2007 IEEE International Conference on
Robotics and Automation
Roma, Italy, 10-14 April 2007

Training and optimization of operating parameters for flash LADAR cameras

Michael Price
EECS Department, MIT
Cambridge, MA 02139
pricem@mit.edu

Jacqueline Kenney and Roger D. Eastman
CS Department, Loyola College in Maryland
Baltimore, MD 21210
{jlkenney,reastman}@loyola.edu

Tsai Hong
Intelligent Systems Division, NIST
Gaithersburg, MD 20899
tsai.hong@nist.gov

Abstract— **Flash LADAR cameras based on continuous-wave, time-of-flight range measurement deliver fast 3D imaging for robot applications including mapping, localization, obstacle detection and object recognition. The accuracy of the range values produced depends on characteristics of the scene as well as dynamically adjustable operating parameters of the cameras. In order to optimally set these parameters during camera operation we have devised and implemented an optimization algorithm in a modular, extensible architecture for real-time applications including robot control. The approach uses two components: offline nonlinear optimization to minimize the range error for a training set of simple scenes followed by an online, real-time algorithm to reference the training data and set camera parameters. We quantify the effectiveness of our approach and highlight topics of interest for future research.**

I. Introduction

In the last few years, flash LADAR cameras have become available as highly integrated cameras connecting to a computer over USB or Firewire. At the National Institute of Standards and Technology (NIST) we have been studying the performance and applicability of these solid-state cameras, which in newer models produce range images of up to 176x144 pixels at frame rates approaching 50 Hz without the use of scanning mirrors found in other LADAR technologies. Flash LADAR cameras provide their own illumination for indoor and some outdoor environments, with a non-ambiguous range of 7.5 to 30 meters in the models we tested. The technology has a large number of potential applications, including robot navigation, object recognition and manipulation, robot safety and general surveillance [1].

In this paper we describe a method for maintaining optimal camera settings during operation so range images are as accurate as possible. Flash LADAR cameras have several parameters that can be adjusted during operation, including the integration time of the demodulation pixels on the CMOS sensor and the modulation frequency of the illumination source. The range cameras produce both systematic and random range errors that depend on these parameters. We perform off-line training to minimize range errors for a variety of scenes, and store the results for each individual model. Afterwards, the results are loaded into a regression tree and accessed during normal use. We standardized the process and used it to control commercially-available models from three manufacturers.

Standardizing an approach to setting range camera parameters supports our objective of developing protocols for evaluating flash LADAR technology. A number of researchers have undertaken the evaluation of LADAR technologies [2], [3], [4], [5], while others have studied how to find and remove noise artifacts in flash LADAR range data [6]. A unified solution would use knowledge of the sensor's noise behavior to continuously monitor and optimize performance.

II. Background: Flash LADAR

It is helpful to understand the operation and error mechanisms of these cameras in order to maximize their effectiveness. We will review how flash LADAR works, the primary causes of range measurement errors, and the meaning of the adjustable settings in currently available cameras. Then we will summarize our approach for minimizing these errors in unsupervised use.

We purchased and tested flash LADARs from CSEM SA, PMDTec GmbH, and Canesta, Inc.[1] All three models use near-infrared LEDs for illumination and rely on the same type of CMOS image sensor. The specifications of the cameras are summarized in Table I.

Manufacturer	CSEM	Canesta	PMDTec
Model	SR2	DP205	PMDvision 19k
Resolution	160x124	64x64	160x120
Interface	USB	USB	Firewire or TCP/IP
Light power (W)	1	1	3
FOV (deg)	46x42	55x55	40x30
Max. range (m)	7.5	11.5	30

TABLE I
Specifications of short-range LADARs used at NIST.

A. Continuous-wave range measurement

Flash LADARs measure the distance to an object by measuring the time needed for light to travel from the instrument to the object and back. Radiated light (usually near infrared) is amplitude modulated at several MHz, so the modulation of the reflected light has a noticeable phase shift. The phase angle is measured by integrating the product of the two signals. Lange [7] provides the fundamentals in his thesis.

[1]NIST does not endorse products discussed within this paper nor manufacturers of those products. Products mentioned are for information only, and are not expressed as an endorsement for them or their manufacturer.

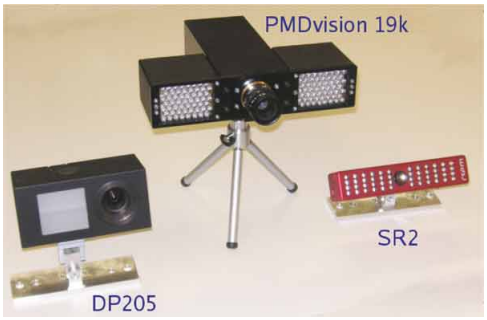

Fig. 1. Example flash LADAR cameras

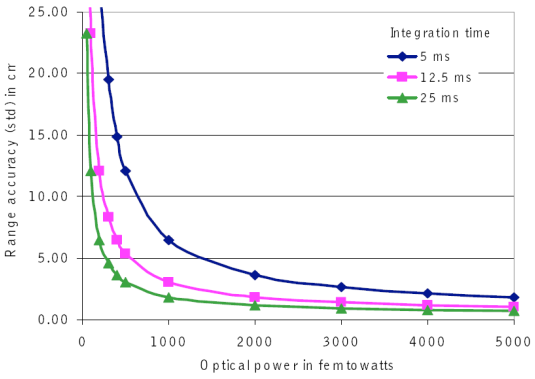

Fig. 2. Theoretical best-case measurement accuracy derived from shot noise and thermal noise [7].

This method gives a unique result only for phase angles between 0 and 180 degrees. For objects more than $\frac{1}{2}$ wavelength away from the camera, such as 7.5 m at 20 MHz, the range measurement is the true range modulo 7.5 m. This limits the useful range of the camera.

Finally, the distance to the object is computed from the phase angle and the modulation frequency. A higher modulation frequency might provide better measurement resolution, but a shorter effective range (for example, 3.75 m at 40 MHz).

B. Error mechanisms

The time-of-flight measurement process is inherently noisy since it can be photon limited with few returns from a distant or dark object using eye-safe illumination sources. Figure 2 shows how the theoretical range accuracy worsens with decreasing illumination [7]. Each individual phase angle measurement, computed from one cycle of the light modulation, can be noisy. To mitigate this problem, the range values returned by the cameras come from measurements averaged over many thousands of cycles of the illumination signal. This measurement period is called the integration time, equivalent to the exposure time of a still camera.

As with film cameras, lengthening the integration time too much causes problems. Besides compromising frame rate in the interest of range error reduction, long integration times allow too much light to arrive at brightly illuminated pixels, saturating the amplifier circuitry. When this happens, the range values develop large errors and the data is often rendered useless. The phenomenon is intensified when the camera's illumination is very directional or very powerful.

C. Settings

Several settings define the operation of each camera. They also form the search space for our optimization procedure, and the dependent variables for regression lookup. Each camera exposes a different set of parameters through its internal logic and driver software. See Table II for a summary of the settings and their adjustment ranges for the CSEM, PMD and Canesta cameras.

Parameter	Unit	Min	Max	Default
CSEM SR2				
Integration time	μs	256	65280	10200
Distance offset	m	0	7.5	0
Amplitude threshold	-	0	65535	192
Canesta DP205				
Integration time 1	ms	1	500	13
Integration time 2	ms	1	500	30
CMR 1	-	0	100	3
CMR 2	-	0	100	0
Modulation frequency	MHz	13	104	26
Light power	mW	0	1000	1000
PMDvision 19k				
Integration time	μs	0	65535	15000
Modulation frequency	MHz	5	30	15

TABLE II
ADJUSTABLE SETTINGS FOR THREE FLASH LADAR CAMERAS

Keeping in mind that the adjustability varies from camera to camera, we summarize the most important parameters here:

1) **Integration time**: how long each pixel accumulates light. A typical range is from 1 to 64 ms. Adjustment is supported by all unit tested. Lower integration times enable faster frame rates but result in noisier data, especially toward the edges of an image or distant targets. The Canesta cameras can use two different integration times and produce a final range map by merging the most accurate pixels from each time.
2) **Modulation frequency**: controls the non-ambiguous range and, to a lesser extent, range resolution. The CSEM SR2 has a frequency of 20 MHz, with a 7.5 m range limit. The PMD cameras allow adjustments in 5 MHz intervals from 5 to 30 MHz; Canesta includes 4 choices between 13 and 104 MHz. This should be adjusted based on the maximum target distance; reflective objects beyond the non-ambiguous range are frequently aliased.
3) **CMR (common mode rejection)**: subtracts out a specified amount of background radiation. Canesta exposes CMR through its API, and PMD implements it in firmware. Outdoor operation in sunlight requires

strong CMR.

4) **Illumination power**: controls the signal radiated by the camera's LEDs. The best setting depends on the distance and reflectivity of the primary targets.
5) **Amplitude thresholds**: the SR2 can automatically eliminate pixels where the returned light signal is outside a defined band. This filtering is used to eliminate outliers caused either by too low or too high of a returned signal. Image processing accomplishes the same thing, but onboard filtering may be useful for increasing frame rates under nonideal conditions.

III. Training and Optimization

Camera setting optimization is too slow to run on-the-fly, so we split the problem into two steps. An offline training process finds the optimum settings for a training set of typical scenes characterized by range and intensity statistics, and a regression system matches the current scene statistics to the training set to return the best settings. The procedure applies to all continuous-wave flash LADARs.

A. Overview

Ideally, one could map a broad range of imaging situations into a reasonable space of "external" parameters, and attach each situation to the best settings – "internal" parameters. Characterizing external parameters would be helped if the training could be performed with a more accurate range camera. Unfortunately, we do not have access to higher performance flash LADARs, so we currently train the camera with a planar target at a known distance and orientation.

We mount the camera on a movable test apparatus looking at a solid, consistent target: a wall. The apparatus has two degrees of freedom, both electronically measured: the distance from the target, and a horizontal rotation angle. By varying these degrees of freedom, we try to simulate a wide variety of situations: different target distances, multiple targets at different distances, and so on. A SICK line scanner (LMS 200) scans across the field of view of the LADAR; the distance and angle to the plane are estimated from its range measurements. We can then generate a reference range image corresponding to the actual target for use as the "ground truth" for error calculations and optimization.

The following sections introduce the relevant variables, then define the training (offline) and lookup (online) functions.

B. Variables

The camera settings are dependent variables of an optimum control function. Choosing the independent variables is tricky. For the purpose of controlling settings, we will describe an image by the mean and variance of its range and amplitude data (whether or not the data is normally distributed). These four statistics are reasonable features for describing the differences between images (see Figure 3).

We categorize settings and image statistics into three general vectors, so that the selection of specific variables

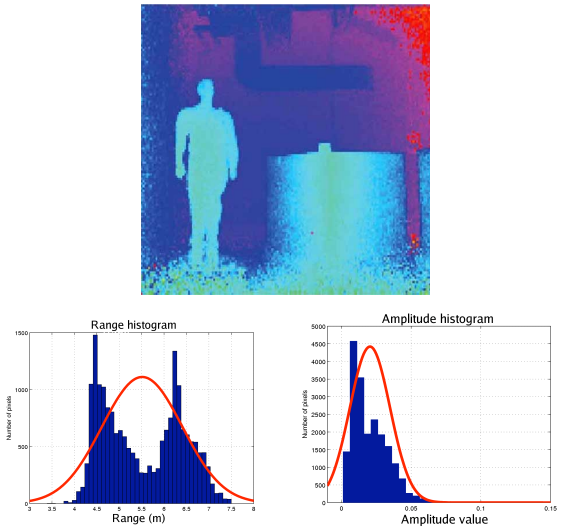

Fig. 3. Typical range image with range and amplitude histograms.

Input	$\mathbf{x}$	Output	$\mathbf{y}$	Auxiliary	$\mathbf{z}$
Range mean	μ_r	Settings vector	$\mathbf{s}$	Error mean	μ_e
Range variance	σ_r^2			Error variance	σ_e^2
Amplitude mean	μ_a				
Amplitude variance	σ_a^2				
Situation vector	$\mathbf{q} = [\mathbf{x}\ \mathbf{y}\ \mathbf{z}]$				
Training set	array of situation vectors $\mathbf{q}[i]$				

TABLE III
Variables and Vector Notation for LADAR Optimization Scheme.

can be expanded or revised within the same framework, as shown in Table III.

Each range imaging scene is described by the input vector $\mathbf{x}$, and the best settings for capturing an image are in the output vector $\mathbf{y}$. The auxiliary vector $\mathbf{z}$ allows a range offset to be subtracted out, and provides an estimate of the measurement uncertainty. A set of situation vectors $\mathbf{q}[i]$ is created during offline training and then referenced for live parameter control. In section IV we will show that the situation vectors produced by training on a simple planar target map effectively to arbitrary scenes.

C. Offline training

In the training process, we subtract the reference ("ground truth") range values from each LADAR range image, and evaluate a cost function on the resulting error image. An unconstrained nonlinear optimizer minimizes this cost by searching through the camera settings:

$$\mathbf{y}_{\text{opt}} = \arg\min_{\mathbf{y}}(\sigma_{\mathbf{e}}^{\mathbf{2}} + \mathbf{A}\mu_{\mathbf{e}}^{\mathbf{2}})$$

where μ_e and σ_e^2 are the mean and variance of the range error in meters, taken from the auxiliary data $\mathbf{z}$. Then a new situation vector $\mathbf{q}[i]$ is formed by attaching $\mathbf{y}_{\text{opt}}$ to the corresponding $\mathbf{x}$ and $\mathbf{z}$ vectors, which are image statistics. The training set is the array of all optimized situation vectors for a particular LADAR.

The quadratic mean error term prevents extreme results such as complete saturation of the sensor; otherwise it is unimportant, because the mean error can be stored and subtracted from future measurements. In practice, a value of $A = 0.1$ works well, and the μ_e term is relatively small. It does not cause a noticeable deviation from the true optimal result.

LADAR camera range data is noisy and dependent on the scene and target material, so the cost function computed from individual frames is also noisy. There is no practical way to solve or approximate this type of function analytically. To accomodate this, the optimizer is based on the Nelder-Mead simplex algorithm [8], which is slow but does not require derivatives of the error function. To help the algorithm deal with the inherent noise in the process, we incorporated modified termination criteria and adjusted the shrinking coefficient σ from 0.5 to 0.9 as suggested in [9]. The simplex, a convex volume in the search space, is initialized to include the most commonly used settings as a starting point. Trade-offs in accuracy can be made by revising the termination condition.

This is an interactive process, with the Nelder-Mead algorithm providing new settings for the camera to use with each successive frame. To build up sufficient information for general-purpose control, we move and rotate the training apparatus as new situation vectors are generated. This spreads the data points across the search space formed by the image statistics in $\mathbf{x}$. At large measurement distances, it may be impractical to fill the field of view with the target, so we use a manually specified region of interest for the error calculations.

Any number of points may be used; a typical training set of 100 points is stored in a 6 KB text file. Cameras with more adjustable settings (such as the Canesta DP205) require more points for the adjustments to be equally effective. The training must be performed separately for each LADAR camera, as we have found their performance can vary even among cameras of the same model.

D. Lookup

"Live" control of a LADAR camera can be accomplished by matching an input $\mathbf{x}_{\text{new}}$ to the most similar situation encountered during training, then returning the optimized settings $\mathbf{y}_{\text{opt}}$ from that situation. The returned settings are in the vector $\mathbf{y}[i]$, which comes from $\mathbf{q}[i]$ – the nearest neighbor among the situation vectors in the training set. The distance metric is based only on the initial image statistics ($\mathbf{x}[i]$ component):

$$\mathbf{y}_{\text{new}} = \mathbf{y}\left[\arg\min_i(||\mathbf{x}[i] - \mathbf{x}_{\text{new}}||)\right]$$

If the training set is dense, $\mathbf{y}_{\text{new}}$ will be very close to a true minimum found by optimization; it can be computed very quickly.

We use a binary regression tree [10] to quickly search through a large amount of training data. Let the training set be a list of scene vectors $\mathbf{q}[i]$, where $i = 0, 1, \ldots, n-1$, loaded from a file. Each leaf of the tree is a list of scene vectors $\mathbf{q}[j]$, where j is a unique subset of i. These are points in the search space defined by $\mathbf{x}$.

The tree expands automatically to keep a small leaf size (such as 16 points). Each branch splits the points into two categories, based on the input variable $\mathbf{x_k}$ with the highest normalized variance over j. As the camera collects images, the amplitude and range statistics contained in $\mathbf{x}$ provide the independent variables for regression lookup. Because of the small number of training points, the memory usage of the regression tree is insignificant – less than 32 KB for up to 1,000 points.

The tree may contain points very close to each other, and changing the settings can cause noticeable delays (as with CSEM and Canesta LADARs). We use mild hysteresis, allowing updates only if the Euclidean distance $||\mathbf{x}_{\text{new}} - \mathbf{x}_{\text{opt}}||$ exceeds a fixed tolerance, to prevent unnecessarily rapid changes and oscillations.

IV. Experiments

We built software for LADAR data collection, analysis, and viewing based on the NIST 4D/RCS control system framework [11]. In that framework, software is implemented as independent modules communicating through messages, either between processes on one CPU or transmitted over an Ethernet network in device-independent format. The optimizer functionality is in a separate module, so it can easily be added to data collection or real-time robot control applications that also operate without it. The software accepts new camera types easily, since camera characteristics are stored in an XML-style configuration file and read in at run time. Extending the software to a new camera model would require only a new configuration file and a program to convert device driver output into RCS messages. We present here preliminary results for the SR2, followed by measurements quantifying the optimizer's choice of integration time.

The optimizer software performs both a steady state calibration (since the correct mean range was known during training) and range error reduction, without the need for detailed information from the LADAR manufacturer or expensive measurement equipment. Figure 4 illustrates the action of the optimizer in setting the depth offset and integration time vs. defaults. The image collected using optimized settings has less noise and better absolute accuracy.

To verify whether the choices of camera settings made by the optimizer were reasonable, we tested the software performance on two scenes. Scene 1 was very basic and emulated the training cases, with a flat planar target covered with white photographic paper. Scene 2 consisted of several objects, including planar surfaces of different reflectances and a clothed mannequin (Figure 5). We focused in these experiments on range error reduction with the intention of comparing the camera setting returned by the optimizer with the true optimal value found by directly exploring the parameter space.

For scene 1, we pointed the camera directly at the target and measured the distance with the SICK laser scanner,

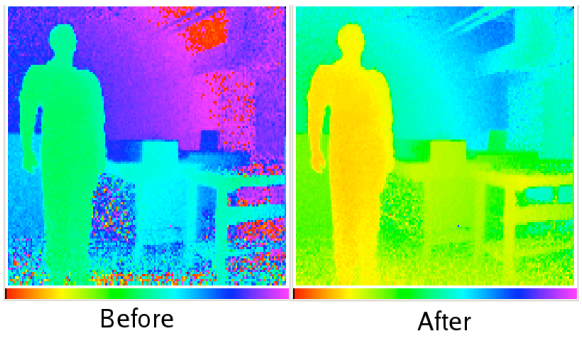

Fig. 4. Images collected with initial and optimized settings

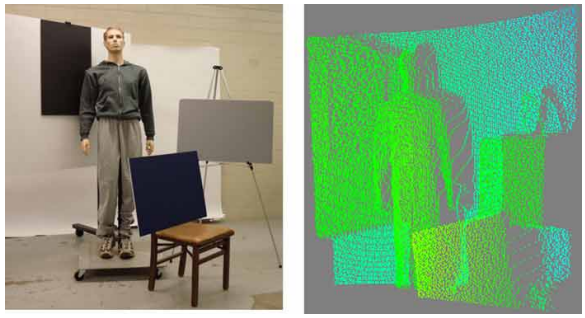

Fig. 5. Scene 2 with mannequin and planar targets: photograph (left) and LADAR range data viewed from a different angle (right).

which provided ground truth. We conducted this test at distances of 2, 3 and 4 meters. At each distance we took 10 images at 25 integration times between 4 and 62 milliseconds. Since the target was a planar surface, we could directly measure the deviation from the known range.

For scene 2, we conducted similar tests but did not have ground truth. We imaged the scene from three locations differing in distance and orientation, and collected the same 10 images at 25 integration times. In this case we used two measurements: a local standard deviation of range values in 5 by 5 neighborhoods, and the standard deviation from individual planar fits to five locally planar, uniform reflectance surface patches extracted by manually drawn masks. An image of the five masks is shown in Figure 6.

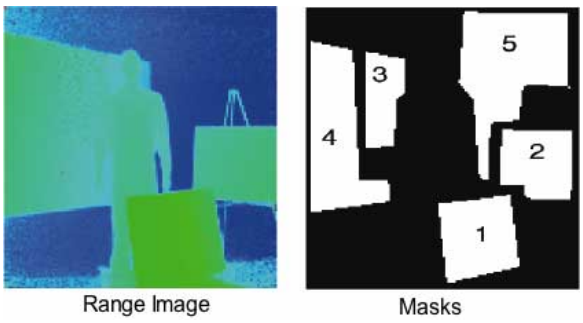

Fig. 6. Scene 2 (left) and masks for locally planar surfaces (right).

For both scenes we can compute a graph of integration time vs. an error measurement and then compare the setting chosen by our optimizer to the true minimum of the error curve. The results of four different tests were consistent. For scene 1, Figure 7 presents the range error statistic σ_e averaged over 10 images at 25 different integration time settings between 4 and 62 ms. The automatically selected integration time, and the corresponding σ_e, are plotted as solid dots. The top graph in Figure 7 shows a bathtub curve at 2 meters, where at low integration times (below 10 ms) the weak light return causes large errors, and at high integration times the light return saturates the sensor. The results at 3 m and 4 m appear similar, without saturation at high integration times. For all three distances there is a relatively wide region of similar near-minimum error dispersal.

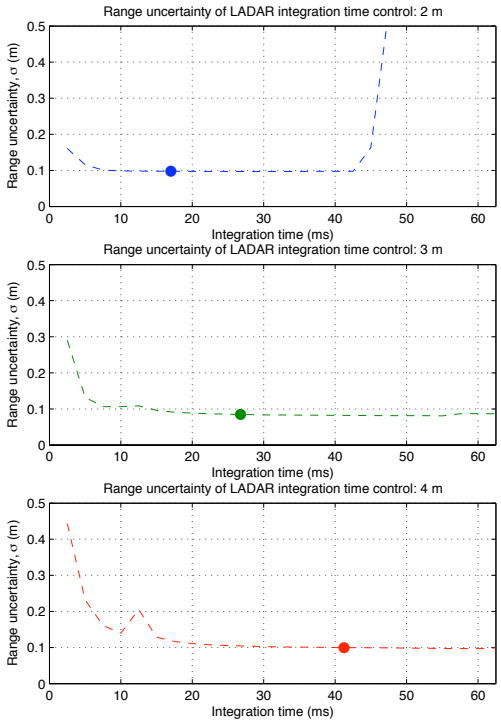

Fig. 7. Scene 1: Test of integration time versus range uncertainty, including optimized settings annotated as filled circles.

Table IV compares the σ_e at optimizer-chosen integration times with the true minimum σ_e across all integration times. The optimizer-chosen integration times are near the left side of a wide region of similar error; they provide sufficiently low error (within 1 cm of the minimum value), trading some error performance for frame rate. This is a consequence of the Nelder-Mead implementation described in section III-C.

For scene 2, the measurements of local and planar error were similar. Figure 8 shows two representative graphs, the first one illustrating the local σ_e and the second one illustrating the masked planar σ_e separately graphed for each planar region (numbered 1 through 5). The value of σ_e varies close to 2.5-fold between the regions, influenced by the reflectance of the surface and its distance from the center of the image (where illumination is strongest).

Table V compares the error measurements for the three

Dist.	Min. error int. time (ms)	Minimum σ_e (cm)	Optimized int. time (ms)	Optimized σ_e (cm)
2 m	33	9.67	18	9.78
3 m	55	8.08	27	8.48
4 m	60	9.68	41	9.97

TABLE IV

SCENE 1: UNCERTAINTY PERFORMANCE (σ_e) OF OPTIMIZED INTEGRATION TIME SETTINGS.

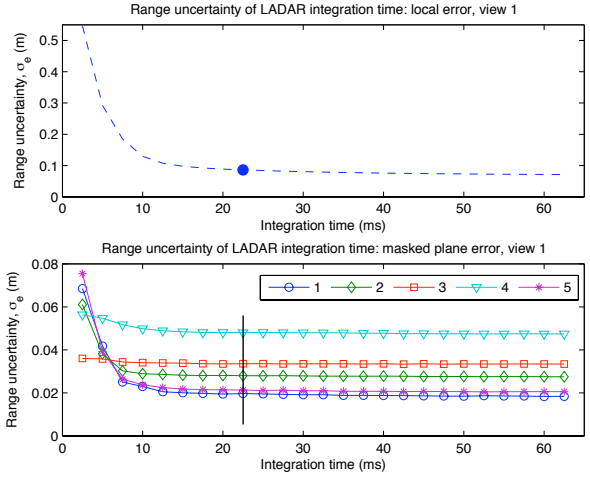

Fig. 8. Scene 2: Test of integration time versus range uncertainty, including the optimized setting of of 22.5 ms annotated as filled circle (top) and as a line (bottom).

views of scene 2 at optimizer-chosen integration times with the minimum local σ_e across all integration times. Again, the optimizer-chosen integration times are near the left side of the region of minimum error, although in this case the difference is as much as 1.8 cm. However, in Table VI (which gives the same results for all five locally planar regions shown in Figure 6), the optimizer-chosen integration time yields a σ_e very close to the global minimum. The value of σ_e varies over 2-fold over the five regions, suggesting that optimization focusing on particular ROIs may be productive.

V. CONCLUSION

In this paper we suggested a robust, simple procedure for the control of LADAR camera settings using standardized structures, applied a nonlinear optimizer to find optimal settings in an offline training environment, and incorporated this

View	Min. error int. time (ms)	Minimum σ_e (cm)	Optimized int. time (ms)	Optimized σ_e (cm)
1	62	7.13	23	8.63
2	62	7.64	32	8.31
3	50	8.11	16	9.96

TABLE V

SCENE 2: UNCERTAINTY PERFORMANCE (σ_e)OF OPTIMIZED INTEGRATION TIME SETTING

Plane	Min. error int. time (ms)	Minimum σ_e (cm)	Optimized int. time (ms)	Optimized σ_e (cm)
1	35	1.89	23	1.97
2	23	2.80	23	2.80
3	20	3.35	23	3.36
4	40	4.78	23	4.80
5	35	2.08	23	2.13

TABLE VI

SCENE 2: UNCERTAINTY PERFORMANCE (PLANAR σ_e)OF OPTIMIZED INTEGRATION TIME SETTINGS.

knowledge in range imaging software. Finally, we conducted preliminary experiments to train and test this control scheme with a flash LADAR camera.

There are alternatives and extensions to our approach that we hope to explore. With the aid of more accurate cameras to use as "ground truth," or other local measures of error, the algorithm may be trained on more complex scenes, or incorporate continuous training in arbitrary scenes. A simple feedback loop maximizing the dynamic range of the intensity image may provide more reliable control of the integration time, which is the most important parameter we encountered, but this algorithm is more general and we believe it can help users improve the performance of highly customizable LADARs. We look forward to investigating these ideas as the technology evolves.

REFERENCES

[1] M. Juberts, N. Dagalakis, J. Stone, J. Gorman, and W. Stone, "Performance analysis of next-generation ladar for manufacturing, construction, and mobility," National Institute of Standards and Technology, Gaithersburg, MD, Tech. Rep. NISTIR-7117, 2004.

[2] M. Herbert and E. Krotkov, "3d measurements from imaging laser radars: how good are they?" *Image and Vision Computing*, vol. 10, no. 3, pp. 170–178, April 1992.

[3] D. Anderson, H. Herman, and A. Kelly, "Experimental characterization of commercial flash ladar devices," in *International Conference on Sensing Technologies*, New Zealand, November 2005.

[4] S. B. Gokturk, H. Yalcin, and C. Bamji, "A time-of-flight depth sensor - system description, issues and solutions," in *IEEE Conference on Computer Vision and Pattern Recognition Workshop*, vol. 3, Washington, DC, USA, 2004, p. 35.

[5] A. M. Lytle, I. Katz, and K. S. Saidi, "Performance evaluation of a high frame-rate 3d sensor for construction applications," in *22nd International Symposium on Automation and Robotics in Construction*, Ferrara, Italy, 2005.

[6] J. Tuley, N. Vandapel, and M. Herbert, "Analysis and removal of artifacts in 3d ladar data," in *IEEE International Conference on Robotics and Automation*, Barcelona, Spain, 2005, pp. 2203–2210.

[7] R. Lange, "3d time-of-flight distance measurement with custom solid-state image sensors in cmos/ccd-technology," Ph.D. dissertation, University of Siegen, 2000.

[8] J. C. Lagarias, J. A. Reeds, M. H. Wright, and P. E. Wright, "Convergence properties of the nelder-mead simplex method in low dimensions," *SIAM Journal Optimization*, vol. 9, pp. 112–147, 1998.

[9] H. G. Neddermeijer, G. J. van Oortmarssen, N. Pierma, R. Dekker, and J. D. F. Habbema, "Adaptive extensions of the nelder and mead simplex method for optimization of stochastic simulation models," Erasmus University Econometric Institute, Tech. Rep. EI2000-22/A, 2000.

[10] L. Breiman, J. Friedman, C. J. Stone, and R. Olshen, *Classification and Regression Trees*. Boca Raton, FL: CRC Press, 1984.

[11] V. Gazi, M. L. Moore, K. M. Passino, W. Shackleford, F. M. Proctor, and J. S. Albus, *The RCS Handbook: Tools for Real-Time Control Systems Software Development*. New York: John Wiley and Sons, 2001.