The Design and Implementation of a Laser Range-Finder Array for Robotics Applications

by

Will Vega-Brown

Submitted to the Department of Mechanical Engineering in partial fulfillment of the requirements for the degree of

Bachelor of Science in Mechanical Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2011

Author ....
Department of Mechanical Engineering
May 12, 2011

Certified by ....
Sangbae Kim
Assistant Professor
Thesis Supervisor

Accepted by ....
John Lienhard
Undergraduate Officer

The Design and Implementation of a Laser Range-Finder Array for Robotics Applications

by

Will Vega-Brown

Submitted to the Department of Mechanical Engineering
on May 12, 2011, in partial fulfillment of the
requirements for the degree of
Bachelor of Science in Mechanical Engineering

Abstract

We introduce the concept of using a laser range finder array to measure height and tilt for mobile robotics applications. We then present a robust, scalable algorithm for extracting height and tilt measurements from the range finder data. We calibrate the sensors using a precision two-axis system, and evaluate the capabilities of the sensors. Finally, we utilize the sensors and the two-axis system for imaging to illustrate their accuracy.

Thesis Supervisor: Sangbae Kim
Title: Assistant Professor

Acknowledgments

The author would like to thank Professor Sangbae Kim for his help and guidance, as well as Albert Wang for his assistance throughout the thesis process.

Contents

List of Figures

1

Introduction

In many undcractuated robotic systems there is a need for independent measurement of the orientation of some part of the robot relative to the ground – independent, that is, in the sense that the data should come directly from sensors and not from reverse kinematics. Generating data from the configuration of the limbs of a robot becomes futile when those limbs are off the ground. An extreme case is a running robot, which may at some point have all legs off the ground at once. In order to generate new control signals, the robot must have access to reliable sensor data, regardless of the configuration and position of the robot. One such sensor system is an array of laser range-finders, and it is this array system which forms the focus of this paper.

The principle of operation is simple; a set of sensors are arranged such that they all face the ground at slightly different angles. The distance from each sensor to the flat ground is then measured, and relevant information, like height and tilt, is extracted using trigonometric and linear algebraic techniques. In this paper, we resolve two barriers to the construction of such a system. First, we explore the methodology of extracting the desired information from sensor data, and present a robust and scalable algorithm for performing this extraction. Second, we calibrate and evaluate one of the laser sensors; this calibration is an essential step towards getting reliable data out of an array. Finally, to demonstrate the calibration, we use the laser sensor for imaging, illustrating its accuracy and ranging abilities.

2

Theoretical Analysis of Sensor Array Data

The ultimate goal of this research is the creation of a sensor array composed of multiple laser rangefinders. If each rangefinder is pointed downwards in a slightly different direction, then so long as the ground is reasonably flat, the array data can be processed to obtain useful information on the location of the array relative to the ground. Specifically, it will be able to discern the distance normal to the ground – the height of the array – as well as its angle relative to the ground, in the form of pitch and roll angles or in some other equivalent form. By utilizing a variety of techniques, the rangefinder array can quickly, accurately and robustly generate this data, for integration into the control loops of mobile robots.

2.1 Transforming measurements

In the most general case, we may have an arbitrary number of sensors n, each with a location and orientation fixed relative to some reference frame attached to the moving platform. The i^{th} sensor, then, may be parameterized by its position $\vec{p}_i$, its heading $\hat{v}$, and the range it reports, x_i. Our goal, then, is to find the orientation and position of the moving reference frame relative to the surface the lasers see. Solving this problem directly via trigonometry is a difficult and messy process which leads to unwieldy solutions for $n \geq 4$; however, by recasting the problem in the language of linear algebra, we can avoid unnecessary trigonometry and make the problem tractable.

First, note that the inverse problem is comparatively easy; given a known position and orientation, finding the expected range reported by a sensor is simply a matter of applying the appropriate sequence of transformations. Taking a single range measurement closes a vector loop which begins and ends in the plane we are measuring against. That is, it creates a chain of vectors connected head-to-tail, which begin and end in a known plane. Letting $\mathbf{P}$ be the position of the mobile frame origin in the lab frame, and $\mathbf{R}$ be the rotation which takes vectors from the lab frame into the mobile frame, the chain can be expressed as

$$\vec{v} = \mathbf{P} + \mathbf{R}\mathbf{p}_i + \mathbf{R}x_i\hat{\mathbf{h}}_i$$

This vector must be confined to the plane we measure against. Thus, if we choose our lab frame such that $\hat{\mathbf{z}}$ is orthogonal to that plane, we have

$$\vec{v} \cdot \hat{\mathbf{z}} = (\mathbf{P} + \mathbf{R}\mathbf{p}_i + \mathbf{R}x_i\hat{\mathbf{h}}_i) \cdot \hat{\mathbf{z}} = 0$$

This reduces to

$$\mathbf{R}_z^T(\mathbf{p}_i + x_i\hat{\mathbf{h}}_i) = 0 \tag{2.1}$$

where $\mathbf{R}_z^T = \begin{pmatrix} \cos(\theta)\sin(\phi) & \cos(\theta)\cos(\phi) & \sin(\theta) & P_z \end{pmatrix}$ is a row vector corresponding to the z-component of the rotation and translation. Here, P_z is the height of the mobile frame above the plane, and we have chosen θ to be roll and ϕ to be pitch. Note $\mathbf{p}_i$ and $\hat{\mathbf{h}}_i$ are four-vectors:

$$\mathbf{p}_i = \begin{pmatrix} \vec{p}_i \\ 1 \end{pmatrix}$$

$$\hat{\mathbf{h}}_i = \begin{pmatrix} \vec{p}_i \\ 0 \end{pmatrix}$$

If we knew the position and orientation of the mobile frame – that is, if we knew θ, ϕ, z, obtaining x_i would simply be a matter of solving equation 2.1. Given n sensors, then, we have a set of n equations relating the orientation of the mobile frame to the n measurements x_i. This can be represented as a single matrix equation:

$$\mathbf{R}_z^T \left[\begin{pmatrix} p_{1x} & p_{2x} & \cdots & p_{nx} \\ p_{1y} & p_{2y} & \cdots & p_{ny} \\ p_{1z} & p_{2z} & \cdots & p_{nz} \\ 1 & 1 & \cdots & 1 \end{pmatrix} + \begin{pmatrix} h_{1x} & h_{2x} & \cdots & h_{nx} \\ h_{1y} & h_{2y} & \cdots & h_{ny} \\ h_{1z} & h_{2z} & \cdots & h_{nz} \\ 0 & 0 & \cdots & 0 \end{pmatrix} \begin{pmatrix} x_1 & 0 & \cdots & 0 \\ 0 & x_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \cdots & x_n \end{pmatrix} \right] = 0 \tag{2.2}$$

To simplify the rest of the problem, we define a matrix $\mathbf{X}$ as

$$\mathbf{X} = \left[\begin{pmatrix} p_{1x} & p_{2x} & \cdots & p_{nx} \\ p_{1y} & p_{2y} & \cdots & p_{ny} \\ p_{1z} & p_{2z} & \cdots & p_{nz} \\ 1 & 1 & \cdots & 1 \end{pmatrix} + \begin{pmatrix} h_{1x} & h_{2x} & \cdots & h_{nx} \\ h_{1y} & h_{2y} & \cdots & h_{ny} \\ h_{1z} & h_{2z} & \cdots & h_{nz} \\ 0 & 0 & \cdots & 0 \end{pmatrix} \begin{pmatrix} x_1 & 0 & \cdots & 0 \\ 0 & x_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \cdots & x_n \end{pmatrix} \right]^T \tag{2.3}$$

Taking the transpose of equation 2.2 we have a reduced matrix equation,

$$\mathbf{X}\mathbf{R}_z = 0 \tag{2.4}$$

Our goal is to obtain the matrix $\mathbf{R}_z$, in which is encoded our orientation information, from a measured matrix $\mathbf{X}$. Note first that the tractability of the problem is dependent on n. As might be expected, for $n < 3$, the system is underdefined, and there are an infinite number of solutions; in this case, there is insufficient information to determine the orientation of the mobile frame. Additionally, for $n > 3$, the system is *overdefined*; since our measurements are subject to noise, our matrix $\mathbf{X}$ will be imperfect and the system will frequently be impossible to solve exactly. Only for $n = 3$ are we strictly guaranteed a solution; we will begin our analysis there.

In this $n = 3$ case, the problem is reduced to finding the nullspace of the matrix $\mathbf{X}$. While there are many methods of doing this, the one which is indicative here is to first diagonalize the system, then recognize that the eigenvector corresponding to a zero eigenvalue lies in the nullspace of $\mathbf{X}$. To do this, we first square the system by multiplying by $\mathbf{X}^T$, then choose an appropriate method for diagonalization.

Most of these problems are mitigated by using a larger number of sensors. In the presence of error, the system will become unsolvable; however, we can employ projective methods to obtain a nearest solution, in the least-squares sense. We begin by squaring the system and defining a quantity ϵ:

$$\epsilon = \mathbf{X}^T\mathbf{X}\mathbf{R}_z \tag{2.5}$$

We then diagonalize $\mathbf{X}^T\mathbf{X}$:

$$\mathbf{X}^T\mathbf{X} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^{-1} \tag{2.6}$$

Note that since $\mathbf{X}^T\mathbf{X}$ is symmetric, we can express the eigenvectors in an orthonormal form, and thus $\mathbf{V}^{-1} = \mathbf{V}^T$.

The best solution to equation 2.4 is then the matrix $\mathbf{R}_z$ which minimizes $|\epsilon|^2$, the magnitude of ϵ:

$$|\epsilon|^2 = \mathbf{R}_z^T\mathbf{V}\mathbf{\Lambda}^2\mathbf{V}^T\mathbf{R}_z \tag{2.7}$$

By inspection, this will be minimized by choosing $\mathbf{R}_z$ as the eigenvector corresponding to the *smallest* eigenvalue of the matrix $\mathbf{X}^T\mathbf{X}$. The fact that the matrix $\mathbf{X}^T\mathbf{X}$ will in general be nonsingular permits us to use a modification of the power method to compute this minimal eigenvector. The power method utilizes the fact that successively multiplying a vector by a matrix will asymptotically map the vector to the eigenvector of that matrix corresponding to its largest eigenvalue:

$$\mathbf{A}^n\mathbf{x} = \sum_{i=1} d\lambda_i^n\mathbf{v}_i \tag{2.8}$$

Here, λ_i denotes an eigenvalue and $\mathbf{v}_i$ an eigenvector. As $n \to \infty$, the largest eigenvalue will dominate the others and the product will converge on an eigenvector. If the matrix **A** is nonsingular, however, we may perform the same process the inverse of **A**; the eigenvalues of the inverse of a matrix are the inverses of the eigenvalues of the original matrix, and as such the largest eigenvalue of the inverse matrix will correspond to the smallest of the original matrix. Thus, to find our eigenvector $\mathbf{R}_z$ above, we simply choose an arbitrary vector $\mathbf{x}$, and perform the multiplication as many times as necessary to achieve our desired accuracy:

$$\mathbf{R}_z = \left(\left(\mathbf{X}^T\mathbf{X}\right)^{-1}\right)^n \mathbf{x} \tag{2.9}$$

The minimal value of n will depend on the typical magnitude of ϵ, and thus will be a function of the accuracy of the sensors. In practice, however, n is typically small, on the order of 2 to 5. One key advantage of this method is that it does not require access to complex numerical libraries; it can be easily coded in any programming language, making it suitable to be run on a microcontroller of the type one might use to drive the sensor system in a mobile robotics application.

We are left with a single degree of freedom in the solution $\mathbf{R}_z^T$: the length of the calculated eigenvector. To obtain the desired result, we impose a normalization on the first three coordinate values of $\mathbf{R}_z^T$.

To do this, we first divide $\mathbf{R}_z^T$ into two parts such that $\mathbf{R}_z^T = \left(\begin{array}{cc} \mathbf{v}_0^T & \mathbf{v}_1^T \end{array} \right)$:

$$\mathbf{v}_0^T = \left(\begin{array}{ccc} \cos(\theta)\sin(\phi) & \cos(\theta)\cos(\phi) & \sin(\theta) \end{array} \right) \tag{2.10}$$

$$\mathbf{v}_1 = P_z \tag{2.11}$$

Trigonometric identities impose

$$\mathbf{v}_0^T\mathbf{v}_0 = 1 \tag{2.12}$$

From this constraint, we can fully deduce the magnitude of $\mathbf{R}_z^T$. If we additionally require P_z to be positive, we restrict the sign of $\mathbf{R}_z^T$; thus, the eigenvector has been completely constrained and the the desired parameters $\{\theta, \phi, z\}$ can be deduced.

The full algorithm can then be defined as below:

1. Collect data from n sensors.
2. Using a previously calculated set of position and heading vectors, calculate the matrix $\mathbf{X}$ as in equation 2.3.
3. Compute $\left(\mathbf{X}^T\mathbf{X}\right)$ and $\left(\mathbf{X}^T\mathbf{X}\right)^{-1}$.
4. Choose an arbitrary vector $\mathbf{x} \in \mathbb{R}^4$. Compute $\mathbf{R}_z$ as in equation 2.9
5. Normalize $\mathbf{R}_z$ as in equation 2.12
6. Extract $\{\phi, \theta, z\}$ from $\mathbf{R}_z$ via trigonometry.

This method is scalable to an arbitrary number of sensors, requires no numerically cumbersome calculations, imposes minimal calculation error, and results in output that is nearly optimal, in the sense that it provides the best estimate of the desired parameters, given a set of data.

2.2 Error Mitigation

The use of a linear algebraic framework opens up a realm of possibilities for interesting ways of mitigating errors in the sensor array, as well as accelerating computation. For instance, by computing the change in coordinates rather than the coordinates directly, we can rapidly develop an estimate of the current state. If we define a matrix of measurements $\mathbf{A} = \mathbf{X}^T\mathbf{X}$, with $\mathbf{X}$ defined as in equation 2.3, then we may solve the problem once explicitly as in section 2.1, then iteratively for later times. Let the subscript i denote a fully solved initial state $\mathbf{A}_i\mathbf{R}_i = \epsilon\mathbf{R}_i$, and let the subscript f denote a later state for which only the measurement matrix $\mathbf{A}_f$ is known. Thus we have

$$\mathbf{A}_i\mathbf{R}_i = \epsilon_i\mathbf{R}_i \tag{2.13}$$

$$\mathbf{A}_f\mathbf{R}_f = \epsilon_f\mathbf{R}_f \tag{2.14}$$

We denote variations by Δ:

$$\epsilon_f = \epsilon_i + \Delta\epsilon \tag{2.15}$$

$$\mathbf{A}_f = \mathbf{A}_i + \Delta\mathbf{A} \tag{2.16}$$

$$\mathbf{R}_f = \mathbf{R}_i + \Delta\mathbf{R} \tag{2.17}$$

If we assume the variations to be small – that is, we assume the update is being applied after a small time step, or in a slowly varying system – we can neglect second-order terms. Moreover, we assume the variation in ϵ to be negligible; ϵ is a small random variable we expect to vary more or less continuously, and variations in small continuous variables are second order terms. Eliminating the f indexed terms in favor of Δ terms and expanding, we find

$$\mathbf{A}_i\mathbf{R}_i + \Delta\mathbf{A}\mathbf{R}_i + \mathbf{A}_i\,\Delta\mathbf{R} = \epsilon_i\mathbf{R}_i + \epsilon_i\,\Delta\mathbf{R} \tag{2.18}$$

This can be solved for $\Delta\mathbf{R}$ by substitution with equation 2.13:

$$\Delta\mathbf{R} = (\epsilon\mathbf{I} - \mathbf{A}_i)^{-1}\,\Delta\mathbf{A}\mathbf{R}_i \tag{2.19}$$

The result provides an accurate estimate of the final coordinate solution $\mathbf{R}$ given the new measurements. This solution can be computed significantly faster than can the eigensolution, and using it as an initial guess could potentially reduce the number of iterations required by the initial algorithm. This speed has an additional advantage. Assume one of the laser beams encounters an obstruction. The resulting data will result in output parameters that dramatically deviate from true values, with no way of knowing *a priori* which beam is giving incorrect data. The only sign of an error will be the large value ϵ generated in the algorithm. However, if the computation can be done quickly enough, an exhaustive search becomes feasible; the algorithm cycles through every combination of beams, starting with all n beams, then all n combinations of $n-1$ beams, and proceeding until ϵ drops below a certain threshold.

1. Use equation 2.19 to estimate the new coordinates $\mathbf{R}_f$ from the new data.

2. Take the product $\mathbf{R}_f^T \mathbf{A}_f \mathbf{R}_f = \epsilon \mathbf{R}_f^T \mathbf{R}_f$. If ϵ is small – with small defined by experiment – then $\mathbf{R}_f$ is likely very close to the true eigenvector. If ϵ is large, it is likely an obstruction has occurred.

3. If ϵ is large, repeat the first step n times, where n is the number of sensors; each time, remove a single sensor from the data. When the result gives a small ϵ, then the obstructed beam has likely been located.

4. Once the beams generating invalid data are obtained, the full algorithm detailed in section 2.1 may be employed.

Obviously, this will only work if three sensors are providing unobstructed data. This suggests a second benefit to including redundant sensors; in addition to reducing the error constant ϵ, they will provide additional redundancy in the case of an uneven floor or the presence of obstacles.

3

Calibration and Evaluation of Sensors

The second phase of our project involved the design and fabrication of a two-axis unit to be used to calibrate and evaluate the laser sensors. Once constructed, the unit was used to measure the variation in sensor output current as the beam was scanned over a wall, allowing us to calibrate the unit for a variety of ranges and angles of incidence. The obtained calibration curve was then used to generate a depth-mapped image of the lab; the image provided fair resolution, and some feature extraction was possible.

3.1 Design of Testing Unit

For the purposes of calibrating and evaluating the laser sensors, we create a two-axis rotating platform, driven by precision servos and laser cut for accurate geometry.

The unit was constructed out of laser-cut acrylic. The first design iteration was meant to be assembled, then clamped and glued together with an adhesive (Figure 3-1). This methodology proved unworkable, however; the need for disassembly to remove the motors and laser sensors made permanent adhesion impractical. Moreover, design flaws in this early iteration became apparent. There was insufficient mechanical support for the pitch axis, which was largely supported by a cantilever action on the motor gear; this resulted in sagging, and made accurate measurement of the position of the laser sensor difficult.

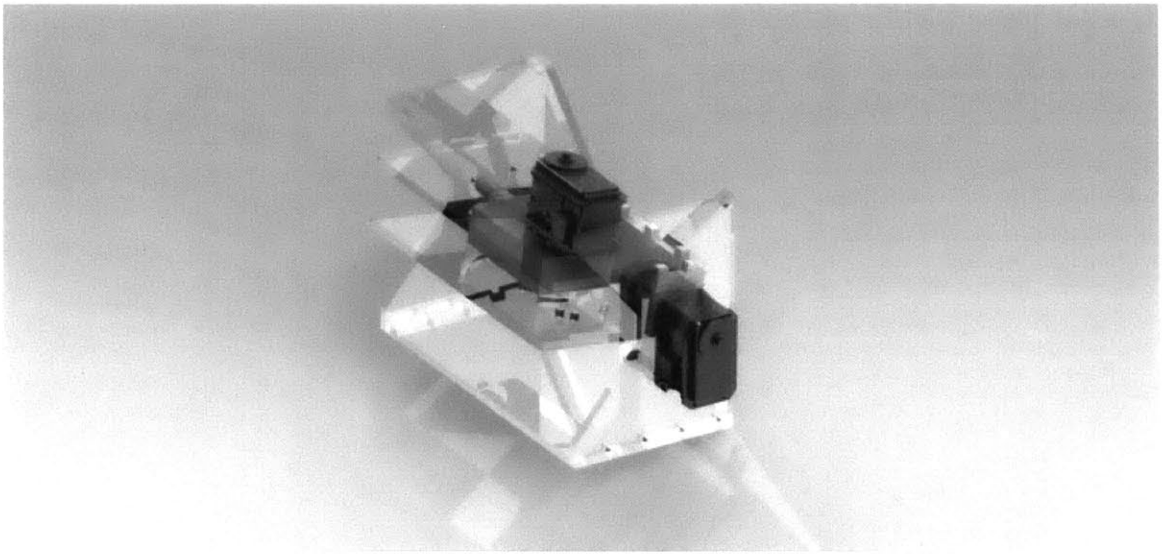


Figure 3-1: First design iteration

A second iteration was built (Figure 3-2). This iteration supported the pitch axis with a bearing, and reinforced the yaw assembly with additional material. Rather than glue, this variant employed laser-cut acrylic struts, meant to mate with holes cut in the main body. This design offered two distinct advantages; first, the snap-fit action of the struts held the structure together without the benefit of adhesives, facilitating assembly and potential disassembly, and second, the struts offered the structure increased rigidity by reinforcing each corner.

The strut-supported assembly did prove troublesome to assemble; because of inconsistencies in the laser kerf, it was difficult to cut mating holes for the struts that interfered enough to provide a good grip, yet not so much as to prohibit assembly. Ideally, multiple sample cuts would have been made and the best size chosen experimentally; in practice, we simply made an initial estimate and found it to be less than ideal, yet satisfactory. Although several cracks formed in the acrylic during assembly, none proved structurally fatal.

To drive the unit, we chose two Dynamixel EX-106+ servos. These were chosen primarily because of their high angular positioning resolution; manufacturer specifications claim a angular precision of 0.06°, and preliminary evaluation suggested that

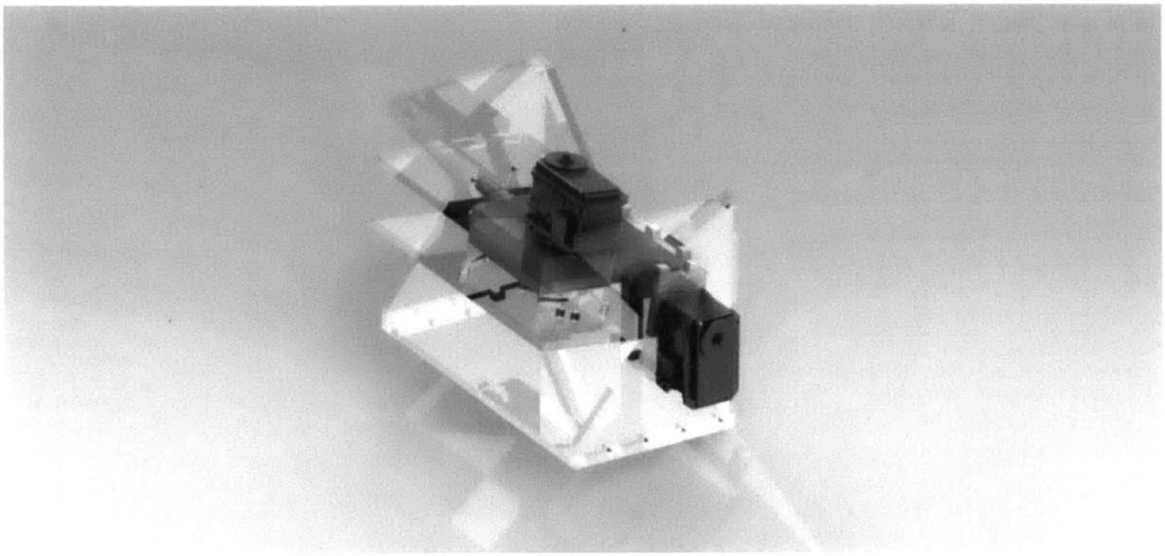


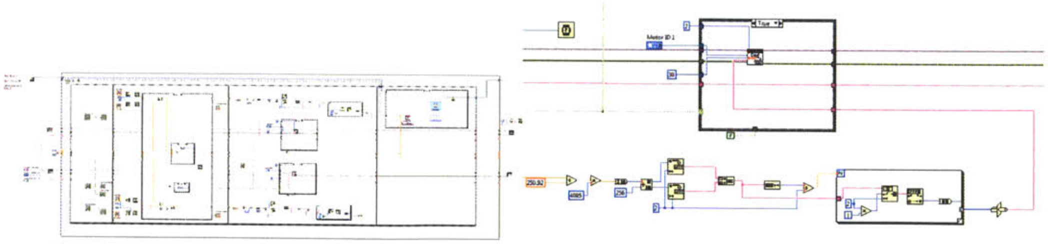
Figure 3-2: Second design iteration

precision could be attained quickly and repeatably, both essential in a precision measurement unit. The servos had additional advantages, as well; they interface nicely with Labview, they can be daisy chained to simplify wiring and control architecture, and they offer high-torque for their size, guaranteeing enough power to move the sensor assembly quickly.

The software implementation was done entirely in National Instruments Labview; the motor drivers, laser data collectors, and high-level control were each written separately, then merged into one larger program (Figure 3-3a). The only complication of note came in the implementation of the motor drivers; the motor accepts commands in hex-code literals, that is, actual four-bit sequences corresponding to the various codes. Outputting such codes is a nontrivial task in Labview, which by default outputs hex code in ASCII characters; the resolution of this issue required a relatively complex subroutine (Figure 3-3b).

3.2 Calibration

To evaluate the accuracy and precision of the Micro-Epsilon range finders, we employed the pan-tilt unit described above. The unit was fixed at a known distance from



(a) Complete Labview block diagram

(b) Labview subroutine for generating hexadecimal output

Figure 3-3: Labview code used to control scans

a flat plane, and measurements were taken at one degree intervals over a wide variety of angles. The resulting data was fitted to a known form using Matlab's optimization tools.

This form was inferred from predictions relating to the geometry of the unit and three measured parameters: the yaw angle θ, the pitch angle ϕ, and the range finder measurement i, measured in mA. The geometry of the unit was obtained from the solid model and verified with direct measurement; the position of the laser point in Cartesian coordinate space can then be obtained via a series of rotations and translations of axes.

We first suppose a linear relationship between the measured current i and the experimental range r; that is, $r = c_1 i + c_2$. If $\vec{v}_1$ is a vector between the center of yaw rotation and the location of the laser emitter, and $\vec{h}$ is the heading vector of the laser in the rotating yaw frame, then the location of the laser point in the yaw frame is given by $\vec{r}_{yaw} = \vec{v}_1 + r\vec{h}$. Rotating the frame by the yaw angle θ and translating by another vector $\vec{v}_2$ gives the laser point location in the rotating pitch frame; a similar procedure gives the laser point location in the lab frame.

The complete transformation is then given by

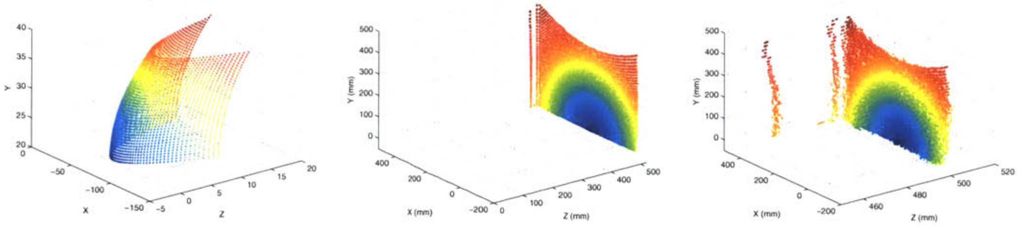
$$\vec{r} = \vec{v}_3 + \mathbf{R}_z(\phi)\left(\vec{v}_2 + \mathbf{R}_y(\theta)(\vec{v}_1 + r\vec{h})\right)$$

By measuring i for a series of angles θ, ϕ, we can obtain the parameters c_1, c_2 which most nearly project the data into the plane; these parameters can in turn be used to interpret the measurements taken by the sensors in other contexts. For instance, the same calibration can be applied to the multi-sensor array, or for use in imaging with the pan-tilt unit.

The unfitted data is shown in Figure 3-4a. The fitted projection is shown in Figure 3-4b. The calibration constants obtained give the relation

$$r = 242.15\frac{\text{mm}}{\text{mA}}i - 718\text{mm} \tag{3.1}$$

The standard deviation of the fitted normal distance is only 4.18mm over a distance of 498mm, or approximately 0.8%. This range is characteristic of the likely ranges sensors would report in an implementation of the array described previously, and the accuracy of the sensor is promising for applications in such a system.



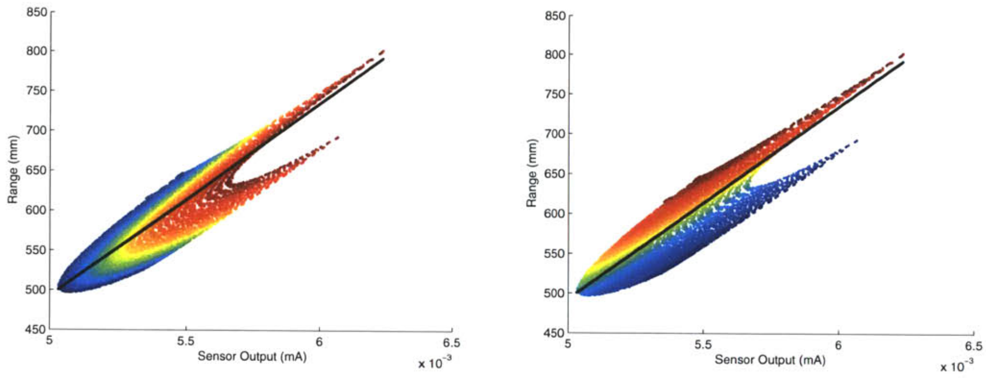
(a) Unfitted laser sensor current (b) Fitted laser sensor output (c) Fitted laser sensor output, scaled

Figure 3-4: Laser sensor ranging a plane. The data is presented as a three dimensional point cloud, with color denoting angle of incidence with the plane

Moreover, the sensor data proved highly repeatable. Repeated measurements at identical angles produced normally distributed data with a standard deviation of just 1.1mm. This suggests that much of the uncertainty in the distribution of points stems from angular dependence; in principle, if we could develop a functional relationship between the angle and the range distortion, we could employ perturbative methods to generate a more accurate measurement of the angle in the sensor array.

Plotting the measured current against the actual range (as obtained via trigonometry) reveals that there is almost no dependence of the error on θ, the angle of rotation about the long dimension of the sensor (Figure 3-5a). However, there is a marked dependence of error on ϕ, the angle of rotation about the short dimension; at large absolute values of ϕ, the measured range drifts further and further from its actual value (Figure 3-5b). Unfortunately, attempts to quantify this disturbance were largely unsuccessful; adding first and second order perturbative terms to the function for r were only capable of reducing its standard deviation by a few microns. The new optimal calibration was found to be

$$r = 241.9\frac{\text{mm}}{\text{mA}}i - 716\text{mm} - 0.188\phi + 1.29\phi^2 \tag{3.2}$$



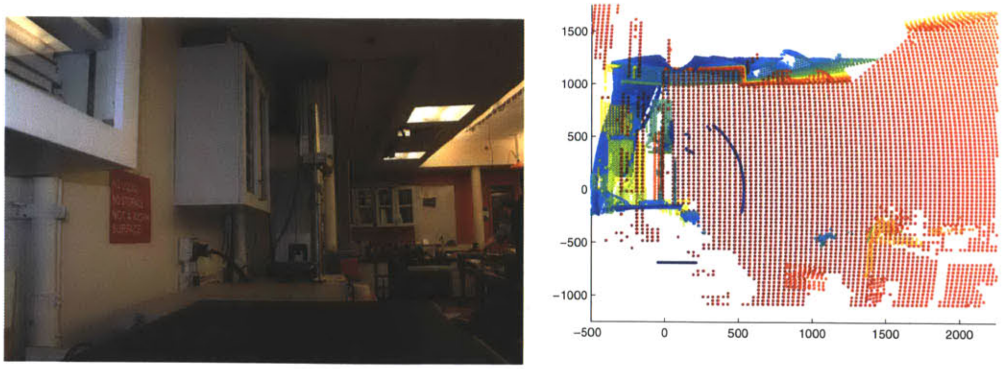
(a) Color indicates θ

(b) Color indicates ϕ

Figure 3-5: Range versus measured current. The solid line is the calibration curve.

The new standard deviation of the fitted normal distance was measured as 4.15mm; this improvement is likely too small to be significant, and no gain is likely to be had by attempting to account for angular effects.

One may question whether some of the error observed in the experiment is endemic to the machine itself, rather than the laser array. While this is certainly possible, it is unlikely; in addition to the manufacturer's claims of accuracy, two simple tests suggest the motion of the machine to be very accurate. First, the machine was able to repeatedly hit a mark on a target wall with the laser beam, even after being taken through its entire range of motion, and even when the wall was a substantial distance – over twenty feet – away. Second, careful measurements of the motion of the laser beam projected on the wall agree to within measurement error with the commands



(a) Photo of imaged area

(b) Depth plot of imaged area

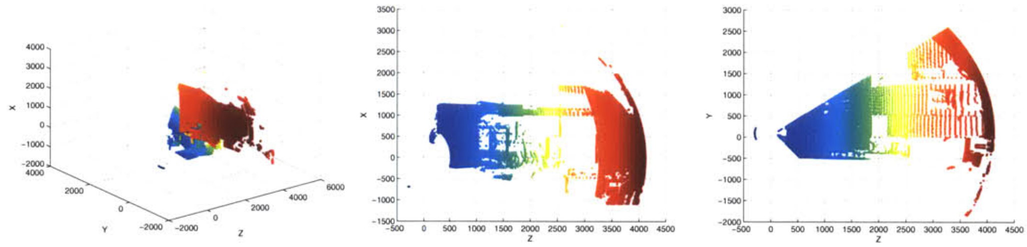
Figure 3-6: Depth mapping of an area of the lab. Color denotes distance from camera.

given. That is to say, when the laser was told to move, for instance, one degree, the observed motion corresponded almost exactly to that predicted by trigonometry. Given this demonstrated accuracy and precision, it seems reasonable to ascribe any error to the laser unit itself, and not the machine.

3.3 Imaging

As an application of the calibrated laser sensor, we use the pan-tilt unit for imaging. This reveals several attributes of the sensor data.

Features can be clearly differentiated at many ranges (on the order of 300mm to 4000mm) in the image in Figure (3-6). For instance, the cupboard and pipes on the left of the photo are visible, as are the drop ceilings and and table on the far right. Accuracy seriously breaks down at longer ranges, however; the red background of the image fails to track the planar shape of the far wall, and is instead roughly dome-shaped, as can be seen in Figure (3-7). This may be due to the exhaustion of the sensor unit's range; the true distance was very nearly the limit of the sensor's advertised 8m range.



(a) Isometric view

(b) Top view

(c) Side view

Figure 3-7: 3D point cloud of imaged

4

Conclusion

Both the experimental and theoretical work done here can serve as useful background and groundwork for the construction of a laser array system. The algorithms presented in the first part of this paper are efficient and in some sense optimal; they extract very nearly as much information as can be extracted from the range-finder data, and perhaps equally important, they can be implemented easily in low-level languages. The most computationally complex step is the inversion of a four-by-four matrix. Moreover, the theoretical framework is scalable and robust, two useful features in a system of this type; to generate more accurate data, we can simply add more sensors, without doing a significant amount of additional work.

The experimental work offers two key results. First, it generated a calibration curve for the sensors, which was found to be very nearly identical in each of several sensors. This curve will allow for more accurate measurement when the sensors are deployed; without it, we would be forced to estimate a calibration from the manufacturer's data sheets, and given how far off our initial guesses were before calibration, this would certainly result in greatly diminished accuracy. Second, the experiments allowed us to get a gauge on how accurate the sensors are. For instance, we measured experimentally the standard deviation of ranges reported, and found it to be quite small. We also were able to measure things like deviations from linearity in the sensors; results indicated that all higher-order terms and all angular dependence are negligible in the regions of interest. This greatly simplifies the later design of the sensor arrays, and suggests such a system can be highly accurate.

There are several concerns not addressed in this paper. One key design parameter for the array system will be its throughput. The target application seeks to operate at a command frequency of roughly 100Hz; the array will only be useful if it can operate at similar speeds. While the sensors themselves have the bandwidth to operate at such speeds, the rest of the system needs to be able to keep up. The National Instruments ADC we used for experiments here – chosen for its compatibility with Labview – would be far too slow at its 'high resolution' setting, and even its 'high speed' setting may not be fast enough. We did an initial investigation of other ADC chips; while high-resolution, high-bandwidth chips do exist, they tend to be expensive and frequently introduce a large time delay. While this is fine for calibration and measurement, it could be extremely detrimental if it occurs inside a command loop.

Finally, it remains to be studied precisely how to best integrate such an array into a command loop. While the method outlined in Section (2.2) does utilize the redundancy of the array, the inclusion of other sensors, such as gyroscopes or tilt sensors, could augment this method to offer even more redundancy. There exist various ways of measuring tilt and height, each with differing advantages; some have better noise characteristics, some offer faster responses, and some are more consistent. An ideal system would incorporate several types of sensors and synthesize their output to generate superior results. The results of this research clearly indicate, however, that a laser range finder array can and should form an integral part of any such ensemble.