the lidar points measured by the lidar L, projected in C_p's image plane. Let $\{C_a\}$ and $\{C_b\}$ be the frames associated with each camera of a stereo camera unit (see Fig. 2). Thus, the constrained

[2]For brevity, we will drop the time-index from the notation wherever possible in subsequent sections.

optimisation problem to calibrate a lidar to a multi-camera unit is defined as

$$\min_{\substack{\mathbf{g}_{C_aL}, \\ \mathbf{g}_{C_bL}}} \quad f_a(\mathbf{g}_{C_aL}) + f_b(\mathbf{g}_{C_bL})$$

$$\text{s.t.} \quad \mathbf{g}_{C_aL} \ominus \mathbf{g}_{C_bL} = \bar{\mathbf{g}}_{C_aC_b}$$

$$\text{where,} \quad f_a \equiv \frac{1}{S_a} \sum_{s_a=1}^{S_a} \text{NID}(X_a, Y_a; \mathbf{g}_{C_aL}) \quad (9)$$

$$f_b \equiv \frac{1}{S_b} \sum_{s_b=1}^{S_b} \text{NID}(X_b, Y_b; \mathbf{g}_{C_bL}).$$

The symbol $\ominus$ is a composition operator. The terms X_a, X_b, Y_a and Y_b denote measurements of the discrete random variables X and Y made by, or projected to, the camera indicated by the subscripts a and b. The scalars S_a and S_b are the number of arbitrarily selected images used to widen the basin of convergence of the cost function [14], [16]. Each camera may be provided a different set of scenes. The known transform $\bar{\mathbf{g}}_{C_aC_b}$ is assumed to be fixed and accurate, and $\mathbf{g}_{C_aL}$ and $\mathbf{g}_{C_bL}$ are the transforms to be optimised.

By using a quadratic penalty term to terminate the equality constraint in (9), we derive the following unconstrained optimisation problem:

$$\min_{\substack{\mathbf{g}_{C_aL}, \\ \mathbf{g}_{C_bL}}} \quad f_a(\mathbf{g}_{C_aL}) + f_b(\mathbf{g}_{C_bL}) + \|\mathbf{e}_{ab}\|^2_{\mathbf{P}_{ab}}$$

$$\text{with,} \quad \mathbf{e}_{ab} = (\mathbf{g}_{C_aL} \ominus \mathbf{g}_{C_bL}) \ominus \bar{\mathbf{g}}_{C_aC_b}, \quad (10)$$

where, the scalar $\|\mathbf{e}_{ab}\|^2_{\mathbf{P}_{ab}} = \mathbf{e}_{ab}^\top [\mathbf{P}_{ab}]^{-1} \mathbf{e}_{ab}$, is the squared *Mahalanobis distance* parameterised by a normal distribution $\mathcal{N}(\mathbf{0}, \mathbf{P}_{ab})$. If the covariance $\mathbf{P}_{ab} = \mathbf{0}$, then the problem in (10) satisfies the strict equality constraint from (9).

Note that f_a and f_b are mutually independent. We use this property to facilitate efficient and fast optimisation by decoupling the problem from (10) into sub-problems that can be solved on different CPU nodes, via a hierarchical alternating optimisation process.

We introduce additional optimisation variables and constraints to decouple the problem in (10) into different hierarchical levels, and define the following problem:

$$\min_{\substack{\mathbf{g}_{C_aL}, \mathbf{g}_{C_bL}, \\ \hat{\mathbf{g}}_{C_aL}, \hat{\mathbf{g}}_{C_bL}}} \quad f_a(\hat{\mathbf{g}}_{C_aL}) + f_b(\hat{\mathbf{g}}_{C_bL}) + \|\mathbf{e}_{ab}\|^2_{\mathbf{P}_{ab}}$$

$$\text{s.t.} \quad \hat{\mathbf{g}}_{C_aL} = \mathbf{g}_{C_aL}, \ \hat{\mathbf{g}}_{C_bL} = \mathbf{g}_{C_bL}. \quad (11)$$

Terminating the equality constraints in (11) by using quadratic penalty terms, we derive the following unconstrained optimisation problem:

$$\min_{\substack{\mathbf{g}_{C_aL}, \mathbf{g}_{C_bL}, \\ \hat{\mathbf{g}}_{C_aL}, \hat{\mathbf{g}}_{C_bL}}} \quad \|\mathbf{e}_{ab}(\mathbf{g}_{C_aL}, \mathbf{g}_{C_bL} : \bar{\mathbf{g}}_{C_aC_b})\|^2_{\mathbf{P}_{ab}}$$

$$+ f_a(\hat{\mathbf{g}}_{C_aL}) + \|\underbrace{\hat{\mathbf{g}}_{C_aL} \ominus \mathbf{g}_{C_aL}}_{\mathbf{e}_{aL}}\|^2_{\mathbf{P}^i_{aL}} \quad (12)$$

$$+ f_b(\hat{\mathbf{g}}_{C_bL}) + \|\underbrace{\hat{\mathbf{g}}_{C_bL} \ominus \mathbf{g}_{C_bL}}_{\mathbf{e}_{bL}}\|^2_{\mathbf{P}^i_{bL}}$$

In (12), the error terms $\mathbf{e}_{aL}$ and $\mathbf{e}_{bL}$ are parameterised by $\mathcal{N}(\mathbf{0}, \mathbf{P}^i_{aL})$ and $\mathcal{N}(\mathbf{0}, \mathbf{P}^i_{bL})$, respectively. The superscript i on $\mathbf{P}^i_{aL}$ and $\mathbf{P}^i_{bL}$ is an optimisation iteration index which indicates that $\mathbf{P}^i_{aL}, \mathbf{P}^i_{bL} \to \mathbf{0}$ as $i \to \infty$. This formulation emphasises solutions within the feasible region. The severity of the penalty levied on infeasible solutions is determined by $\mathbf{P}^i_{aL}$ and $\mathbf{P}^i_{bL}$ [20]. As $\mathbf{P}^i_{aL}$ and $\mathbf{P}^i_{bL}$ decrease, the unconstrained problem in (12) accurately replicates the constrained problem in (11).

C. Calibration via Alternating Optimisation

We solve the problem in (12) using an alternating optimisation algorithm [21]. Note that, for each camera C_p, the penalty term $\|\mathbf{e}_{pL}\|^2_{\mathbf{P}^i_{pL}}$ is a function of $\mathbf{g}_{C_pL}$ and $\hat{\mathbf{g}}_{C_pL}$. Thus, to solve the problem defined in (12), we utilise the quadratic penalty terms $\|\mathbf{e}_{aL}\|^2_{\mathbf{P}^i_{aL}}$ and $\|\mathbf{e}_{bL}\|^2_{\mathbf{P}^i_{bL}}$ to link the two alternating optimisation levels.

1) Lower-level Optimisers: For each $p = 1 \ldots N_C$, independent optimisers first optimise $\hat{\mathbf{g}}_{C_pL}$ by solving the following problem:

$$\min_{\hat{\mathbf{g}}_{C_pL}} \quad f_p(\hat{\mathbf{g}}_{C_pL}) + \|\mathbf{e}_{pL}(\hat{\mathbf{g}}_{C_pL} : \mathbf{g}_{C_pL})\|^2_{\mathbf{P}^i_{pL}} \quad (13)$$

Note that Eqn. (13) is a sub-problem decoupled from Eqn. (12), i.e. the second or the third row of (12), depending on the camera index. We call Eqn. (13) as a lower-level optimiser. There are $N_C N_L$ lower-level optimisers for each lidar camera pair. While solving (13), $\mathbf{g}_{C_pL}$ is held constant.

2) Upper-level Optimiser: For the problem defined in Eqn. (12), the lower-level optimiser provides optimised solutions of $\hat{\mathbf{g}}_{C_aL}$ and $\hat{\mathbf{g}}_{C_bL}$. Using $\hat{\mathbf{g}}_{C_aL}$ and $\hat{\mathbf{g}}_{C_bL}$ as known constants, we define the upper-level optimiser as

$$\min_{\substack{\mathbf{g}_{C_aL}, \\ \mathbf{g}_{C_bL}}} \quad \|\mathbf{e}_{ab}(\mathbf{g}_{C_aL}, \mathbf{g}_{C_bL} : \bar{\mathbf{g}}_{C_aC_b})\|^2_{\mathbf{P}_{ab}}$$

$$+ \|\mathbf{e}_{aL}(\mathbf{g}_{C_aL} : \hat{\mathbf{g}}_{C_aL})\|^2_{\mathbf{P}^i_{aL}} \quad (14)$$

$$+ \|\mathbf{e}_{bL}(\mathbf{g}_{C_bL} : \hat{\mathbf{g}}_{C_bL})\|^2_{\mathbf{P}^i_{bL}}$$

Note that the problem in (14) is a sub-problem decoupled from Eqn. (12). We utilise the alternating optimisation formulation to solve the problem in (12) as a set of hierarchical, closed-loop, and sequential optimisation problem. Herein, the solutions of the lower-level optimisers drive the upper-level optimiser, and vice versa.

Remark 1: Since the lower-level optimisers are mutually independent, they can be solved in a distributed setting on multiple CPU nodes. We expect the parallelisation of the lower-level optimisers to aid an online implementation of the proposed method in the near future. ◇

Remark 2: By defining a camera base frame $\{C\}$ aligned with frame $\{C_a\}$ of the multi-camera unit, we get $\mathbf{g}_{C_aL} = \mathbf{g}_{CL} \iff \mathbf{G}_{CC_a} = \mathbf{I}$. Thus, the problem in Eqn. (12) can be rewritten in terms of $\mathbf{g}_{CL}$, which makes the problem consistent with the formulation in Eqns. (3) and (4). ◇
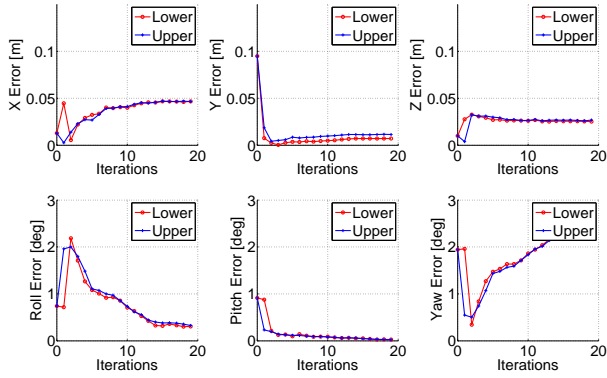
(a) Left camera - Laser calibration $\mathbf{g}_{C_a L}$
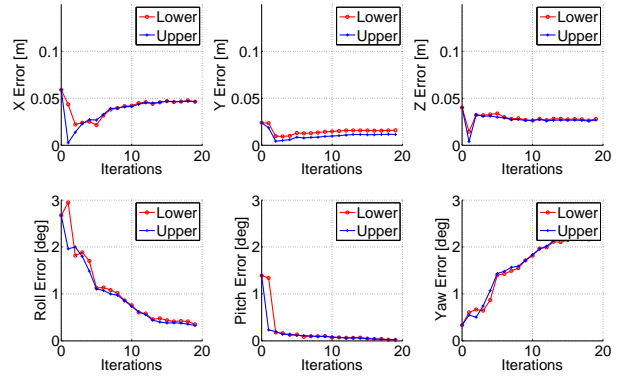
(b) Right camera - Laser calibration $\mathbf{g}_{C_b L}$

Fig. 4: Absolute Error *w.r.t* Ground Truth: Fig. shows the solution achieved by the proposed alternating method at each global iteration. The red and blue lines with colour coordinated circular markers show the lower- and upper-level solutions per iteration. Note that in the limit, both optimiser levels tend to converge to the same optimum. This is the effect of the quadratic penalty term coupling the two levels.
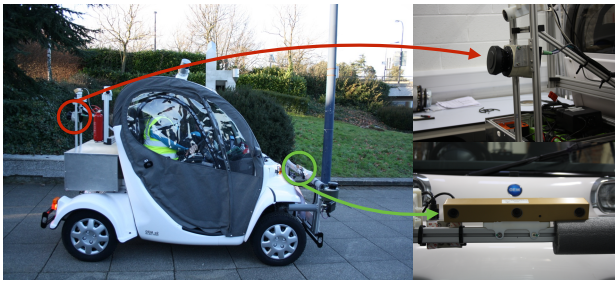

Fig. 5: Setup: Fig. shows the sensors to be calibrated mounted on the platform such that they do not have overlapping FoV. We use the stereo camera mounted at the front of the vehicle to estimate vehicle pose using VO. The data used for calibration was collected in Milton Keynes, UK.
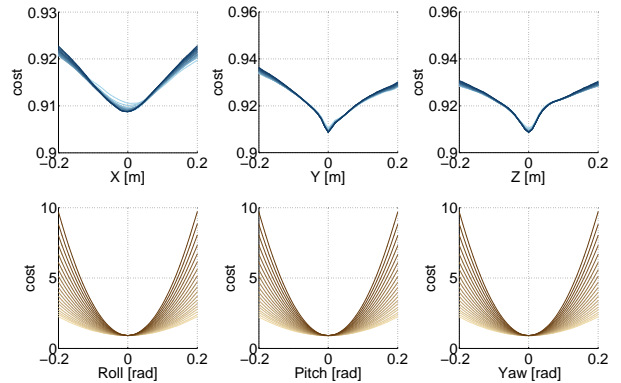

Fig. 6: NID Cost: Fig. shows the cost function minimised in the lower-level optimiser over each iteration of the alternating method. Light to dark graphs indicate the evolution of the optimisation. The cost function is sampled around the current solution of the lower-level optimiser.

IV. EXPERIMENTS AND RESULTS

The platform used for collecting the data on which the proposed algorithm was implemented can be seen in Fig. 5, wherein, the positions of the sensors to be calibrated are highlighted. Fig. 5 shows that the two sensors have non-overlapping fields-of-view.

A. Setup

The vehicle used has the stereo camera mounted in front, facing forward, and tilted downward by approximately $18°$. The lidar is positioned such that it is tilted back by approximately $9°$. The lidar scans are generated in a plane that is offset by approximately $9°$ with respect to the plane, orthogonal to the direction of horizontal planar motion. We refer to this as a 'push-broom' configuration. The point cloud generated using such a configuration is illustrated in Fig. 1.

We use a Point Grey Research (PGR) Bumblebee XB3 multi-baseline stereo camera to perform our experiments. PGR provide sub-millimetre accuracy for the inter-camera transform between the individual cameras of the stereo unit [22]. As explained in Section III, we can exploit an accurate inter-camera transform provided by the manufacturer, as a known parameter to perform lidar-camera calibration.

Following the procedure in II-B, we generate point clouds for 30 different scenes, each 10m long. We test our procedure over these data.

B. Performance Evaluation

To evaluate the proposed algorithm, we employ the proven method of calibration between the lidar and camera using a checkerboard, as documented in [7], to obtain reference values. Since the lidar and camera do not have overlapping FoVs in our case, we use an accurate (less than 0.5mm translation error) motion tracking system [23] to locate the plane of the checkerboard while being observed by each sensor. Thus, the lidar and each camera of a multi-camera unit can both be calibrated very accurately with respect to the frame of the motion tracker. For the purposes of this paper, this calibration has been established as 'ground truth'.

Fig. 4 shows the results from an optimisation run over 30 frames, with a fixed number iterations for the lower and upper-level optimisers. The lower-level optimiser runs for 200 iterations, and the upper-level for 20 iterations. Note that we run the lower-level for each camera in the stereo rig. Each graph illustrates how the solution for alternating
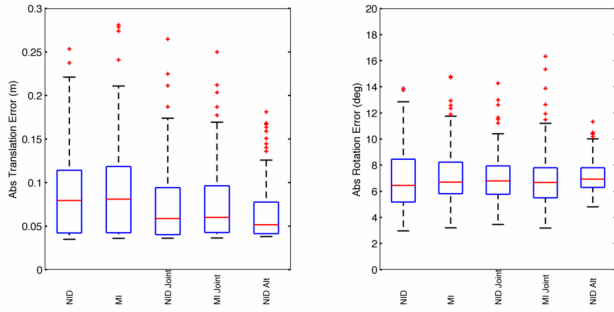
Fig. 7: Error Analysis: Box plots showing the RSE in translation (left) and rotation (right) when using different cost functions. Moreover, we compare them in the context of different optimisation schemes: using once-off NID and MI registration, once-off registration using a cost function that sums the NID and MI over both cameras (NID and MI Joint) and our proposed alternating optimisation (NID ALT).

Fig. 9: Solution Quality: Fig. shows the projection of a 3D point cloud on a reference image. The red dots show the non-optimal initial-seed, and the blue dots show the optimised solution for $\mathbf{g}_{C_aL}$. The displayed solution is obtained after 10 iterations of the alternating method, and shows considerable improvement on visual inspection.
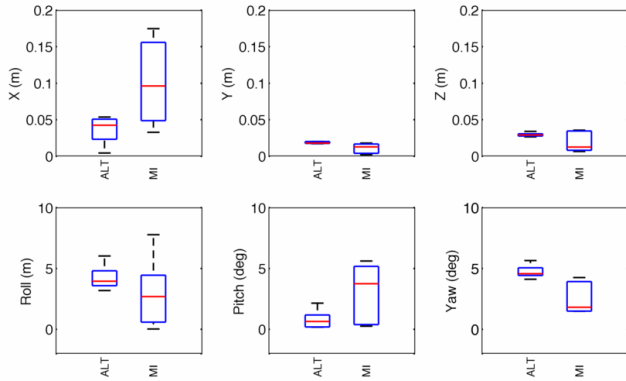
Fig. 8: The box plot indicates the distribution of absolute errors using the proposed method (ALT), compared to the MI based implementation of [14]. The tops and bottoms of each boxplot are the 25th and 75th percentiles of the samples, respectively. The line in the middle of each box is the error median.

optimisation per each degree-of-freedom evolves over the number of iterations. As described in section III-B, the upper and lower-level optimisers are coupled by quadratic penalty terms. Therefore, the solutions obtained at both levels tend to converge asymptotically. Fig. 4 illustrates this behaviour.

The initial values for the covariance $\mathbf{P}^0_{aL}, \mathbf{P}^0_{bL}$ are set up with standard deviations of $\sigma_t = 5$m and $\sigma_\phi = 5$ radians for the translation and the rotation parameters, respectively. This allows the calibration parameters to move freely during the first few iterations. At each i-th iteration, we follow the rule suggested in [20], where the covariance matrix is updated as $\mathbf{P}^{i+1}_{aL} = \omega \; \mathbf{P}^i_{aL}$, with $\omega \in [0\,1]$.

Fig. 6 plots the evolution of the lower-level NID cost function. The effect of the quadratic penalty in Eqn. (12) becomes apparent as the parameter is shifted away from the minimum; large step sizes in the parameters are penalised.

As an extension of our evaluation, we compare our method with the method proposed in [14]. Analogous to our NID-based approach, the method therein maximises the Mutual Information (MI) between the two sensor modalities. However, the original method imposes co-visibility of the observations gathered by the different sensors, restricting the application to $360°$ lidars. To deal with this restriction, we generate 3D point clouds using the 2D lidar mounted at

the rear of the vehicle (see Fig. 5), by moving through the world. This procedure is described in section II-B. For this experiment, we use the source code provided by the authors.[3]

Fig. 8 shows the error distribution for a set of final solutions obtained for 5 different initial parameter values. Our assessment shows that the proposed alternating method generates similar median errors to the MI-based method from [14]. In fact, the variance is significantly reduced in all the degrees of freedom. It is worth mentioning that the solutions obtained are affected by the different factors. For instance, the optimisation in [14] trusts on a gradient descend algorithm while our approach is formulated as a unconstrained nonlinear optimisation and uses a simplex search method. For a fair comparison, we added to our calibration pipeline the possibility to use an MI-based cost function at the lower level step of the optimisation. Fig. 7 shows the distributions of the Root Square Errors (RSE) in translation and rotation for the different cost functions under three optimisation schemes: with no upper level optimisation (i.e NID and MI), alternating optimisation (NID ALT). The 'Joint' tag indicates image-laser registration where the cost function is the sum of the NID or MI from both cameras. This is achieved by presuming a known, certain and fixed left-to-right camera transform. The alternating methods differs in that it relaxes the requirement for the transform to be known with certainty.

The error plots indicate that our alternating approach produces more accurate or comparable estimates with a lower error variance.

Fig. 9 provides a qualitative illustration of the solution computed by the alternating method. The point cloud generated using the initial parameter values is superimposed on a reference image as red dots. The point cloud generated using the optimised parameter values is superimposed on the same reference image as blue dots. A visual inspection of Fig. 9

[3]http://robots.engin.umich.edu/SoftwareData/ExtrinsicCalib

indicates a marked improvement in lidar camera calibration.

V. Conclusion and Future Work

We propose a target-less, automatic, data-driven method for calibrating a 2D push-broom lidar to a multi-camera unit, using a hierarchical, closed-loop, alternating optimisation algorithm, distributed over different optimisation levels. The lower-level minimises multiple *Normalised Information Distance*-based (NID) cost functions, while the upper-level implements a nonlinear least-squares based refinement step. The two levels are linked together using quadratic penalty terms that determine the penalty associated with computing infeasible solutions. The two optimisation levels work hierarchically, in cohesion, to converge to a solution. This approach is applicable, but not restricted to, stereo camera units or other multi-camera configurations, e.g. Point Grey Research's Bumblebee and Ladybug cameras. The method can be applied to systems as long as an overlapping FoV exists or is simulated, and if an accurate rigid-body transform between each sensor of the multi-sensor unit is available.

We provide implementation results along with a detailed performance analysis on real-world data. Performance of the proposed method is evaluated against calibration parameters obtained from a highly-accurate commercial calibration system. The results show that the proposed approach performs at least as well as prior art, without requiring sensor co-visibility of the scenes used for calibration.

As for most real-world applications, the function to be minimised for the proposed method is only locally convex, and is data-dependent. Each image used for calibration may provide a different amount of information. This can affect the basin of convergence for the selected cost function. Thus, learning a calibration cost function from the data, and utilising information-based image selection and weighting schemes are interesting problems for future research.

In [16], calibration is performed by creating a synthetic lidar image through interpolation of lidar reflectance values. This lidar reflectance interpolation step is computationally expensive, and we believe, is unsuitable for extension to on-line calibration approaches. Extending the proposed method to an online implementation is worth investigating and within future scope.

References

[1] J. Leonard *et al.*, "A perception-driven autonomous urban vehicle," in *Journal of Field Robotics Special Issue on the 2007 DARPA Urban Challenge, Part III*, vol. 25, Tokyo, Japan, Oct 2008, p. 727–774.

[2] C. Urmson *et al.*, "Autonomous driving in urban environments: Boss and the urban challenge," in *Journal of Field Robotics Special Issue on the 2007 DARPA Urban Challenge, Part I*, vol. 25, Tokyo, Japan, June 2008, pp. 425–466.

[3] S. Thrun, "Winning the darpa grand challenge: A robot race through the mojave desert," in *IEEE/ACM International Conference on Automated Software Engineering (ASE)*, Tokyo, Japan, Sept 2006, pp. –.

[4] Q. Zhang and R. Pless, "Extrinsic calibration of a camera and laser range finder (improves camera calibration)," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, vol. 3. IEEE, 2004, pp. 2301–2306.

[5] O. Naroditsky, A. Patterson, and K. Daniilidis, "Automatic alignment of a camera with a line scan lidar system," in *IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China, May 2011, pp. 3429–3434.

[6] L. Huang and M. Barth, "A novel multi-planar lidar and computer vision calibration procedure using 2d patterns for automated navigation," in *IEEE Intelligent Vehicles Symposium (IVS)*, Xi'an, China, June 2009, pp. 117–122.

[7] G. Pandey, J. R. McBride, S. Savarese, and R. Eustice, "Extrinsic calibration of a 3d laser scanner and an omnidirectional camera," in *IFAC Symposium on Intelligent Autonomous Vehicles (IAV)*, Sep 2010.

[8] R. Unnikrishnan and M. Hebert, "Fast extrinsic calibration of a laser rangefinder to a camera," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Carnegie Mellon University, 2005.

[9] A. Geiger, F. Moosmann, O. Car, and B. Schuster, "Automatic camera and range sensor calibration using a single shot," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*. IEEE, 2012, pp. 3936–3943.

[10] F. M. Mirzaei, D. G. Kottas, and S. I. Roumeliotis, "3d lidar-camera intrinsic and extrinsic calibration: Identifiability and analytical least-squares-based initialization," in *The International Journal of Robotics Research (IJRR)*, Sep 2012, pp. 452–467.

[11] D. Scaramuzza, A. Harati, and R. Siegwart, "Extrinsic self calibration of a camera and a 3d laser range finder from natural scenes," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2007, pp. 4164–4169.

[12] L. Quan and Z. Lan, "Linear n-point camera pose determination," in *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 21, no. 8, Xi'an, China, Aug 1999, pp. 774–780.

[13] J. Levinson and S. Thrun, "Automatic online calibration of cameras and lasers." in *Robotics: Science and Systems*, 2013.

[14] G. Pandey, J. R. McBride, S. Savarese, and R. Eustice, "Automatic extrinsic calibration of vision and lidar by maximizing mutual information," in *Journal of Field Robotics (JFR)*, Sep 2014, pp. 1–27.

[15] S. Boyd and L. Vandenberghe, "Duality," in *Convex Optimization*, 2004, pp. 215–287.

[16] A. Napier, P. Corke, and P. Newman, "Cross-calibration of push-broom 2d lidars and cameras in natural scenes," in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, Karlsruhe, Germany, May 2013.

[17] I. Baldwin and P. Newman, "Localising transportable apparatus," Apr 2013, wO Patent App. PCT/GB2012/052,381. [Online]. Available: http://www.google.com/patents/WO2013045917A1?cl=en

[18] A. Kraskov and P. Grassberger, "Mic: Mutual information based hierarchical clustering," in *Information Theory and Statistical Learning*, 2009, pp. 101–123.

[19] T. M. Cover and J. A. Thomas, "Entropy, relative entropy, and mutual information," in *Elements of Information Theory*, vol. 2, 1991, pp. 1–55.

[20] D. P. Bertsekas, "The method of multipliers for equality constrained problems," in *Constrained Optimization and Lagrange Multiplier Methods*, 1996, pp. 95–157.

[21] J. C. Bezdek and R. J. Hathaway, "Convergence of alternating optimization," in *Neural, Parallel and Scientific Computations (NPSC)*, vol. 11, no. 4, Dec 2003, pp. 351–368.

[22] P. G. Research. (2014) Bumblebee2 and xb3 datasheet. [Online]. Available: http://www.ptgrey.com/support/downloads/10132/

[23] P. T. Inc. (2014) Vz4000. [Online]. Available: http://www.ptiphoenix.com/?prod-trackers-post=vz4000