Active Planning based Extrinsic Calibration of Exteroceptive Sensors in Unknown Environments

Varun Murali, Carlos Nieto, Siddharth Choudhary, Henrik I. Christensen

***Abstract*— Existing Simultaneous Localization and Mapping systems require an extensive manual pre-calibration process. Non-manual calibration procedures use manipulators to create known patterns in order to estimate the unknown calibration. Calibration is often time-consuming and involves humans performing repetitive tasks such as aligning a known calibration target at different poses with respect to the sensor. We propose an algorithm that plans a trajectory which actively reduces the uncertainty of the robot's calibration given a rough initial calibration estimate. Calibration is performed autonomously in a previously unknown environment by maintaining the belief over landmarks, poses, and the calibration parameters. We present experimental results to demonstrate the approach's ability to autonomously calibrate the exteroceptive sensor in simulated and real environments. We show that even a greedy approach can reduce the effort needed to perform calibration every time the robot is reconfigured for autonomous tasks and mitigates the possibility of human error added into the calibration.**

I. Introduction

Autonomous robots depend on high-quality extrinsic calibration of their sensors to reliably perform their tasks. As a motivating example, one of the tasks typically performed by mobile robots is inferring the structure of the environment using sensor measurements. Given a representation of the world, the robot could generate a plan of actions to accomplish a given goal like navigation.

We study the self-calibration problem to autonomously estimate the extrinsic parameters of the sensors. Simultaneous Calibration Localization and Mapping problem attempt to jointly estimate the location of landmarks, the trajectory of the robot and the extrinsic parameters of the sensors on the robot. We exploit the structure of this problem to greedily plan a path that would reduce the uncertainty of the extrinsic parameters of the robot. Fig. 1 shows the output of such a system where the calibration parameters, trajectory, and landmarks are jointly estimated.

Often a human operator is given a task with easy objectives such as repeatedly aligning the robot's field of view with the calibration grid, a process that could be sped up through automation. This would also allow the robot to be robust to sensor alignment changes between different experiments due to either intentional or accidental reconfigurations.

We propose a novel algorithm that allows the robot to perform a self-calibration routine through autonomous investigative movements around a previously unknown environment that would then enable the robot to autonomously perform other tasks. The main contribution of this paper is to show the possibility of a generalized, multi-modal approach to autonomously calibrate extrinsic parameters of available sensors on the robot by leveraging a greedy planner to quickly estimate the extrinsic sensor transform. This would greatly reduce the amount of time and effort required to setup an autonomous robot. We further demonstrate the utility of planning towards the application of simultaneous localization calibration and mapping.

V. Murali, C. Nieto, S. Choudhary and H.I. Christensen are with the College of Computing, Georgia Institute of Technology, Atlanta, GA, USA {varunm, carlos.nieto, siddharth.choudhary}@gatech.edu, hic@cc.gatech.edu

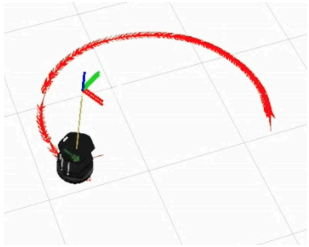

Fig. 1: The figure above shows the robot's estimated trajectory and the two transforms represent the ground truth transform and the estimated sensor transform with respect to base at that time.

II. Related Work

In the last years, there has been substantial research to solve the challenging problem to calibrate sensors on a robot. Extrinsic calibration techniques has been studied in depth by Zhang [1] [2], Heikkilä [3]. Classical methods just calibrate one sensor at the time to guarantee accurate results.

An alternative approach is to calibrate the system with many sensors at the same time. Roy and Thrun [4] proposed a statistical method which uses the robot's sensor to automatically calibrate the robot as it operates. This approach also enables the robot to adapt to changes in its kinematics on-line. Le and Ng [5] presented a unified framework that jointly calibrates many sensors simultaneously. Their algorithm satisfies precision requirements for tasks and takes less human time compared to older methods.

The presence of landmarks or synthetic objects allowed a multi-sensor system to be calibrated by itself [6] where the only input in the calibration is a sequence of images with easily detectable spots acquired from low cost synchronized the multi-camera system. Other techniques [7] [8] [9] [10] focused on calibrate lidar sensors with respect to a camera for data fusion purposes. Gao and Spletzer [11] proposed and automated on-line approach to calibrating multiple LIDARs mounted in a car. This technique supports *"real time"* calibration operations and the convex relaxation

is computationally efficient than more standard non-linear least squares approaches. Heng et al. [12] proposed an alternative approach called infrastructure-based calibration that is efficient, requires no modification of the calibration area and is completely unsupervised. The use of this system does not assume an overlapping field of view between any two cameras and does not require an initial guess of any extrinsic parameter.

In [13] Foxlin discusses the design of a very general architectural framework for navigation and tracking that fuse dead-reckoning sensors with environment-referenced sensors. His framework can handle full SLACAM (Simultaneous Localization and Calibration for Mapping) into a complementary EKF (Extended Kalman Filter) with explicit states for intrinsic and extrinsic parameters of sensors and targets attached to the mobile platform or to the map. Caltabiano [14] presented a localization algorithm for mobile robots able to calculate absolute position and orientation of the platform to estimate the odometry parameters. The computational time of this method is low compared with other classical approaches because the odometry parameters are automatically estimated while the platform is in motion.

Levinson and Thrun [15] [16] introduces two new real-time techniques that enable camera-laser calibration online, automatically, and in arbitrary environments. The constant background monitoring algorithm detects sudden miscalibrations which allow the robot to rely on the acquired sensor data. In addition, the technique is able to track gradual drift of sensor pose over time, without performing computationally intensive global optimizations over the entire search space.

Kümmerle et al. [17] [18] introduced a Simultaneous Parameter Calibration Localization and Mapping method wich performs on-line estimation of the calibration parameters while performing Simultaneous Localization and Mapping (SLAM). This method does not require prior knowledge of the environment and relies on a rough initial guess of the robot parameters.

We can generalize motion trajectories suitable for autonomous calibration by taking advantage of planning within the Generalized Belief Space, which is related to POMDP, or a Belief Road Map. A Belief Road Map assumes a known environment with uncertain trajectory and encodes the information as a cost of moving along graphs' edge. This may be extended in some cases to partially unknown environments that have only been partially explored [19]. An analogous system given only a nominal policy that covers the target area, without any other a priori environmental information, has been able to plan and navigate using a Rapidly-exploring Random Graph (RRG). The Generalized Belief Space removes the dependence on even a nominal policy but retains the need for a good initial estimate [20]. While assuming maximum-likelihood is easiest with this method, there may be some cost benefit to not making this assumption [21].

III. Approach

As the robot moves in an unknown environment, it continuously maps using the simultaneous localization and mapping (SLAM) pipeline presented by Trevor et al. [22]. However, in contrast to the known extrinsic sensor calibration as assumed in general, we model the extrinsic calibration as an unknown parameter and estimate its uncertainty along with other unknown parameters like landmarks and poses. At each instant, the robot plans a trajectory which actively reduces the uncertainty of extrinsic calibration parameters. As a result, the robot autonomously calibrates its sensor by actively moving in the direction of the maximum uncertainty reduction.

In this section, we briefly review simultaneous localization and mapping, explain the theoretical details behind simultaneous calibration, localization, and mapping and propose an approach for autonomous and active planning in belief space for extrinsic calibration.

A. Simultaneous Localization and Mapping (SLAM)

In landmark-based SLAM, a robot, while navigating, tries to localize itself and at the same time build a map of the environment (represented using landmarks). Assuming a pose of the robot at the i^{th} time step is x_i with $i \in 0 \ldots M$, a landmark is l_j with $j \in 0 \ldots N$ and a measurement is z_k, with $k \in 0 \ldots K$, the joint probability model is given as, $P(X,L,Z) = P(x_o)\prod_{i=1}^{M} P(x_i|x_{i-1},u_i)\prod_{k=1}^{K} P\left(z_k|x_{ik},l_{jk}\right)$ where $P(x_o)$ is a prior on the initial state, $P(x_i|x_{i-1},u_i)$ is the motion model, parametrized by a control input u_i and $P(z_k|x_{ik},l_{jk})$ is the landmark measurement model, x_{ik} and l_{jk} corresponds to measurement z_k. Assuming the motion and measurement models are Gaussian, $P(x_i|x_{i-1},u_i) \propto \exp -\frac{1}{2}\|f_i(x_{i-1},u_i) - x_i\|^2_{\Lambda_i}$ and $P(z_k|x_{ik},l_{jk}) \propto \exp -\frac{1}{2}\|h_k(x_{ik},l_{jk}) - z_k\|^2_{\Sigma_k}$ where $f()$ is the robot motion equation and $h()$ is a landmark measurement equation with Λ_i and Σ_k as the respective covariances.

We use a factor graph to represent the joint probability model $P(X,L,Z)$ where each factor represents either $P(x_o)$ or $P(x_i|x_{i-1},u_i)$ or $P\left(z_k|x_{ik},l_{jk}\right)$. Therefore the joint probability model can be written as $P(X,L,Z) \propto g(\Theta) = \prod_i g_i(\Theta_i)$ where Θ_i is the set of variables θ_j adjacent to the factor g_i. Fig. 2 shows the corresponding factor graph. Given all the measurements, we obtain the maximum a posteriori (MAP) estimate by maximizing the joint probability $P(X,L,Z)$.

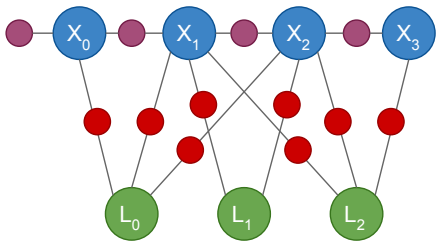

Fig. 2: Factor graph representation of the SLAM problem. Blue circles denote the poses (X) and green circles denote landmarks (L). Small purple circles represent odometry constraints and red circles represent landmark-pose constraint.

$$\Theta^* = \arg\max_{\Theta} P(X,L|Z) = \arg\min_{\Theta}(-\log g(\Theta)) \quad (1)$$

which leads to the following non-linear least squares problem: $\Theta^* = \arg\min_{\Theta} \sum_{i=1}^{M} \|f_i(x_{i-1},u_i) - x_i\|^2_{\Lambda_i}$ $+ \sum_{k=1}^{K} \|h_k(x_{ik},l_{jk}) - z_k\|^2_{\Sigma_k}$.

The non-linear least squares problem is solved using a non-linear optimization method such as the Levenberg-Marquardt algorithm which solves a succession of linear approximations in order to approach the minimum. We use Square Root SAM (Dellaert and Kaess [23]) to optimize the resulting factor graph.

B. Simultaneous Calibration, Localization and Mapping

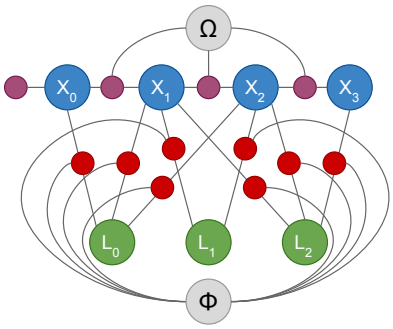

Fig. 3: Factor graph representing the Simultaneous Calibration, Localization and Mapping problem. Gray circles represent the additional calibration parameters (Ω,Φ) which are connected to other factors resulting in joint optimization over landmarks, poses and calibration parameters

In order to perform simultaneous calibration, localization and mapping we explicitly represent the additional calibration parameters in the joint probability model as $P(X,L,Z,\Omega,\Phi) = P(x_o)\prod_{i=1}^{M} P(x_i|x_{i-1},u_i,\Omega)$ $\prod_{k=1}^{K} P(z_k|x_{ik},l_{jk},\Phi)$ where Ω represents odometry bias parameters and Φ represents the extrinsic transformation between the robot base and the corresponding exteroceptive sensor which has to be calibrated. Fig. 3 shows the corresponding factor graph. Each sensor will have a different extrinsic transformation Φ but for simplicity we use a single Φ corresponding to a single sensor. Optimization can be done in a similar manner as shown in the previous section with additional variables representing the extrinsic calibration Φ and odometry bias Ω. The resulting non-linear optimization is $\Theta^* = \arg\min_{\Theta} \sum_{i=1}^{M} \|f_i(x_{i-1},u_i,\Omega) - x_i\|^2_{\Lambda_i}$ $+ \sum_{k=1}^{K} \|h_k(x_{ik},l_{jk},\Phi) - z_k\|^2_{\Sigma_k}$

C. Autonomous Extrinsic Calibration using Active Planning

We use active planning in belief space for autonomous extrinsic calibration of an exteroceptive sensor. Planning is done such that it actively reduces calibration parameter uncertainty at each instant in a previously unknown environment. To achieve that end, we integrate simultaneous calibration, localization and mapping with an active planning framework resulting in *planning for calibration*.

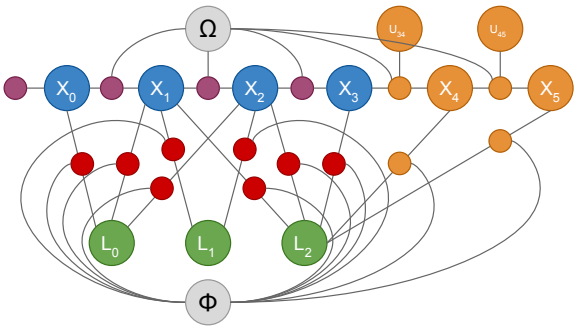

Fig. 4: Factor graph representing the Planning for Calibration problem. Orange circles to the right represent the future control inputs and predicted poses. Maximum likelihood observations are represented by small orange circles.

In order to compute the optimal control action over the s lookahead steps, we compute the predicted belief over the time horizon. However we don't know the observations $Z_{M+1:M+s}$ ahead of time, given M is the current time step. We assume the maximum likelihood observation as future observation given the current belief over landmarks and poses and future control inputs. Modeling future observations as unknown has an insignificant increase in efficiency [21]. We also assume that the number of landmarks does not change when we predict the belief over the time horizon.

Given the estimated extrinsic calibration parameter Φ_M and odometry bias Ω_M, estimated poses and landmarks up to time M, current observations $Z_{1:M}$, control inputs $U_{M+1:M+s}$, and the corresponding predicted observations $Z_{M+1:M+s}$, the predicted calibration belief Φ_{M+s}, Ω_{M+s} at a future time step l is given as

$$gb(\Phi_{M+s},\Omega_{M+s}) =$$
$$P(\Phi_{M+s},\Omega_{M+s}|X_{1:M+s},L_{1:M+s},Z_{1:M+s},U_{M+1:M+s})$$

Fig. 4 shows the corresponding factor graph. The control action minimizes the general objective function $J(U_{M+s:M+s})$ over s look ahead steps

$$J(U_{M+1:M+s}) = c_l(gb(\Phi_{M+s},\Omega_{M+s}))$$

where c_s is defined as the determinant of predicted joint covariance of the calibration parameters Φ_{M+s}, Ω_{M_s}. At each step the general objective function selects the control action that results in the minimum predicted uncertainty of the calibration parameter. In turn, the resulting planning algorithm will ensure that the robot moves in a direction that will reduce the associated calibration uncertainty. Given the control action, the robot is moved in the corresponding direction followed by re-estimation of landmarks and pose beliefs. Using the new added poses and landmarks, we again estimate the predicted belief over the next s look ahead steps. This algorithm is run until the change in the uncertainty determinant of the calibration parameters is below a certain threshold or the maximum number of steps are taken. Algorithm 1 summarizes the process.

Algorithm 1: Active Planning based Extrinsic Calibration

$\{X, L\} \leftarrow$ Initialize by taking random step
repeat
  Select control action that minimizes the objective function: $\mathbf{U} = \min_{U_{M+1:M+s}} J(U_{M+1:M+s})$
  Move the robot in the corresponding direction.
  Re-estimate the pose and landmark beliefs
until stopping criterion is met

a) Monte-Carlo Sampling: To select the control action that minimizes the objective function $J(U_{M+1:M+s})$ we perform Monte-Carlo sampling where we randomly the select sequence of control actions from among a discrete set of control action sequences and estimates the corresponding value of the objective function. The control action which results in the minimum value of the objective function is taken as the optimal control action. Algorithm 2 summarizes it.

We use random sampling in the discrete control space, as opposed to using a gradient descent in continuous space to reduce the probability of being stuck in local minima and to allow for better back-tracking. To sample in control space for the mobile robot experiments, we randomly generate samples in the velocity space (x, θ) and predict the position based on applying these velocities for a pre-defined time using the dynamics of the robot.

Algorithm 2: Monte-Carlo Sampling(X,L)

$V_{min} \leftarrow \infty, U_{min} \leftarrow \emptyset$
for K iterations **do**
  $U \leftarrow$ Select random sequence of controls $U_{M+1:M+s}$.
  Forward simulate the robot poses given the controls.
  Add Maximum-Likelihood observations.
  Generate predicted graph $g(\Theta)$ and estimates.
  $V \leftarrow$ Estimate the value of objective function $J(U)$.
  if $V \leq V_{min}$ **then**
    $V_{min} \leftarrow V$
    $U_{min} \leftarrow U$
return U_{min}

b) Uncertainty Prediction: Instead of re-optimizing the trajectory in order to evaluate the objective function $J(U)$, we use an approximation to predict the joint covariance of the calibration parameters required for the evaluation. Future pose estimates are predicted given the latest pose and control inputs. We linearize the current graph around the current estimate and future pose estimates to generate the Jacobian A and the corresponding Hessian $A^T A$. In order to compute the covariance of Ω, Φ, we eliminate rest of the variables (non-calibration parameters: X, L) from the Hessian $A^T A$ using QR factorization and evaluate the determinant of remaining matrix corresponding to probability distribution $P(\Omega, \Phi)$ as shown in Equation 3. Algorithm 3 summarizes it.

$$|\Sigma|^{-1} = |A^T A| = \left| \begin{bmatrix} R & T \\ & A_{\Omega,\Phi} \end{bmatrix}^T \begin{bmatrix} R & T \\ & A_{\Omega,\Phi} \end{bmatrix} \right| \quad (2)$$

$$P(X, L, \Omega, \Phi) = P(X, L|\Omega, \Phi) P(\Omega, \Phi) \quad (3)$$

Algorithm 3: Uncertainty Prediction($g(\Theta)$,$\hat{X}$,$\hat{L}$,$\hat{\Omega}$,$\hat{\Phi}$)

$G \leftarrow$ Linearize $g(\Theta)$ around $\hat{X}$,$\hat{L}$,$\hat{\Omega}$,$\hat{\Phi}$
$(R, A_{\Omega,\Phi}) \leftarrow$ Eliminate X, L from G
$logdet \leftarrow \ln|A_{\Omega,\Phi} A_{\Omega,\Phi}^{-1}| = -2 \times \ln|A_{\Omega,\Phi}|$
return $\exp(logdet)$

The above-described algorithm plans a trajectory that minimizes the calibration uncertainty. The resulting plan will ensure that the robot moves in a direction that actively calibrates the sensor transforms.

IV. Experimental Results

In this section, we describe the experimental setup, define the metrics used for evaluation and show the results.

A. Setup

We evaluate our algorithm in a simulated environment with known data association and ground-truth parameter values and real environment using a robot platform. In rest of the experiments we assume that odometry bias (Ω) is known since the focus of this paper is to perform extrinsic calibration of an exteroceptive sensor. However, this algorithm can be easily extended to estimating odometry biases as well at the cost of increased non-linearity in the resulting objective function.

1) Simulated Environment: In simulation, we directly compare the result of our autonomous calibration algorithm against known extrinsic calibration parameter values for the exteroceptive sensor. It allows us to analyze the performance of the algorithm with different scenarios and parameter initializations, and noise in a controlled setting.

The Simultaneous Calibration Localization and Mapping algorithm is implemented using a factor graph approach as described in section III. For the planning, we evaluate the algorithm against the random walk algorithm as there is no an existing state of the art for this particular problem. In random walk, at each instant the robot moves in a random direction instead of moving in a direction which results in maximum reduction in uncertainty.

For the first set of experiments, we present four different 2D scenarios to compare the proposed approach against the random walk algorithm. These scenarios are shown in Fig. 5. In each scenario, the robot starts at the same location for both random walk algorithm and our proposed approach and we compare the algorithms based on the evaluation metrics proposed in section IV-B. We assume that there are no false positive data-associations as the robot moves through the scene. The robot is assumed to be a 2D point robot with a range-bearing sensor attached to the robot

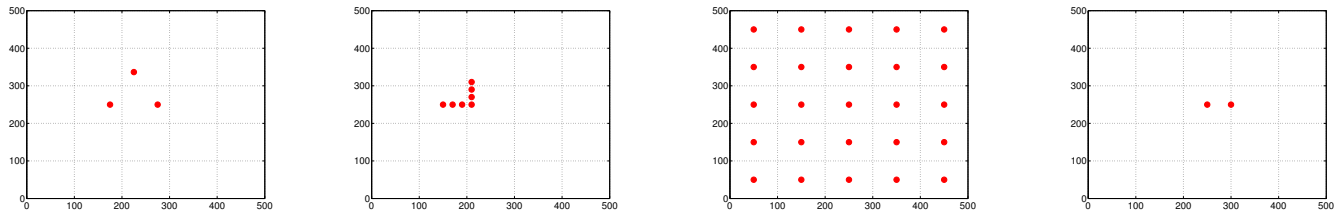

Fig. 5: Simulated Scenarios used for Experiments. Red dots represent the landmarks. Start location of the robot is near the center of each scene. The size of the world is 500x500.

with a known extrinsic transformation. For each scenario, we run each algorithm for 10 independent trials given the same start location and measure the covariance determinant of the estimated extrinsic transformation and the median transformation error to the known ground truth calibration. Evaluation metrics are explained in detail below. We log the evolution of these metrics with the number of planning iterations and demonstrate that using active planning results in faster convergence as compared to random walk algorithm. The default parameter values are explained in Table I.

Parameter	Value
#steps	1
step size	50
angular resolution	1 deg
#samples	360
sensor location	$(0,50,0)$
sensor initialization	$(0+\mathcal{N}(0,10),50+\mathcal{N}(0,10),0+\mathcal{N}(0,0.1rad))$

TABLE I: Default parameter values used in the simulated environment. #steps refer to the number of looks ahead steps considered during each iteration. step size is the size of each step taken in any direction. angular resolution divides 360 degrees into a set of directions which are considered at each time step. #samples refer to the number of monte-carlo samples considered. sensor location is the default ground truth value at which sensor is placed and sensor initialization is the noisy initial sensor location used as input to the algorithm. $\mathcal{N}(\mu,\Sigma)$ is a normally distributed pseudorandom number generator having mean μ and variance Σ.

2) Gazebo Environment: For the second set of experiments in simulation, we use Gazebo[24] simulation environment to run our algorithm on a turtlebot. It is modified to attach a Hokuyo laser scanner at a certain height from the turtlebot base. Instead of maintaining landmarks in the environment and using landmark-pose constraints, we estimate pose-pose constraints through scan matching and add the corresponding pose-pose-transform factor similar to the odometry bias factors. Omnimapper [22] is used for mapping. We do not have any assumption on data-association as well.

In this experiment, we sample the control in the velocity space, and estimate the change in the confidence of the sensor extrinsic parameters to choose the next control. The default parameters for this are shown in Table II. Ground-truth (x,y,z) sensor location is at $(0,0,1.36m)$ and the sensor initialization used is $(1,-3,1.36m)$ which is chosen at random. We measured the rate of the convergence of the error of the estimate for 5 independent iterations with random start location.

Fig. 6: The figures above show an example reconfiguration for the experiments conducted. The figure on the top left shows the hokuyo laser on the top of the robot and the figure on the top right shows the Hokuyo laser towards the right of the robot. The figure on the bottom shows another possible configuration of the robot mounted with two Kinect sensors, a Roboteye RE05 and a Hokuyo laser.

Parameter	Value
#steps	1
#samples	100
range min	0.10 m
range max	30.0 m
measurement res	0.01 deg

TABLE II: Default parameter values used in Gazebo and Real environment. #steps refer to the number of look ahead steps considered during each iteration. #samples refer to the number of monte-carlo samples considered. Range min refers to the minimum range of the sensor. Range max refers to the maximum range of the sensor. measurement res is the resolution of the sensor.

3) Real Environment: For experiments in a real environment, we run our algorithm on a physical robot. The robot platform consists of a Segway RMP-200 mobile base, which has been modified to be statically stable. For experiments, we mounted a Hokuyo UTM30LX-EW laser Range Finder at a certain height. Computation is performed on an onboard laptop. To evaluate the system, the robot was allowed to attempt self-calibration in different locations in an indoor lab environment. The robot is shown in Fig. 6. This was

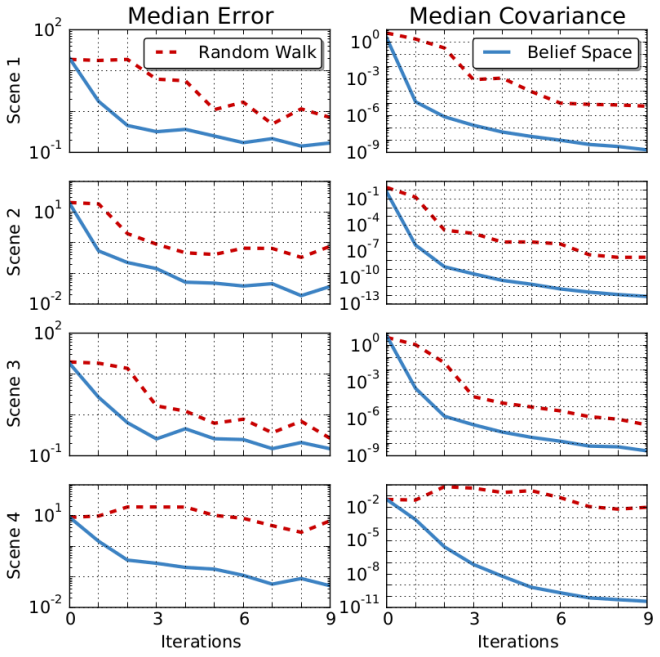

Fig. 7: Plots showing evolution of median transformation error (left) and median covariance determinant (right) over planning iterations. Blue line shows the results using our approach and Red dotted line shows the result using Simultaneous Calibration Localization and Mapping (SCLAM) + random walk algorithm. Median error and covariance in all the scenes converge faster to the groundtruth values using our appraoch as compared to using random walk.

an example scenario for the experiment where the system was evaluated to test the variance against the reconfiguration. Similar to the Gazebo environment, we use scan matching to add pose-pose constraints instead of adding landmark-pose constraints.

B. Evaluation Metrics

Below we explain the evaluation metrics used to compare our algorithm against random walk algorithm.

1) Median transformation error: It is evaluated by computing the median over the translation errors when comparing the ground truth sensor translation values to the estimated sensor translation values. The translation errors are computed after every planning iteration and the median translation error is computed over 10 independent trials of the algorithm.

2) Median covariance determinant: Covariance determinant is evaluated by computing the determinant of the marginal uncertainty of the extrinsic transformation parameter. Similar to the previous metric, we compute covariance determinant after every planning iteration and compute its median over 10 independent trials of the algorithm.

C. Results

1) Simulated Environment: We compared our algorithm against random walk algorithm and plotted the evolution of median transformation error and median covariance error

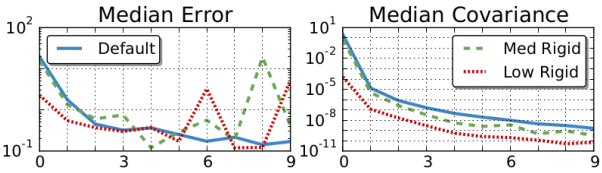

Fig. 8: Plots showing evolution of error (left) and covariance (right) for different plan rigidities. Blue plot uses default parameter value, green dotted plot uses medium plan rigidity and red dotted plot uses low plan rigidity.

as shown in Fig. 7. We used the default parameters as summarized in Table I and ran it on scenes shown in Fig. 5.

In median covariance comparison (fig. 7), the median of the covariance of the sensor transform for all of the 10 runs is plotted with the x-axis being the planning iteration. The proposed method performs better on all of the scenes. Scene 3 is the only one where Simultaneous Calibration Localization and Mapping along with random walk performs comparably.

Similarly, in median error comparison (fig. 7), the median of the absolute error (translation error between the estimate and the ground truth) for all of the 10 runs is plotted with the x-axis being the planning iteration. The proposed method performs significantly better in all the scenes except 3. This is because Scene 3 is a grid of landmarks with a lot of landmarks being visible as the robot moves in the scene. This provides good calibration uncertainty convergence irrespective of the used strategy (random walk or active planning based calibration).

Below we analyze the effect of varying parameter values on the evolution of median error and covariance values.

a) Rigidity of plan in each iteration: We reduce the rigidity of plan in each iteration by reducing the step size and increasing the number of steps. We experiment with three configurations of rigidity, default: (step size: 50, #steps: 1), medium rigidity: (step size: 25, #steps: 2), low rigidity: (step size: 10, #steps: 5). Fig. 8 summarizes the results. Reducing the rigidity of the plan increases the variance in median error but has better median covariance than more rigid plans. The reason behind this is that reducing the rigidity might result in robot getting stuck in local minima even though it leads to faster covariance reduction.

b) Angular Resolution: We experimented with three different values of angular resolution, default: 1 degree, medium: 2 degrees, high: 4 degrees. The angular resolution affects the accuracy of the measurement and can result in missed landmarks. Fig. 9 show the results. The varying angular resolution has an insignificant effect on the number of median error and covariance values.

c) Number of Monte-Carlo Samples: We experimented with three configurations of the number of monte-carlo samples used in the Algorithm 2. The configurations we experimented with are default: 360, medium: 180, low: 90. Fig. 10 show the results. Reducing the number of samples have an insignificant effect on the error and covariance values.

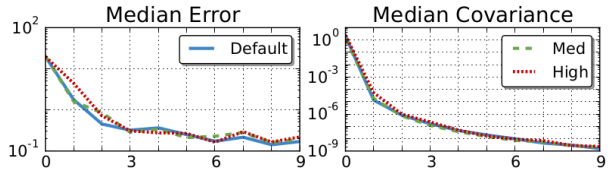

Fig. 9: Plots showing evolution of error (left) and covariance (right) for different angular resolution. Blue plot uses the default parameter value, green dotted plot uses 2 degrees angular resolution and red dotted plot uses 4 degrees angular resolution.

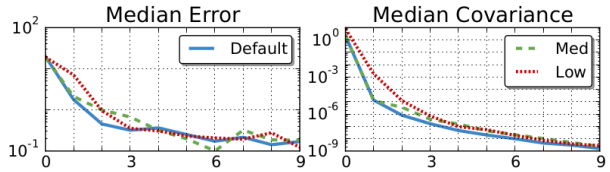

Fig. 10: Plots showing evolution of error (left) and covariance (right) for different number of samples. Blue plot uses the default (360) number of samples, green dotted plot uses 180 samples and red dotted plot uses 90 samples.

d) Robustness to bad initialization: To analyze the robustness of our algorithm to bad initializations, we experiment with three configurations of sensor location initializations, default: $(0+\mathcal{N}(0,10), 50+\mathcal{N}(0,10), 0+\mathcal{N}(0,0.1rad))$, medium noise: $(0+\mathcal{N}(0,20), 50+\mathcal{N}(0,20), 0+\mathcal{N}(0,0.5rad))$, large noise: $(0+\mathcal{N}(0,40), 50+\mathcal{N}(0,40), 0+\mathcal{N}(0,1rad)$. $\mathcal{N}(\mu,\Sigma)$ is a normally distributed pseudo random number generator having mean μ and variance Σ. Fig. 11 summarizes the result. Increasing the initialization noise has an insignificant effect on the convergence. This implies that the objective function has a wide basin of attraction and can converge given bad initializations as well.

2) Gazebo Environment: We experimented in the simulated Gazebo environment with 10 trials with random initial locations and random initial poses of the sensor. We measured the convergence of the absolute error with respect to the known ground truth over time. This is summarized in the Fig. 12. This shows that the robot is able to estimate the extrinsic of the sensor in a timely fashion and to high degrees of accuracy. This can consistently be seen across the random restarts and this shows that the algorithm is able to converge quickly for each of the test cases. As can be seen in the graph, the robot can autonomously calibrate the sensor with required accuracy within 300 seconds.

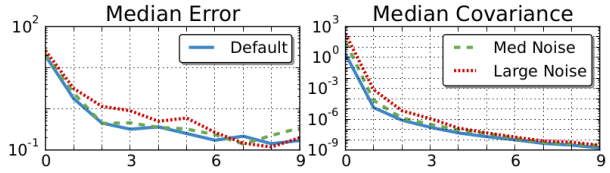

Fig. 11: Plots showing evolution of error (left) and covariance (right) for different sensor initializations. Blue plot uses the default initialization, green dotted plot uses medium initialization noise and red dotted plot uses high initialization noise.

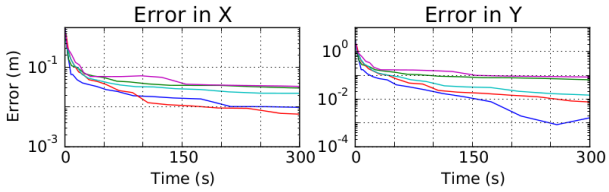

Fig. 12: Plots showing evolution of error over time for 5 separate gazebo simulations. The y axis is in m and the x axis is time in seconds. The lines represent the error over time for each of the random restarts.

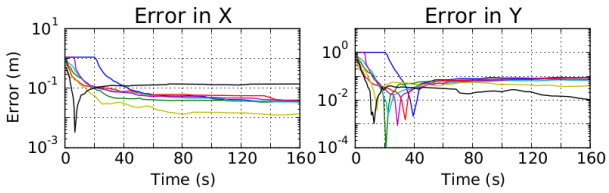

Fig. 13: Plots showing evolution of error over time for 7 separate real robot experiments. The y axis is in m and the x axis is time in seconds. The lines represent the error over time for each of the random restarts and different configurations of the robot.

3) Real Environment: For the real experiment, we measured the ground truth extrinsic calibration by hand. We experimented with different configurations to measure the absolute error with respect to time. All the sensors in the different configurations are calibrated separately. This is summarized in Fig. 13. This shows that the approach can be used effectively to estimate the extrinsic parameters of the sensor on the real robot quickly. The estimation of the sensor is also invariant to changing the pose of the sensor with respect to the base. This is consistent with the theoretical analysis and the simulations.

V. Conclusion and Future Work

In robotics, proper extrinsic calibration is often mission critical and the effects of this are apparent. For instance, in the DARPA robotics challenge, a fall could result in throwing off the extrinsic calibration. This, in turn, affects other tasks such as the teleoperation, and perception. Online techniques that have the ability to self-calibrate can help mitigate such issues to a certain extent. However, this would be more useful if the robot could perform a self-calibration routine before performing any operation.

The laborious and human error prone task of calibration requires a researcher to collect data for a camera given a calibration object manually. This is often random motion but does presuppose some knowledge of the calibration procedure with some level of experience, and so may not be purely random. As seen from the results, however, the proposed algorithm performs better than merely a random walk approach that thought of as a novice human performing

manual calibration, but more so in accordance of an expert who's each motion serves to reduce a specific uncertainty.

When calibration is performed one of the points of note is that the person performing the calibration will attempt to collect data at as many unique poses transforms between the sensor and the calibration targets. This intuitively can be seen in the plots from the trajectories generated in the simulation that the robot chooses to reduce its uncertainty.

Although a human has an intuition of what kind of motion helps generate better results for the camera calibration problem, this is not equivalent to having a numerical estimate of the variance in the change in error between steps. This often means that an overabundance of data is collected to ensure the quality of the calibration. Performing this task manually every time a configuration change occurs on the robot is a time-consuming task as well as challenging from a repeatability and reproducibility standpoint.

In the proposed algorithm, we use the idea of a look ahead to see what the consequence of an action is, and this allows us to estimate the parameter very close to the ground truth in fewer steps. Some of the constraints of this type of setup are that the amount of information available in the world is limited to what we can observe and what we have observed. This means that we can have an initial belief state estimate where the world is completely unknown and the belief state can be updated over time. The number of possible belief states is large and searching through all possible situations is not possible. This restricts the amount of look ahead that will actually help the algorithm. This highlights one of the main differences between our work and similar past research. We believe that this research will greatly benefit the mobile robot community as a whole and allow us to reduce the time taken on calibration routines between experiments.

One of the interesting observations, while performing the experiments was that there were certain emergent behaviors. For instance, the robot initially prefers taking paths that are cyclic and this could be attributed to a notion of closing the loop. This is an interesting behavior as it relates the active planning to intuition.

In this paper, we presented a novel solution that can quickly and accurately measure the extrinsic parameters of a sensor using active planning. One of the most important aspects of this work is the ability to estimate the pose of the sensor with respect to the base of the robot in an unknown world in a timely fashion. This eliminates the need of known calibration objects and the requirement of having a human in the loop performing this task. This can help eliminate the human error involved in performing this task and reduce the amount of repetitive manual labor required. For future work, we propose to show that this work could be used to reason about the failure of a sensor by continuously estimating the transform and comparing it against its history.

ACKNOWLEDGMENT

This work was generously supported by U.S. Army Research Lab (ARL) trough the MAST-CTA project 329420.

REFERENCES

[1] Z. Zhang, "Flexible camera calibration by viewing a plane from unknown orientations," in *Int. Conf. on Computer Vision ICCV*, 1999.

[2] ——, "A flexible new technique for camera calibration." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, 2000.

[3] J. Heikkilä, "Geometric camera calibration using circular control points," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 10, 2000.

[4] N. Roy and S. Thrun, "Online self-calibration for mobile robots," in *1999 IEEE Int. Conf. on Robotics and Automation*, 1999.

[5] Q. V. Le and A. Y. Ng, "Joint calibration of multiple sensors." in *Proceedings on the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2009.

[6] T. Svoboda, H. Hug, and L. Van Gool, "ViRoom — low cost synchronized multicamera system and its self-calibration," in *Pattern Recognition, 24th DAGM Symposium*, ser. LNCS, no. 2449. Springer, 2002.

[7] R. Unnikrishnan and M. Hebert, "Fast extrinsic calibration of a laser rangefinder to a camera," *CMU-RI-TR-05-09*, 2005.

[8] Q. Zhang and R. Pless, "Extrinsic calibration of a camera and laser range finder (improves camera calibration)." in *Proceedings on the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2004.

[9] C. Mei and P. Rives, "Calibration between a central catadioptric camera and a laser range finder for robotic applications." in *ICRA*, 2006.

[10] D. Scaramuzza, A. Harati, and R. Siegwart, "Extrinsic self calibration of a camera and a 3d laser range finder from natural scenes," in *Proc. of The IEEE Int. Conf. on Intelligent Robots and Systems (IROS)*, 2007.

[11] C. Gao and J. R. Spletzer, "On-line calibration of multiple lidars on a mobile vehicle platform," in *IEEE Int. Conf. on Robotics and Automation, (ICRA)*, 2010.

[12] L. Heng, P. T. Furgale, and M. Pollefeys, "Leveraging image-based localization for infrastructure-based calibration of a multi-camera rig," *J. Field Robotics*, vol. 32, no. 5, 2015.

[13] E. M. Foxlin, "Generalized architecture for simultaneous localization, auto-calibration, and map-building," in *Proceedings on the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2002.

[14] D. Caltabiano, G. Muscato, and F. Russo, "Localization and self-calibration of a robot for volcano exploration," in *Proceedings on the IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2004.

[15] S. T. Jesse Levinson, "Automatic online calibration of cameras and lasers," in *Robotics: Science and Systems*, 2013.

[16] J. Levinson and S. Thrun, "Automatic calibration of cameras and lasers in arbitrary environments," in *International Symposium on Experimental Robotics*, 2012.

[17] R. Kuemmerle, G. Grisetti, and W. Burgard, "Simultaneous calibration, localization, and mapping," in *Proceedings of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems IROS*, 2011.

[18] ——, "Simultaneous parameter calibration, localization, and mapping," vol. 26, no. 17, 2012.

[19] R. Valencia, M. Morta, J. Andrade-Cetto, and J. Porta, "Planning reliable paths with pose slam," *IEEE Transactions on Robotics*, vol. 29, no. 4, Aug 2013.

[20] V. Indelman, L. Carlone, and F. Dellaert, "Planning under uncertainty in the continuous domain: a generalized belief space approach," in *Proceedings on the IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2014.

[21] J. van den Berg, S. Patil, and R. Alterovitz, "Motion planning under uncertainty using iterative local optimization in belief space," *The International Journal of Robotics Research*, vol. 31, no. 11, 2012.

[22] A. J. Trevor, J. G. Rogers III, and H. I. Christensen, "Omnimapper: A modular multimodal mapping framework," in *Proceedings on the IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2014, pp. 1983–1990.

[23] F. Dellaert and M. Kaess, "Square Root SAM: Simultaneous localization and mapping via square root information smoothing," *The International Journal of Robotics Research*, vol. 25, no. 12, Dec 2006.

[24] N. Koenig and A. Howard, "Design and use paradigms for gazebo, an open-source multi-robot simulator," in *Proceedings on the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2004.