Geometric Calibration of Single-Pixel Distance Sensors

Carter Sifferman[1], Dev Mehrotra[1], Mohit Gupta[1], and Michael Gleicher[1]

***Abstract*—Single-pixel distance sensors are a low-power, low-cost option for distance ranging, and are often attached to robots for collision detection and avoidance. The relative sensor pose, i.e., its position and orientation relative to the robot, must be known to relate its measurements to 3D scene geometry. However, sensor pose is difficult to measure accurately, which has precluded the use of single-pixel sensors from applications such as environment mapping and precise collision avoidance. In this work, we provide a calibration procedure that can accurately determine the pose of a single-pixel distance sensor given only the known motion of the robot and an unknown planar target. We establish a geometric relationship between the relative sensor pose, robot motion, and an arbitrary plane, and show that the plane and sensor parameters can be recovered via nonlinear optimization. The result is a practical procedure for sensor calibration. We evaluate the procedure in simulation and in real world experiments, and provide an open source implementation. We consider two commonly available sensors (ST VL6180X and ST VL53L3CX) and characterize them to show that while they deviate from the idealized model used in our derivation, their poses can be recovered precisely and used for effective 3D scene reconstruction.**

***Index Terms*—Calibration and Identification, Localization, Range Sensing**

I. Introduction

SINGLE-PIXEL distance sensors return only a single approximate distance measurement. Despite being low-fidelity and having only single-pixel resolution, these sensors find several applications due to their small size, low cost and minimal compute and power requirements. For example, they are often attached to robots and vehicles as proximity sensors for collision detection and avoidance. However, the location and pose of these sensors relative to the platform that they are mounted on (e.g., robot, vehicle) is typically not known precisely. This makes it challenging to relate sensor measurements to the precise 3D geometry of the environment, as a small error in estimated sensor pose can result in large errors in the reconstructed geometry. For example, if the sensor is rotated by a few degrees, a robot might incorrectly estimate the location of an obstacle by many centimeters. Because of this weakness, tasks such as environment mapping, localization, and precise collision avoidance are infeasible with single-pixel distance sensors alone.

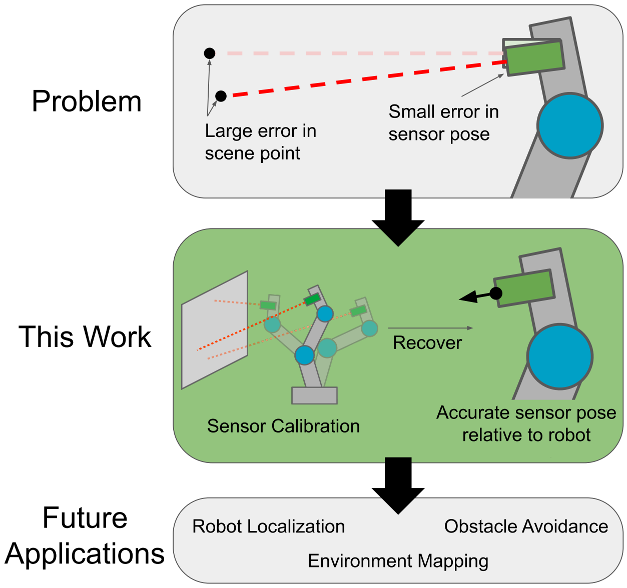

Fig. 1. **This work enables extrinsic calibration of distance sensors attached to a robot arm.** Accurate sensor calibration is critical, as small errors in sensor pose can lead to large errors in scene points. We provide a calibration procedure that uses known robot motion to accurately determine the sensor pose based on observations of an arbitrary plane. Using this recovered sensor pose, more meaningful interpretation of sensor measurements is possible. This is a step towards using distance sensors for precise tasks.

In this work, we consider a single-pixel distance sensor attached to a robot arm, and show that it is possible to recover the sensor's 5D pose relative to the robot via an easy-to-perform calibration procedure. Standard camera calibration procedures do not apply because a single-pixel sensor cannot provide correspondences. Therefore, in order to perform calibration we derive a relationship between the pose (position and orientation) of a distance sensor relative to a robot segment, the motion of the same robot segment, and a calibration plane at which the sensor is pointed. Assuming the sensor makes multiple observations of the plane, this relationship enables us to recover the 5D pose of the sensor and 3D pose of the plane without initial estimates or constraints on their poses. We demonstrate this calibration procedure both in simulations and in the real world with a single-pixel distance sensor attached to a robot arm.

The geometric relationship that we derive is based on an idealized sensor model in which distance is measured along

Manuscript received: February, 24, 2022; Revised May, 10, 2022; Accepted April, 26, 2022.

This paper was recommended for publication by Editor L. Pallottino upon evaluation of the Associate Editor and Reviewers' comments. This work was supported in part by NSF Awards 1830242, 1943139, 2107060, and 2003129, and a UW-Madison Vilas Award.

[1] Carter Sifferman, Dev Mehrotra, Mohit Gupta, and Michael Gleicher are with the Department of Computer Sciences, University of Wisconsin-Madison, Madison 53706, USA [sifferman|mehrotra|mohitg|gleicher]@cs.wisc.edu

Digital Object Identifier (DOI): see top of this page.

a single direction, assuming an infinitesimal field-of-view (FOV) along a single ray. In practice, single-pixel distance sensors have a finite conical FOV, as well as several other noise sources that corrupt their distance measurements. To understand how these deviations might affect our calibration procedure, we empirically evaluate two low-cost, low-power distance sensors (ST VL6180X[1] and ST VL53L3CX[2]). In Section IV, we show empirical sensor characterization results (precision and robustness to oblique angles), and in Section VI, we demonstrate that our proposed geometric recovery techniques are robust to these deviations in the real world.

Implications– Applications in which single-pixel distance sensors are attached to a robot will benefit from our calibration procedure as it determines the accurate pose of the sensor relative to the robot. Our work enables applications that require geometric accuracy, such as localization, mapping and precise collision avoidance. Accurate calibration also allows combining multiple sensors to create a "distributed camera" where the known relationship between sensors enables combining their observations into a single geometric model. The calibration procedure is particularly useful in development and laboratory settings, where sensors are attached by hand and may be relocated often.

II. Related Work

A. Single-Pixel Distance Sensors on Robots

Single-pixel distance sensors have been used on robot arms for obstacle detection. Early work used infrared or ultrasonic sensors to avoid collisions [1], [2]. Avanzini et. al. [3] used a CAD model of a robot and optimization to find optimal distance sensor placements on the robot and implemented a control strategy which avoids collision with humans. While this previous work considers *where* to place distance sensors, it does not consider *how* to determine their location once they are placed. Tsuji et. al. [4] created an array of 54 single-pixel distance sensors to mount on a robot, and implemented a control strategy to prevent collisions. Similarly, rings of distance sensors have been characterized for their precision and theoretical performance at obstacle avoidance [5], [6], [7]. Himmelsbach et. al. [8] considered the problem of a distance sensor detecting the robot itself and introduced a method for predicting self-detection. In each of these cases, distance sensors are placed manually on the respective robot, and no geometric calibration is performed before using them to detect and avoid collisions. Because of this, sensor poses are not well-known, and these approaches cannot take full advantage of the distance measurements.

Gandhi and Cervera [7] introduced a way to perform collision avoidance without explicitly determining the locations and poses of the sensors on the robot. Their approach uses Q-learning to determine appropriate actions for avoiding a collision without estimating where the sensors are. Since they do not aim to recover the positions of the sensors, this method is not suitable for other downstream tasks such as 3D mapping of the environment and path planning.

B. Sensor Calibration

Extrinsic camera calibration is a classical problem in computer vision and robotics [9]. In robotics, extrinsic calibration of a camera in the robot's coordinate frame is necessary to relate 2D points in the image to 3D points in the robot's environment. Existing approaches rely on finding feature and 2D keypoint correspondences between images [10], [11]. Other approaches calibrate full depth cameras by matching dense 3D geometry of moving objects across frames [12]. These approaches are not applicable to single-pixel sensors where each measurement is only a single distance value rather than a 3D point cloud or 2D image, making it impossible to extract image features or perform dense geometry matching. Shiu and Ahmad [13] introduced a method for using robot motions to calibrate a sensor attached to a robot. However, this requires that the sensor be able to detect the 6D pose of some calibration object from each robot position. The closest related work to our approach is by Watanabe et. al. [14] who recover the 6D pose of a single-pixel distance sensor on a robot arm by coupling the sensor with an inertial measurement unit (IMU), and relating IMU measurements to robot motions. In contrast, our work does not require an IMU, but instead uses a calibration plane. To our knowledge, our work is the first to perform geometric extrinsic calibration of a single-pixel distance sensor on a robot using measurements from the sensor itself.

The geometric constraint that we use was introduced by Thanh et. al. [15] to extrinsically calibrate a single-point laser range finder and camera pair. Using this constraint, they recover the pose of a range finder which is facing a known moving plane. While we use the same geometric relationship as Thanh et. al., our approach enables extrinsic calibration of the sensor relative to the robot, rather than relative to a camera, and is able to simultaneously recover the sensor pose and plane pose.

C. Sensor Characterization

The sensors that we use are based on single photon avalanche diode (SPAD) technology [16]. Recently, researchers have performed detailed characterization of such single-pixel distance sensors. Jans et. al. [17] characterize the STMicroelectronics ST VL53L1X SPAD sensor array, which, like our sensor, utilizes SPAD technology, but has an array of receivers compared to the single receiver in the VL6180X and VL53L3CX. They present results for distance measurement accuracy, noise, error, and range of ambient illumination under which the sensor performs reliably. Some of these results are likely to hold true for other SPAD sensors, such as the sensors we use. In this work, we evaluate the geometric properties of the VL6180X and VL53L3CX that are likely to affect their performance in our calibration procedure.

III. Geometric Constraint

A. Problem Overview

In this section, we describe our calibration procedure based on moving the robot arm while the sensor observes a planar surface as calibration target, and derive the associated

[1] https://www.st.com/resource/en/datasheet/vl6180x.pdf

[2] https://www.st.com/resource/en/datasheet/vl53l3cx.pdf

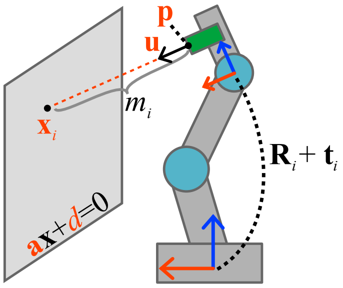

Fig. 2. **Visualization of notation used in Section III-B.** We consider a distance sensor attached to a robot arm. The sensor's pose in the local coordinates of the segment it is attached to is given by its position $\mathbf{p}$ and orientation as a unit vector $\mathbf{u}$. Robot motions are given as a transform $\mathbf{R}_i, \mathbf{t}_i$, which relates some global coordinate system (shown at robot base) to the local coordinate system of the segment. Sensor readings are given by m_i, and the point which the sensor sees is given by $\mathbf{x}_i$. The plane which the sensor sees is described by the parameters $\mathbf{a}$ and d in the equation $\mathbf{ax} + d = 0$. Unknown variables are shown in red.

geometric constraint. For ease of exposition, we derive this constraint assuming an idealized "point-imaging" model (i.e., sensor measuring distance along a single direction).

Consider a single-pixel distance sensor attached to a rigid segment of a robot whose movements are known. The sensor can be described by its position $\mathbf{p} = \langle p_x, p_y, p_z \rangle$ and orientation, given by a unit vector $\mathbf{u} = \langle u_x, u_y, u_z \rangle$ relative to the coordinate system of the robot segment to which it is attached (illustrated by colored axes at the end effector in Figure 2). The motions of the segment can be described by a rotation $\mathbf{R}$ and translation $\mathbf{t}$ between a global coordinate system (usually defined by the robot at its base) and the segment's local coordinate system.

B. Derivation of Geometric Constraint

Suppose the robot segment is moved to n positions given by $\mathbf{R}_1 ... \mathbf{R}_n$ and $\mathbf{t}_1 ... \mathbf{t}_n$. Then, for the i^{th} robot segment pose, the position of the sensor in global coordinates is given by $\mathbf{R}_i \mathbf{p} + \mathbf{t}_i$, and its orientation is given by $\mathbf{R}_i \mathbf{u}$.

Suppose the sensor is facing a planar surface in the world, described by its normal vector $\mathbf{a} = \langle a_x, a_y, a_z \rangle$ and offset d in global coordinates, via the equation $a_x x + a_y y + a_z z + d = 0$. For the i^{th} robot segment pose ($1 \leq i \leq n$), let the distance to the plane measured by the sensor be m_i. Assuming no measurement noise, the coordinates of the point $\mathbf{x}_i$ where the sensor 'sees' the plane is given by:

$$\mathbf{x}_i = \mathbf{R}_i \mathbf{p} + \mathbf{t}_i + m_i \mathbf{R}_i \mathbf{u} \, . \tag{1}$$

We also know that $\mathbf{x}_i$ lies on the plane, so:

$$\mathbf{a} \cdot \mathbf{x}_i + d = 0 \, . \tag{2}$$

Substituting 1 into 2 and rearranging provides:

$$(\mathbf{a} \cdot \mathbf{R}_i \mathbf{p}) + \mathbf{a} \cdot (m_i \mathbf{R}_i \mathbf{u}) = -d - \mathbf{a}\mathbf{t}_i \, . \tag{3}$$

This equation is the key geometric constraint that relates sensor pose, sensor distance measurements, plane parameters, and robot motion. We assume that robot motions ($\mathbf{R}_1 ... \mathbf{R}_n$ and $\mathbf{t}_1 ... \mathbf{t}_n$) are exactly known via precise forward kinematics available on industrial robots. Distance measurements ($m_1 ... m_n$) are also known, although with noise. Our goal is to recover the pose of the distance sensor via the unknown parameters $\mathbf{p}$ and $\mathbf{u}$, which amounts to extrinsic calibration of the sensor relative to the robot. We also recover the parameters $(\mathbf{a}, d)$ of the plane that the sensor is facing.

C. Nonlinear Solution

In the case where neither the sensor pose ($\mathbf{p}$, $\mathbf{u}$) nor the plane pose ($\mathbf{a}$, d) are known, but the robot motions are known, the distance between the points $\mathbf{x}_1 ... \mathbf{x}_n$ and the plane can be minimized directly as follows:

$$\begin{aligned} \min_{\mathbf{p},\mathbf{u},\mathbf{a},d} \quad & \sum_{i=1}^{n} ((\mathbf{a} \cdot \mathbf{R}_i \mathbf{p}) + \mathbf{a} \cdot (m_i \mathbf{R}_i \mathbf{u}) + d + \mathbf{a}\mathbf{t}_i)^2 \\ \text{subject to} \quad & ||\mathbf{u}||_2 = 1, \\ & ||\mathbf{a}||_2 = 1 \end{aligned} \tag{4}$$

The constraints on the length of $\mathbf{a}$ and $\mathbf{u}$ eliminate extraneous degrees of freedom in the parameterization of the sensor orientation and plane pose, respectively. Without them, an optimal solution would be to set all free variables to 0.

To implement this nonlinear solver, we use the `scipy.optimize.minimize` function in the SciPy Python library[3], with `solver="slsqp"`. The SLSQP Solver [18] uses a quasi-Newton approach that is well suited for such non-linear least squares problems. Our open source implementation is available online[4].

Iterative non-linear solvers, such as SLSQP, require an initial estimate for free variables, and SLSQP is not guaranteed to find the global minimum for any one set of initial estimates. We prefer not to rely on an initial estimate of the sensor pose. To avoid local minima, we use a search process that solves from each of a pre-determined set of initial estimates and chooses the found solution with the lowest loss. We search over the following values for $\mathbf{u}$: $\langle 1,0,0 \rangle, \langle 0,1,0 \rangle, \langle 0,0,1 \rangle, \langle -1,0,0 \rangle, \langle 0,-1,0 \rangle, \langle 0,0,-1 \rangle$, while maintaining that $\mathbf{a} = \langle 0,0,-1 \rangle$, $\mathbf{p} = \langle 0,0,0 \rangle$ and $d = 0$. Through experimentation, we found that searching over values for $\mathbf{u}$ allows for a smaller search set than other variables. The choice of $\mathbf{a}$ and d is arbitrary, and has little effect. While this approach provides no guarantee of finding a global optimum, we find that it is effective in practice. Section V-A and Section VI show that the process finds good solutions efficiently and reliably in simulation and the real world.

D. Degenerate Cases

The nonlinear system given in Equation 4 has eight degrees of freedom in the unknowns; three DoF in sensor position, two DoF in sensor orientation, and three DoF in the plane parameters. Therefore, in the absence of noise, eight observations are sufficient to find a solution for all unknowns. However, there exist degenerate cases in which more observations are needed.

Consider the scenario where the plane parameters $\mathbf{a}$ and d are known, which amounts to knowing the pose of the

[3] https://scipy.org/

[4] https://github.com/uwgraphics/SPD-Geometric-Calibration

calibration plane. In this case, Equation 3 can be expressed as a linear system $\mathbf{Ax} = \mathbf{b}$ where $\mathbf{r}_{ji}$ refers to the j^{th} column of $\mathbf{R}_i$:

$$\mathbf{A} = \begin{bmatrix} \mathbf{ar}_{11} & \mathbf{ar}_{21} & \mathbf{ar}_{31} & m_1\mathbf{ar}_{11} & m_1\mathbf{ar}_{21} & m_1\mathbf{ar}_{31} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{ar}_{1n} & \mathbf{ar}_{2n} & \mathbf{ar}_{3n} & m_n\mathbf{ar}_{1n} & m_n\mathbf{ar}_{2n} & m_n\mathbf{ar}_{3n} \end{bmatrix}$$

$$\mathbf{x} = \begin{bmatrix} p_x \\ p_y \\ p_z \\ u_x \\ u_y \\ u_z \end{bmatrix}, \mathbf{b} = \begin{bmatrix} -d - \mathbf{at}_1 \\ \vdots \\ -d - \mathbf{at}_n \end{bmatrix} \tag{5}$$

Although this resulting linear system has 6 constraints, there are only 5 DoF when the additional nonlinear constraint $||\mathbf{u}||_2 = 1$ is included, as in Equation 4. Therefore, $rank(\mathbf{A}) >= 5$ is required for the system to have a unique solution. If this linear system is underdetermined, then the nonlinear system (Equation 4) to solve for both sensor and plane parameters is also underdetermined; if the nonlinear system has no unique solution with two of its parameters ($\mathbf{a}$ and d) fixed, then it has no unique solution with them free.

1) No rotations: If no rotations are present in the robot motions ($\mathbf{R}_1 ... \mathbf{R}_n$ are all equal), when solving the linear system given in Equation 5, the coefficient matrix $\mathbf{A}$ has the following form:

$$\mathbf{A} = \begin{bmatrix} \mathbf{ar}_{11} & \mathbf{ar}_{21} & \mathbf{ar}_{31} & m_1\mathbf{ar}_{11} & m_1\mathbf{ar}_{21} & m_1\mathbf{ar}_{31} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{ar}_{11} & \mathbf{ar}_{21} & \mathbf{ar}_{31} & m_n\mathbf{ar}_{11} & m_n\mathbf{ar}_{21} & m_n\mathbf{ar}_{31} \end{bmatrix}$$

Columns 1, 2, and 3 are linearly dependent, as are columns 4, 5, and 6, meaning $rank(\mathbf{A}) = 2$. As described previously, this means the linear system is underdetermined, and the nonlinear system given in Equation 4 is also underdetermined.

2) Measurements are same distance: Similarly to “no-rotation” degeneracy, if each distance measurement is the same ($m_1 ... m_n$ are all equal), the coefficient matrix $\mathbf{A}$ as given in Equation 5 is rank deficient, with the following form:

$$\mathbf{A} = \begin{bmatrix} \mathbf{ar}_{11} & \mathbf{ar}_{21} & \mathbf{ar}_{31} & m\mathbf{ar}_{11} & m\mathbf{ar}_{21} & m\mathbf{ar}_{31} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{ar}_{1n} & \mathbf{ar}_{2n} & \mathbf{ar}_{3n} & m\mathbf{ar}_{1n} & m\mathbf{ar}_{2n} & m\mathbf{ar}_{3n} \end{bmatrix}$$

Columns 4, 5, and 6 are equal to columns 1, 2, and 3 scaled by m, so $rank(\mathbf{A}) = 3$. Again, this means that the containing nonlinear optimization problem given in Equation 4 is underdetermined.

3) No spread on calibration plane: If the points $\mathbf{x}_1 ... \mathbf{x}_n$ where the sensor senses the plane lie on a line, the plane's true parameters are ambiguous, as there are infinitely many planes which pass through any one line. In this case, it is impossible to solve for $\mathbf{a}$ or d.

In practice, completely degenerate cases can be avoided by choosing random robot motions. In our real world tests in Section VI, we did not encounter a completely degenerate case, despite making no specific effort to avoid them beyond moving the robot with some randomness. In Section V, we verify these degenerate cases in simulation and show that accuracy is harmed when *near* a degenerate case.

IV. Sensor Characterization

Our theoretical model described so far assumes an ideal sensor that measures distance along a single ray and provides an exact measurement. In practice, real sensors measure distances over a finite cone, and provide noisy measurements. Sources of noise include finite geometric effects caused by the finite measurement cone, practical issues such as quantization, and the inherent randomness in measurement processes. In this section we examine two representative commercially available sensors to evaluate how far they deviate from the ideal sensor model and understand their effectiveness when used with our calibration procedure.

The sensors we examine are the ST VL6180X and the ST VL53L3CX. The VL6180X has been widely used in prior robotics applications [4], [19], [20], while the VL53L3CX is a newer sensor that offers a greater distance range. Both sensors are based on single photon avalanche diode (SPAD) technology [16], and use time of flight (ToF) to measure distance. SPADs come coupled with a light source which sends out a cone of light, and the sensor measures the return time of many photons over some interval, gathers an internal histogram of photon times-of-flight from that cone, and returns a single value by summarizing that histogram. This process leads to specific artifacts (systematic errors and noise) in the resulting measurements. In the following, we investigate how both sensors deviate from the idealized model in ways that are relevant to our calibration procedure.

A. Precision

While we expect our calibration procedure to smooth out noise over many measurements, understanding the magnitude of that noise is helpful for running simulated experiments in Section V. To measure this noise, we position either sensor orthogonal to a plane at various distances, and measure the standard deviation over 1000 distance measurements in Table I. The measurements are taken in a well-lit, windowless room against a light colored dry wall, the same environment in which we evaluate the calibration procedure itself in Section VI. We see that the VL53L3CX is better tuned to longer distances, while the opposite is true for the VL6180X.

B. Oblique Measurements

Because SPADs have a finite conical field-of-view (quoted at 25° for both the sensors we use), distance measurements of a plane may vary depending on the angle of incidence. We measure this effect with both sensors by pointing the sensors at a plane and changing the angle of incidence only, while maintaining the same idealized distance by tilting the plane along an axis which runs through the point where an idealized sensor “sees” the plane ($\mathbf{x}_i$ in Section III-B). We find that the VL53L3CX is robust to ± 40 °, and begins to under-report distance at more extreme angles, while the VL6180X

TABLE I
PRECISION OF SENSOR DISTANCE MEASUREMENTS

Distance (mm)	Standard Deviation (mm)	
	VL53L3CX	VL6180X
10	out of range	1.21
20	1.45	1.49
100	0.64	1.79
250	0.78	2.84
500	1.51	out of range

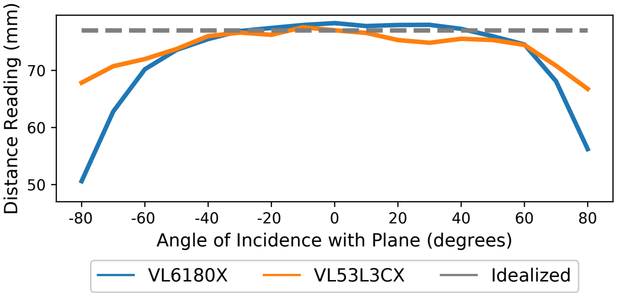

Fig. 3. **VL53L3CX distance sensor is more robust to oblique measurements of a plane.** Sensors were pointed at a planar surface, and the angle of incidence with the surface was changed, while maintaining a constant distance. While both sensors read near the idealized distance at low angles-of-incidence, after 40 degrees, both sensors' readings become inaccurate. We see larger inaccuracies at oblique angles in the VL6180X.

is robust to ± 40 °and under-reports distances more drastically. Therefore, when performing the calibration procedure, we avoid oblique angles beyond what the sensors are robust to, to prevent systematic errors.

C. Other Noise Considerations

While both the VL6180X and VL53L3CX are subject to temperature- and voltage-dependent drift, we operate at a constant voltage and temperature, and perform offset calibration under these conditions as instructed by the manufacturer before using either sensor. Both sensors are also subject to quantization noise, as they both quantize at 1mm. To overcome this, we perform macro-averaging over one second (about 30 measurements) for each distance measurement. Small amounts of noise in the readings effectively serve as dithering and average out to reduce quantization noise in the macro-averaged result.

V. SIMULATED EXPERIMENTS

We utilize the geometric relationship derived in Section III to recover sensor and plane poses from known robot motions. All simulations are run in Python, and motions generated are modeled as moving coordinate frames, and are not necessarily kinematically feasible on a given robot arm. We demonstrate the procedure on a real robot, with kinematically feasible motions in Section VI.

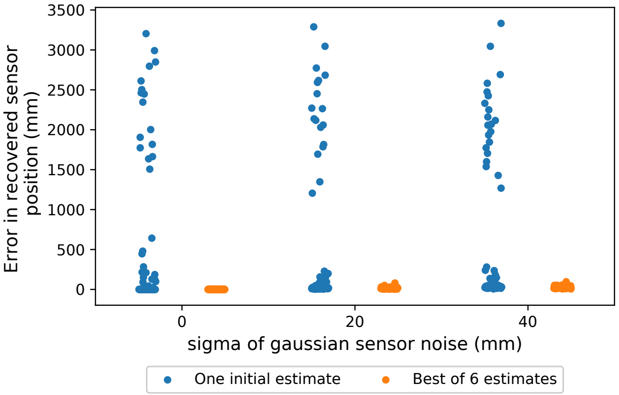

Fig. 4. **Search of six initial estimates for optimization solver results in lower error in simulation.** Blue points represent performing the optimization procedure with only one initial estimate for the free variables, while orange points represent picking the best of six initial estimates, based on loss. Solutions with a very high error rate are eliminated by search over initial estimates. 100 trials per column, 32 observations per trial.

A. Search of Initial Estimates

As discussed in Section III-C, the nonlinear SLSQP solver has the best chance of finding a global minimum when given a reasonable starting estimate for the free variables $\mathbf{p}$, $\mathbf{u}$, $\mathbf{a}$, and d. In practice, gathering this starting estimate is cumbersome, and negates the convenience of our calibration procedure. In Section III-C, we describe the search that we perform over initial estimates. Through simulation, we empirically observe that performing this search of initial estimates results in a good solution 100% of the time for 10,000 randomly generated possible robot motions under reasonable zero-centered Gaussian sensor noise models ($\sigma \leq 40$mm), as shown in Figure 4. When solving with an initial estimate search, average solution time is 0.7 seconds on a laptop with an Intel Core i7-10750H CPU at 2.6 GHz. This solution time is reasonable given that the calibration procedure only needs to be run once per sensor, and physically performing the robot motions takes much longer than running the solver. We use this search based approach throughout; all results shown are without the help of initial estimates.

B. Degeneracy Analysis

We simulate the three degeneracy conditions described in Section III-D to confirm that they are degenerate and evaluate how solution accuracy is affected when *near* a degenerate case. We simulate 100 randomly generated trials at varying proximity to each degenerate case. Simulations of no-rotation degeneracy are shown in Figure 5 (left), same-distance degeneracy in Figure 5 (center), and no-spread degeneracy in Figure 5 (right). We perform the experiments at a reasonable scale, using robot motions which are feasible with a Universal Robots UR5, and sensor noise similar to that found in Table I. In each case, we see that solution accuracy decreases as each degenerate case is approached. However, even at small distances from each degenerate case, error in the recovered parameters is reduced dramatically.

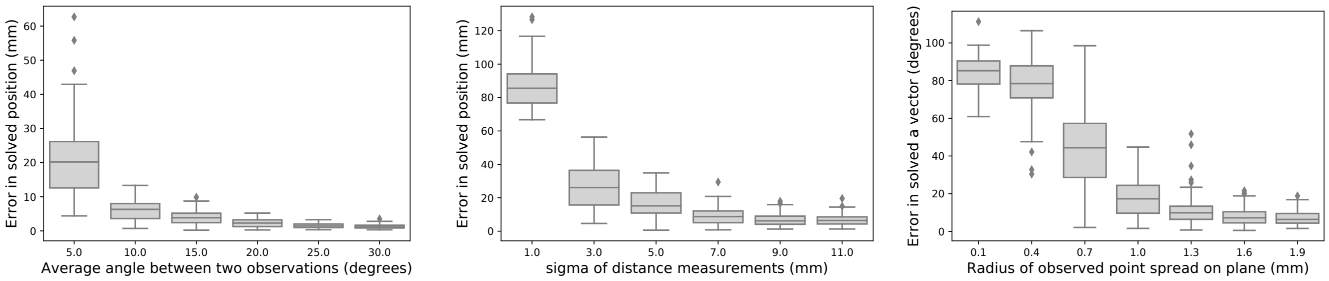

Fig. 5. **Simulated experiments demonstrate no-rotation, same-distance, and no-spread degeneracy.** For each type of degeneracy as described in Section III-D, error in the solved parameter (y axis) increases as the degenerate case is approached. However, relatively small deviations from each degenerate case dramatically reduce solution error. Each box represents 100 trials, each with 32 observations and Gaussian measurement noise ($\mu = 0$mm, $\sigma = 0.5$mm). Diamonds indicate outliers.

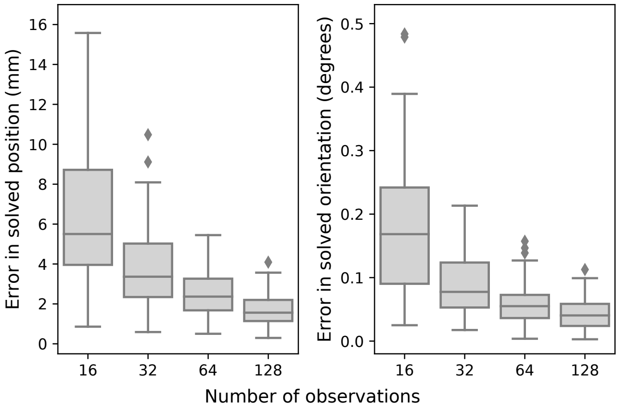

Fig. 6. **More observations leads to lower error in recovered sensor pose.** Our solver is able to overcome zero-centered sensor noise by averaging over many observations; the more observations that are available, the more effectively noise can be smoothed and an accurate solution found. Each box represents 100 trials with Gaussian measurement noise ($\mu = 0$mm, $\sigma = 0.5$mm). A few outliers in the 16 observation case have been hidden to allow reasonable y-axis scale. Diamonds indicate outliers.

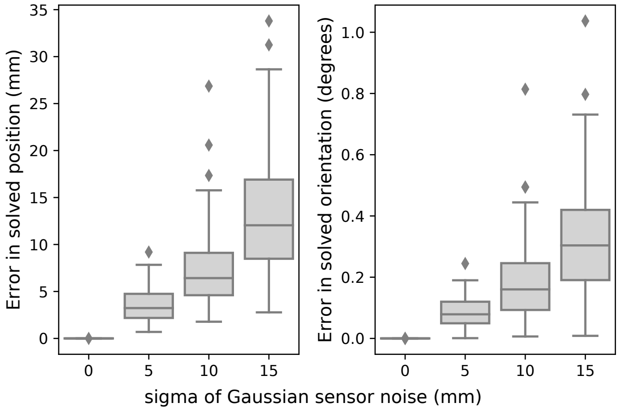

Fig. 7. **Error in solved sensor position increases with sensor noise.** We see a roughly linear relationship between sensor noise and solution error in both the **p** and **u** parameters, and errors in position are at the same order of magnitude as input noise. Each box represents 100 trials with 32 observations per trial. Diamonds indicate outliers.

C. Effect of Varying Number of Observations

Given the presence of zero-centered sensor noise, it is expected that, as the number of observations used to solve for the sensor pose increases, averaging effects will smooth over the sensor noise and the error in the solved sensor pose will decrease. To verify this, we run the calibration procedure in simulation with 16, 32, 64, and 128 observations and a constant ($\sigma = 0.5$mm) sensor noise. Robot motions are randomly generated and keep the sensor within one meter of its starting position, while looking at a square plane two meters across. The results of these simulations are shown in Figure 6. As expected, we find that error decreases as the number of observations increases.

D. Effect of Varying Sensor Noise

We expect noise in any known parameters (robot motions $\mathbf{R}_1 ... \mathbf{R}_n$ or sensor measurements $m_1 ... m_n$) to negatively impact the accuracy of the recovered sensor pose $(\mathbf{p}, \mathbf{u})$ and plane pose $(\mathbf{a}, d)$. Industrial robots, such as our test case, the Universal Robots UR5, are highly precise in their motions (quoted at ± 0.1mm [5]). Of most concern is the effect that noise in sensor measurements will have on the recovery of the sensor pose. To evaluate this effect, we generate scenarios in the same manner as in Section V-C, but keep the number of observations fixed, and vary sensor noise from $\sigma = 0$mm to $\sigma = 15$mm. The results of these simulations are shown in Figure 7. As expected, a higher σ leads to higher error in the recovered sensor pose. Additionally, we find that output error is on the same order of magnitude as input error, given 32 observations of the plane.

VI. Real World Experiments

We evaluate our calibration procedure in the real world by attaching a distance sensor (VL6180X or VL53L3CX) to a Universal Robots UR5 robot arm, as shown in Figure 8. The sensor is attached to a custom 3D printed end effector to ensure rigidity and consistency between trials. We interface with the sensor through an Arduino Nano connected to a

[5] https://www.universal-robots.com/media/50588/ur5_en.pdf

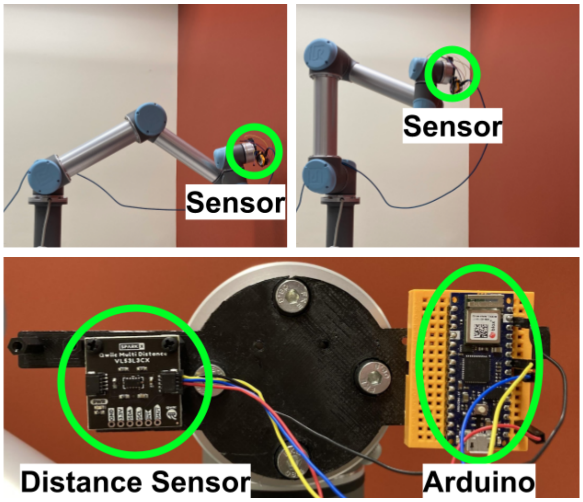

Fig. 8. **We test our calibration procedure in the real world by attaching distance sensors to a robot arm**. Top: robot arm in two different poses during calibration procedure; sensor is mounted at the end effector. Bottom: close-up view of sensor (VL53L3CX) and Arduino on the end-effector of the robot, as used to conduct real world experiments

Linux computer, and control the robot by sending end effector goals as transforms via the URX Python library [6]. We run the procedure a total of 16 times, each one with a different combination of sensor model, sensor pose, robot motions, and plane pose, and with 32 robot motions per trial. The robot motions are selected by hand to ensure kinematic feasibility. Attention was paid to make sure that for each robot motion: 1) the sensor points at the plane, 2) the sensor's angle-of-incidence to the plane is less than 50°, and 3) each pose is superficially different than the last. No specific effort was made to avoid degenerate robot motions.

A. *Solution Precision*

For both of our sensors, we perform the geometric calibration procedure eight times, given different robot motions, plane poses, and sensor poses. We then cross-validate between trials in which the sensor pose did not change to get a measure of the calibration procedure's *precision* in recovering the sensor pose. Average precision results are shown in Table II as the average deviation between solved sensor poses. We observe that the solved poses of the more optically robust VL53L3CX sensor are more precise than those of the VL6180X. In the case of the VL53L3CX, precision in position is on the order of magnitude of the size of the SPAD unit itself (which is 4.4 x 2.4 mm). Our real-world position precision results are comparable to what the IMU-based approach of [14] is able to achieve *in simulation*. Real-world quantitative results are not provided for the IMU-based approach. Our orientation precision results cannot be directly compared to [14], as their approach recovers three rotational degrees of freedom, while ours recovers two.

TABLE II
PRECISION OF SOLVED SENSOR POSE IN REAL WORLD TRIALS

Sensor	Average Deviation	
	Position (**p**)	Orientation (**u**)
VL53L3CX	3.18 mm	0.61°
VL6180X	7.29 mm	2.01°

B. *3D Reconstruction Using Single-Pixel Sensors*

While Table II quantifies the precision of our procedure, it does not directly establish the method's accuracy. Because the exact sensor pose is unknown, we assess the accuracy of our calibrated poses indirectly, by assessing their performance in a downstream task: 3D scene reconstruction. For each of 16 calibrated sensor poses, we point the sensor at a previously unseen plane from 32 different robot poses. For each robot pose, we then project a point onto the plane according to the true observed measurement and the calibrated sensor pose. We fit a plane to these 32 points, and measure the average orthogonal distance from the points to the best-fit plane (average residual). These average residuals are plotted as blue dots in Figure 9. In each trial with the VL53L3CX sensor, the average residual is $<$ 2mm, indicating a high-quality reconstruction of previously unseen planar object, and indicating that our calibration procedure is effective.

In a world with no noise, and a perfectly calibrated sensor, the average residual would be zero in every case, as every projected point would lie perfectly on the best-fit plane. In the real world, even with a perfectly calibrated sensor, the average residual will be non-zero due to sensor noise. To test whether our non-zero average residuals are due to sensor noise or to poor calibration, we add small perturbations (in the range $\mathbf{p}\pm$ 1cm, $\mathbf{u}\pm$ 10°) to our calibrated poses and perform the same projection and plane-fitting procedure. If our calibrated poses are inaccurate, we would expect some of the perturbed poses to better reconstruct the plane, and result in a lower average residual. The average residuals resulting from these perturbed sensor poses are plotted as orange dots in Figure 9. We find that, in almost all cases, a small perturbation causes a higher average residual to the best-fit plane. In the case of the VL53L3CX sensor, only two of 600 perturbed sensor poses better reconstruct a plane than the calibrated pose, across all trials. Similarly encouraging results are found for the VL6180X, aside from trial 14. This experiment also highlights the importance of accurate extrinsic calibration; it demonstrates that, given only small inaccuracies in sensor pose, the quality of a reconstructed scene object decreases significantly.

VII. Discussion

This paper presents a geometric relationship between sensor pose, robot motions, and plane pose for a single-pixel distance sensor attached to a robot arm observing a plane. We show how this relationship allows us to perform extrinsic calibration to simultaneously determine the 5D pose of the sensor and

[6] https://github.com/SintefManufacturing/python-urx

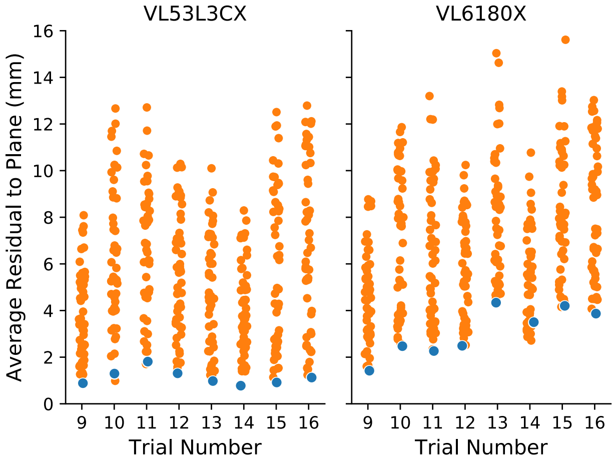

Fig. 9. **Calibrated sensor pose reconstructs a previously unseen plane better than nearly any other sensor pose**: Blue points indicate our calibrated sensor poses, while orange points indicate random perturbations of that pose in the range **p**± 1cm, **u**± 10°. We find that a never-before-seen plane reconstructed using our calibrated sensor pose tends to have a lower residual from the point cloud used to construct it when compared to small perturbations of the same pose. This indicates that our pose is more accurate than those perturbations, and that an inaccurate sensor pose leads to a poor reconstruction of a plane.

3D pose of the plane. We characterize how a practical sensor deviates from the idealized model used in our derivation. We show empirically that our approach is able to combine multiple observations to smooth over noise and precisely and usefully recover sensor pose, both in simulation and the real world.

Limitations– Our sensor characterization experiments identify a key source of structured error when at an oblique angle-of-incidence with the detected plane. While our method is functionally robust to this error to a certain extent, our simulations do not account for it, and additional understanding is needed to fully mitigate its effects. Additionally, further analysis of the effect that the selection of robot motions has on solution precision could allow quicker and more accurate procedures with automatically generated robot motions. Our work also assumes that the sensor is always facing the calibration plane, and is not practical in unknown environments without human supervision; in a less supervised setting, the solver may need to be modified to better handle outliers. Additionally, our work considers only a single sensor; while a set of sensors could be treated individually, combining information across multiple sensors and incorporating constraints on their relative motions could result in higher accuracy of the recovered sensor poses. Lastly, our experiments only consider a sensor mounted on a robot joint which has a full six degrees-of-freedom. Additional constraints may need to be implemented to recover the pose of a sensor mounted elsewhere on the arm, as the motion of the segment alone might not be diverse enough to resolve all ambiguities.

In its current form, our work provides a practical procedure to calibrate a single-pixel distance sensor attached to a robot arm knowing only that the observations measure the same plane. Our work allows location of the sensor and plane without access to precise ground truth. While improvements to the method will improve performance, we already envision using the approach to calibrate sensors on robot arms for use in applications that require precise interpretations of single-pixel distance sensor measurements, such as creating collision maps for planning.

References

[1] E. Cheung and V. Lumelsky, “Development of sensitive skin for a 3d robot arm operating in an uncertain environment,” in *International Conference on Robotics and Automation*, 1989.

[2] J. Llata, E. Sarabia, J. Arce, and J. Oria, “Fuzzy controller for obstacle avoidance in robotic manipulators using ultrasonic sensors,” in *AMC’98 - Coimbra. 1998 5th International Workshop on Advanced Motion Control. Proceedings (Cat. No.98TH8354)*, 1998, pp. 647–652.

[3] G. Buizza Avanzini, N. M. Ceriani, A. M. Zanchettin, P. Rocco, and L. Bascetta, “Safety control of industrial robots based on a distributed distance sensor,” *IEEE Transactions on Control Systems Technology*, vol. 22, no. 6, pp. 2127–2140, 2014.

[4] S. Tsuji and T. Kohama, “Proximity skin sensor using time-of-flight sensor for human collaborative robot,” *IEEE Sensors Journal*, vol. 19, no. 14, pp. 5859–5864, 2019.

[5] P. Chen, W. Li, Y. Jiang, D. Bai, X. Zhu, S. Togo, and H. Yokoi, “Safety control for robotic arm in narrow space based on distance sensor,” in *2018 IEEE International Conference on Intelligence and Safety for Robotics (ISR)*, 2018, pp. 66–70.

[6] O. A. Adamides, A. S. Modur, S. Kumar, and F. Sahin, “A time-of-flight on-robot proximity sensing system to achieve human detection for collaborative robots,” in *IEEE International Conference on Automation Science and Engineering (CASE)*, 2019.

[7] D. Gandhi and E. Cervera, “Sensor covering of a robot arm for collision avoidance,” in *IEEE International Conference on Systems, Man and Cybernetics*, vol. 5, 2003.

[8] U. B. Himmelsbach, T. M. Wendt, N. Hangst, and P. Gawron, “Single pixel time-of-flight sensors for object detection and self-detection in three-sectional single-arm robot manipulators,” in *IEEE International Conference on Robotic Computing (IRC)*, 2019, pp. 250–253.

[9] Z. Zhang, “A flexible new technique for camera calibration,” *IEEE Trans. PAMI*, vol. 22, no. 11, 2000.

[10] J. Domhof, J. F. P. Kooij, and D. M. Gavrila, “An extrinsic calibration tool for radar, camera and lidar,” in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 8107–8113.

[11] T. Kühner and J. Kümmerle, “Extrinsic multi sensor calibration under uncertainties,” in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, 2019, pp. 3921–3927.

[12] S. Miller, A. Teichman, and S. Thrun, “Unsupervised extrinsic calibration of depth sensors in dynamic scenes,” in *International Conference on Intelligent Robots and Systems*, 2013.

[13] Y. Shiu and S. Ahmad, “Calibration of wrist-mounted robotic sensors by solving homogeneous transform equations of the form ax=xb,” *IEEE Transactions on Robotics and Automation*, vol. 5, no. 1, pp. 16–29, 1989.

[14] K. Watanabe, M. Strong, M. West, K. Chaitanya, C. Escobedo, and A. Roncone, “Self-contained kinematic calibration of a novel whole-body artificial skin for collaborative robotics,” in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021.

[15] T. Nguyen and G. Reitmayr, “Calibrating setups with a single-point laser range finder and a camera,” in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2013, pp. 1801–1806.

[16] A. Rochas, “Single photon avalanche diodes in cmos technology,” *PhD thesis, EPFL*, 2003.

[17] R. M. Jans, A. S. Green, and L. J. Koerner, “Characterization of a miniaturized ir depth sensor with a programmable region-of-interest that enables hazard mapping applications,” *IEEE Sensors Journal*, vol. 20, no. 10, pp. 5213–5220, 2020.

[18] D. Kraft, “A software package for sequential quadratic programming,” DLR German Aerospace Center – Institute for Flight Mechanics, Tech. Rep. DFVLR-FB 88-28, 1988.

[19] S. Zaheer, T. Gulrez, and I. Paramabath, “From sensor-space to eigenspace – a novel real-time obstacle avoidance method for mobile robots,” *IETE Journal of Research*, pp. 1–13, 09 2019.

[20] S. Tsuji and T. Kohama, “Proximity and contact sensor for human cooperative robot by combining time-of-flight and self-capacitance sensors,” *IEEE Sensors Journal*, vol. 20, no. 10, pp. 5519–5526, 2020.