6-14-2012

Cross Hallway Detection and Indoor Localization Using Flash Laser Detection and Ranging

Istvan M. Prileszky

CROSS HALLWAY DETECTION AND INDOOR LOCALIZATION USING FLASH LASER DETECTION AND RANGING

THESIS

István Prileszky, Second Lieutenant, USAF

AFIT/GE/ENG/12-34

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT/GE/ENG/12-34

CROSS HALLWAY DETECTION AND INDOOR LOCALIZATION USING FLASH LASER DETECTION AND RANGING

THESIS

Presented to the Faculty

Department of Electrical and Computer Engineering

Graduate School of Engineering and Management

Air Force Insitute of Technology

Air University

Air Education and Training Command

in Partial Fulfillment of the Requirements for the

Degree of Master of Science in Electrical Engineering

István Prileszky, B.S.E.E.

Second Lieutenant, USAF

June 2012

AFIT/GE/ENG/12-34

CROSS HALLWAY DETECTION AND INDOOR LOCALIZATION USING FLASH LASER DETECTION AND RANGING

István Prileszky, B.S.E.E.
Second Lieutenant, USAF

Approved:

Lt Col Michael Stepaniak (Chairman)	Date
Maj Ken Fisher (Member)	Date
Dr. Meir Pachter (Member)	Date

Abstract

A flash LADAR is investigated as a source of navigation information to support cross-hallway detection and relative localization. To accomplish this, a dynamic, flexible simulation was developed that simulated the LADAR and the noise of a LADAR system. Using simulated LADAR data, algorithms were developed that were shown to be effective at detecting cross hallways in simulated ideal environments and in simulated environments with noise. Relative position was determined in the same situations. A SwissRanger SR4000 flash LADAR was then used to collect real data and to verify algorithm performance in real environments. Hallway detection was shown to be possible in all real data sets, and the relative position-finding algorithm was shown to be accurate when compared to the absolute accuracy of the LADAR. Thus, flash LADAR is concluded to be an effective source for indoor navigation information.

To my parents, friends, and mentors

Table of Contents

List of Figures

List of Tables

Cross Hallway Detection and Indoor Localization Using Flash Laser Detection and Ranging

1 Introduction

INDOOR navigation has always presented a unique problem. Many applications for indoor navigation exist, to include unmanned vehicle control, rescue operations, fire fighting, or certain combat operations. A large number of navigation techniques utilize some form of satellite or outside navigation, like the global positioning system (GPS). Indoors, however, GPS is often either unreliable or completely unavailable. Also, extremely accurate systems are often too large or power consumptive to use in these environments. Indoor navigation has thus relied upon methods such as inertial navigation or vision-aided navigation. However, the introduction of new technologies often provides new methods for this navigation to occur.

Recent development of new sensor technology has pushed several new devices into a position where their use is now practical in real-world situations. Namely, flash laser detection and ranging (LADAR) systems, a successor of radar, have been improved to the point where their use in navigation has become a possibility. These systems provide range data with very good accuracy using lasers. They work on the principle of time of flight in conceptually the same manner as radar. However, these systems are far more suited for use indoors than radar or other ranging systems.

Ranging systems in the past, like interferometry or line-scanning LADAR have a varied history of utilization. However, flash LADAR provides several advantages over these systems that make them far more useful for indoor navigation. They update faster and have fewer of the motion-induced errors that these systems have. Their lack of

moving parts makes them much more reliable in situations with high dynamics or movement. They are smaller than many alternatives.

The problem of indoor navigation is extremely complex. This research focuses on using a flash LADAR camera, the MesaImaging SR4000, to detect potential passageways in indoor environments. This information can be extremely useful if combined with navigation solution data. These cross hallways represent potential pathways through buildings where *a priori* information of the layout is lacking or outdated. Thus, knowing where the cross hallways are is useful for navigating through these areas when human interaction is minimal. Such detection can be used in localization and mapping procedures. Also, knowing vehicle position relative to the hallway is useful. Measuring this position can provide information to other navigation systems that allow for control input development or reduction of error.

This document is organized into several chapters. Chapter 2 provides background information that laid the groundwork necessary for this research. Chapter 3 describes the methods used and the algorithms implemented to complete this task. Chapter 4 discusses the outputs and results of these algorithms and experiments. Chapter 5 gives conclusions and suggestions for future research.

2 Background Information

THIS chapter introduces the motivation for using flash LADAR for navigation. Also, basic information about ranging methods and Flash LADAR is provided. The systems and current research are introduced.

2.1 Motivation for the Use of Flash LADAR

Flash LADAR presents one possible solution to the technical problem of navigating indoors. Soldiers, rescue teams, and unmanned vehicles of all form factors are often called upon or designed to navigate indoors to complete tasks. Conventional methods of navigation often rely on Global Positioning System (GPS) signals to navigate with reasonable accuracy. However, indoor environments often preclude the use of GPS due to the obstruction of signals by blockage and multipath interference. Thus, it is commonly necessary to augment or replace traditional navigation methods such as GPS with other, often novel schemes. Flash LADAR can provide certain benefits to navigation in indoor environments. The National Institute of Standards and Technology has expressed interest in using LADAR as a sensor in automated indoor tasks, suggesting they could have a major impact on navigation indoors [26].

2.2 Navigation on Small Vehicles

Many solutions exist for the navigation of vehicles through three-dimensional space. Many vehicles that require navigation use a combination of GPS and inertial navigation systems (INS) consisting of gyroscopes and accelerometers. These navigation systems are capable enough to navigate through indoor environments by way of “dead reckoning.” However, the systems that utilize these sensors are often limited by cost, size, or power constraints to low-accuracy sensors. Inaccurate micro-electromechanical systems (MEMS) that are much less accurate are often used on small vehicles. These systems

experience large errors that grow over time, quickly degrading the performance of the navigation. GPS is a common tool for mitigating the drift of these sensors. It is useful to aid these systems with additional measurements from systems with similarly small form factors and low costs, especially in situations, like indoor navigation, that preclude the use of satellite navigation systems.

2.3 Inertial Navigation System Aiding

As aforementioned, MEMS inertial navigation systems can be used to navigate small vehicles, but these are subject to large amounts of drift over time. Adding information by monitoring the environment through computer vision is one way to mitigate and constrain this drift. LADAR has been used for the navigation problem in the past. For instance, line-scanning LADAR has been successfully used to help constrain the drift of INS outputs [4].

Flash LADAR is a relatively new imaging system that provides unique advantages over other forms of computer vision, such as scanning LADAR. Flash LADAR is a fast and relatively accurate means of obtaining range data from a scene. Combining the Ladar-based computer vision solution with the inertial system could allow for successful indoor navigation where GPS is not available. Flash LADAR has been used for INS aiding as well in preliminary research that used a SwissRanger SR3000 in order to constrain INS drift. This research was completed at less than the ambiguous range of the camera, but demonstrates the potential use of flash LADAR in this capacity [15].

2.4 Introduction to Light Ranging Methods

There are many methods for determining the range of an object in a 3D scene using radiation in the light spectrum, with wavelengths of 0.5-1 μm. These include interferometry, triangulation, and time-of-flight (TOF). Interferometry and triangulation

have their own benefits, but as this research focuses on the use of a TOF camera, it will be described in more detail. All flash LADARs are limited to this approach.

2.4.1 Triangulation. The basis of triangulation is stereo vision. It is used by many systems, including many organic systems. Human depth perception works on this basic concept. The process relies on measuring angles to an object or target in order to use geometry to determine the range to the object. When the measuring system has many known features, these angles can be found or computed. The process can be done passively with two cameras, or actively with one camera and a light source.

Passive triangulation is the method of triangulation that human vision relies upon for depth perception. The object whose range is unknown at point C is viewed from two points A and B, the distance between which is well known. The angles to the point C with respect to the base of the triangle AB are then measured as α and β, as seen in Figure 2.1.

This method relies on several things. First, the image in question must have high contrast. This is due to the fact that in computer vision, the point must be identified using a correlation. If the contrast is too low, the correlation may identify the wrong point. Also, in order to do fast rangefinding, the computer must be able to perform the correlation quickly, a feat that is difficult on mobile systems. Stereovision is also limited to certain surfaces, and does not handle irregular surfaces well [17].

Active triangulation relies on the lasing or illumination of a target. Usually accomplished with a laser, this technique relies on similar triangles to find the range to an object. The object to be ranged is illuminated, and the reflection is passed through a lens before a detector. If the detector is well-characterized, the angle to the object can be measured, as shown in Figure 2.2. The range and range resolution to the object are functions of the sensor geometry and the range to the target.

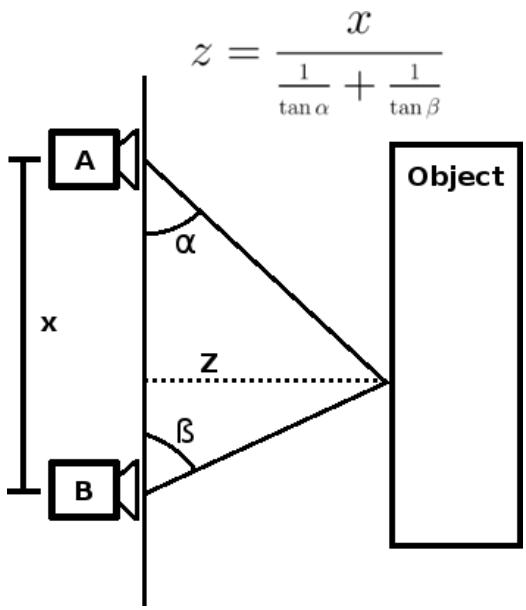


Figure 2.1: Basic Principle of Passive Triangulation

The range resolution equation makes very clear the primary problem with this method of rangefinding. In order to achieve a reasonable ranging resolution, there are two factors that must be considered. First, there must be a good local detector resolution $\delta x'$. Also, the actual detector must be larger for increased resolution, as $\delta x'$ can be defined as a function of x. The size of sensor A is directly correlated with the accuracy of the system. For mobile solutions, the large size of active rangefinding systems can be prohibitive [17].

2.4.2 Interferometry. Interferometry works by superimposing two monochromatic (identical frequency) waves of light. This is a technique commonly used in navigation, but not as a computer vision method. Instead, this technique is often used for ring-laser gyroscopes. Also, larger systems have been placed on satellites and used to provide high-accuracy depth information. Interferometry begins with two waves that are modulated at the same frequency, f. Because they are monochromatic, they also have identical wavelength λ. They are usually generated by creating one laser pulse that is

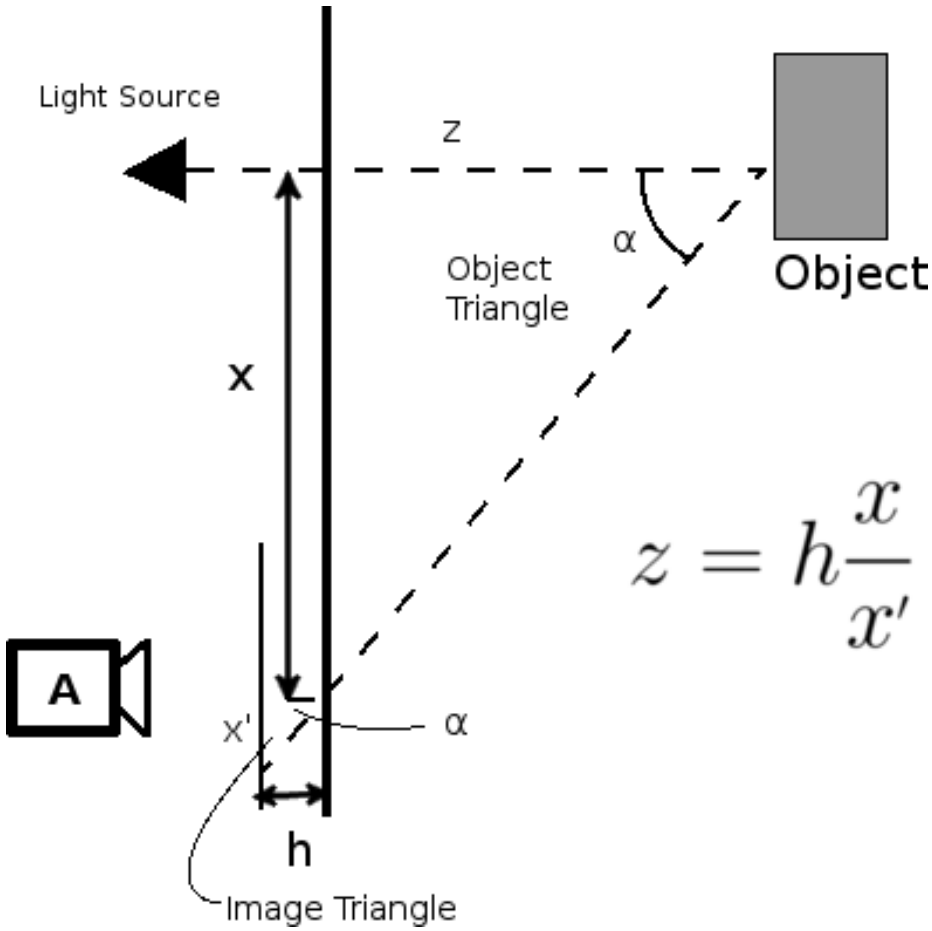


Figure 2.2: Basic Principle of Active Triangulation

passed through a beam splitter. As shown in Figure 2.3, one of the split beams is passed to a mirror ninety degrees from the target. The distance to this mirror is known very precisely. The other beam is passed to the object in question, at an unknown distance. Both beams, the reference beam and the measurement beam, are passed back to the beam splitter and picked up by an integrating detector.

The result is a light intensity that depends on the distance to the target. If the difference between the reference distance and the target distance is half a wavelength away, the intensity at the detector is maximal due to constructive interference. At a quarter wavelength difference, the intensity is minimal due to destructive interference [17].

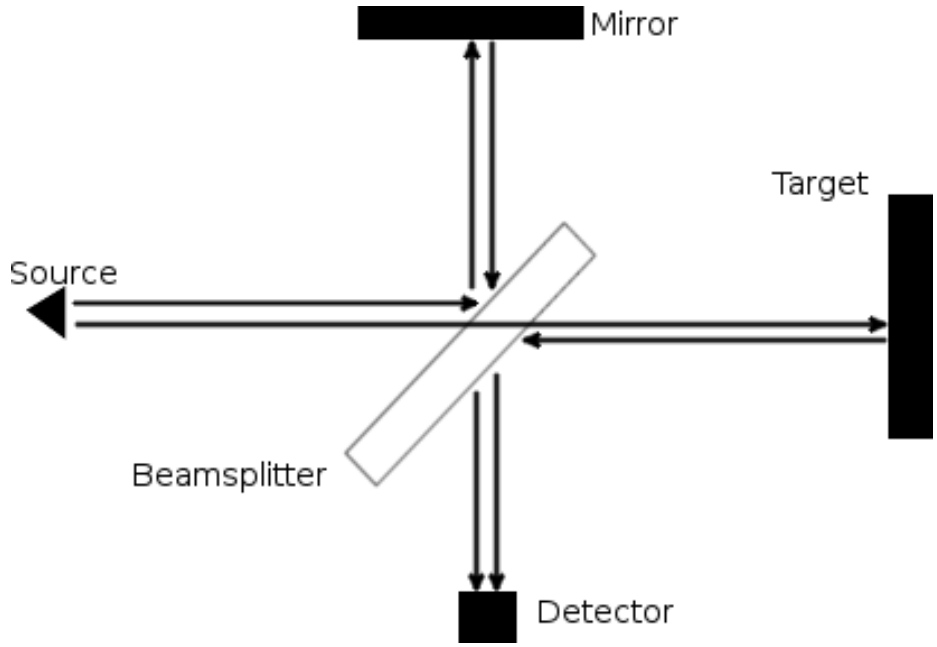


Figure 2.3: Basic Michelson Interferometer

Interferometry is not suitable for navigation outside of its use in gyroscopes and space-based surveying solutions for several reasons. They offer impeccable accuracy of $\lambda/100$ or even $\lambda/1000$, but this comes at a considerable cost. Because the distance measurements are reliant on phase difference, the non-ambiguous measurement distance is only half of a wavelength of the light signal used. Thus, at any distance beyond this, measurements will begin to appear as closer than they actually are. While this behavior is identical to that time-of-flight systems, the wavelengths used on small systems are prohibitively short for navigation. The wavelengths on large systems are much longer, but rely on a large baseline for accuracy [10]. Also, this type of system is limited by its space requirements. The space-based systems used for remote sensing, while accurate, are both very large and very susceptible to miscalibration in the presence of dynamics beyond that of space. Though other form factors of interferometers exist, none are easily expansible to measure a useful amount of a scene. Interferometers are also very susceptible to

calibration problems after movement. Thus, they are better suited for laboratory applications requiring high accuracy.

2.4.3 Time of Flight. Time of flight works by measuring the absolute distance between a target and a detector. In the case of LADAR, this is indirectly measured by recording the time it takes for a pulse of light to reach a target in the scene and return. At its most basic, the camera emits a pulse. This pulse triggers the beginning of a timer. The light of this pulse travels to some target object and is returned to the camera, which triggers the measurement of the elapsed time on the stopwatch. Because the speed of light is well known, the amount of time the pulse takes to leave and return corresponds to a fixed distance from the camera. In this case, the measurement occurs in one dimension. In order to obtain three-dimensional range data, the illumination source must send out pulses in two axes. To obtain a high-accuracy measurement, the timer must also be highly accurate [17].

The vast majority of LADAR sensors function by first emitting an intensity modulated light signal to illuminate the scene. In some cases, this can be with a single, collimated laser. However, in the case of flash LADAR, this is done with light-emitting diodes (LEDs) that function similarly to a camera flash. This light is reflected from a target object and a portion of it, dependent on reflectivity and distance of the target, is returned to the detector. The time from the emission of the light to the reception of the beam is measured. This indirect measurement is then converted to a distance, which represents the distance to a part of the 3D scene. In the case of flash LADAR, this data can also be turned into three-space coordinates for each point, which will be discussed in a later section. There also exists an optical shutter approach to time of flight measurements, but as very little public research has been done into this method, it will be largely ignored [16].

The signal may be modulated in one of two ways, pulsed-wave or continuous-wave modulation. The first, more intuitive method of modulation is the pulsed-wave modulation technique. This utilizes a pulse of light that is emitted and then received. This signal is then correlated with an internal signal that acts as a timer. The result of the correlation provides the trip time of the signal. This offers the advantage of high power over a short time, but requires a highly dynamic light source that can provide pulses at a high rate. The alternative, continuous-wave modulation, does not require the distinct pulse generation that the pulsed-waved cameras do. This sort of camera functions by measuring the phase difference between the transmitted and received signals. This approach allows for simultaneous measurement of range for all the pixels in an image, and is the method used by the SR4000 camera used in this research and further described in Section 2.5.2.

The frequency modulation approach to TOF measurements uses modulated light, often in the infrared or near-infrared spectrums. The illumination source modulates the light at a given radian frequency, ω. The output signal is both transmitted to the scene and reflected back to the sensor in order to create a reference pulse with a phase offset $g(t + \tau)$, where τ is the phase offset introduced to the reference signal. This concept is shown in Figure 2.4. The incident signal $s(t)$, which is returned from the target object is detected by the sensor and correlated with the reference signal, usually on the imaging chip. The signal g is defined by

$$g(t) = \cos(\omega t) \tag{2.1}$$

The signal s is similarly defined as

$$s(t) = b + a\cos(\omega t + \phi) \tag{2.2}$$

where b is the correlation bias introduced by the correlator and a is the amplitude of the reflected signal at the detector. ϕ is the phase offset that results from the trip to and from the target object. This phase offset corresponds to the distance of the object. The

correlation of these two signals, $c(\tau)$ is given by

$$c(\tau) = s \otimes g = \int_{-\frac{T}{2}}^{\frac{T}{2}} s(t) \cdot g(t+\tau)dt \tag{2.3}$$

The correlation, which is simplified with trigonometric calculus, is then found using four phase images offset by a quarter wave, i.e.:

$$c(\tau) = \frac{a}{2}cos(\omega\tau + \phi) + b, \quad A_i = c(i\frac{\pi}{2}), \quad i = 1, \ldots, 3 \tag{2.4}$$

.

$$a = \frac{\sqrt{(A_3 - A_1)^2 + (A_0 - A_2)^2}}{2} \tag{2.5}$$

$$\phi = \arctan\left(\frac{A_3 - A_1}{A_0 - A_2}\right), \quad I = \frac{A_0 + A_1 + A_2 + A_3}{4} \tag{2.6}$$

where the resultant a is the amplitude factor shown in Equation 2.4, and I is the calculated approximate returned intensity of the light. Once ϕ is found, because the speed of light is known to be $c = 2.99 \times 10^8 \ \frac{m}{s}$. It is a simple calculation to find the distance of the object away, using the equation

$$d = \frac{c\phi}{4\pi\omega} \tag{2.7}$$

where each variable is the same as above.[16]

2.4.4 Time of Flight Errors. Time of flight systems are prone to several errors that are difficult to mitigate that result in range measurement noise. Much of the error can be modeled as a function of several image and target parameters. Besides this error, there are first errors that are a result of the imperfections of the system in hand, including systematic distance error. These errors can often be mitigated with better equipment or application of the technology in areas where these errors are not exacerbated. There are also unmitigable errors that are inherent in the method of image capture, such as multipath.

The basic error model for a time-of-flight camera can be described as a function of wavelength, range, reflectivity of the target, and angle of incidence. This model can be

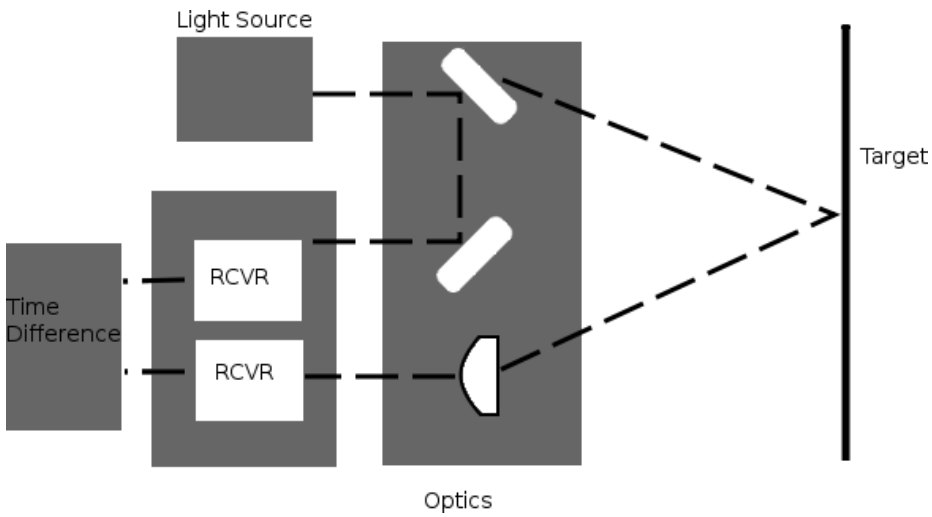


Figure 2.4: Basic Time of Flight Camera Principle

expressed as

$$\sigma_r \propto \frac{\lambda R^2}{\rho \cos \alpha} \tag{2.8}$$

where R is the range to the target, λ is the wavelength of the light used, ρ is the reflectivity, α is the angle of incidence, and σ_r is the standard deviation of the range error [1]. Range error increases with the square of the range, and the error standard deviation grows with range. The fit lines in the figure use the model $\sigma = kR^2$, where k is a constant determined from experimental data. Angle of incidence is another error inherent in flash LADAR data. This error increases as the angle of incidence increases. The error standard deviation grows with angle of incidence, correlated with the secant of the angle of incidence. This data is fit to the model $\sigma = k \sec \phi$. A final error is introduced by varying intensity at the detector due to varying reflectivities of the target material. However, this can sometimes be tied to physical limitations of certain semiconductors and camera circuits [16]. Reflectivity of the target has on the range error for a time-of-flight camera, an effect both shown in research for both a MesaImaging SwissRanger 2 and a CanestaVision flash

LADAR. Range error for a targets with different reflectivity show that varying reflectivity varies the standard deviation of the error [1].

In addition to the basic error, the limitations of physical systems result in some errors. First is the systematic distance error, also called “wiggling.” This occurs because the signal measured is assumed to be a perfect sinusoid. However, physical systems cannot create a perfect sinusoidal wave, so error is introduced in this way. This error is inherent in this method of modulating the signals, but the error can be mitigated with better hardware to the point of the error becoming insignificant. However, on mobile systems, this is impractical. Another error is introduced because there are multiple detectors operating near to each other. When in close proximity, the measurements of each detector suffer from interference. Also, because the sensor relies on light, noise from outside sources is also possible.

The errors introduced when multiple nearby pixels are recorded is a significant source of error when feature extraction is a goal. Because the detection chips are so closely located, the return signals from surfaces interfere with each other. The radiation that returns from the scene interferes with itself as it returns to the sensor. The result is measurable range error. This error occurs because coherent light like that of a flash LADAR is susceptible to interference [17]. This effect is also noticed when multiple cameras that use the same frequency are utilized in near proximity [16]. This error can be mitigated with larger geometries, but this solution is also impractical for small mobile systems.

As mentioned above, one noise source for flash LADAR is the noise introduced by alternate light sources. Because the flash emitted by most flash LADAR cameras available today is in the infrared spectrum, most light sources will introduce noise. This is especially true of light sources that are relatively wideband, as is the case with sunlight or fluorescent lighting. Because most of the testing of the physical system occurs where

fluorescent lighting is omnipresent, it is important to be aware of this error source. While the noise due to varying reflectivities is documented, the very similar noise due to additional light is not. This noise is discussed in greater detail in Chapter 3.

There also exists errors inherent in all time of flight camera systems. First, there is the error introduced by multiple reflections, often referred to as "multipath," and this interference occurs in two methods. Internal reflections in the camera affect both the measurement of the reference beam and of the beam from the target. External reflections happen entirely outside the camera, occurring as reflections from objects in the scene [14][16]. Figure 2.5 shows how multipath error is introduced. The black lines are the direct signal path and shortest range to the target. The red line is a potentially interfering signal that results from the signal bouncing in the scene before returning to the camera. If these two signals both reach the camera, the result will be an erroneous range measurement. This erroneous measurement will have greater range magnitude than the original signal in most cases.

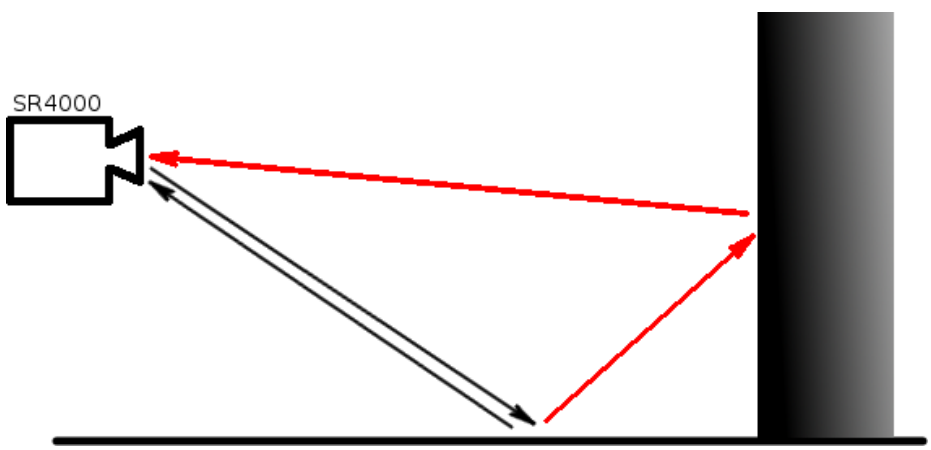


Figure 2.5: Demonstration of Multipath Error

Another such source of error, which results from aspects of the particular scene being image and the specifications of the camera, is depth inhomogeneity. When one pixel

receives light from two different sources in the scene at different ranges, the result is the measurement of “flying pixels,” at boundaries such as corners or points. Motion artifacting will also occur at object boundaries. Motion artifacting is a result of the camera integrating received signals over a given integration interval. If the camera moves during this integration period (usually on the order of 10^{-2} seconds), the data will be somewhat corrupted, especially at object edges. This suggests that flash LADAR is better suited for mobile situations where the scene to be measured is not feature-rich. Figure 2.6 shows how flash LADAR measures flying pixels.. It is often assumed that the laser spot on the ranging target is a single point. This is not the case, as the spot has some finite size. At corners, flying pixels result from the measurement of the surface on both sides of the corner, resulting in a mixed range result [1].

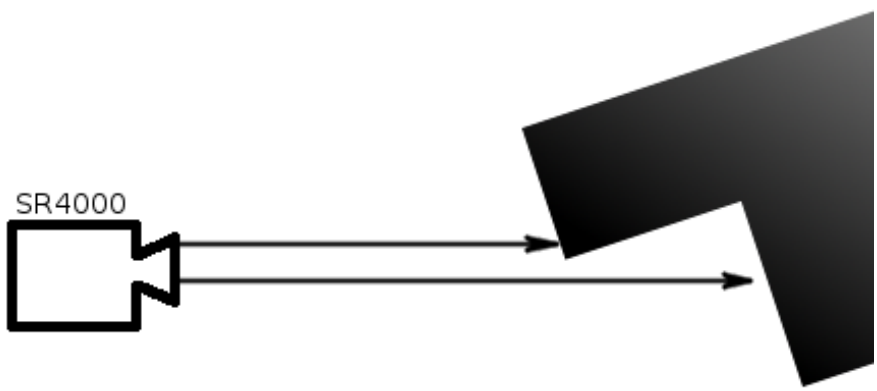


Figure 2.6: Flying Pixel Production

Another error is direct interference from several planes. Because a pixel does not have infinitely small size, it can receive the return from two different planes. Whereas this

fact results in the flying pixels mentioned above in convex surfaces, the result is different for concave surfaces. For instance, if the camera receives returns from two different planes in one corner, the range will be shorter. Thus, corners will appear rounded and closer. The concept is illustrated in Figure 2.7.

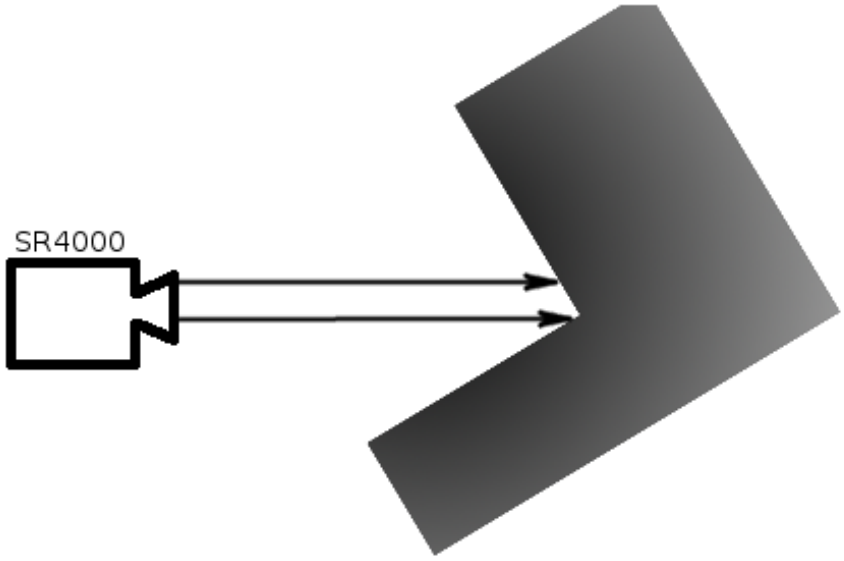


Figure 2.7: Direct Interference Principle

The error of a time-of-flight system also varies with angle of incidence. Because the intensity returned is lower when the angle of incidence is increased, the accuracy is decreased. The secant of the angle of incidence describes how the error increases with angle of incidence. The error changes with the secant of the angle of incidence. This error is difficult to remove, as its removal depends on reliable measurement of angle of incidence.

2.4.5 Range Ambiguity. There also exists a range ambiguity in flash LADAR data. This ambiguity occurs because, as described in Section 2.4.3, the light is modulated at a constant frequency. For instance, the SR4000 has its frequency modulated at thirty

megahertz, which for light corresponds to a wavelength of almost exactly ten meters [21]. The correlation of this signal is only unambiguous to one half wavelength of this signal. Thus, beyond five meters in range, the camera cannot resolve the range ambiguity and returns a range result of five meters less. For example, an object six meters away will be seen at one meter away by the camera. The same happens for each ambiguity "bracket," each multiple of five meters away, meaning the range estimate will be the same for an object eleven meters away. Because it is useful to know the absolute rather than the unambiguous range to objects in the scene, especially for navigation and localization, it is useful to consider the methods that have been researched for removing or resolving this ambiguity.

One method first uses edge detection in order to determine where a likely range ambiguity occurs. Because it is known that where the range ambiguity occurs there is a large edge, this step allows for the identification of the ambiguity relatively reliably. This data is used to segment the image. These segments are then reduced in number in order to simplify processing. Then these segments average intensity return is measured, under the basic assumption that the returns will be much greater for the closer intensity brackets. Based on this information, each segment can be classified by its range bracket. Thus the range of the pixels in each segment can be corrected [20].

Another method of removing range ambiguity relies on the use of two cameras. The range ambiguity occurs at one half wavelength of the modulation frequency of one camera. Thus, if a second camera is used with a different modulation frequency, the range ambiguity will occur at a different range. If the data from these two cameras is combined, the range ambiguity can be resolved. For each pixel, only one ambiguity range bracket produces a similar result. This concept is shown in Figure 2.8. However, this method was not completed in experiment with two different cameras. A single, custom-built camera was used in the same location twice, meaning that the same method would need further

work to develop an algorithm to match pixels from two differently located cameras, or practical multi-frequency cameras need to be developed. The camera used was larger and more complex than any commercially available LADAR system [24]. This suggests that multiple-frequency cameras may be extremely useful in the future of flash LADAR, but currently the possibilities are limited.

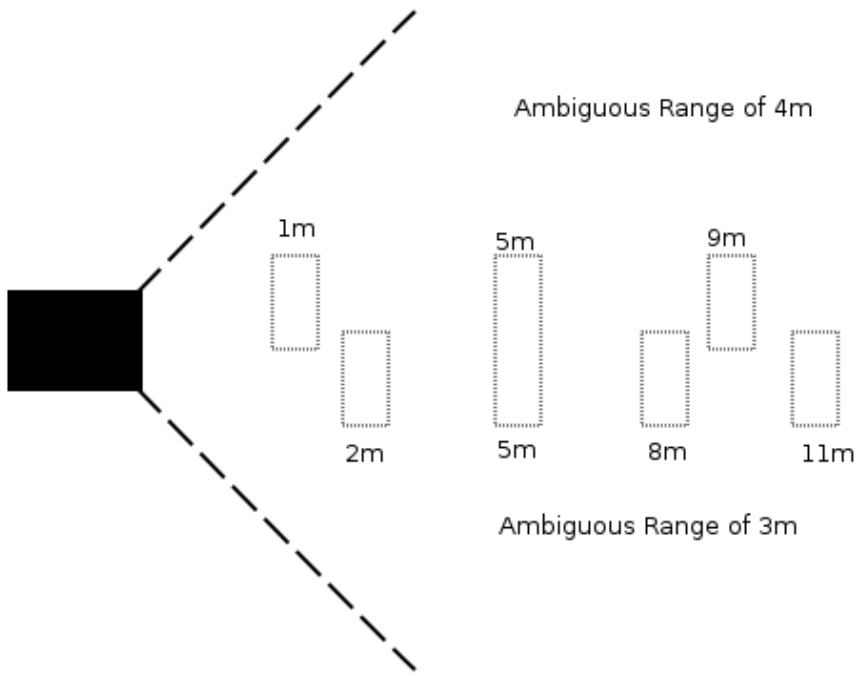


Figure 2.8: Range Ambiguity Resolution Using Two Different Frequencies

Range ambiguity must be removed in order to effectively use LADAR data for navigation. The removal of range ambiguity is further discussed in Chapter 3.

2.4.6 Time of Flight Advantages. Time of flight has many advantages over interferometry and triangulation for navigation. Triangulation, while it offers higher accuracy, requires two cameras. Also, it is less accurate over certain ranges. Triangulation also requires more processing per pixel than time of flight does. Interferometry has the

highest absolute accuracy of all three approaches, but it also has the shortest non-ambiguous range. This limitation alone makes its use limited to laboratory applications.

Time of flight is suited for this application mostly because it offers a small, lightweight, and compact means of providing both range and intensity data. It offers relatively fast imaging speeds, which makes it better for application to mobile vehicles. It is somewhat vulnerable to motion artifacting and noise from outside sources, but much of this can be mitigated indoors.

2.5 Introduction to LADAR

Laser detection and ranging (LADAR) is a computer vision method that uses information from the electromagnetic spectrum to gain depth and intensity information from its surroundings. LADAR systems consist of a laser light emitter and a camera. The laser is used to illuminate a target with coherent light. The camera uses photodiodes to record the intensity and phase of the returned light. In this way, a LADAR system is nearly identical to a conventional radio detection and ranging (RADAR) system. The systems differ in the portion of the electromagnetic spectrum they use.

There are many schemes for obtaining information from the data provided by a laser light emitter/receiver pair. Line-scanning LADAR uses a single laser beam to sequentially illuminate an entire scene. Flash LADAR uses a flash of coherent light to illuminate an entire scene at once. LADAR allows for the measurement of both the range and the intensity of a three-dimensional (3D) scene. The range is simply the distance from the detector to a portion of the scene, as measured by the detector. The intensity measurement reflects how much light is returned from the scene to the detector. LADAR systems, like the MesaImaging SR4000, use a time-of-flight (TOF) method for determining range information.

2.5.1 Line-Scanning LADAR. Line-scanning LADAR is a laser camera system that uses a single collimated beam in order to find the range to a single point at a time. As the name suggests, the laser is panned across a line or multiple lines to form an image. At its simplest, it produces a single line of pixels with range and intensity data. There has been much use of line-scanning LADARs in navigation and success in using them. However, there are problems inherent in the line-scanning method. Most importantly, the laser must scan the scene, moving from one pixel to the next. Thus, for a relatively high-resolution image, the scanning process can take a considerable, non-trivial length of time. This results in relative motion error. In this situation, the pixels of the image that update earlier do not necessarily correspond with those from a later epoch due to movement or rotation of the body. This makes data processing much more difficult, especially if attitude changes occur during the LADAR update. Also, update rates rely on the speed of the scanning laser. Thus, the update rate is limited mostly by the physical capabilities of the sensor. Line-scanning LADARs are also mechanical systems, which means they are usually poor choices for highly-dynamic movement or systems for which maintenance is troublesome.

2.5.2 Flash LADAR and the SR4000. Flash LADAR is a variation of LADAR that updates in three dimensions every update epoch. At its simplest, it is a two-dimensional array of detectors that each provide range to a target. This array, called a focal-plane array, provides simultaneous updates for each detector at each time epoch, as mentioned above.

In both visible light cameras and flash LADARs, a single silicon chip (often a CMOS chip) provides the detectors necessary for image capture. As opposed to a conventional camera, however, an additional step is required in processing the input to a time-of-flight sensor as used in the SR4000 and other LADARs. The correlation in Equation 2.3 must be completed for each pixel. The simplest and most effective method for completing this calculation is to evaluate the correlation at the sensor itself. The necessity for correlator

circuitry in the silicon detector array results in large pixel "footprints." The result is that flash LADAR cameras that are limited in size are also limited in resolution, which is one of the greatest disadvantages to using flash LADARs. For instance, the SR4000 has a resolution of 176 pixels horizontally by 144 pixels vertically. This resolution is much lower than that of even low-quality visible light cameras without ranging ability. This has led to the processing of LADAR data alongside high-resolution true-color images.

2.5.3 SR4000 Flash LADAR Camera. The SR4000 is a flash LADAR camera that is both compact and lightweight. It uses an ethernet connection to transfer data via a cable to a nearby computer or other device. This camera is one of only a few commercially available flash LADAR imaging devices. It is far smaller than most line-scanning LADAR cameras; it is little more than a tenth the weight of the SICK LMS-200 laser measurement sensor. Another, more competitive line-scanning LADAR, the Hokuyo UBX-04 weighs only 0.16 kg and uses only 4 W of power, and its specifications suggest it has similar accuracy to that of the SR4000, despite having a larger angular resolution [23]. The SR4000, however, has no moving parts and has no maintenance requirements besides cleaning the lens of the sensor. Other competing sensors are the Canesta XZ422 and the PMDTech CamCube. These cameras and the SR4000 all return both an intensity image and a range image, both of which are useful for processing for purposes of navigation or tracking.

The field of view mentioned in Table 2.1 is the angular measure of the camera's capability to see the environment. Each pixel of the camera is assigned an elevation and azimuth angle corresponding to its position on the sensor, which is calibrated with the manufacturer. The field of view is visualized in Figure 2.9, which shows the SR4000 field of view. This image also shows that images in the scene are projected to the focal-plane array at the far left, and that each pixel corresponds to a specific point in the scene.

Table 2.1: Manufacturer-provided Specifications of the SwissRanger SR4000 Flash LADAR

Parameter	Specification
Communication Interface	Ethernet
Modulation Frequency	30 MHz
Detection Range	0.1 - 5 m
Calibrated Range	0.8 - 5 m
Absolute Accuracy	+/- 10 mm
Pixel Array Size	176(h) x 144(v)
Field of View	69 ° (h) x 56 ° (v)
Weight	510 g

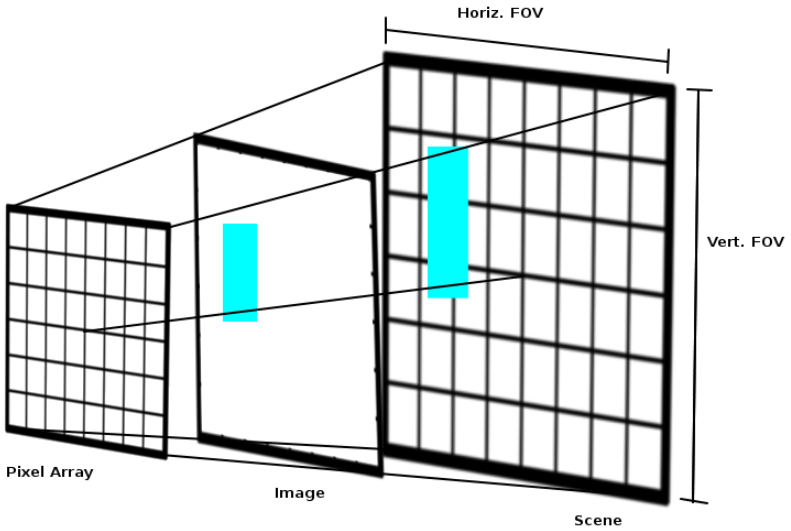


Figure 2.9: SR4000 Field of View

The SR4000 calibrates incoming images using data compiled by the manufacturers, MesaImaging. This data is unique for each camera and is generated at the factory. The

calibration occurs at the hardware level, and thus each captured image is automatically calibrated. The calibration occurs on an FPGA in the camera, and cannot thus be stopped. The data provides a relatively good calibration of the system. However, Chapter 3 describes how this calibration is somewhat flawed. For this research, it is assumed that the calibration data is acceptable, but it is also apparent that user calibration could provide a better solution.

This camera outputs data in a 3-dimensional Cartesian frame relative to the body of the camera. The three axes, X, Y, and Z form a right-handed coordinate frame. As shown in Figure 2.10, the three axes are all perpendicular. The Z axis points out from the camera. The X and Y axes point to the camera's right and top, respectively. Thus, the X and Y axes form a plane with the same normal vector as the camera's orientation. The X and Y coordinates are measured as positive or negative distance from the center of the camera's field of view. This is hereafter referred to as the body frame, as it is assumed the camera is represented by a point at the center of this frame of reference.

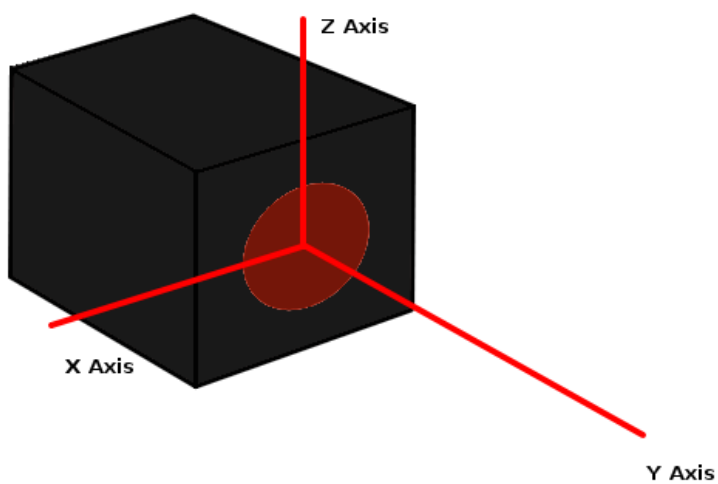


Figure 2.10: SR4000 Body Axes

Given that flash LADAR is relatively young, the SR4000 represents potentially the best commercially available camera of its type. Thus, it is perfectly reasonable that it be used to be used to test the utilization of flash LADAR sensors to detect cross hallways.

2.6 Past Research in use of Ranging Methods

The use of ranging techniques in navigation has a long pedigree. Radar, which has existed in some form or another for decades, has long been used for navigation. LADAR, while a newer technology, has since been used many times for navigation. Flash LADAR, even newer than its counterparts, has seen less use and research into its utilization in navigation.

2.6.1 Line-Scanning LADAR Research. Line-scanning LADAR has a longer history of use than flash LADAR, as these systems have existed in a practical form for a longer period of time. Line-scanning LADAR has been successfully used to augment other navigation sensors in situations such as the first DARPA Grand Challenge in 2005. Because the output of flash and line-scanning LADAR cameras is very similar, many of the same data processing methods can be used. Thus, it is useful to consider how line-scanning LADAR is used in navigation.

Many teams utilized LADAR in the Grand Challenge. Researchers for Team CIMAR used several line-scanning LADAR systems in order to determine obstacle position and optimal path. In order to determine a suitable path, plane detection was used to find the easiest terrain to cross. This data was also processed to provide control inputs to the vehicle based on its position relative to the optimal path [9]. The TerraMax team from the Ohio State University also used line-scanning LADARs for pathfinding and navigation [2]. Carnegie-Mellon University's Red team utilized a combination of line-scanning LADAR and radar [25]. In all three of these cases, GPS provided coarse location information that

would likely be unusable in an indoor setting. Thus, for the purposes of this research, it is also necessary to consider the use of LADAR as an alternative to GPS navigation.

Line-scanning LADAR has often been used to detect roads or pathways for vehicles. This past research is pertinent to the problem of cross-hallway detection, as the techniques and assumptions used can be adapted for the purpose of cross-hallway detection. In one case, LADAR was used as the principle sensor in road detection and vectorization. Road detection entailed identification and location of potential road surfaces. Vectorization was the processing of data to determine the direction of the road. To achieve this, prior information about the road, including its height and reflectivity, were used in a road model that was the applied to the detection algorithm. This road model was dependent on the data received. For instance, with low-resolution data, a line is the best approximation of a road, but with high-resolution data, a plane segment is a much better model. In order to apply this to the cross-pathway detection problem, it is important to know what data is received from the sensor. This is addressed further in Chapter 3 [5].

The same research discusses several methods for detecting roads. Because roads can be likened to walls, roofs, or ceilings, this research can be applied to hallway detection. One potential method is the detection of two boundary lines for each plane that describes two of the plane's edges. Many of these methods relied on some method of sensor fusion or operator involvement to detect road edges. Also, the research details an algorithm that classifies all points as "road" or "non-road" points, an approach that could easily be extended to wall detection [5].

Further research into road detection suggests that the use of intensity data from LADAR images could be used to determine which points are or are not road surfaces. This approach can also be applied to the detection of hallways. By creating an artificial set of intensity thresholds for walls, points that do not meet the given criterium could be filtered out [6]. This reduces the number of points to process and reduces the complexity

of any feature detection. However, this research was carried out from much greater range than any indoor setting. This provided an advantage in that the intensity of return from each pixel did not vary greatly with distance. However, separate research suggests that intensity can be corrected for distance through image processing techniques. By using a model of the intensity return change with distance, the intensity image could be calibrated even in an indoor setting [22]. Using such techniques, the intensity image might successfully be used for such filtering.

Another method of feature detection in LADAR data is corner detection in line-scanning LADAR outputs. This method uses existing corner detectors such as the Kanade-Tomasi and Harris corner detectors. These features are described as "general purpose," as they can be used as features for tracking or navigation algorithms. For instance, the algorithm was also tuned to detect trees. The weakness of this approach to feature detection with flash LADAR is that it usually requires a higher resolution camera than flash LADAR can provide. This is due to the fact that flash LADAR simply provides fewer good features than higher-resolution alternatives. This method is better suited for large line-scanning LADARs.

2.6.2 Extension to Navigation. Recent research has included the use of line-scanning LADARs for the purpose of navigating small, automated vehicles indoors. These micro air vehicles (MAVs) would benefit greatly from the use of a single sensor that could provide range data, especially in indoor areas where GPS signals are unreliable or unavailable. By using LADAR as an aiding sensor, navigation and localization indoors were improved. In this case, LADAR was used as a completely stand-alone source for both heading and position. The LADAR was simulated and the simulation was used to provide estimates of position and heading. When coupled with a simulated inertial measurement unit (IMU), the LADAR system provided a navigation solution. Image 2.11 demonstrates the navigation solution arrived at by using LADAR [4]. This research shows

that line-scanning LADAR can be used successfully used for indoor navigation. It can be assumed that this result can be extended to flash LADAR results.

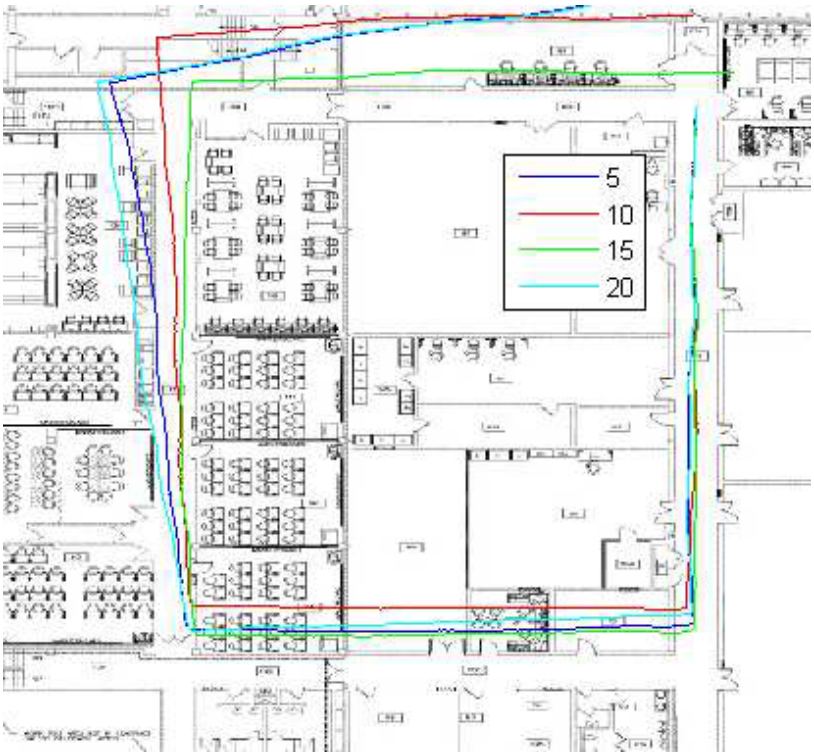


Figure 2.11: Demonstration of navigation solution utilizing line-Scanning LADAR Data [4]

Other research has used the SR4000's predecessor, the SwissRanger SR3000 for navigation. The basic approach used the identification of static planar surfaces to localize the vehicle in an area. This approach was shown to be successful in research. If these planes are assumed to be truly static, the vehicle can be localized using these as features [27]. It is assumed that a similar approach using static planes detected in LADAR data could be extended to localization in a hallway or other indoor environment.

The use of both RGB and LADAR data together has been done experimentally several times. As mentioned in Section 2.5.2, this is one way to mitigate the problem of

low resolutions in the data output by existing flash LADAR cameras. This method allows for several potential different methodologies. In one experimental setup, field-programmable gate array (FPGA) firmware was developed that pre-processed 2D RGB video data and 3D flash LADAR together, applying range data to the 2D image before it was output to a computer. This allowed for fast conversion of the 2D image to 3D [18]. Other research in the same direction used a complex algorithm in order to map depth to each surface in the RGB image. By correlating surfaces in the depth, intensity, and RGB images, depth can be relatively accurately assigned to parts of the high-resolution image. This increases the ability to utilize the RGB image [19]. However, both of these methods are computationally intensive.

3 Methodology

THIS research was completed by a combination of simulation results and experimental results. The LADAR was simulated in order to create test data that was simple to acquire. This simulation was verified against real data in order to confirm that it was an acceptable simulation.

3.1 Assumptions

Several assumptions were made to make the problem and its solution simpler.

3.1.1 Manhattan World. The first assumption made when developing the cross-hallway algorithm was the Manhattan World assumption. This assumption suggests that the world is predominantly comprised of orthogonal surfaces and regular planes. This assumption simplifies the problem and allows for the simulated rooms and hallways to be simpler in construction.

From a stochastic viewpoint, this assumption holds for a variety of environments. These areas have statistical regularities that can be used for processing images. When the orthogonal world assumption is utilized, edges or features in the image usually either form or follow an existing orthogonal grid. This grid can be used to identify an axis system, which in turn defines an orthogonal reference frame. Research has shown that this reference frame can be measured for a myriad of different scenes, to include urban and rural areas [7]. Orthogonality is present in most situations. If this reference frame is assumed to exist, certain information processing methods can be assumed to be effective.

The Manhattan world assumption was developed for "urban canyons," areas defined by buildings and streets arranged on an orthogonal grid pattern. However, it is often applicable to indoor hallways and buildings as well. Research has also suggested that this assumption can be applied to indoor and outdoor outside of clearly defined hallways, such

as rural roads or open planform rooms [8]. Because the Manhattan world assumption is extensible to situations outside of the average urban area, the algorithm developed to determine the position of cross hallways could potentially be expanded to many situations.

The use of this assumption allows for several simplifications of the cross-hallway detection algorithm. It can be reasonably assumed that cross hallways are orthogonal or nearly orthogonal to the vehicle's current hallway. Also, this assumption can be used for localization of the vehicle within the hallway. With this assumption, the center of the hallway in a single axis can be found as the point directly between the two planes. Thus, this assumption is central to the development of the position-finding algorithm.

The problem with using this assumption is that it limits the use of the detection algorithm to situations where this holds. While most man-made structures follow this assumption, it is not the case for all areas. Chapter 4 discusses a case where the algorithm works successfully without this assumption, but the result not ideal.

3.1.2 Static Plane. It is assumed that the environment to be measured consists of planes that can be identified as walls, floors, and ceilings. This is valid in urban areas that mostly consist of planar walls, floors, and ceilings. Even in cases where there is no cross hallway, the existence of such planes allows for relative position finding within the hallway. Also assumed is that these planes remain stationary between frames. Thus, even if the vehicle or camera moves between frames, the same planes can be identified in each time epoch and treated as the same walls and ceilings.

3.1.3 Calibration. As mentioned above in Section 2.5.3, the SR4000 automatically calibrates the data that it receives based on factory calibration data. However, this calibration is not perfect, as can be clearly witnessed around the edges of an image. The edges of the image consistently have a positive range error. In some cases, even, the range at the edge of an image is measured at ranges greater multiples of the

ambiguous range, which is clearly impossible. The manufacturer's specification is for noise with a standard deviation of 10 mm, but the standard deviation of the error shown in Figure 3.1 is measured at 5 cm. However, the central pixels in the image fit the specified error statistic. At the corners, the range error can reach a standard deviation of 10 cm or more. Because of this disparity in error, there is the potential option of simply removing the points in the corners that display large amounts of error. This was not done, because even the inaccurate data points provide information on the scene. Figure 3.1 shows one such image from the camera, captured at one meter from the wall. This is an image captured from a flat wall with no corners. This data is plotted in green alongside the ideal simulation data, which can be treated as truth and is plotted as blue points. At the top and bottom of the image are the corners of the range image, which display considerable error. The top of the blue line represents the far right corner of the simulated image. The green points at the top of the image correspond directly with the top of this blue line, but the error apparent in their position means they are not colocated. Demonstrated is the extreme error in the corner of the image. The standard deviation of the error in this image is less than five centimeters.

3.1.4 Assumptions of Noise. All of the noise modeled in this simulation was first assumed to be white and Gaussian. This is a reasonable assumption because many noise sources can be modeled as Gaussian noise sources. Also, many probability density functions can be represented as the sum of Gaussian probability density functions. Thus, if the dominant sources of noise are well modeled with Gaussian functions, the white and Gaussian model for noise is acceptable.

The Gaussian noise distribution for the range error was verified using range information from the camera. By determining the error at each point, error statistics can be determined over a series of frames. For instance, Figure 3.2 is a histogram of the range

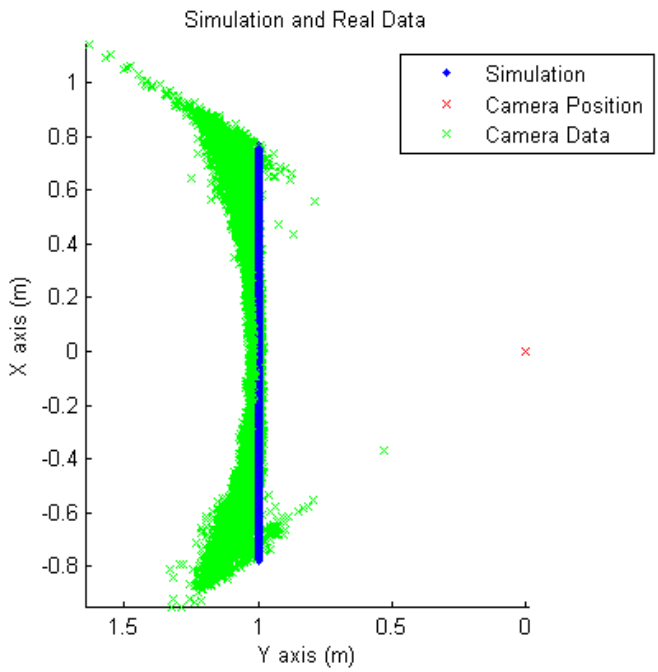


Figure 3.1: Simulated and Actual Data for One Meter

error for a single pixel, collected from 132 frames of data. The Gaussian in Figure 3.2, plotted as a blue line, is the Gaussian defined by the measured mean and standard deviation of the error. This noise distribution was also seen in other pixels in the image. This confirms that the error can be represented as a Gaussian at each point. The bias evident in this noise was handled as the mean of the noise at each pixel, whereas the spread of the Gaussian was encapsulated in a standard deviation.

3.2 SR4000 Characterization

3.2.1 Interference of Outside Noise. Many of the noises sources inherent in the data output by flash LADAR sensors were well-documented. However, the noise introduced by interference from outside sources was not. When the LADAR sensor received information on a scene illuminated by certain kinds of light, there is considerable

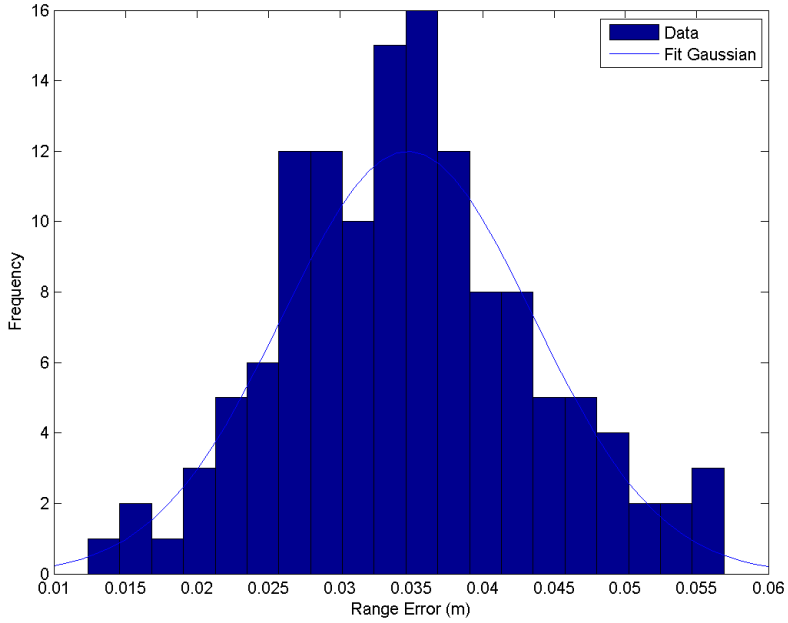
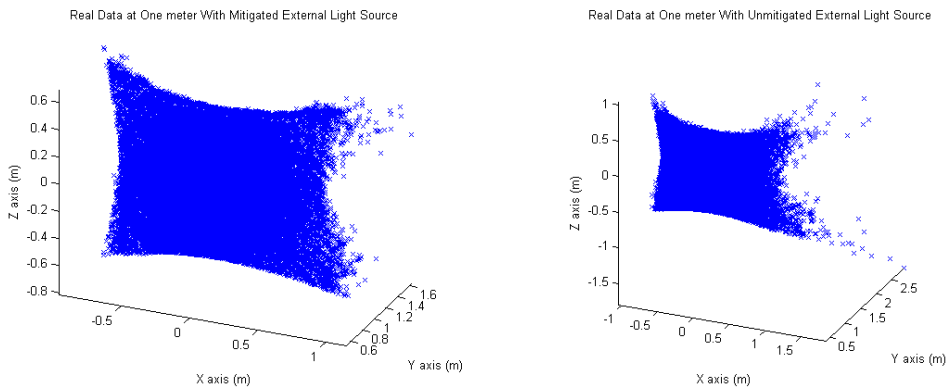


Figure 3.2: Histogram of Error at Single Point with Gaussian Approximation

noise introduced. For example, Figure 3.3 shows two images. In Figure 3.3.3a, outside light was shaded from the scene, essentially removing a noise source. Figure 3.3.3b shows the exact same scene. The camera was not relocated, but the scene was lit with sunlight in the far right side of the image. Both the maximum range error and the standard deviation of the error were significantly increased in this portion of the image.

Another image of a plane at 2 meters from the camera was also interfered with by sunlight. This example, shown in Figure 3.4, displays the increased error due to sun. The green points are the data acquired from the camera. The blue points are the simulated truth data. The red bracket shows the area where the noise has a larger bias than a similar image with no sunlight. The image is top-down, and it is clear that along the right side of the range image, there is considerable error. When this range image was captured, there was sunlight shining on the surface being recorded on the right side. The apparent result was



(a) Outside Light Mitigated

(b) Outside Light Unmitigated

Figure 3.3: Point Cloud from One Meter With and Without Mitigation

greatly increased error. The effect is also noticeable with fluorescent lighting, but is less pronounced.

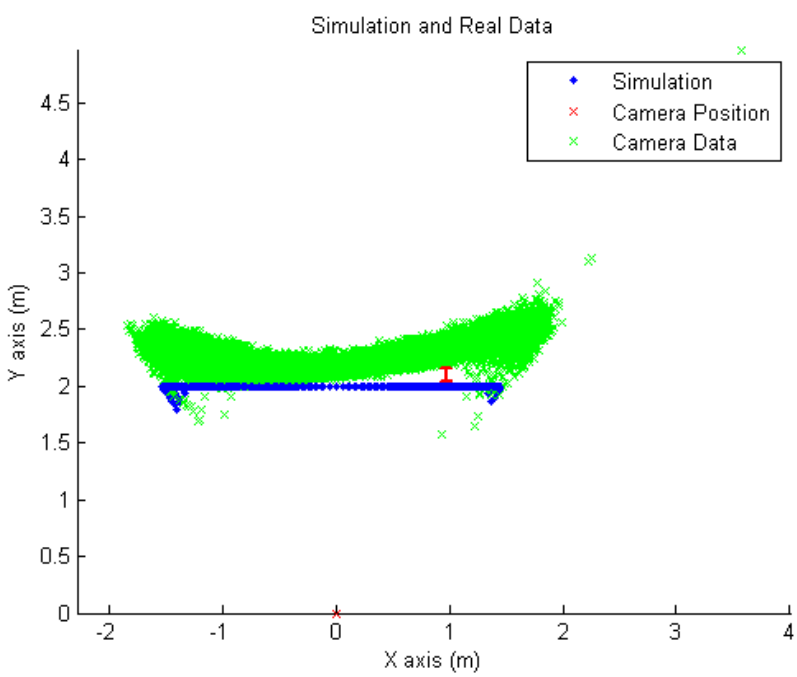


Figure 3.4: Demonstration of Sunshine interference at Two Meters

3.3 Simulation Development

In order to more simply develop and test an algorithm for cross hallway detection, a simulation was developed that mimicked the output of the SwissRanger SR4000 camera, and could optionally be adapted for other flash LADAR cameras. It was important that this simulation meet several requirements, which will be discussed. The simulation was to be used for a variety of verification and algorithm development tasks.

3.3.1 Simulation Requirements. The simulation was required to be fully three-dimensional. Because flash LADAR is mostly useful because of its fast, simultaneous updates, the simulation needed to be capable of providing the same. Thus, the simulation had to provide updates for a simulated focal plane array and output them as a single range image. This output had to represent each pixel and the position of its measured return. In the case of simulating the SR4000, this meant that each simulated measurement would require 176×144, or 25,344 points as a point cloud. This point cloud also had to be represented as a set of Cartesian coordinates, as the camera outputs its range data already converted to Cartesian coordinates and calibrated based on factory data.

The three-dimensional output requirement was related to a few other requirements. The Cartesian coordinate outputs could not be generated directly. Instead, they were required to be determined in the same fashion the camera generated them. Thus, each point in the point cloud would first need to be simulated as a range, azimuth, and elevation. This data would then be converted to Cartesian coordinates. Another requirement for the simulation derived from these is that the simulation must be equipped to handle different orientations and positions in space. Virtually rotating the camera should be possible, as should changing the position of the camera.

The simulation also had to support vehicle dynamics. The code had to be able to simulate a camera with a given “flight path.” Without this functionality, it would be

impossible to test the algorithms dynamically. Because the algorithms are intended for potential moving vehicles, this is necessary.

The simulation also needed to be able to simulate the range ambiguity of the sensor. In order to remove the ambiguity, it is simpler to test any algorithm on simulated data before attempting to remove the ambiguity on real data. The noise in real data makes it more difficult to accurately remove the ambiguity, so testing the algorithm first on simulated data will ensure the algorithm is at least sound.

The intensity image output was deemed unnecessary for the simulation. Because the intensity image is largely variable with environment, lighting conditions, temperature, and camera orientation, it is an unreliable source of information for what is intended to be an algorithm that can be applied generally.

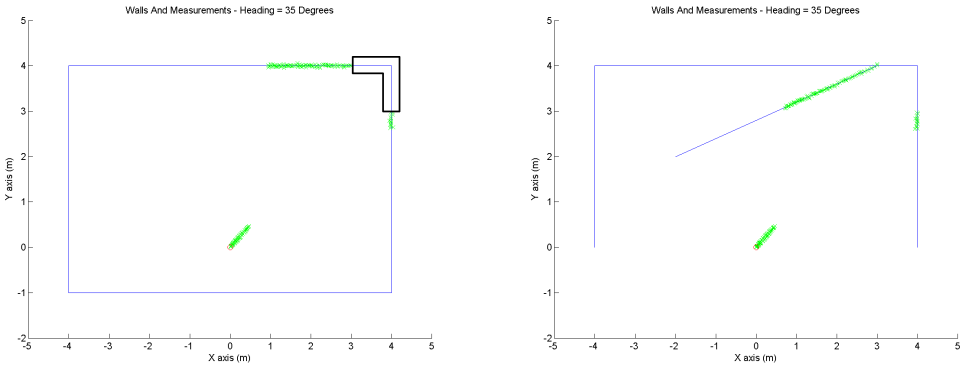
3.3.2 Two-Dimensional Simulation. In order to develop the full three-dimensional simulation, it was first useful to create a two-dimensional version of the simulation. In this way, some of the basic design principles of the three-dimensional simulation could be decided before it was coded.

This simulation operates first by loading a room, which uses basic information to create a two-dimensional room of lines that represent walls. Each of these walls is defined by its two endpoints. In this way, the room can be quickly changed or built by simply defining new walls in this standard format. Because the endpoints are two-dimensional, the line segments are simply defined by four values and can be updated simply.

The simulation operates by creating rays that project from the camera. These rays relate to the direction that each pixel of a LADAR will face, but in a single line as opposed to a two-dimensional array. Then, the intersection of these lines and the room line segments is computed. The range is computed, and if it is desired, noise is added to the computed value. This new noisy value is used to find the nominal coordinates in the two

axes. If the option to preserve the range ambiguity is selected, the range is adjusted before being reconstructed as a Cartesian coordinate.

This simulation provided basic functionality that was extended to the development of a three-dimensional simulation. The same basic principles were utilized, but extended to three dimensions for the full simulation. Figures 3.3.5a and 3.3.5b demonstrate both the range ambiguity capability and the adjustable rooms. The first image shows that beyond a certain range, the result is decreased by the ambiguous range, as would be expected in real data. The points near the center of the image map to the corner of the room bounded by the black box. The second demonstrates a room with an extra wall in order to show the capability of the simulation to handle a variety of rooms. Finally, Figure 3.6 shows the same room and camera orientation, but without the ambiguity implemented.



(a) Basic Two-dimensional Room with Ambiguity(b) Adjusted Two-dimensional Room with Ambiguity

Figure 3.5: Two-Dimensional Simulation Examples with Ambiguity

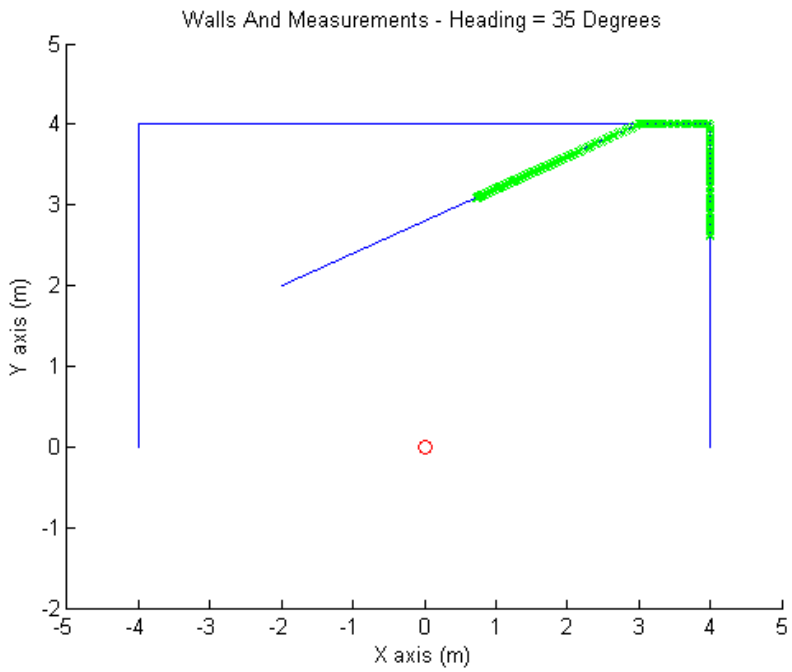


Figure 3.6: Adjusted Two-dimensional Room without Ambiguity Enabled

3.4 Three-Dimensional Simulation

The basic conception of the three-dimensional simulation is essentially the same as that of the two-dimensional simulation. The image is scanned in two dimensions, creating a set of points that represents each pixel of the detector. Each pixel has a corresponding azimuth and elevation, just as with the actual sensor.

3.4.1 Basic Simulation Algorithm. The simulation works as described above. It scans through azimuth, elevation, and walls and uses the best intersection point for each row and each wall. The pseudocode shown in Pseudocode 1 shows the basic process used to generate this data. The *MaxRows* and *MaxCols* variables describe the vertical and horizontal resolutions of the camera, respectively. The number of walls is determined by those planes defined in the algorithm that generates the room. The function isNotOpening

is necessary because the simulation is generated using infinite planes. This is discussed in Section 3.4.3 below.

Pseudocode 1 Simplified Simulation Algorithm

```
for row = 1 → MaxRows
    for col = 1 → MaxCols
        for wall = 1 → NumWalls
            Intersection ← findIntersect(row, col, Wall)
            if isNotOpening(Intersection, Room)
              range ← norm(Intersection)
              if range < measurement
                measurement ← range
              end
            end
        end
        measurement ← measurement + noise
        col ← col + 1
    end
    row ← row + 1
end
```

3.4.2 Room Construction. The three-dimensional room cannot be so simply constructed as with the two-dimensional simulation. In the three-dimensional simulation, the line segments must be replaced by three-dimensional polygons. However, the computational burden of creating these polygons is unacceptable. Thus, infinite planes are

created instead. These infinite planes can be used to create a myriad of rooms and hallways. However, because the walls are infinite, the simulation had to be adjusted to ignore certain parts of the planes. A basic algorithm was created to achieve this.

The user must first provide the program with a series of four three-dimensional points for each plane. The coordinate frame used to define the room is not the same as the coordinate frame that the camera uses. The room frame represents the same frame as that of the vehicle whose position and trajectory can be set. Also, if any polygons are to be ignored for more complex rooms, the walls they are a part of and the polygons must be set to be flagged. These points and polygons fully define the room. As the simulation runs, the room is transformed to the body frame, depending on the position and orientation of the vehicle and camera. This allows for the simulation to change the room dynamically as the vehicle moves or rotates between time epochs.

The room's coordinates are not defined in the body frame, which is the frame the camera outputs its results in. Thus, in order for the room to be usable by the simulation, it must be converted to the body frame. Code was written that both translated and rotated the room coordinates based on the rotation and position of the camera. The result is a room whose walls and features are defined the same way the camera outputs them.

3.4.3 Point Cloud Construction. The actual determination of the coordinates of the point cloud happens in a very similar way to that of the two-dimensional simulation. The three-dimensional planes are defined and an intersection is found. A sample intersection is shown in Figure 3.7. This intersection is used to generate a range value. If noise is enabled, the range is corrupted by noise. If ambiguity is enabled, the range is decreased by the unambiguous range value until the range is less than the unambiguous range.

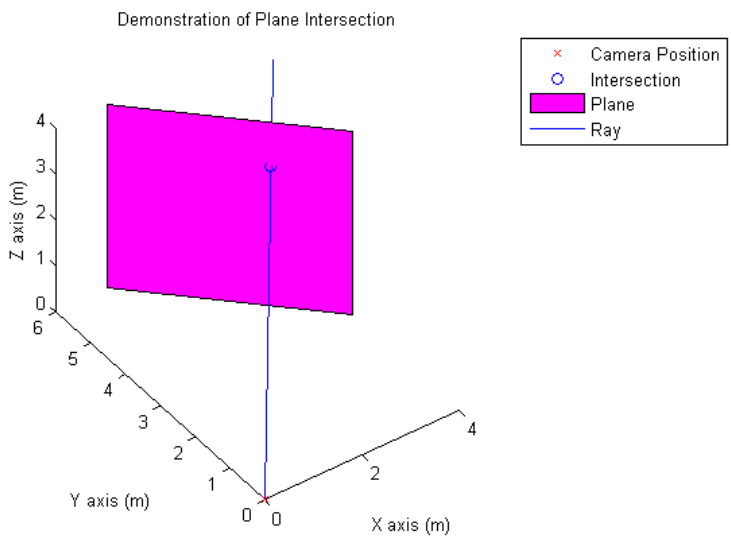


Figure 3.7: Example of Intersection Between Ray and Plane

The simulation algorithm, however, must determine whether or not the intersection occurs at a point that should be ignored. This can occur for several reasons. First of all, the same ray may intersect several planes. In order to resolve this problem, the shortest of the ranges is assumed to be the most correct. This is a valid assumption, because the intersections occur in an idealized situation. Thus, only the shortest of the ranges can be valid. This handles the vast majority of ignored intersection points. However, because the intersections are generated using infinite planes, it would be impossible to create more than the most basic of rooms without creating a way to ignore certain points. This is done by defining areas to ignore in the room definition and then ignoring these measurements in the simulation. Walls with areas to exclude are flagged. Any intersection with these walls is checked to ensure that it should not be ignored. Pseudocode 2 shows how this check is completed. The intersection point is checked to ensure it falls on the plane in question, and then it is determined whether or not the point falls within the bounds of the polygon defined in the room definition. The variable *polygon* is the set of Cartesian points that

define the boundaries of the polygon, and $polygon(1)$ is the first of this set of points. $Intersection$ is the calculated intersection point between the ray that defines the sensitive direction of the detector pixel and the wall plane. The variables $test1$, $test3$, and $test3$ are boolean arrays. The test also ensures that the point is coplanar with the polygon in question. If the point is in the polygon to be excluded, the point is ignored. This particular test adds to the computational burden of the simulation, but makes it much more extensible.

Pseudocode 2 isNotOpening Algorithm

```
if currentWall = flaggedWall
  then
      polygon ← flaggedPolygon
      test1 ← Intersection ≥ min(polygon)
      test2 ← Intersection ≤ max(polygon)
      test3 ← Intersection = polygon(1)
      if isCoplanar(Intersection, polygon)  &&  sum((test1&test2)||test3)  =  3
        then
            ignore ← true
        else
            ignore ← false
        end
    else
        ignore ← false
      end
```

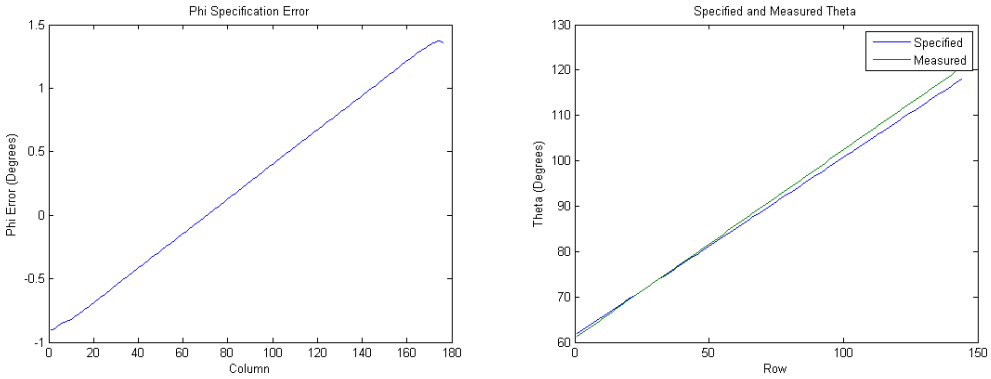
After finding the intersection and the range to the point, the simulation uses the potentially noise-corrupted and ambiguous range output by these algorithms and converts them to Cartesian coordinates as the camera would, using azimuth, elevation, and returned range.

3.4.4 Azimuth and Elevation Selection. While scanning through the rows and columns of the image, each pixel has a defined azimuth and elevation. The simplest approach is to assume that these are well-defined and well-known values, and can be linked directly to the specifications in Table 2.1. This was how the azimuth and elevation were initially treated. However, it became quickly clear that the specified values for both field of view and angular resolution were provided for the ideal case and differed from reality.

Using measurements from the camera, the actual azimuth and elevation of each pixel was determined. Range was immaterial in this determination, so range error was also excluded. There still exists, however, some error due to miscalibration of the camera. This is assumed to be insignificant, for reasons explained in Section 3.1.3.

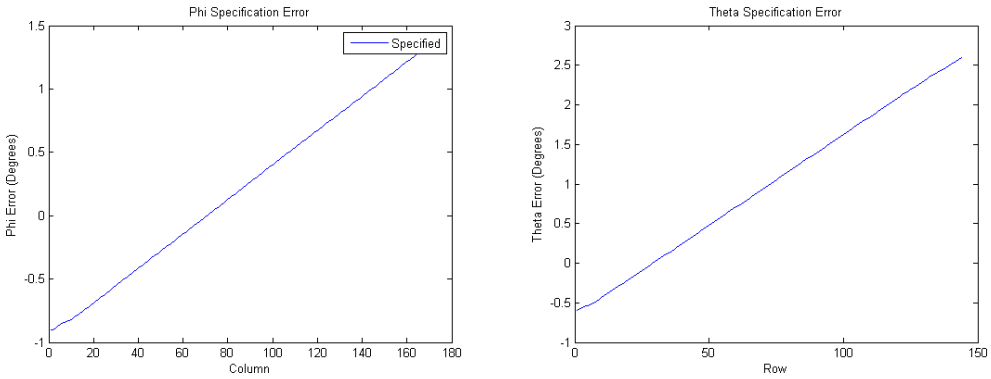
The published specifications were somewhat incorrect for the given images. Figure 3.8 shows both the measured and specified azimuth and elevation data, while Figure 3.9 shows the difference between the two. Table 3.1 compares the specified to the actual data. The azimuth and elevation vary with both row and column. Thus, an array was formed wherein was stored information about the azimuth and elevation data for each pixel. This data is used in the simulation for the generation of the point cloud.

3.4.5 Dynamic Simulation. Because the simulation was required to be dynamic, the simplest approach to making it so was making the simulation a standalone function. This allows the function calling it to update the position and orientation of the vehicle in



(a) Azimuth (b) Elevation

Figure 3.8: Actual and Specified Azimuth and Elevation Information



(a) Azimuth Error (b) Elevation Error

Figure 3.9: Error in Azimuth and Elevation

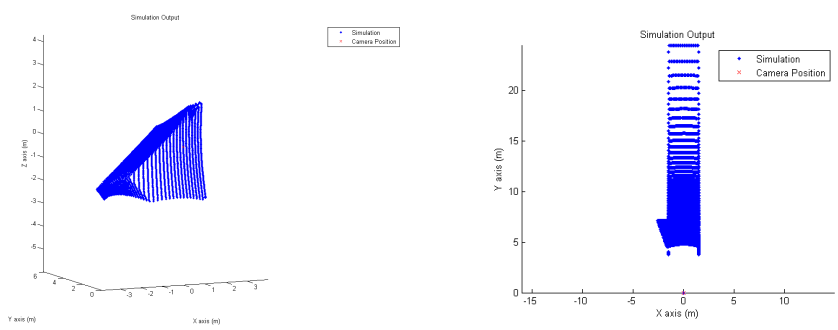
between simulation time steps. The result is a fully dynamic simulation that provides results for any time epoch.

3.4.6 Simulation Sample Output. The simulation is capable of producing output after the vehicle has been rotated or translated through space. The simulation is also

Table 3.1: Actual Field of View Characteristics and Specifications of the SR4000

Parameter	Specification	Actual
Field of View (h)	69 °	71.3 °
Field of View (v)	56 °	71.3 °
Angular Resolution (h)	0.39 °/pixel	0.407 °/pixel(Avg.)
Angular Resolution (v)	0.39 °/pixel	0.412 °/pixel(Avg.)

capable of producing noise outputs and mimicking the miscalibration of the camera. Figure 3.10 shows two different outputs of the simulation as verification of its operation. Figure 3.3.10a shows the output if the environment is a rectangular room and the camera is both translated into the corner and pointed toward the opposite corner. The other, Figure 3.3.10b, shows the bird's-eye view of a hallway and cross hallway simulated with the code. The simulation also supports the simulated output of other cameras, if specifics about their operation such as update rate and field of view are known. The simulation can also be applied dynamically.



(a) Corner of Square Room

(b) Top View of Hallway with Crossing Hallway

Figure 3.10: Samples of Simulation Output

3.5 Data Collection

Data from the sensor was found in two different ways: statically and dynamically. The static cases were intended to collect data to characterize the sensor noise. The dynamic collections were intended as data to test the hall detection and position- finding algorithms.

The static data collections were acquired with a planar surface filling the entire field of view. Past research has used small surfaces to measure the noise of the sensor. However, because the sensor has relatively low resolution, modeling the noise of a small number of pixels is not adequate. The images for this data collection were acquired with the camera positioned perpendicular to the wall. The camera was level to the floor. Its orientation was verified using a combination of level and t-square. The camera was mounted on a surface tall enough to ensure that only the target surface was measured. The distance to the target was measured. Data was acquired over half-meter intervals from one meter to three meters. At least one hundred images were acquired at each range to ensure the noise statistics were fully captured. The target was always the same, to ensure similarities in data. The target was a white, featureless wall. This wall was lit by a minimum of artificial light and at times when sunlight was at a minimum.

The dynamic data collections were completed by placing the camera on a cart. The camera was 0.875 meters from the floor on this cart. This cart was rolled through a hallway at the best approximation of a constant velocity as possible. Basic hallways, which can be defined by flat, featureless, and orthogonal walls, were used as often as possible. Considering the errors introduced by fluorescent light and sunlight, these datasets were obtained when the fluorescent lights were mostly turned off and in hallways where sunlight could not reach. One of the datasets, which was obtained near large windows, was measured in the evening to reduce any impacts of sunlight in the data. One dataset was taken with a minimum of clutter in the scene. Featureless walls with nothing

else in the scene were measured. A second dataset was taken with other objects in the scene. For instance, a large trash can resided in the middle of the hallway. Chairs occluded the camera's vision of the cross hallway. A third dataset was collected with the camera at a measured 30° angle. For all of the datasets, the camera was placed in the measured center of the hallway and aligned with the primary axis of the hall. The cart was pushed down the center of the hallway and images were collected at the highest capture rate of the camera.

3.6 Range Ambiguity Resolution

Using real-life data from the SR4000 has the added difficulty of the range ambiguity. It is nearly impossible to glean information from a scene while the range ambiguity is included in the data. Therefore, a simple algorithm was developed to remove the majority of this ambiguity.

By the nature of the problem, the range ambiguity will result in edges in the range image that have the magnitude $\lambda/2$. Thus it is possible, knowing this, to detect these edges and move points into the correct ambiguity bracket. This algorithm simply works by scanning through the image for edges of a magnitude near to that of the range ambiguity. This is completed for each frame of the data collected. The results of this ambiguity resolution are shown in Section 4.4.

3.7 Hallway Detection

The detection of cross hallways is useful for navigation because it would allow for the identification and use of alternate pathways in environments not mapped with *a priori* knowledge. In such cases, especially in dynamic situations, fast, automatic identification of possible paths can assist in unaided or unmanned navigation.

3.7.1 Feature Extraction. It was decided that the best method for approaching feature extraction in LADAR data is plane detection. Especially considering the

Manhattan world world assumption in Section 3.1.1, plane detection seemed the best way to handle the variety of errors and shortcomings of flash LADAR. LADAR's odd behavior when viewing corners and edges is compounded when the angular resolution is as low as the resolution most flash LADAR cameras' sensors can provide. Thus, features like edges and corners are unreliable in flash LADAR data. Also, as mentioned in Chapter 2, plane detection has already been used with flash LADAR data for successful navigation.

Other methods such as optical flow or feature tracking algorithms like scale-invariant feature transform (SIFT) or speeded up robust feature detection (SURF) are computationally intensive. Thus, they were ignored. While these algorithms are very powerful, they lack the simplicity of plane detection algorithms. Also, because a Manhattan world is assumed, a cross hallway or alternate pathway can be approximated by a plane. Plane detection algorithms can be used on surfaces that are only somewhat planar.

The plane detection approach was also a good choice considering the possible application of this algorithm to mobile vehicles. Certain algorithms, especially the random sample consensus algorithm (RANSAC) can be parallelized and adapted for mobile processors like field-programmable gate arrays (FPGAs) and application specific integrated circuits [11]. Also, lightweight, fast variants of the RANSAC algorithm have been applied to mobile vehicles in the past, even used for localization and navigation of mobile vehicles [13]. This makes this algorithm a very good selection for feature extraction using LADAR data on a mobile platform.

3.7.2 RANSAC Algorithm. The RANSAC algorithm works by randomly selecting points from a point cloud fitting a feature to these points. In two-dimensional point clouds, it is very good at detecting lines or edges. In three-dimensional point clouds, it can find many features, one of the simplest being the plane. The algorithm selects random points and tries to fit a plane to them. If it succeeds, it finds the other points in the image that fit this computed plane [12]. The basic idea of the plane detection RANSAC algorithm can

be found in Pseudocode 3. It works by selecting three points, determining the equation of the plane it forms, and determining whether the standard deviation of the distance to the plane is small enough that the identified plane is likely a good feature. This code can be run for any number of iterations, as long as there are points in the point cloud. The result is an equation for the plane and a mask that tells which points in the point cloud are part of that plane.

Pseudocode 3 RANSAC Pseudocode

```
bestSupport = 0; bestPlane = [0, 0, 0];
bestStd = inf;
alpha = 0.9;
epsil = 1 − forseeable_support/length(point_list)
N = round(log(1 − alpha) = log(1 − (1 − epsilon)^3))
for i = 1 : N
    j = pick3pointsrandomlyamong(point − list)
    pl = pts2plane(j)
    dis = dist2plane(pl, point − list)
    s = nd(abs(dis) <= t)
    st = Standard − deviation(s)
    if (length(s) > bestSupport)or(length(s) = bestSupportandst < bestStd)then
      bestSupport = length(s)
      bestPlane = pl; bestStd = st
    end
end
```

The RANSAC algorithm has several parameters that can be tuned. First is the σ, which is the assumed noise inherent in the measurement. In this case, the Gaussian assumption is required. This assumption is acceptable here because the noise from the sensor can be loosely modeled as a Gaussian. Increasing this number means that points that are farther away are more likely to be included in identified planes. Another parameter to tune is the P_{Inlier}, which is the probability that points within a certain noise threshold are members of the plane or not. This also increases the number of points accepted into each plane when it is increased by assuming more often that a point belongs in the plane identified. Finally, there is ϵ. This term describes the probability that no iterations will detect any planes. This value is used to decide the number of minimum iterations the algorithm will run before deciding on the optimal plane. Increasing ϵ will, on average, improve the results, but it will always increase the number of iterations necessary to detect planes in the scene. These three parameters are essential to the scene.

Another parameter that does not tune the algorithm but is important to the efficiency of the algorithm is the minimum iterations. As the algorithm searches for good planes, it will attempt to find the best candidate. The minimum iterations number forces the algorithm to continue looking for a better candidate even if it finds an acceptable one, until the algorithm reaches the required number of iterations.

This computational burden of the RANSAC algorithm grows exponentially with the number of points in the point cloud. In this case, the low resolution of flash LADAR allows for a relatively fast algorithm. Also, to take advantage of this fact, the extraction of planes from the range data will be quickly followed by removal of those members from the point cloud. In this way, the computation time can be kept to a minimum.

The algorithm, when applied to the hallway scene above, produces the results shown in Figure 3.11. The green points are those that are a member of the identified plane. The

blue points are those not identified as such. This plane could now be used as an identifiable feature in navigation or localization algorithms.

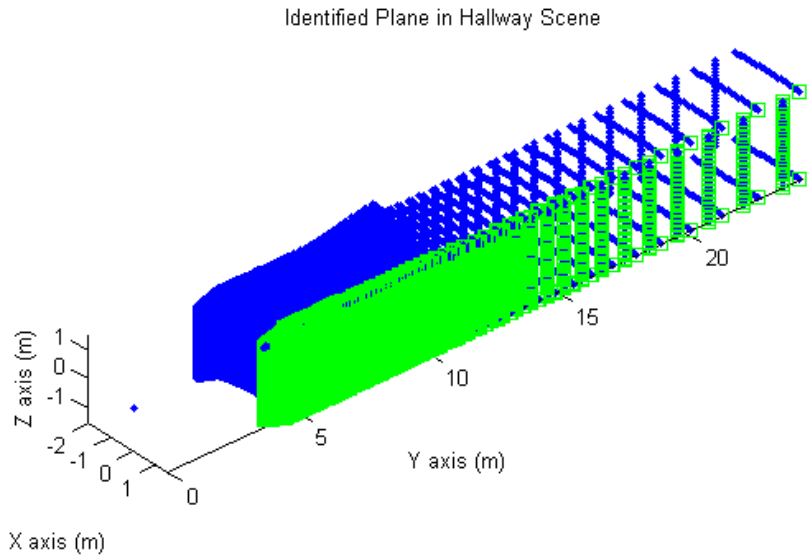


Figure 3.11: Hallway Wall Identified in Simulation Data

3.7.3 Detection Algorithm. In order to detect a cross hallway, several assumptions about potential hallways must be made. This allows the constraint of the problem to a simpler one. First, the hallway can be assumed to be bounded by the same type or construction of walls that define the main hallway or thoroughfare. This is important, as it means that the RANSAC algorithm can be tuned for a single set of factors. This simplifies both the algorithm by ensuring the RANSAC algorithm does not require retuning. This is a considered a valid assumption because it is thought unlikely that the cross hallway be of entirely different construction. Also, the cross hallway is assumed to be at least partially occluded or blocked from vision by the wall of the main hallway. If it is known or

assumed that at least part of the wall is blocked from view, the hallway detection algorithm can use this information to better discriminate cross hallways. Finally, the cross hallways to be detected are assumed to be at least nearly perpendicular to the main hallway. As with the previous assumption, this information can be used to further discriminate cross hallways from other features in the scene.

The cross hallway detection algorithm developed uses the three assumptions above to develop three criteria. The first is that the cross hallway be identified as a plane in the scene. This plane can then be found by the RANSAC algorithm and identified as a candidate for a cross hallway.

Second, any potential detected cross hallway, if it is partially occluded, will change in size as the camera approaches it. Other planes in the image will most likely either remain the same size from frame to frame or change unreliably. Cross hallways should grow from frame to frame. The original algorithm implemented a test to ensure identified planes were growing before identifying them as cross hallways.

Another criteria for cross hallway detection was that the planes detected had to be near parallel to the axis of the hallway. This would ensure that the planes detected represented the boundaries of cross hallways. This criteria is tested by finding the cross product of the plane's unit normal vector and the hallway parallel. If the cross product is small, the plane's normal vector is close to parallel with the hallway's axis. The threshold for acceptance of this is tunable, so cross hallways that are not exactly parallel may be identified as well.

An additional criteria was included primarily for the purposes of computation. In order to avoid identifying extremely small planes as hallways, a size threshold was also included. This represents another tunable parameter of the algorithm. This size threshold was initially based on size information from the simulation, but it was adjusted based on data from the real hallway. This threshold was not included for performance, but

Chapter 4.3.3 includes analysis on how this threshold was instrumental in removing false detections from results.

Finally, the RANSAC algorithm was run a certain number of times. This number could be set by the user. If the number was not set, the algorithm would run until there were not enough points in the image to reliably identify a plane. However, this would result in large amounts of excess computation for very little benefit. This number was originally set to five, but was soon after raised to eight when it was determined that the potential cross hallways were sometimes not detected until after some erroneous planes. This was most often the case when the cross hallway was mostly occluded by its opposing wall and the plane was still small.

The result was an algorithm with the tunable parameters of hallway size and the threshold for perpendicularity. This portion of the algorithm then stores the planar information. Then, the second half of the algorithm works to identify specific planes as potential cross hallways. The algorithm checks planes from the current frame of the image and sees if they have been identified in previous frames. If they have, a counter is increased that represents how many frames the plane has appeared in. If the counter reaches a sufficient number, the plane is flagged as a potential cross hallway, plotted, and its information is returned to the workspace. The algorithm is shown in Pseudocode 4.

3.8 Hallway Localization

Also useful for navigation is localization within the hallway. Localization is the determination of relative position in an environment. This differs from navigation in that is is of smaller scope. To develop this algorithm, many of the same assumptions used in the hallway detection algorithm were used. The Manhattan world assumption is assumed to hold, and is relied upon more heavily for this algorithm. It is assumed that there exist two parallel walls and a floor and ceiling that are parallel. These walls do not need to be

Pseudocode 4 Simplified Hallway Detection Algorithm

```
for index = 1 → Frames
    Planes = RANSAC(image(index))
    for Current = 1 → size(Planes)
        this_plane = Planes(Current)
        cross = this_plane.Normal / ||this_plane.Normal|| × Hallway_Major_Axis
        if norm(cross) < Perpendicular_Threshold
          Plane_size = mesh_size(this_plane)
          if Plane_Size > Size_Threshold
            Store_Plane(this_plane)
          end
        end
        for index2 = 1 → size(Store_Planes)
            count ← 0
            for index3 = 1 → size(Stored_Planes) − index2
                if location(Stored_Planes(index2)) = location(Stored_Planes(index3))
                  count ← count + 1
                end
            end
            if count > count_threshold
              return   Stored_Planes(index2)
            end
        end
    end
```

strictly perpendicular, and the threshold for these planes being parallel can be set by the user.

By either using data from the image itself or information determined in the simulation of the scene, the rotation matrix of the vehicle is used to transform the hallway out of the body frame. The resulting Cartesian coordinates can be used to more simply determine the vehicle's position in the hall.

Utilizing the assumption mention above, the simulation first attempts to find the two largest sets of parallel planes. These are most often the walls and the ceiling in a Manhattan world, assuming the vehicle or camera is not too close to another object. The minimum distance to each of these planes is found. Because the planes were rotated out of the body frame, the coordinate of the intersection can be used to determine the position of the vehicle.

Because the field of view of the camera is limited, distance to the measured points of these planes would be inaccurate. Thus, infinite three-dimensional planes are created that have the same properties as the measured planes. The vehicle's distance from these planes is measured and recorded.

The algorithm is shown below. It first cycles through each identified plane, starting with the smallest ones and progressing to larger planes. This plane is checked against other identified planes. If the two planes are parallel to each other and the nominal floor or wall, the coordinates are logged. These are used to determine width and height of the hallway and the relative position of the vehicle in the hallway.

Pseudocode 5 Simplified Hallway Localization Algorithm

```
for index = 1 → Frames
    Planes = RANSAC(image(index))
    for Current = size(Planes) → 2
        for Second = (Current − 1) → 1
            if areParallel(Planes(Current), Planes(Second))
              if areParallel(Planes(Current), LEFTWALL)
                pt1 = memberOf(Planes(Current))
                pt2 = memberOf(Planes(Second))
                x_coords = [pt1(1)pt2(1)];
                width = max(x_coords) − min(x_coords);
              elsif areParallel(Planes(Current), TOPWALL)
                    pt1 = memberOf(Planes(Current))
                    pt2 = memberOf(Planes(Second))
                    z_coords = [pt1(3)pt2(3)];
                    height = max(z_coords) − min(z_coords);
              end
            end
        end
    end
```

4 Results

This chapter focuses on the results of the research. The simulation's performance is analyzed and verified. The use of the algorithm for cross-hallway detection is discussed. The effectiveness of the position-finding algorithm is examined.

4.1 Noise Models

The noise of the SR4000 has several parts that can be easily modeled. First the error contains a bias that increases with range. To find this bias, images were taken of featureless walls at varying distances from one meter to three meters at half-meter intervals. The range error was measured by comparing the expected range for each azimuth and elevation against the actual range. The mean error was found to determine the bias. Over 100 images were utilized for each depth to ensure that the noise was reliably measured and fully characterized. Experimentally, it was determined that the noise grew linearly, and the coefficient of this bias was dtermined to be 0.0244 meters/meter. This compares favorably with other experimental data that suggests the bias grows with a coefficient of 0.026 m/m [3].

Another simply modeled source of error is the range uncertainty that grows with range. As detailed above in Section 2.4.4, the standard deviation of the noise grows with the square of range. The noise characteristics we determined using over 200 images from the camera to ensure the noise was well represented. The range noise standard deviation and mean were plotted against both the angle from the center of the image and the range measurement. In the case of the noise standard deviation, it was simpler to first determine the description of the noise's dependence on the angle from the center. A model for the standard deviation was found by basic quadratic regression. Quadratic regression was used because the basic noise model in Equation 2.8 suggests the noise strength increases

with the square of range. Because Gaussian noise is additive, this function was then subtracted from the error and the regression was performed with respect to the range.

Finally, the noise that changes with reflectivity could be modeled, but was neglected. Including this model would require both a much more complex simulation and far more complicated data analysis using the camera. Using values from prior research into the the effect of reflectivity, this noise was included in the simulation as additive noise [1].

4.2 Simulation Verification

The simulation's performance must be verified to ensure its validity. In order to be valid, the simulation had to meet several requirements.

4.2.1 Simulation Noise Model. The noise measured in Chapter 3 was determined to be varying with the square of range. Figure 4.4.1a displays both the range error standard deviation as a function of the angle from the center and the regressed model for the standard deviation. The model used for the standard deviation is given by

$$\sigma \propto 0.498\theta^2 - 0.296\theta + 0.045 \tag{4.1}$$

where θ represents the angle from the center. This portion of the noise was then removed and a model was found in a similar fashion for the range-dependent error. The result of the regression was given by

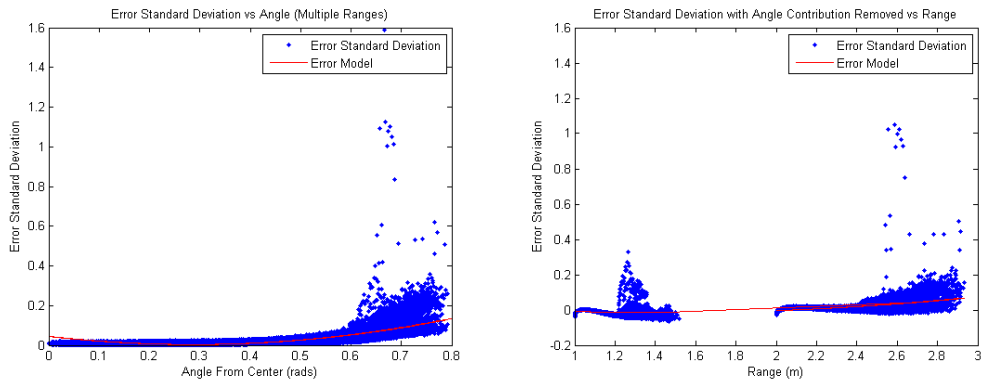
$$\sigma \propto 0.037R^2 - 0.106R + 0.061 \tag{4.2}$$

This regression and the range data can be seen plotted in Figure 4.4.1b. The error mean was determined in the same way, but it was simpler to determine the range-dependent error mean first in this case. The models for the mean are shown by

$$\mu \propto 0.149R^2 - 0.324R + 0.194 \tag{4.3}$$

$$\mu \propto 0.325\theta^2 - 0.120\theta - 0.016 \tag{4.4}$$

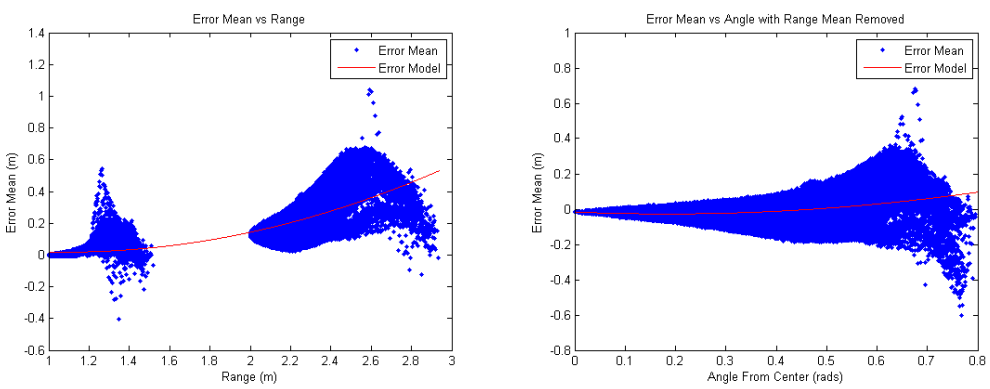
and the plotted data and models are shown in Figure 4.2. The standard deviation and mean are both highly variable, but inspection of error at different pixels suggests the error can be simply modeled by one or two Gaussian functions.



(a) Error σ as a Function of Angle

(b) Error σ as a Function of Range

Figure 4.1: Error Standard Deviation Models



(a) Error μ as a Function of Range

(b) Error μ as a Function of Angle

Figure 4.2: Error Mean Models

The simulation's noise model was verified against real data to ensure that it is a good representation of the uncertainty inherent in the sensor's measurements. Consider Figure 4.3, which, by inspection, seems to have similar error properties to that of the captured image. However, this provides no metrics into the validity of this claim. Thus, in order to verify the noise model, an ensemble of noisy images was generated and the error statistics of this ensemble compared to that of the measured data.

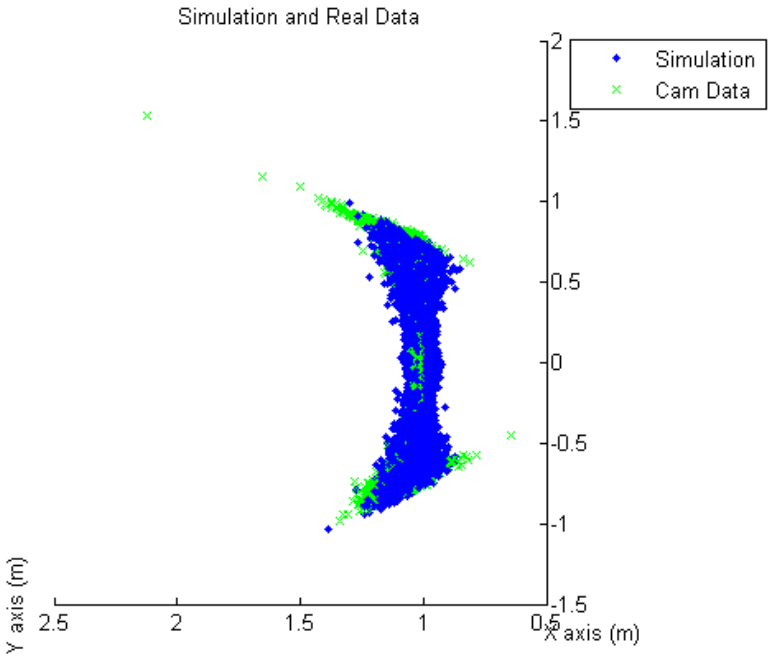
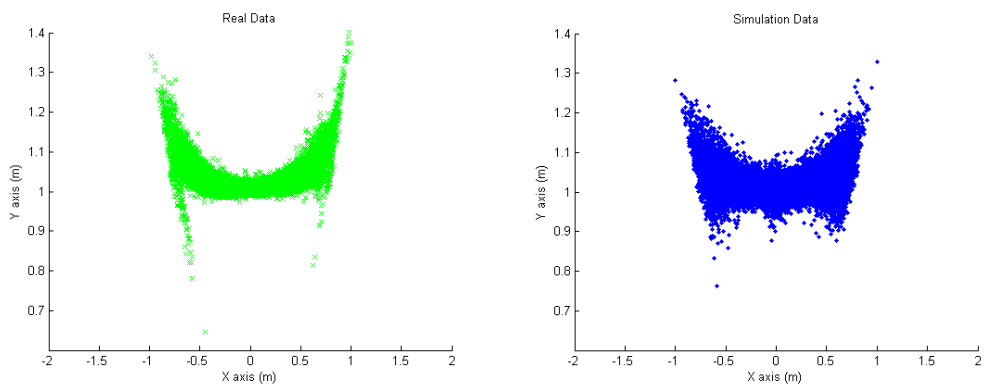


Figure 4.3: Simulation Data with Noise and Measured Data

The noise generated by the simulation has error characteristics that can be measured most accurately through Monte Carlo analysis. This ensures that the noise models' characteristics are fully shown. In order to accurately test the simulation noise, at least thirty examples are needed. This ensures the probability density functions of the noise can be reasonably characterized and compared to that of the actual data.

Thirty scans of a wall one meter away were generated. The ranges to each pixel were compared to ideal ranges generated by the simulation with no noise. The same was done for the experimental data. The standard deviation and mean of the error represent the best way to model the noise and compare it to real data. Sample scans from both the real and the simulated data are shown in Figure 4.4. By inspection, the simulation seems to correctly model the noise.



(a) Real Data

(b) Simulation

Figure 4.4: Samples of Real and Simulated Output

The noise statistics of the simulation were found to be extremely similar to those of the real-world data. The mean and standard deviation of noise at each pixel closely followed the model used. Based on the assumption of Gaussian noise, this implies the noise is correctly described by the model. For this reason, it can be assumed that the noise model used was acceptable.

4.3 Hallway Detection Verification

The hallway detection algorithm was tested first on simulation data of a hallway in order to tune the algorithm and determine the ideal values for the tuning parameters to

ensure efficiency. The parameters of the RANSAC algorithm were a special focus, as these have a large impact both on the performance and the computation of the result. The hallway detection algorithm works exceedingly well on the simulated data. In some cases, no retuning was necessary before the algorithm was used with real-time data.

4.3.1 Detection in Simulated Data. When applied to simulation data, the hallway detection algorithm identified and localized the hallway in the body frame of the vehicle. The result was an algorithm that could find both perpendicular hallways and hallways that were not perpendicular but fell with the threshold set by the simulation. These hallways were tracked from frame to frame.

The first simulation test was completed with the simulated camera pointed directly down the hallway. The simulated results output can be found in Appendix A, which demonstrates the MATLAB output of the algorithm. This output shows several features of the algorithm. It is shown that it takes several frames before the algorithm determines that the cross hallway is present. This is by design, as it ensures the same plane is detected several times before being flagged as indicative of a cross pathway. Also, as the camera passes the hall completely, the hallway is no longer detected. Figure 4.5 depicts the simulation output before any detection. Figure 4.6 displays the simulation output when a potential cross hallway has been detected, before it has been positively identified. The red points in this image are the potential cross hallway's far wall. Figure 4.7 shows a sample frame in which a cross hallway is detected. from this data processing. The green points are the detected hallway. The blue points are those not detected as being a member of the hallway.

The second test simulated the camera being oriented differently. As opposed to being perfectly parallel with the major axis of the hallway, the camera was turned fifteen degrees to the left. The rotation matrix for the vehicle's orientation was approximated and used to

determine the vector parallel to the hallway. The result was a successful detection of the cross hallway. When compared with the results for the first test, the result with the camera turned was far better. The hall was detected much sooner and with more certainty. However, any hall on the other side of the passageway would likely have been ignored.

Tests were completed for several different rotations and several different orientations. Notably, hallways were simulated of AFIT hallway size and of 3m square size, where the height and width of every hallway is uniform. Each simulation differed in speed of the vehicle, position of the camera in the hallway, and orientation of the camera. It was noted that the hallway of AFIT size had much shorter detection ranges in every case.

Several simulations were run with varying headings, which had the greatest impact on detection besides hallway size. All detection parameters were constant for every run. In every case, the hallway was detected correctly. Differences were noted between the detections in each case, especially in the detection range. The cross hallway was detected at different ranges in every case, but it was most dependent on hallway size. The hallway of AFIT proportions resulted in much shorter range detections. The detection range of these situations is listed in Tables 4.1 and 4.2. One aspect this data shows is that increasing the yaw decreases the detection range, which differs from results of real data analysis in Section 4.3.2. This is the result of geometry, as when the camera is turned, more of the cross hallway is occluded by the hallway walls.

Table 4.1: Angle vs. Detection Range for Hall with AFIT size

Yaw Angle (°)	Detection Range (m)
45	4.1731
30	4.0879
15	4.407
0	4.5033

Table 4.2: Angle vs. Detection Range for Hall of 3m Width and Height

Angle (°)	Detection Range (m)
45	7.3873
30	7.4546
15	7.56
0	7.6702

4.3.2 Detection in Real Data. Essentially the same process was used to generate the hallway detection algorithm for the real data. The algorithm was applied to real data and the results were recorded. One point that became immediately apparent is that data in this environment was far more cluttered than in the situations used for the noise modeling, despite efforts to avoid sources of noise. All of the scans were taken when fluorescent lighting was at a minimum, and several datasets were taken with the explicit intent of avoiding surfaces that tend to be noisy in flash LADAR. In many cases where range ambiguity should not have been encountered, there were pixels whose range was returned as a minimal value. Also, in places where there was very little or no return to the camera, such as the center of the hallway, the result was almost always a seemingly random value that "floated" in the middle of the hallway. These effects are shown in the Figures 4.8(a) and (b). All of the points in these images are points measured by the LADAR in a basic hallway. The black lines denote the walls. The left image shows points in green whose range not only exceeds the ambiguous range of the camera, but exist where there were walls at much shorter ranges. The right image shows the floating pixels in red. These are due to both the range ambiguity and noise to the camera. When the camera receives no returns from surfaces in the distance, the noise represents the only return. Thus, surfaces are detected when there are none.

The real hallway differed from simulated hallways in several ways. The were differences in returns due to differing reflectivity of surfaces, introducing an unmodeled source of noise. For instance, wall hangings and floor tiles all presented a different surface to the camera, with different returns. Also, the angle of incidence impacted the range error. As the camera scanned walls that were farther away, the very large angle of incidence resulted in increased range error. Also, there existed other objects in the scene. For instance, chairs were included in some scans of the hallway that made detecting the cross hallway more difficult. These objects also ostensibly create more floating pixels and noisy measurements.

Due to the limited size of the passages and hallways in question, the range ambiguity was largely left in the image data. A rudimentary algorithm for range ambiguity resolution that relied on basic edge detection was developed. However, while many ambiguous points were placed at the correct range, many others were not.

One troublesome area in the real data was the cross hallway itself. At the top and bottom of this region, the range of the pixels measured was well beyond any surface in that direction. Figure 4.9 shows these points circled in red, while the cross hallway on the left of the hall is shown in green. The black lines show where the walls are. The blue points in the middle of the image are the roof and ceiling points. These are more numerous than those shown in Figure 4.8(a), suggesting the existence of another error source. These could be the result of multipath error due to reflections in the hallway, due to the fact that almost all of these points have a positive range error. This may be a limitation on the use of LADAR on indoor areas, especially areas where reflectivity results in large amounts of multipath error.

The first test case was a hallway with very little clutter in the image and primarily flat, plain walls. This allows for the algorithm to work in a best-case scenario while detecting the cross hallways. This case had no false detections. The cross hallway was

correctly identified and tracked for multiple frames. Figure 4.10 shows an example of the cross hallway, which is on the right side of the image. The green points are those points identified as members of the cross hallway. The black lines are the walls.

The second test case was the same hallway, but with items blocking the view of the sensor and a larger number of reflective surfaces. Also, there existed clutter in the cross hallway that reduced the already low number of points included These highly reflective surfaces were noted to produce large amounts of error in preliminary scans of this and similar errors. However, despite the expectation that this would be the worst dataset, the detection algorithm correctly identified the cross hallway.

In many cases, the plane-finding algorithm also mistakenly adds points from the walls to this plane, artificially increasing its size. These points, while they do increase the size, only increase the surface area of a mesh created from the points. Because the erroneously included points do not represent a plane or volume when this mesh is created, the result does not increase the plane's size noticeably. Figure 4.11 shows two detections. In the left image, the red points extend across the hallway for this reason. The same is true for the right image in the green points. Also, the location of this mesh is not changed substantially, as many more points are found in the detected hallway.

The algorithm was first run on this dataset with the settings originally found to work best on the data in the simulation. Without any tuning, the correct cross hallway was identified in three separate frames. However, this dataset made it apparent that the greatest weakness to this method is the fact that the narrow nature of this hallway makes the occlusion of the cross hallway the limiting factor. Also, in this dataset, the existence of a large number of points that were measured in the wrong ambiguity bracket and a large object that was nearly with this ambiguity in the center of the image led to the false positive identification of a plane. Once the object in the foreground was passed, however, the correct cross hallway was identified until it, too, was passed. These two different

hallway detections can be seen in Figure 4.11. The left image shows the erroneous detection of a trashcan. The red points are what the algorithm defined as a cross hallway. The correct hallway is shown as a green line. In the right image, the correct plane is detected as green points. In both images, the black lines are the true walls.

A very small number of points are actually identified with the cross hallway, making detection of the plane difficult. There are several ways this problem could be mitigated. First, if the side of the passageway where the cross hallway is expected is known, traversing the opposite side of the hall could ensure that the cross hallway is less occluded. However, this is somewhat difficult with no *a priori* information about the environment. Another way to mitigate the effect of this problem is proper sensor design and selection. A sensor with a higher angular resolution, or higher angular resolution at the edges of the sensor than in the center would ensure that more points from the cross hallway were included. This would serve an additional benefit for navigation, if it is assumed that features usable for navigation are more likely to be found at the edge of a hallway than in the center. Finally, an algorithm to exclude statistical outliers in the data would improve the chances that the cross hallway is detected.

In order to verify that the ability of the algorithm to detect halls in these datasets is due to the combination of hall geometry and sensor characteristics, a similar hallway was simulated, with dimensions identical to those of the real hall. The hallway detection algorithm was run on this data using the same settings for size threshold and the RANSAC algorithm. The hallway was detected at 4.99 meters away in the y-axis, the axis that is longitudinal to the camera. In the real data from the hallway, the hallway was detected at a similar distance of about 4.6 meters away in the y-axis. However, on simulated hallways where the width is 3 meters wide and the cross hallway is also 3 meters wide, the hallways were found 11 meters away. When the narrow hallway was simulated with a sensor that had 200 pixels in the horizontal, the range was not increased unless the size threshold was

adjusted. However, this could prove problematic in real data, where objects in the scene could approach the size of a hallway. In light of the existence of noise the simulation does not model, this data is taken to indicate that the hallway geometries impact the detection algorithm.

Another dataset was recorded in order to potentially avoid the floating pixels in the middle of the hallway by turning the camera to the side. Also, it was hoped that by decreasing the angle of incidence with the wall at the edge of the image, flying pixels in the vicinity of the boundaries would be less common. However, the opposite was true. Because the edge of the sensor's field of view, the range ambiguity resulted in a mass of pixels on the left side of the image that made it impossible to identify any cross hallways. This dataset also used a smaller cross hallway wall, which, compounded with the above problems resulted in an image that had fewer detections than the other scenes. The "cross hallway" involved was only an alcove, but served as a good proxy for what is assumed to be the form of cross hallways in situations such as caves. Despite this, the algorithm still correctly identified the alcove. The camera was required to be much closer to the alcove to identify it, but it was identified for a greater distance than in either of the two other cases. Figure 4.12 shows a sample detection for this dataset. As before, green is the identified cross hallway. The points farther away from the rest of the plane on the right side of the figure are range measurements taken through a window that happened to be coplanar with the alcove.

The fact that the hallway was detected for a longer duration raises an interesting problem. There exists a tradeoff between early detection and sustained tracking of a cross hallway, which compete with a field of view that includes both sides of the passageway. In the first dataset, the hallway was detected when it was 3.675 meters away. The third dataset, which had about 30° of rotation, detected the alcove at 4.38 meters away. The second dataset had its first good detection at 4.02 meters away. However, the third dataset

contained exactly no information about cross hallways on the other side of the hallway. Also, the third dataset, which included rotation, had the surface tracked for a full half-meter longer than either of the other two scenarios. It can be assumed, however, that there exists an optimal angle for different applications.

The output positions of these cross hallways could be extremely useful in navigation. The position solutions for detected hallways were returned in the camera body frame. These positions represented the center of the detected plane that was part of the cross hallway. While the positions varied due to the random nature of the RANSAC algorithm, they were always on the right plane and the correct side of the hallway. This data could be the basis for development of control inputs for a vehicle.

4.3.3 Improving Detection Results. In order to ensure a minimum of false positives and a maximum of good detections, the hall detection and RANSAC algorithms were tuned to ensure that the best results were found.

The RANSAC algorithm settings were adjusted in several ways. First, by increasing ϵ, fewer planes were detected. This resulted in fewer false positives and stopped the RANSAC algorithm from running more times than necessary. However, despite the fewer calls to the RANSAC algorithm, each call took longer and the processing was overall much longer. Changing P_{Inlier} changed the results in several ways. Increasing the parameter above the default value of 0.99 also increased the number of points in planes found in each run of the RANSAC algorithm. This resulted in more points being removed before each successive run. Thus, the process was much faster. However, the calculated locations and directions of the directed planes were far more corrupted with noise. This, then, required that some of the parameters of the hallway detection algorithm be relaxed in order to correctly identify the planes. In several cases, the hallway size threshold became completely irrelevant because of the number of planes that reached the hallway size required to be counted. Also, because the sizes of the resulting planes was so great, the

algorithm was unable to determine which plane was unique between frames. Many planes were assumed to be identical when they were not. Decreasing the parameter P_{Inlier} below about 0.92 resulted in computation times that were far longer and many planes that existed in the image were not detected at all, much less correctly.

Changing the parameter σ was the most reasonable. By increasing this parameter beyond the measured mean of the sensor error, planes were detected quickly. However, contrary to the case when changing P_{Inlier}, the planes were more reliably identified. Because the planes had enough member points, increasing σ only increased the number of points identified by the algorithm. These points contributed to an increased uncertainty in planar coordinates and directions, but over multiple frames, the effect was minimal. Decreasing σ decreased the number of points in each plane and increased the computation time, but only marginally. Changing this parameter did not affect whether or not the critical planes were detected or not.

Adjusting the minimum repetitions of the RANSAC algorithm forced a computational floor on each frame. This increased computation provided minimal improvement in plane detection. In fact, no difference was noticed in the planes detected with or without the minimum iterations. Instead, the same results were output, but the time necessary to compute it was about 120%. In light of this, it was deemed far better to remove any minimum number of iterations.

Often, changing the settings of the RANSAC algorithm would not result in a better solution. Making the algorithm more lenient in finding planes usually resulted in ambiguous points being included to a greater degree. Making the RANSAC process more strictly define planes sometimes resulted in fewer false positives, but it also resulted fewer positive detections. In this way, too, there is a tradeoff. It was determined that the ideal values for the RANSAC algorithm parameters were very different from the default values. ϵ was set at 1×10^{-6}, which is three orders of magnitude smaller than the default. The

value for σ was taken from early indications of noise standard deviation, and had a value of 0.002075 meters/meter. This value was actually nearly identical to the default value for the algorithm. Finally, the value for P_{Inlier} was set to 0.96, much smaller than the default of 0.99. These values produced the most reliable plane detection.

The hallway detection algorithm also needed to be correctly tuned to detect planes. The size parameter was the most important parameter in tuning this algorithm. For each hallway, the size of the cross hallway differed based on several things. The second dataset had clutter occluding the cross hallway and a trash can blocking the view of much of the camera. The size threshold on this dataset was the most important in improving the results. Without changing the threshold, the trash can's side was identified as a cross hallway. However, this was fixed by judiciously setting the size threshold to larger than the side of the trash can. The result was a complete removal of false positives on the second dataset.

It was noted after using the detection algorithm on real data that the size of the cross hallway plane does not grow monotonically. Also, the code utilized to measure the area of each plane was unreliable. Thus, the algorithm was adjusted to exclude only those surfaces that were shrinking quickly, as opposed to all those surfaces that did not grow. The result was a more robust algorithm. This size unreliability was due to the fact that points across the hallway are included in the plane detection of the cross hallway. This problem did not occur in simulated data, as the walls and ceiling were always detected first and as such were not included in detection of the cross hallway. However, this did not present a large problem, as the size of the cross hallway was not significantly impacted by the added points.

Also important was the threshold for perpendicularity. Because of the noise and range ambiguity, many of the detected planes were not as perfectly perpendicular to the main hallway as they could be. In order to reduce the number of false positives, the threshold was increased to 10° of difference between the hall's axis and the normal of

potential cross hallways. This parameter had a large impact on efficiency of the algorithm, but as long as it was not increased past about 45°, the result was essentially the same. If increased beyond this number, planes detected entirely in the noisy points were identified as cross hallways.

4.3.4 Hallway Detection of Non-perpendicular Hallways. It is useful to discern whether or not this algorithm can be extended to hallways that are not perpendicular with the path of vehicle travel. This would be the case outside of a Manhattan world, which describes many real-world environments. However, without a real-world place to test this, it must be tested in simulation. A hallway was created with a cross hallway that was off-axis by 30 degrees. An idealized view of this hallway is shown in Figure 4.13. The cross hallway is found at the top of the image. However, this hallway should be tested using the noise model included in the simulation. In order to obtain the best result, the range ambiguity is left out.

By changing the thresholds of the detection algorithm, especially the threshold for perpendicularity, the hallway could in fact be detected in the noisy data. Because cross hallways could be detected in data with this threshold set at this level or higher in real data, it can be assumed that a similar hallway in real data could be identified.

Also, in this case, identifying the actually hallway was somewhat simpler. The size threshold of the algorithm could be set very high. This is because the angle of the hallway results in a very large detected plane. Thus, with a large size threshold, the hallway could be detected without any false positives.

If there is *a priori* knowledge of the cross hallway, the threshold could be set against a known direction for the plane. In such a case, the hallway detection algorithm would ostensibly perform as well as with perpendicular halls.

4.4 Range Ambiguity Removal

Removing the ambiguity, even unreliably and with a rudimentary algorithm, improved the results of the detection algorithm greatly. Removing the ambiguity is almost necessary for proper analysis of the data. This research did not focus on a robust method for removing this ambiguity, but an algorithm was developed that removed a considerable amount of the ambiguous points in some of the datasets.

The removal of this ambiguity greatly improved the results of the hallway detection. Before ambiguity removal, the first and second datasets had one hallway detection between them, and it was only identified as a potential cross hallway as opposed to a confirmed cross hallway. Because many of the points affected by the ambiguity were assumed to be in the center of the hallway, planes were often detected there where none existed.

Basic ambiguity resolution on the principle of edge detection moved many of the ambiguous points to a different ambiguity bracket. Figure 4.14 shows an example of the ambiguity resolution. Many of the points, shown in red, were incorrectly measured as very near the sensor. These were converted to range, increased by one ambiguity bracket, and transformed back into Cartesian coordinates. The green points in the image are those points that were moved into a different ambiguity bracket. In each case, about 4% of the total points are moved into a different ambiguity bracket. The result was an improved range image with fewer points in the closest ambiguous bracket.

The weakness of the range ambiguity resolution was the fact that it often did not place points beyond the second ambiguity bracket. Thus, anything measured beyond ten meters was often ignored or unmeasured. This was not a great limitation, as the number of points measured beyond ten meters was either low, or the points did not impact hallway detection.

4.5 Position Finding

The position-finding algorithm measured the position of the vehicle in the hallway. Both the height and width of the hallway were measured. The lateral and vertical position were also measured.

Simulated data, even with noise included, always resulted in a near-perfect estimation of hallway size and relative position. This is most likely due to lack of clutter and outside noise sources. The hallway width in the simulation was three meters, and the position-finding algorithm measured the hall at 2.98 meters with a standard deviation of only 0.011 meters. This certainty suggests that the algorithm itself is sound for Manhattan world environment. The height of the hallway was similarly accurately measured.

The vehicle's horizontal and vertical positions were also measured correctly within three tenths of a meter in both cases. Assuming the vehicle was in the exact center of the hallway, the horizontal error was always within 0.25 meters. The mean error was only 0.0327 meters. The average vertical position across both datasets with useful position finding information was in error by a very small 2 millimeters. The maximum vertical error was only 0.1077 meters. The standard deviation of this measure was only 0.0632 meters. This data suggests that the position-finding algorithm was successful in real data.

The hallway was measured at two and a half meters in width. The position-finding algorithm provided several estimations of the hall width. For the first dataset, the result was underestimated with a mean of 2.48 meters, with a standard deviation of 0.154 meters. The second dataset resulted in a mean of 2.497 meters with a standard deviation of 0.119 meters. The mean across both datasets for hall width was 2.500 meters, with a combined standard deviation of only 0.133 meters. The third dataset returned a width measurement, but only one wall was truly measured. The other wall that the camera measured was made up of points the range ambiguity algorithm had not removed.

This information suggests that the SR4000 can be used to measure its surroundings in a dynamic setting. Despite the relatively large standard deviation of this data, the hallway's width was found. Also, the horizontal position of the vehicle was consistently in the middle of the hallway, as it was intended to be.

The height of the hall is also measured. The actual height of the hall was nine feet, or about 2.74 meters. However, the algorithm underestimated the hall height in every case. For the first dataset, the estimated height had a mean of 2.648 meters and a standard deviation of 0.110 meters. The second dataset had a measured mean of 2.644 meters and a measured standard deviation of 0.084 meters. These are low estimates because of the nature of LADAR measurements in corners. As mentioned in Chapter 2, the range estimates in corners are lower than the actual value. This was noticed more strongly in the ceiling and floor because these had fewer points measured. The camera's larger horizontal field of view and the fact that the hallway was taller than it was wide resulted in many more points being measured on the walls than on the floor or ceiling. In addition, further noise was introduced because the floor was highly reflective, increasing the probability of range error. Also, the ceiling had many highly reflective features that would incur error in the range data, namely fluorescent light fixtures and varying surfaces of varying reflectivity. Thus, it is to be expected that there be more error in estimating hall height and relative vertical position.

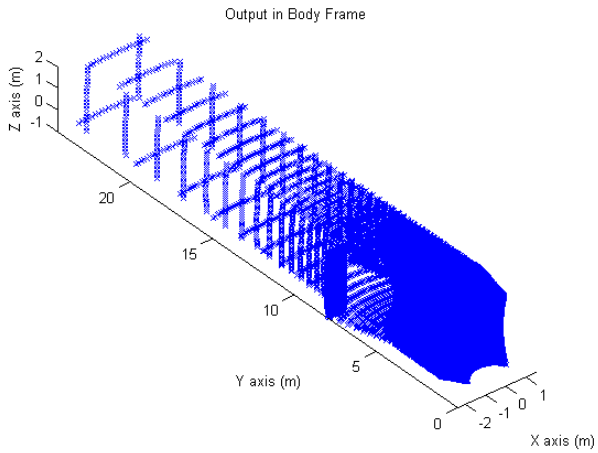


Figure 4.5: Simulation Data with No Hallway Detected

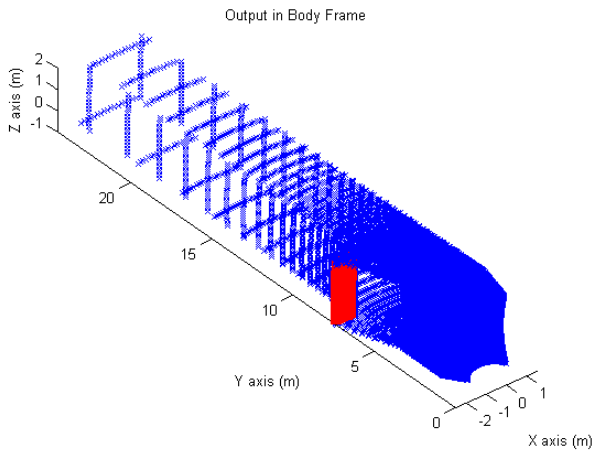


Figure 4.6: Simulation Data with Potential Hallway Detected

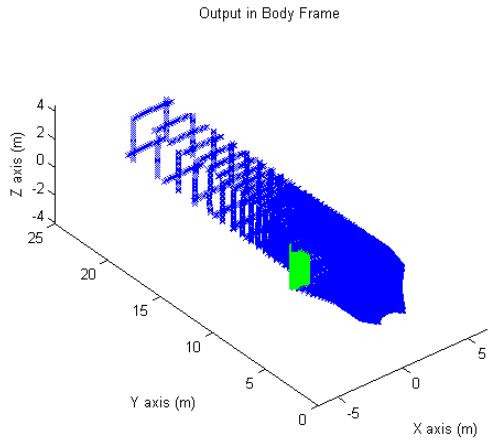
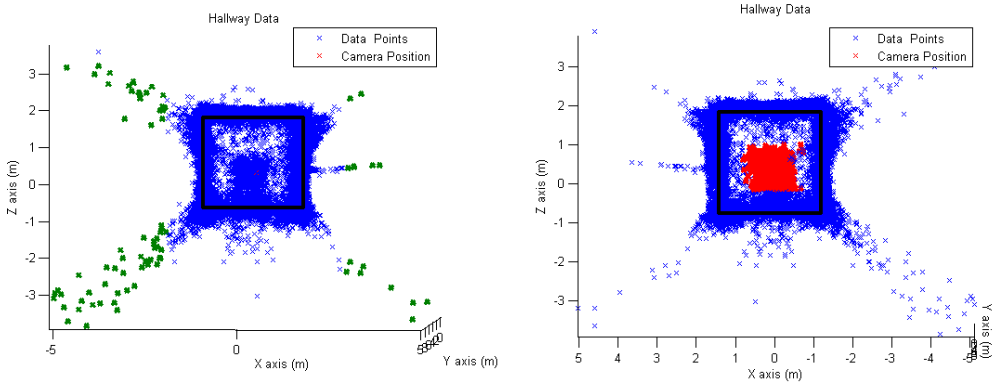


Figure 4.7: Simulation Data with Hallway Detected



(a) Points Detected Erroneously

(b) Floating Pixels

Figure 4.8: Samples of Real Data Errors

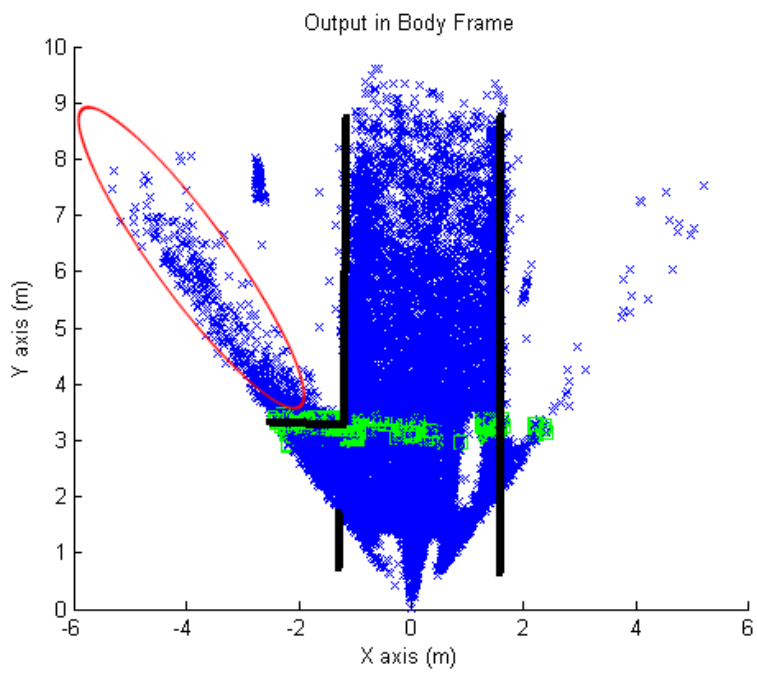


Figure 4.9: Hallway Detected with Erroneous Ranges

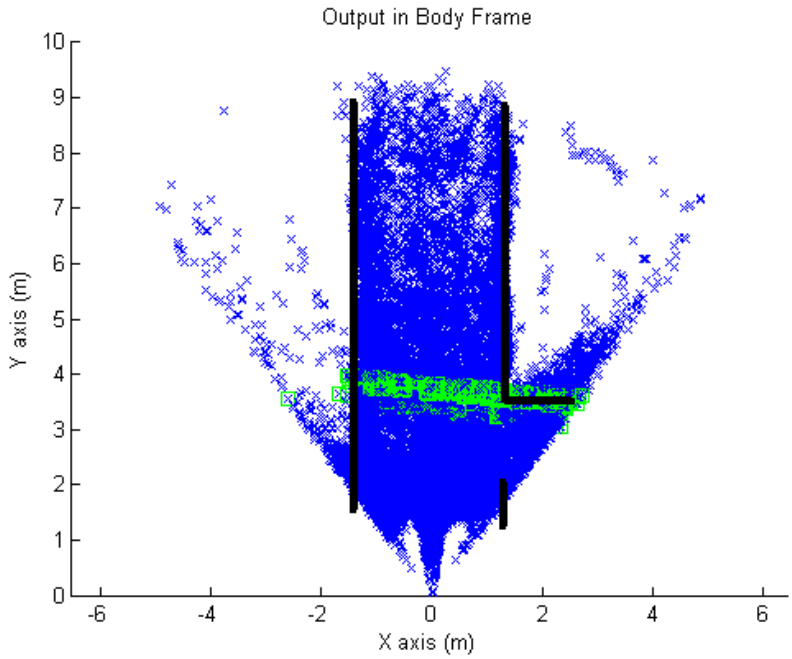
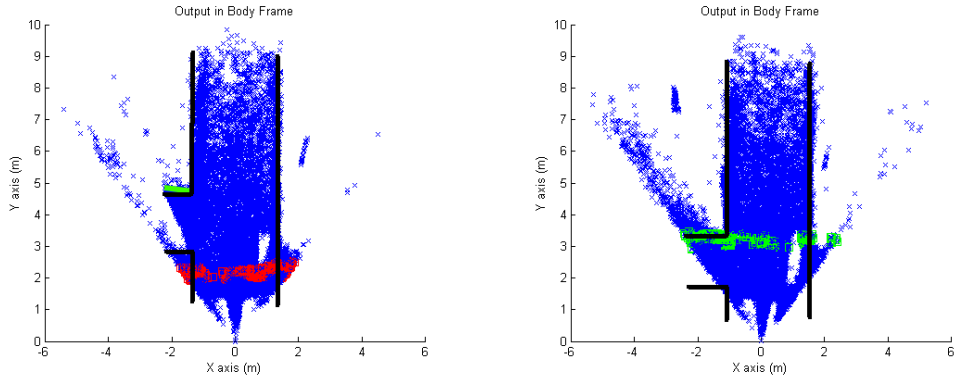


Figure 4.10: First Dataset Frame with Hallway Detected



(a) Erroneous Detection

(b) True Detection

Figure 4.11: Samples of Incorrect and Correct Output

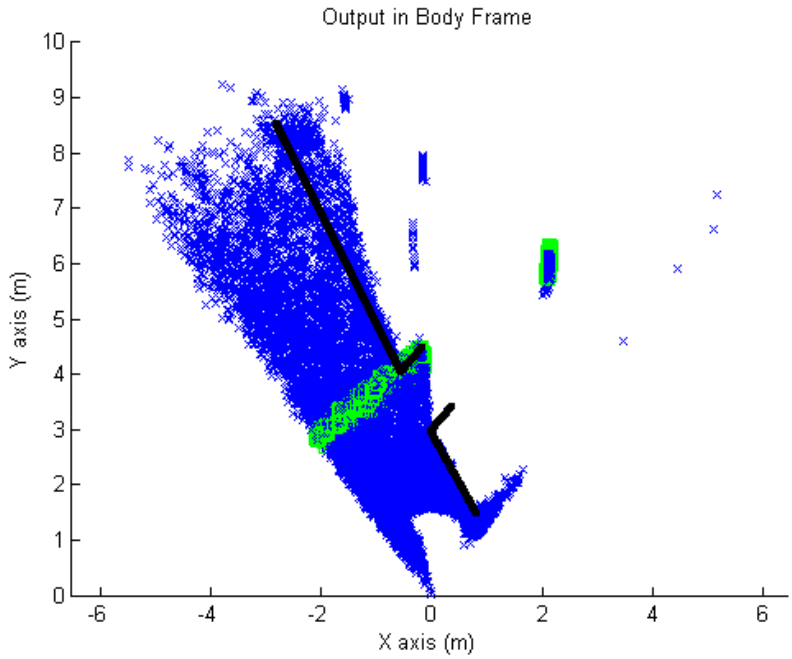


Figure 4.12: Simulation Data from Dataset 3 with Hallway Detected

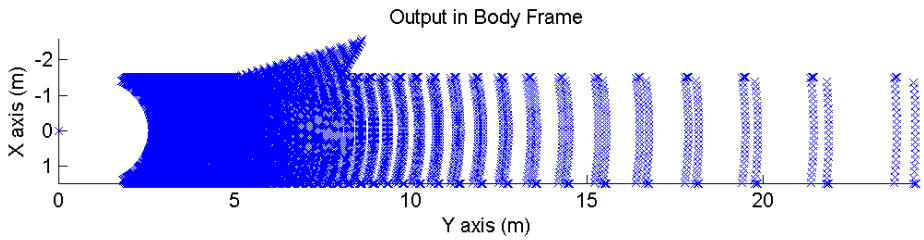


Figure 4.13: Idealized Simulation Output for Hall with 30° Cross Hallway

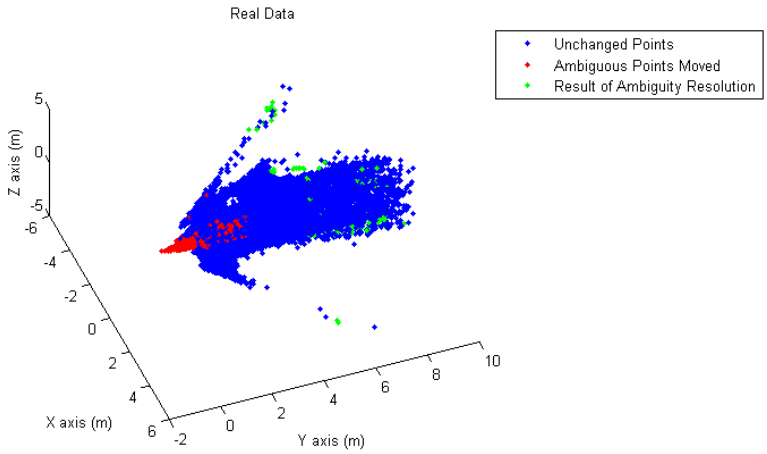


Figure 4.14: Range Ambiguity Removal on Real Data

5 Conclusions

THIS chapter discusses conclusions on the performance and utilization of the algorithms developed. Also, possible future research is discussed. Overall, the simulation and algorithms created were considered successful.

5.1 Simulation Performance

The simulation created to model the LADAR performed well in mimicking the SR4000 output. The simulation created data that was successfully used to design and test the hallway detection and position finding algorithms. The simulation is highly adaptable, allowing for different noise models, sensors, environments, and trajectories. The fact that it is completely three-dimensional means that it can be used for any number of situations.

5.1.1 Simulation Outputs. The simulation output exactly the same information as the camera did. It provided body-frame position of each pixel, and provided them in the same structure as the camera itself. Each pixel had the same angular position in the simulation as in the real camera, and the ranges were simulated in the same way. Noise was added according to the noise model experimentally determined, and it was added as a range rather than as a position error.

The simulation did not include the intensity image. However, utilization of the intensity image was not planned for any of the algorithms utilized. Thus, it was deemed unimportant.

This simulation was found to be a successful approximation of the camera. Its use enable more data processing than would be possible with only real data, and allowed for development of algorithms using far more situations. Thus, it is concluded that the simulation was a success.

5.1.2 Noise Model. The simulation's noise model closely approximated the predictable noise of the LADAR sensor. The range- and angle-induced error were well modeled and included. The result was a simulation that could be used to reliably test algorithms from a stochastically sound viewpoint. For this reason, the noise model is thought to be a successful representation of the noise present in LADAR data.

The model did not include many sources of error, such as reflectivity, floating or flying pixels, or movement smearing. However, several of these sources are very difficult, if not impossible to model. Correcting them may have improved the fidelity of the simulation, but the overall impact is believed to be minimal. For instance, consider the floating pixels that underestimate the range to a corner. The simulation completely ignored this problem. However, this error source, when it was encountered, did not hamper the plane detection in real data. Thus, it can be assumed that modeling it would have been unnecessary computational burden. Thus, ignoring these sources of error is thought to be acceptable for the purposes of this research.

Overall, the noise model was acceptable for this research. While not complete, it encapsulated enough of the error to ensure that the simulated models could be reliably used as a surrogate for real data. Thus, the simulation's noise model was a good approximation of real data.

5.1.3 Computation. The computational burden of the simulation was great, with each frame taking about a minute to create. In this way, the simulation is quite limited. Creation of large datasets with many frames would be prohibitively slow. Also, increasing the resolution or number of points measured would greatly increase the computation time as well. For this reason, the simulation in its current form is best used modeling the flash LADAR systems discussed in this paper, as their low resolution means they can be simulated quickly.

5.2 Hallway Detection Algorithm Performance

The hallway detection algorithm was successful in identifying hallways that fit the Manhattan world assumption of passageways. Their positions and orientations were measured correctly as well, meaning they are features that could be used in navigation. These features could be noisy, off-axis, or measured ambiguously, and the algorithm could find them. This in turn suggests that flash LADAR is an acceptable sensor for finding these cross hallways.

5.2.1 Simulated Performance. The hallway detection algorithm worked in every test on simulated data. The thresholds for cross hallway size and orientation could be reliably set very tightly, close to the simulated value. The hallways were detected in simulation data even when the noise was increased beyond the model.

The simulation allowed for the testing of the algorithm on many non-standard situations, which the algorithm successfully responded to. Small hallways, which were shorter or narrower, were able to be tested with the algorithm. It was able to detect cross hallways in these situations. Also, the detection of cross hallways that were off-axis with the main hallway could not be tested in the real world, but the simulation ensured that the algorithm could find them. Situations with multiple cross hallways were difficult to find in the real world, so the simulation provided data that ensured that the algorithm successfully detected multiple hallways and identified them as unique.

Many errors were ignored in simulation. First, in most tests, the ambiguity was completely ignored. Thus, the point cloud was very clean. No unambiguous points appeared at small ranges. These points sometimes affected plane detection in the real data, so it can be determined that lacking these points improved the results. Also, ignoring many noise sources removed many of the points of data that were essentially useless. For instance, the camera calibration created some points in real data that should have been impossible for the camera to measure, as their range was beyond the camera's

non-ambiguous range. These points were not included in the simulation and thus improved the results.

The performance of the hallway detection algorithm in simulated data was strong enough to allow for the application of the algorithm to real data.

5.2.2 Hall Finding in Real Data. The hallway finding algorithm was ultimately successful in detecting cross hallways in real data. The real data was far noisier and more cluttered than the data produced by the simulation. However, the fact that the detection and RANSAC algorithms were very tunable allowed for the adjustment necessary for correct detection.

Before its removal, the range ambiguity had a great impact on the effectiveness of the algorithm to correctly detect hallways. When it was partially removed, the result was greatly improved. This leads to the conclusion that the range ambiguity must be removed before the real data can be used for navigation or localization. Ideally, a sensor with an unambiguous range beyond features of note would be used. With greater unambiguous range also comes greater range, though, so a trade space would need to be considered.

Clutter in the images had the greatest impact on the algorithm. Light fixtures and drinking fountains introduced large numbers of completely erroneous points. These points were often included in the plane detections and in some conditions produced erroneous results. However, properly tuning the algorithm would result in no false positive detections.

The hall finding algorithm was also successfully applied when the camera was rotated. The hallway detected was detected earlier and longer, at the expense of no vision of the other side of the hall.

5.2.3 Limitations. This algorithm was designed to work under the assumptions of the Manhattan world. Outside of this environment, this algorithm would be less effective.

However, according to research mentioned in Section 3.1.1, the Manhattan world assumption holds relatively well outside of urban or industrialized areas. Thus, while the algorithm would no doubt perform worse outside a hallway or other indoor environment, there exists the possibility it could be successfully applied.

The algorithm itself did poorly when other planes exist in the image that could approximate hallways. On a dataset where a closed alcove was measured, the algorithm detected it as a hallway, despite the fact that it ended after about a meter and a half. However, this is also a result of the sensor itself. The sensor, in this case, did detect the boundary of the hallway until it had been identified as a cross hallway for several frames. Thus, the sensor was completely unable to determine that the hallway had an end.

The algorithm is also extremely slow. Each frame takes about three seconds to process on a capable desktop computer, a time that would not allow for real-time application. The current algorithms are also poorly written for speed. In order to speed up the algorithm, several algorithms would need major code reworking.

The RANSAC algorithm, while it produced a relatively low computational burden in this case, has a computational load that grows exponentially with the addition of more points. Also, while the RANSAC algorithm did not take long, it was the slowest part of the hallway detection algorithm. While this impact can be deferred by using fast adaptations of RANSAC algorithms or parallelization of the RANSAC process, the fact remains that a large number of points will slow the plane detection to a point that would limit its use in real-time situations. However, for the SR4000 and other flash LADAR cameras, their low resolution does not exacerbate this problem.

5.2.4 Improvements. The detection algorithm would benefit from the inclusion of segmentation. As it stood, the RANSAC algorithm often included points in detected planes that were across the hallway or part of another plane. The result was planes whose detected size were larger than they should have been, making proper selection of the

algorithm's size threshold difficult. If the planes were segmented, this problem could be mitigated.

The hallway detection algorithm would be greatly improved with the addition of code to detect the end of an identified hallway. As mentioned above, the detection of closed alcoves could be problematic. Without the ability to determine the boundary of the cross-hallway, false positives could result.

The RANSAC algorithm could be greatly improved in both design and implementation. There exist many fast or efficient alternatives that use the RANSAC algorithm as a base. Also, one of the major advantages of the algorithm is that fact that it can be parallelized, or run on several processors. The RANSAC algorithm could be implemented on a parallel architecture or on a piece of hardware, like an FPGA or an ASIC. These things would greatly increase the speed of the algorithm, to the point where it could potentially be used in real-time.

The hallway detection algorithm as a whole could be made to be more efficient. In many instances, the code used was slow and inefficient, resulting in a slow and inefficient hallway detection algorithm.

5.3 Position-Finding Algorithm

The position-finding algorithm was exceptionally accurate in finding the position of the vehicle in the hallway. This localization could be exceptionally useful in indoor navigation.

5.3.1 Simulated Results. In the simulation, the position-finding algorithm had extremely accurate results. Unsurprisingly, when noise was not included, the position estimates were nearly exact. The only reason they were not exact was the inclusion of cross hallway points while estimating the detection of the hallway wall.

With noise included, the results were also good, with the algorithm again almost exactly determining the position of the simulated camera. These results were expected to be better than real data due to the lack of image clutter.

5.3.2 Real Data Results. The algorithm was also extremely accurate in real data. The horizontal and vertical positions of the vehicle were accurate to within a decimeter on average. This suggests that, in a Manhattan world, this is an extremely viable approach to finding a vehicle's position in a hallway. The extreme accuracy of the algorithm implies that this algorithm is a successful localization tool.

5.3.3 Limitations. This algorithm relies on the fact that the primary hallway walls are the largest planes detected in the images. If this were not the case, as with the dataset with the rotated camera, the result is incorrect. In this dataset, the algorithm attempted to define ambiguously-ranged points as a wall of the hallway. This was obviously incorrect. However, in this same dataset, the ceiling and floor were correctly found.

Also, the output of this algorithm is position relative to the hallway. Outside of this situation, this terminology has little meaning. The horizontal and vertical positions are relative to the hallway walls, which assumes the existence of hallway walls. Thus, this algorithm could not be utilized in an area where there were no ceiling or walls on either side.

5.4 Continuation

There are several ways this research could be extended. Primarily, this research could be extended to navigation systems that utilize LADAR as a sensor.

5.4.1 Estimation of Orientation. In every case where the orientation of the camera was not assumed to be directly down the hall, user input was required to create a direction cosine matrix (DCM) that showed the rotation of the vehicle. Producing this DCM using

the range data would be preferred. If the orientation of the vehicle within the hallway were known, not only would user input be minimized, but the position solution could be used to create a full navigation solution or aiding inputs to a navigation device.

5.4.2 Integration with IMU. Inertial measurement units can be used to provide a full navigation solution. However, the result of such navigation has error that grows over time. In IMUs used in mobile vehicles, this drifting error can be extreme.

The data output from these algorithms is useless if it is not used alongside other navigation tools. If this algorithm were paired with and inertial measurement unit, the position finding algorithm could provide useful data that could be integrated with the inertial data. This could be used to create a navigation solution whose drift is constrained. By constraining the drift, in the vertical and horizontal directions, the IMU's navigation solution could be greatly improved. Because the position finding algorithm is so accurate, it can be assumed that there would be extremely strong aiding of the IMU.

Bibliography

[1] Anderson, Dean, Herman Herman, and Alonzo Kelly. "Experimental Characterization of Commercial Flash Ladar Devices". *Proceedings of the International Conference of Sensing and Technology*, 17–22. November 2005.

[2] Chen, Q., U. Ozguner, and K. Redmill. "Ohio State University at the 2004 DARPA Grand Challenge: developing a completely autonomous vehicle". *Intelligent Systems, IEEE*, 19(5):8–11, 2004.

[3] Chiabrando, F., D. Piatta, and F. Rinaudo. "SR-4000 TOF Camera: Further Experimental Tests And First Applications To Metric Surveys". *Proceedings of the 2010 International Society for Photogrammetry and Remote Sensing Workshop*, 149–154. 2010.

[4] Christel, Brady. *Two Dimensional Positioning and Heading Solution for Flying Vehicles using a Line-Scanning Laser Radar (LADAR).* Master's thesis, Air Force Institute of Technology, March 2011.

[5] Clode, S., F. Rottensteiner, P. Kootsookos, and Emanuel Emil Zelniker. "Detection and vectorization of roads from lidar data". *Photogrammetric Engineering and Remote Sensing*, 73(5):517–535, 2007.

[6] Clode, Simon, Peter J. Kootsookos, and Franz Rottensteiner. "The Automatic Extraction of Roads from LIDAR data". *Proceedings of the 2004 International Society for Photogrammetry and Remote Sensing Workshop*, 12–23. 2004.

[7] Coughlan, J. M. and A. L. Yuille. "Manhattan World: compass direction from a single image by Bayesian inference". *The Proceedings of the Seventh IEEE International Conference on Computer Vision*, volume 2, 941. 1999.

[8] Coughlan, James M. and A. L. Yuille. "The Manhattan World Assumption: Regularities in scene statistics which enable Bayesian inference". *Proceedings of the 2000 Neural Information Processing Systems Conference*, 845–851. 2000.

[9] Crane, Carl D. III, David G. Jr Armstrong, Mel W. Torrie, and Sarah A. Gray. "Autonomous Ground Vehicle Technologies Applied to the DARPA Grand Challenge". *Proceedings of the 4th International Conference on Control, Automation, and Systems*, 1–5. 2004.

[10] Ferretti, A., C. Prati, and F. Rocca. "Permanent scatterers in SAR interferometry". *IEEE Transactions on Geoscience and Remote Sensing*, 39(1):8 –20, Jan 2001. ISSN 0196-2892.

[11] Fijany, A. and F. Hosseini. "Image processing applications on a low power highly parallel SIMD architecture". *Proceedings of the 2011 IEEE Aerospace Conference*, 1–12. 2011. ISBN 1095-323X.

[12] Fischler, Martin A and Robert C Bolles. "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography". *Communications of the Association for Computing Machinery*, 24(6):381–395, 1981. URL http://portal.acm.org/citation.cfm?doid=358669.358692.

[13] Fontanelli, D., L. Ricciato, and S. Soatto. "A Fast RANSAC-Based Registration Algorithm for Accurate Localization in Unknown Environments using LIDAR Measurements". *Proceedings of the 2007 IEEE International Conference on Automation Science and Engineering*, 597–602. 2007.

[14] Gudmundsson, SA and Kongens Lyngby. *Robot Vision Applications Using the CSEM SwissRanger Camera*. Master's thesis, Technical University of Denmark, July 2006.

[15] Uijt de Haag, M., D. Venable, and A. Soloviev. "Implementation of a Flash-LADAR aided inertial navigator". *Proceedings of the 2008 IEEE/ION Position, Location and Navigation Symposium*, 560 –567. May 2008.

[16] Kolb, A., E. Barth, R. Koch, and R. Larsen. "Time-of-Flight Sensors in Computer Graphics". M. Pauly and G. Greiner (editors), *Eurographics 2009 - State of the Art Reports*, 119–134. Eurographics Association; Eurographics, Munich, Germany, Mar 2009. ISBN 1017-4656. URL http://www.cg.informatik.uni-siegen.de/data/Publications/2009/kolb09tof-star.pdf.

[17] Lange, Robert. *3D Time-of-Flight Distance Measurement with Custom Solid-State Image Sensors in CMOS/CCD-Technology*. Ph.D. thesis, University of Siegen, Department of Electrical Engineering and Computer Science, 2000.

[18] Linarth, A.G., J. Penne, B. Liu, and O. Jesorsky. "Fast fusion of range and video sensor data". *Advanced Microsystems for Automotive Applications*, XVI:119–134, 2007.

[19] Lindner, M., A. Kolb, and K. Hartmann. "Data-Fusion of PMD-Based Distance-Information and High-Resolution RGB-Images". *Proceedings of the 2007 International Symposium on Signals, Circuits and Systems*, volume 1, 1 –4. July 2007.

[20] McClure, Shane H., Michael J. Cree, Adrian A. Dorrington, and Andrew D. Payne. "Resolving depth-measurement ambiguity with commercially available range imaging cameras". volume 7538, 75380K. SPIE, 2010. URL http://link.aip.org/link/?PSI/7538/75380K/1.

[21] Mesa Imaging, AG. "SR4000 Data Sheet".

[22] Oprisescu, S., D. Falie, M. Ciuc, and V. Buzuloiu. "Measurements with ToE Cameras and Their Necessary Corrections". *Proceedings of the 2007 International Symposium on Signals, Circuits and Systems*, volume 1, 1–4. 2007.

[23] Park, Chan-Soo, Doik Kim, Bum-Jae You, and Sang-Rok Oh. "Characterization of the Hokuyo UBG-04LX-F01 2D laser rangefinder". *Proceedings of the 2010 International Symposium on Robots and Human Interactive Communications*, 385 –390. Sept 2010. ISSN 1944-9445.

[24] Payne, A. D., A. P. P. Jongenelen, A. A. Dorrington, M. J. Cree, and D. A. Carnegie. "Multiple frequency range imaging to remove measurement ambiguity". *Proceedings of the 9th Conference on Optical 3-D Measurement Techniques*, 139–148. 2009. URL http://researchcommons.waikato.ac.nz/handle/10289/4032.

[25] Racing, Red Team. "Red Team DARPA Grand Challenge 2005 Technical Paper", 2005. URL http://www.darpa.mil/grandchallenge/TechPapers/RedTeam.pdf.

[26] Stone, William C., Maris Juberts, Nick Dagalakis, Jack Stone, Jason Gorman, Phillip J. Bond, Under Secretary, and Arden L. Bement. *Performance Analysis of Next-Generation LADAR for Manufacturing, Construction, and Mobility*. NIST Interagency/Internal Report (NISTIR) 7117, NIST, 2004. URL http://www.nist.gov/manuscript-publication-search.cfm?pub_id=822493.

[27] Venable, Donald T. *Implementation of a 3D Imaging Sensor Aided Inertial Measurement Unit Navigation System*. Master's thesis, Russ College of Engineering and Technology of Ohio University, 2008.

Appendix: Sample Output of Hallway Detection Algorithm on Simulated Data

Frame: 5 of 80; Time elapsed: 13.220234

Cross Hallway Found. Frame: 7;Position: (-2.021301,11.400198,0.039567)

Cross Hallway Found. Frame: 8;Position: (-2.003571,11.300198,0.039220)

Cross Hallway Found. Frame: 9;Position: (-1.985841,11.200198,0.038873)

Cross Hallway Found. Frame: 10;Position: (-2.047000,11.100220,0.038574)

Frame: 10 of 80; Time elapsed: 26.511330

Cross Hallway Found. Frame: 11;Position: (-2.028559,11.000220,0.038226)

Cross Hallway Found. Frame: 12;Position: (-2.010118,10.900220,0.037879)

Cross Hallway Found. Frame: 13;Position: (-2.068630,10.800244,0.037580)

Cross Hallway Found. Frame: 14;Position: (-2.049477,10.700244,0.037232)

Cross Hallway Found. Frame: 15;Position: (-2.030323,10.600244,0.036884)

Frame: 15 of 80; Time elapsed: 39.976107

...

Frame: 60 of 80; Time elapsed: 176.385585

Frame: 65 of 80; Time elapsed: 189.709948

Frame: 70 of 80; Time elapsed: 202.754406

Frame: 75 of 80; Time elapsed: 215.842229

Frame: 80 of 80; Time elapsed: 229.063870

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 074-0188

1. REPORT DATE *(DD-MM-YYYY)*	**2. REPORT TYPE**	**3. DATES COVERED** *(From – To)*
14-06-2012	**Master's Thesis**	Sept 2010 – Jun 2012

4. TITLE AND SUBTITLE Cross Hallway Detection and Indoor Localization Using Flash Laser Detection and Ranging	**5a. CONTRACT NUMBER** **5b. GRANT NUMBER** **5c. PROGRAM ELEMENT NUMBER**
6. AUTHOR(S) Prileszky, Istvan M., 2d Lieutenant, USAF	**5d. PROJECT NUMBER** **5e. TASK NUMBER** **5f. WORK UNIT NUMBER**
7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(S) Air Force Institute of Technology Graduate School of Engineering and Management (AFIT/EN) 2950 Hobson Way WPAFB OH 45433-7765	**8. PERFORMING ORGANIZATION REPORT NUMBER** AFIT/GE/ENG/12-34
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Intentionally Left Blank	**10. SPONSOR/MONITOR'S ACRONYM(S)** **11. SPONSOR/MONITOR'S REPORT NUMBER(S)**

12. DISTRIBUTION/AVAILABILITY STATEMENT
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

13. SUPPLEMENTARY NOTES
This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

14. ABSTRACT

A flash LADAR is investigated as a source of navigation information to support cross-hallway detection and relative localization. To accomplish this, a dynamic, flexible simulation was developed that simulated the LADAR and the noise of a LADAR system. Using simulated LADAR data, algorithms were developed that were shown to be effective at detecting cross hallways in simulated ideal environments and in simulated environments with noise. Relative position was determined in the same situations. A SwissRanger SR4000 flash LADAR was then used to collect real data and to verify algorithm performance in real environments. Hallway detection was shown to be possible in all real data sets, and the relative position-finding algorithm was shown to be accurate when compared to the absolute accuracy of the LADAR. Thus, flash LADAR is concluded to be an effective source for indoor navigation information.

15. SUBJECT TERMS
Flash LADAR, Indoor Navigation, Hallway Detection, Relative Position

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
					Michael J. Stepaniak, Lt Col, USAF (ENG)
REPORT	**ABSTRACT**	**c. THIS PAGE**			**19b. TELEPHONE NUMBER** *(Include area code)*
U	**U**	**U**	**UU**	106	(937) 255-3636 x4603

Standard Form 298 (Rev: 8-98)
Prescribed by ANSI Std. Z39-18