Project Report
LSP-221

Blind Compensation of Angle Jitter for Satellite-Based Ground-Imaging Ladar: FY17 Line-Supported Integrated Systems Program

E.J. Phelps
C.A. Primmerman

22 December 2017

Lincoln Laboratory
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
LEXINGTON, MASSACHUSETTS

This material is based upon work supported by the Assistant Secretary of Defense for Research and Engineering under Air Force Contract No. FA8721-05-C-0002 and/or FA8702-15-D-0001.

This report is the result of studies performed at Lincoln Laboratory, a federally funded research and development center operated by Massachusetts Institute of Technology. This material is based on work supported by the Assistant Secretary of Defense for Research and Engineering under Air Force Contract No. FA8721-05-C-0002 and/or FA8702-15-D-0001. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the Assistant Secretary of Defense for Research and Engineering.

Massachusetts Institute of Technology
Lincoln Laboratory

Blind Compensation of Angle Jitter for Satellite-Based Ground Imaging Ladar: FY17 Line-Supported Integrated Systems Program

E.J. Phelps
Group 106

C.A. Primmerman
Division 9

Project Report LSP-221

22 December 2017

Lexington Massachusetts

This page intentionally left blank.

ABSTRACT

Space-based ground-imaging ladar has become more feasible with recent technological advances. Compact fiber-optic lasers and single-photon-sensitive Geiger-mode detector arrays push the design towards low pulse energies and high pulse rates. A challenge in implementing such a system is imperfect pointing knowledge caused by angular jitter, exacerbated by long distances between satellite and ground. Without mitigation, angular jitter would cause significant blurring of the 3D data products. Reducing angular jitter to avoid such problems might require extreme mechanical isolation, advanced inertial measurement units (IMUs), star trackers, or auxiliary passive optical sensors. These mitigations can increase cost and size, weight, and power (SWaP) considerably. An alternative approach is demonstrated that uses only the ladar data to mitigate the unknown jitter, similar to blind deconvolution. Expectation Maximization is used to jointly estimate the 2-axis-jitter time series and the 2D ground surface. Reasonable assumptions about the jitter spectrum and the spatial-frequency spectrum of the ground are incorporated as prior distributions in a way consistent with Bayes' rule.

This page intentionally left blank.

TABLE OF CONTENTS

This page intentionally left blank.

LIST OF ILLUSTRATIONS

This page intentionally left blank.

LIST OF TABLES

This page intentionally left blank.

1. INTRODUCTION

MIT Lincoln Laboratory has pioneered the development of high-performance 3D-ladar systems using photon-counting avalanche photodiode (APD) arrays. We have successfully fielded several airborne 3D-ladar systems.[1] Currently, we are designing a space-based 3D ladar to go on a small-sat bus.

The basic concept of the ground-imaging ladar is shown in Figure 1. A short-pulse (~1 ns) high-repetition-rate laser illuminates a small patch on the ground (of order 70 m × 70 m). The backscattered light is imaged on a moderate-sized (128 × 128 pixel) APD array. The APD pixels make precise timing measurements, from which surface heights can be calculated. To generate a surface-height map over a large area (of order 10 km^2), the laser beam and APD-array field of view are rapidly scanned across the ground.

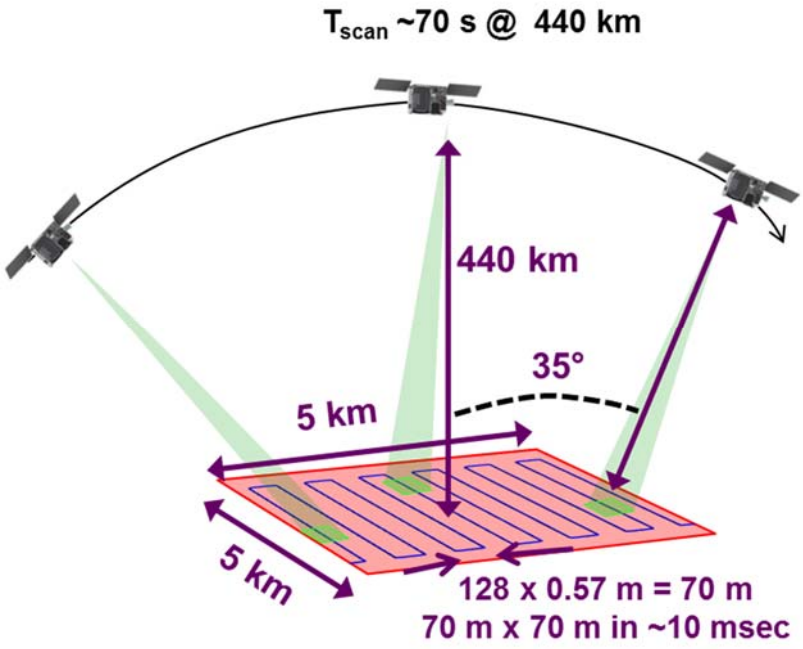

Figure 1. Nominal concept for space-based ground-imaging 3D ladar.

[1] *Laser Radar, Progress and Opportunities in Active Electro-Optical Sensing*, The National Academies Press (2014).

A requirement for developing an accurate surface-height map over a large area is that the majority of the individual height measurements are registered to a small fraction of a pixel (say <1/3 pixel). We have achieved this registration accuracy for airborne systems, but achieving it in a space-based system is much more challenging for three reasons.

First, the airborne systems did not need to operate diffraction limited, whereas the much longer-range space ladar will need to operate close to the diffraction limit to get adequate cross-range resolution. For the airborne systems, the Ground Sampling Distance (GSD) was greater than the diffraction-limited Ground Resolved Distance (GRD). For the space-based system, on the other hand, we need to have $GSD \lesssim GRD$.

Second, for the airborne systems, the range was a few kilometers; for a space-based system, the range will be a few hundred kilometers. This difference in range means that, for equivalent ground resolution, a space-based system will be 100 times more sensitive to jitter than an airborne system. For an airborne system, a pointing jitter of 100 μrad might be adequate; a space-based system may need a pointing jitter <1 μrad.

Third, for our airborne systems, the pointing knowledge was provided by highly precise, pointing mirrors, but for the small-sat system, it would be desirable to eliminate the need for large, heavy, expensive pointing mirrors.

To achieve the required registration accuracy, we have been investigating image-based registration—that is, using the ladar measurements themselves to estimate the jitter. Image-based registration has its own challenges. A particular problem for our scenario is that the relatively low-pulse-energy, high-repetition-rate laser (which is selected to maximize laser power efficiency) means that there are very few detections (~10) across the entire APD array per laser pulse. Thus, measurements from many pulses need to be integrated to make one height measurement.

In this report, we describe the development and initial testing of a novel algorithm to concurrently solve for both the pulse-by-pulse angle jitter and the pixel-by-pixel true heights of the scene.

2. DESIGN CONSIDERATIONS

As the satellite passes over a region of interest, the ladar will do a raster scan over the region. The transmitted beam will illuminate the whole field of view (FOV) of the receive array. A narrower beam would make the signal less challenging to integrate, but would increase the system complexity and mechanical requirements. The raster scan can be divided into multiple horizontal sweeps that can be processed independently. This report will focus on the processing of one horizontal sweep or the sensor staring at one location, both of which can be handled similarly.

TABLE 1

Example Parameters of Small-Sat Ladar

Parameter	Value
Array size (fully illuminated)	128 × 128 pixels
PRF	1 MHz
Pulsewidth (FWHM)	0.5 ns
Pulse energy	0.2 mJ
Wavelength	1 μm
Receive aperture diameter	1 m
Receive efficiency	0.2
Oversampling factor, Q	1 (GSD = GRD)
Background count rate	2 kHz
Surface reflectivity	0.1
Altitude	440 km
Jitter correlation time	10 ms
Jitter variance	50 μrad^2
Integration time per FOV	12.5 ms

Table 1 lists the parameters of the current system design. Based on these parameters, the ground sampling distance (GSD) at zenith is

$$\frac{1.22\lambda}{D}\frac{R}{Q} = 0.57\ m,$$

where λ is the wavelength, D is the aperture diameter, R is the range, and Q is the oversampling factor. The per-pulse photoelectron (PE) return rate across the whole array is

$$E_{pulse}\frac{\lambda}{hc}\frac{\rho}{2\pi R^2}\pi\left(\frac{D}{2}\right)^2\gamma = 13.8\,PE/pulse,$$

where additionally, E_{pulse} is the pulse energy in Joules, h is Planck's constant, c is the speed of light, ρ is the target reflectivity, and γ is the receive efficiency, which combines quantum and optical efficiency. When distributed over 128^2 pixels, this return rate becomes $8.4*10^{-4}$ PE/pixel/pulse, which is very low, especially compared to noise rates. The low return rate necessitates long integration times, which makes it necessary to compensate for jitter to avoid smearing.

Since the system is still in the early design stage, the angular jitter characteristics are not known. As a reference, Figure 2 shows sample power spectral densities (PSDs) for angular jitter for various spacecraft. A conservative baseline PSD for angular jitter was determined based on these spectra. The second largest of the PSDs (the ICD system) was selected and reduced by a factor of two based on engineering judgement. The parameters of a random process were then determined from this baseline PSD, such that the new PSD matches the baseline at frequencies above the reciprocal of the integration time, but flattens out at lower frequencies, which would be indistinguishable from DC. The random process is a 1st order Gauss-Markov process, and its associated parameters are listed in Table 1.

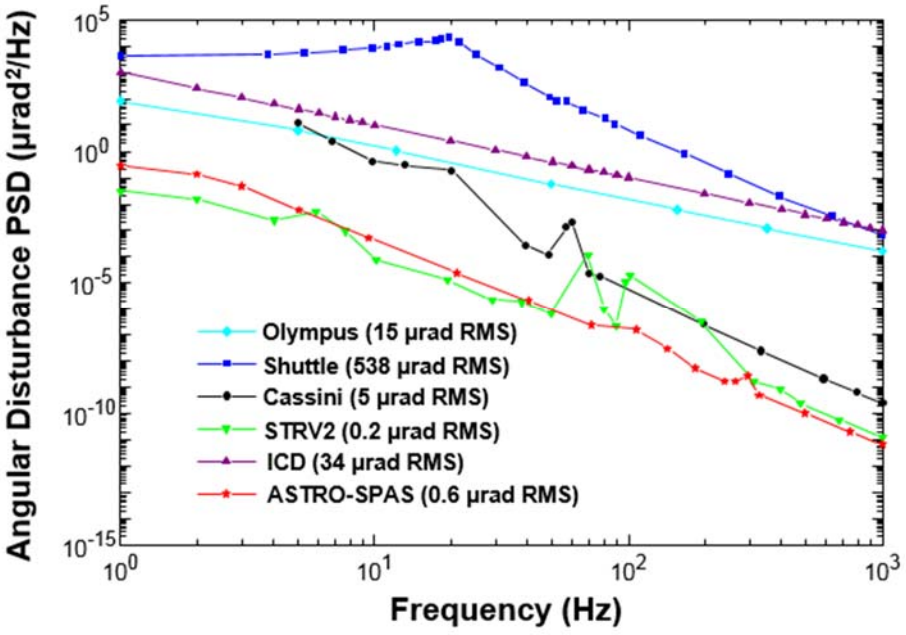

Figure 2. Angular jitter power spectral densities for various spacecraft.

A small angle approximation is used to translate the angular jitter into lateral motion of the boresight on the surface, i.e., $\Delta x \approx R\theta$. Using the parameters from Table 1, the per-axis RMS jitter at the surface would be

$$\Delta x \approx 440 \cdot 10^3 \sqrt{50 \cdot 10^{-12}}\, m = 3.1\, m$$

Actually, because the variance is defined as a long-term average, but the integration time is limited, AC fluctuation in the jitter is only about half of that value, at 1.5 meters. This is still multiple GSDs.

This page intentionally left blank.

3. PROBLEM FORMULATION

3.1 PHYSICAL MODELING

The problem has been modeled in a way that is simplified, but retains the most important physical characteristics. As the satellite passes over the region of interest, it adjusts its attitude to scan the field of view over the area. The satellite trajectory and attitude are both known to limited precision from inertial measurement unit (IMU) measurements. The combination of the trajectory and attitude determines the projection of the field of view onto the ground. To simplify the problem, an orthographic projection model is used, which assumes that the change in angular pointing is small. The region of interest can be divided into sufficiently small pieces that this assumption is valid. With the orthographic projection model, angles are replaced by their projection onto the observed scene.

The satellite position and the pointing, as reported by the IMU, are combined into a single 3D vector function of time:

$$g(t) = \begin{bmatrix} g_x(t) \\ g_y(t) \\ g_z(t) \end{bmatrix} \tag{1}$$

The coordinate frame is fixed relative to the ground, with the x-y plane orthogonal to the line of sight, and the z axis pointing along the line of sight towards the satellite. The z component can be considered the range, due to the orthographic projection approximation. The x-y plane, also called the reference plane, is defined such that it is close to the surface, notionally at the average surface level. The pixels of the detector array will project onto the reference plane in a grid with spacing equal to 1 GSD.

The 3D scene is modeled as a surface height relative to the reference plane, as a function of the lateral coordinates, that is:

$$h(x, y) \tag{2}$$

This is sometimes referred to as a 2.5D image. Real 3D scenes can of course be more complex than this, with multiple hard surfaces per GSD and fine-grained structure such as foliage, that allows ladar returns from a variety of ranges over a small line-of-sight change. Nevertheless, we believe this model to be a good approximation, as there is often a strong return from the first surface, and it is likely that a large portion of the scene will be well-represented by the model. The goal is to represent the surface accurately enough to facilitate the estimation of the jitter.

The jitter is modeled as a 2D function of time that represents a lateral shift of the boresight along the reference plane:

$$j(t) = \begin{bmatrix} j_x(t) \\ j_y(t) \end{bmatrix} \tag{3}$$

The x and y jitter functions are assumed to be independent 1st order Gauss-Markov random processes. This type of process has a power spectral density (PSD) that decreases proportionally to the square of frequency, which is typical of many real systems. This random process is equivalent to Gaussian white noise passed through a single-pole low pass filter.

Figure 3 illustrates the surface height with the projected pixel grid, and the jitter path. Over the integration time interval, the center of the pixel grid would traverse the jitter path.

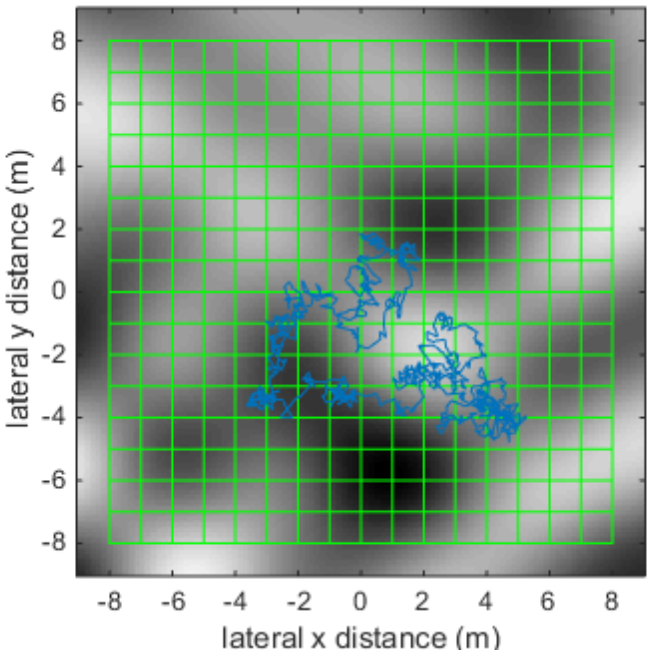

Figure 3. Surface height, pixel grid, and jitter path.

3.2 TRANSMITTED PULSES

The ladar transmits pulses that are assumed to have a large enough angular divergence that the whole field of view is illuminated. The pulse shape is assumed to be Gaussian in time. The transmit time of each pulse is recorded, which represents the center of the pulse. If the pulse energy is varied, then this is also recorded.

A transmitted pulse travels at the speed of light towards the surface, reflects off the surface, and returns towards the ladar. It is convenient to define the time and range at which a pulse would reflect off the reference plane. We will call them the nominal reflection time and nominal reflection range. These are found by taking the intersection of the outgoing pulse's range vs. time curve and $g_z(t)$, the range vs. time of the reference plane. This is shown in Figure 4. It is valid to treat the problem as if the satellite is stationary and the ground is moving. The intersection can be computed by the following steps:

1. Define a rotated coordinate frame such that the range vs. time trajectory of an outgoing pulse is a vertical line. This rotation will be 1/8 of a cycle (45 degrees) if the units of range are chosen to be the distance that light travels in one unit of time.

2. Convert the range vs. time function $g_z(t)$ into a function of the abscissa variable of the rotated frame. This is guaranteed to be a proper function because the range rate must be less than the speed of light.

3. Interpolate the rotated range vs. time function at the abscissa values of the vertical lines of the outgoing pulses.

4. Rotate each ordinate-abscissa pair back to the original frame to get the time and range of reflection.

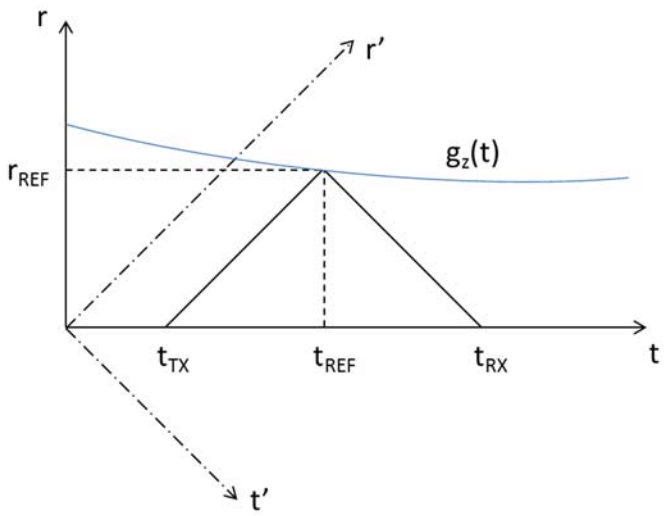

Figure 4. Determination of reflection range and time via interpolation in rotated frame.

The nominal reflection time and range for the i[th] pulse are related by

$$t_{REF\,i} = t_{TX\,i} + \frac{r_{REF\,i}}{c} \tag{4}$$

where c is the speed of light, and the remaining variables are defined in Table 2. Given the nominal reflection time, the nominal receive time is

$$t_{RX\,i} = t_{TX\,i} + 2(t_{REF\,i} - t_{TX\,i}) \tag{5}$$

The nominal receive energy is

$$e_{RX\,i} = \frac{\rho A_{RX} \gamma_{RX}}{2\pi} \frac{e_{TX\,i}}{r_{REF\,i}^2} \tag{6}$$

where ρ is the surface reflectivity (assumed constant), A_{RX} is the receive aperture area, and γ_{RX} is the combined optical efficiency and quantum efficiency. It is assumed here that the transmitted energy is distributed uniformly over the scene, and that the return energy spreads uniformly over a hemisphere. If $e_{RX\,i}$ is expressed in units of counts, then it is the expected number of detections for the i[th] pulse, across all pixels.

In simulating measurement data, a binary indicator variable is randomly generated, which represents whether each pulse generates a detection – called a hit – in a particular pixel. This variable has a Bernoulli distribution, with parameter $e_{RX\,i}/N_{pixels}$ expressed in units of counts. Low per pixel signal hit rate is assumed here, i.e., $e_{RX\,i}/N_{pixels} \ll 1$.

$$q_{RX\,i\,k} \sim Bernoulli\left(\frac{e_{RX\,i}}{N_{pixels}}\right) \tag{7}$$

This indicator variable is generated for every pixel (index k) and for every pulse (index i). The set of variables associated with each pulse is listed in Table 2.

TABLE 2

Properties of Pulses

Variable	Description
$t_{TX\,i}$	transmit time (center of pulse)
$e_{TX\,i}$	transmit energy
$t_{REF\,i}$	nominal reflection time (center of pulse)
$r_{REF\,i}$	nominal reflection range
$t_{RX\,i}$	nominal receive time (center of pulse)
$e_{RX\,i}$	nominal receive energy
$q_{RX\,i\,k}$	hit indicator for pixel k

3.3 MEASUREMENT DATA

Measurements are taken over the time interval [0, T]. The sensor data contains the time and pixel number of each hit, which may arise from signal or background noise. For simulation of signal hits for pixel k, the hit indicator variable $q_{RX\,i\,k}$ determines whether pulse i causes a hit. For each of these pulses with $q_{RX\,i\,k} = 1$, the hit time is modeled as

$$\zeta_{j\,|\,signal} = t_{RX\,i(j)} - \frac{2}{c}h(x_j, y_j) + v_j \tag{8}$$

where j is the hit index and $i(j)$ is the corresponding pulse index. v_j represents when the received photon was emitted relative to the center of the pulse, which is distributed as

$$v_j \sim Gaussian(0, \sigma_{PW}^2) \tag{9}$$

where σ_{PW}^2 is the variance representing the Gaussian pulse. Low per pixel signal hit rate is assumed here, i.e., $e_{RX\,i}/N_{pixels} \ll 1$, to avoid blocking-induced skewing of the pdf relative to the pulse shape.

The reflection coordinates at which the surface height is evaluated are defined as

$$x_j = g_x(t_{RX\,i(j)}) + j_x(t_{RX\,i(j)}) + c_x(n_j) + a_{x\,j} + b_{x\,j} \tag{10}$$

$$y_j = g_y\left(t_{RX\ i(j)}\right) + j_y\left(t_{RX\ i(j)}\right) + c_y\left(n_j\right) + a_{y\ j} + b_{y\ j} \tag{11}$$

Both of these equations are the sum of the following five terms:

- Reported boresight projection at the nominal receive time ($g_x\left(t_{RX\ i(j)}\right)$ and $g_y\left(t_{RX\ i(j)}\right)$)
- Jitter at the nominal receive time ($j_x\left(t_{RX\ i(j)}\right)$ and $j_y\left(t_{RX\ i(j)}\right)$)
- Offset of the pixel center relative to the boresight ($c_x\left(n_j\right)$ and $c_y\left(n_j\right)$)
- Random location of arrival within the pixel ($a_{x\ j}$ and $a_{y\ j}$)
- Optical blur ($b_{x\ j}$ and $b_{y\ j}$)

The functions $c_x\left(n_j\right)$ and $c_y\left(n_j\right)$ return the x and y coordinates, respectively, of the center of pixel n_j relative to the boresight. $a_{x\ j}$ and $a_{y\ j}$ are uniformly distributed random variables for the location of photon arrival within the pixel, relative to its center.

$$a_{x\ j} \sim Uniform\left(\frac{-\Delta x_{pixel}}{2}, \frac{+\Delta x_{pixel}}{2}\right) \tag{12}$$

$$a_{y\ j} \sim Uniform\left(\frac{-\Delta y_{pixel}}{2}, \frac{+\Delta y_{pixel}}{2}\right) \tag{13}$$

where Δx_{pixel} and Δy_{pixel} are the pixel widths, which will be equal for square pixels. $b_{x\ j}$ and $b_{y\ j}$ are Gaussian-distributed random variables representing optical blur. The Gaussian distribution is an approximation to an Airy function.

$$b_{x\ j} \sim Gaussian\left(0, \sigma_{blur}^2\right) \tag{14}$$

$$b_{y\ j} \sim Gaussian\left(0, \sigma_{blur}^2\right) \tag{15}$$

Background noise can come from dark counts or from sources outside the sensor that are uncorrelated with the pulses. Background noise is modeled as a Poisson point process, so the number of hits within the integration time for a given pixel is Poisson distributed:

$$N_{noise\ k} \sim Poisson\left(\ f_{noise\ k} T\right) \tag{16}$$

where $f_{noise\ k}$ is the background count rate for pixel k, and T is the integration time. The $N_{noise\ k}$ individual hit times are each independent and uniformly distributed over the integration time.

$$\zeta_{j\ |\ noise} \sim Uniform(0, T) \tag{17}$$

This is repeated to generate the noise hits for all pixels.

Once all signal and noise hits have been generated, they are merged by concatenating the two sequences, and then sorting into time order. The measurement data is the sequence of pixel number – hit time pairs for all hits within the integration time:

$$\{n_j, \zeta_j\}_{j=1:N}$$

TABLE 3

Properties of Hits

Variable	Description
ζ_j	hit time (time of detection of a photoelectron)
n_j	pixel number
s_j	source indicator (noise or signal)
x_j	x coordinate of surface reflection of photon (valid only for signal hits)
y_j	y coordinate of surface reflection of photon (valid only for signal hits)

This page intentionally left blank.

4. ESTIMATION APPROACH

4.1 STATE REPRESENTATION

The state is defined as a vector with three parts, representing the x jitter, y jitter, and surface height:

$$\theta = \begin{bmatrix} \theta_x \\ \theta_y \\ \theta_h \end{bmatrix} \tag{18}$$

The jitter and surface height functions are defined in a parametric form as the weighted sum of regularly spaced kernel functions, with the state vector containing the weights. The x and y jitter functions are

$$j_x(t;\theta_x) = \sum_i \theta_x(i)\varphi_{jitter}\left(\frac{t - t_{grid}(i)}{\Delta t_{grid}}\right) \tag{19}$$

$$j_y(t;\theta_y) = \sum_i \theta_y(i)\varphi_{jitter}\left(\frac{t - t_{grid}(i)}{\Delta t_{grid}}\right) \tag{20}$$

where t_{grid} is a vector of uniformly spaced times with interval Δt_{grid}. The jitter kernel is the triangle function, which makes the jitter functions equivalent to linear interpolation of the state vector elements.

$$\varphi_{jitter}(t) = \mathrm{tri}(t) = \begin{cases} 1 - |t| & |t| < 1 \\ 0 & otherwise \end{cases} \tag{21}$$

The sampling interval for defining the jitter sequences is chosen to be 10 pulses. At this time scale, the variation of the jitter random processes is very small, allowing accurate representation.

The height function is

$$h(x,y;\theta_h) = \sum_{i,j} \theta_h(i,j)\varphi_{surface}\left(\frac{x - x_{grid}(i)}{\Delta x_{grid}}, \frac{y - y_{grid}(j)}{\Delta y_{grid}}\right) \tag{22}$$

where x_{grid} and y_{grid} are uniformly spaced distance vectors with intervals Δx_{grid} and Δy_{grid}, respectively. The double indexing of the state vector should be interpreted as if the vector were reshaped into the size of the 2D grid. The surface kernel is a 2D Gaussian function, which can be expressed as the product of two 1D Gaussian functions:

$$\varphi_{surface}(x,y) = \varphi_G(x)\varphi_G(y) \tag{23}$$

$$\varphi_G(w) = \exp\left(-\frac{1}{2}\left(\frac{w}{\sigma}\right)^2\right) \tag{24}$$

The kernel width is determined by the standard deviation, which is chosen to be

$$\sigma = \frac{1}{2} \tag{25}$$

The lateral sampling interval for defining the surface is chosen to be 3 GSDs. This represents a compromise between good performance and fast/reliable convergence. A finer sampling grid would allow more accurate representation of the surface, which would potentially improve estimation performance at the expense of increased compute time. However, it has been observed that too fine of a sampling grid can greatly hinder algorithm convergence. Limitations on the lateral sampling interval are related to the size of the jitter and the spatial frequency content of the scene. Table 4 describes the size and spacing of the sampling grids, and therefore the sizes of the state vector components. The state vector is quite large, with multiple thousands of elements, so care must be taken in the algorithm implementation.

TABLE 4

Sampling Grids and Interpolation Kernels

	Dimension	Sampling interval	Kernel
x jitter	1401	10 pulses	triangular (linear interpolation)
y jitter	1401	10 pulses	triangular (linear interpolation)
surface	70 × 70	3 GSDs	2D Gaussian (σ = 1/2 sampling interval)

4.2 PRIORS

The jitter and surface height functions are expected to be slowly varying, and this represents prior information that can be exploited. The jitter functions are modeled as two independent uniformly sampled 1st order Gauss-Markov processes, so the θ_x and θ_y parts of the state vector are i.i.d. (independent, identically distributed) Gaussian random vectors with zero mean.

$$\theta_x \sim Gaussian(0, Q_x) \tag{26}$$

$$\theta_y \sim Gaussian(0, Q_y) \tag{27}$$

$$Q_x = Q_y \tag{28}$$

The covariance matrix is large and non-sparse, but its inverse is conveniently sparse. It is a tridiagonal matrix with only three unique non-zero values, given by

$$Q_x^{-1} = Q_y^{-1} = \begin{cases} \dfrac{1}{(1-\rho^2)\sigma^2} & first\ and\ last\ diagonal\ elements \\ \dfrac{1+\rho^2}{(1-\rho^2)\sigma^2} & elsewhere\ on\ diagonal \\ \dfrac{-\rho}{(1-\rho^2)\sigma^2} & on\ first\ off-diagonals \\ 0 & elsewhere \end{cases} \tag{29}$$

where ρ is the correlation coefficient between adjacent samples, given by

$$\rho = \exp\left(\frac{-\left|\Delta t_{grid}\right|}{\tau}\right) \tag{30}$$

The variance σ^2 and correlation time τ are chosen to match the expected PSD of the jitter.

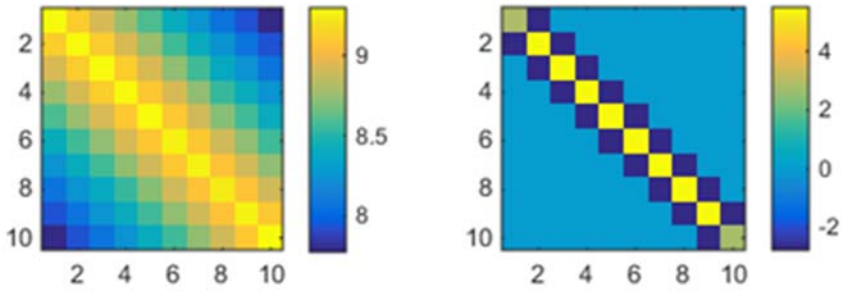

Figure 5. Covariance (left) and inverse covariance (right) of 10 samples of a 1D 1st order Gauss-Markov process.

The surface height function is modeled as a 2D uniformly sampled 1st order Gauss-Markov process (Markov random field). The vectorized array of height coefficients θ_h is also a Gaussian random vector.

$$\theta_h \sim Gaussian(\bar{\theta}_h, Q_h) \tag{31}$$

The mean, $\bar{\theta}_h$, might be set by some preexisting knowledge of the surface, such as a digital elevation model (DEM), or it might be zero. The covariance is the Kronecker product of two matrices:

$$Q_h = \left(\frac{1}{\sigma} Q_{hy}\right) \otimes \left(\frac{1}{\sigma} Q_{hx}\right) \tag{32}$$

Here, Q_{hx} and Q_{hy} are the covariance matrices of 1D slices of the height coefficient array along the x and y directions, respectively. They have the same form as the jitter prior covariances Q_x and Q_y but different values for σ^2 and ρ. The variance is chosen to match the expected height variation in the scene. The correlation coefficient is computed in a similar way as for the jitter, except using a distance interval and correlation distance rather than a time interval and correlation time. The correlation distance is also chosen to represent the expected scene. Q_{hx} and Q_{hy} should share the same parameters (σ^2 and ρ), but will differ in size if the height coefficient array is not square. The composite covariance Q_h is again non-sparse, but its inverse is sparse. Q_h^{-1} is a 9-banded array with six unique non-zero values. Since the inverse of a Kronecker product is the Kronecker product of the inverses, the inverse covariance can be written as

$$Q_h^{-1} = \left(\sigma Q_{hy}^{-1}\right) \otimes \left(\sigma Q_{hx}^{-1}\right) \tag{33}$$

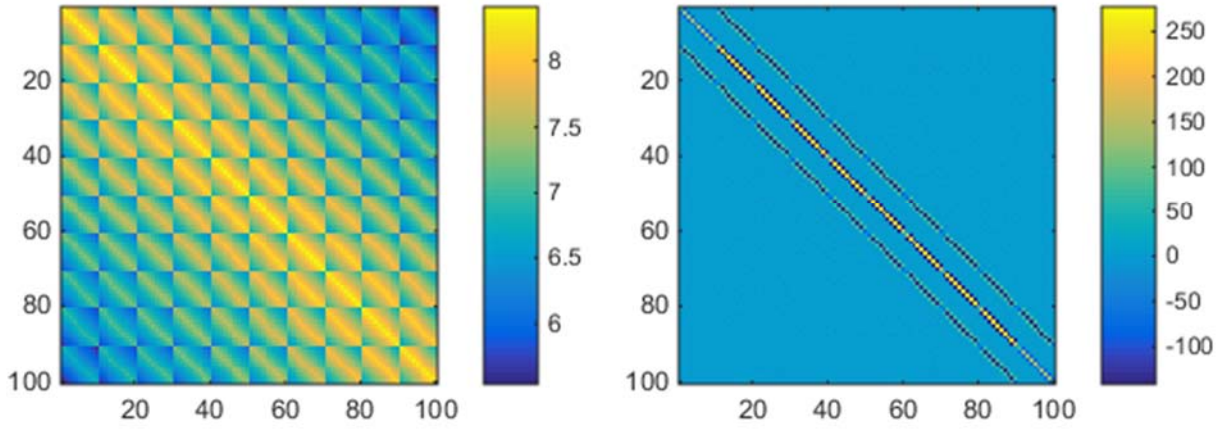

Figure 6. Covariance (left) and inverse covariance (right) of a 10 × 10 sample array of a 2D 1st order Gauss-Markov process

Since the three parts of the state vector are independent (in the absence of measurements), the total prior can be written as a product:

$$p(\theta) = p(\theta_x) p\left(\theta_y\right) p(\theta_h) \tag{34}$$

The product and each of the three terms is a multivariate Gaussian with the form

$$p(\theta) = Gaussian(\bar{\theta}, Q) = \exp\left(-\frac{1}{2}[\log(|2\pi Q|) + (\theta - \bar{\theta})^T Q^{-1}(\theta - \bar{\theta})]\right) \quad (35)$$

4.3 EM ALGORITHM

The state estimation uses the Expectation Maximization (EM) algorithm to deal with the uncertainty of whether each hit arose from noise or signal, in a robust way. "Noise" and "signal" are considered to be two components in a mixture pdf. The EM algorithm estimates the weights (prior probabilities) of the components and the parameters of each component. For this application, only the signal component has parameters to estimate, since the pdf for the noise component is uniform over a predefined interval. A key aspect of EM is the estimation of the "membership" of each measurement. This is a probability vector for each hit, representing whether it is noise or signal.

The steps in EM are

$$\{w, \theta\} = initialize(\quad)$$

$$m = E\_step(w, \theta)$$

$$\{w, \theta\} = M\_step(m, \theta)$$

Here, w is the estimated prior probability vector of the components, θ is the estimated state vector, and m is the estimated membership array. All three steps have access to the measurements and assorted parameters. After initialization, the E and M steps are cycled through repeatedly until a stopping criterion is met. Given enough iterations, the EM algorithm is guaranteed to converge to a local maximum of the likelihood function. In many applications of EM, the M step has a simple closed-form solution for minimizing its cost function. When this is not the case, as in this application, Generalized Expectation Maximization (GEM) can be used, which settles for a mere reduction in the cost function. GEM does not have the local maximum guarantee, but still has good convergence properties. To determine when to stop iterating, any of the following criteria or a combination of them can be used:

- Change in θ
- Change in m
- Change in total likelihood of data (available in E step)
- Number of iterations

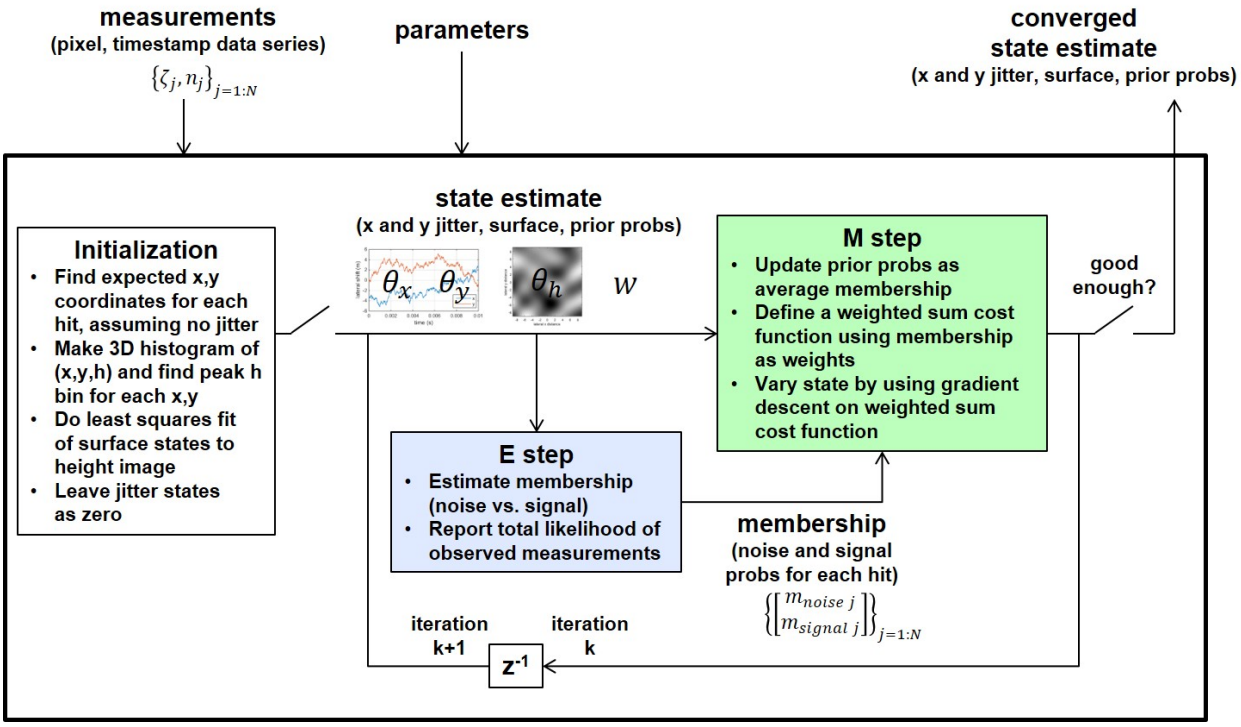

Figure 7. EM algorithm flow.

4.4 INITIALIZATION

Algorithm performance and speed are both improved by a careful initialization. The jitter states θ_x and θ_y are initialized as zeros. The height states θ_h are initialized by forming a height image without jitter compensation, and doing a least squares fit to the image.

For each hit, the measured height is calculated as

$$h_j = -\frac{c}{2}\left(\zeta_j - t_{RX\,i(j)}\right) \tag{36}$$

$i(j)$ is defined here as the index of the nearest pulse to the observed hit time, that is:

$$i(j) = \underset{i}{\operatorname{argmin}}\left(\left|\zeta_j - t_{RX\,i}\right|\right) \tag{37}$$

The expected ground sample coordinates are calculated as:

$$\bar{x}_j = g_x\left(t_{RX\,i(j)}\right) + c_x\left(n_j\right) \tag{38}$$

$$\bar{y}_j = g_y\left(t_{RX\,i(j)}\right) + c_y\left(n_j\right) \tag{39}$$

Next, a 3D histogram of the ($\bar{x}_j$, $\bar{y}_j$, h_j) triplets is made. The "measured" height image is formed by taking the height bin with the most counts for each x,y bin. The choice of bin sizes is affected by signal-to-noise ratio (SNR). If the SNR is very low, larger bins can be used to improve the peak finding, at the expense of resolution.

The height function from equation (**22**) can be rewritten as a dot product, and then used to write a set of measurement equations. The scalar measurement equation for the kth element of the measured height image is:

$$h_k = \left(\Phi_y(y_k) \otimes \Phi_x(x_k)\right)^T \theta_h + v_k \tag{40}$$

where the functions $\Phi_x(x)$ and $\Phi_y(y)$ are vectors of shifted 1D kernel functions given in the appendix. These functions are evaluated at x_k and y_k, which are the x,y coordinates of the center of element k of the height image. v_k represents measurement noise, which can be assumed i.i.d. unless there are reasons for it to be otherwise:

$$v_k \sim Gaussian(0, \sigma^2) \tag{41}$$

The variance σ^2 represents the error of the measured height image, and is related to the bin size.

The prior information about the surface can be easily incorporated into the initialization. This has the added benefit of ensuring that the system of equations is fully determined. The prior provides one equation for each element of the state vector to be estimated. In vector form, the equation for the prior is:

$$\bar{\theta}_h = I\theta_h + w \tag{42}$$

where w represents Gaussian noise distributed as

$$w \sim Gaussian(0, Q_h) \tag{43}$$

The combined set of equations (with noise statistics) representing the measurements and prior information can now be solved using linear least squares to get the initial estimate of θ_h. This set of equations can be large, so it may be helpful to exploit the (approximate) sparsity of the measurements, and to use reduced precision arithmetic.

4.5 E STEP

The purpose of the E step is to update the membership of each hit. To define membership, first consider a discrete variable s_j that indicates whether a particular hit arose from noise or signal.

$$s_j \in \{noise, signal\} \tag{44}$$

This indicator variable is not known, but can be estimated. Define the membership of hit j as a vector m_j that represents the conditional probability distribution of s_j:

$$m_j = \begin{bmatrix} m_{noise\ j} \\ m_{signal\ j} \end{bmatrix} = \begin{bmatrix} P(s_j = noise|\zeta_j) \\ P(s_j = signal|\zeta_j) \end{bmatrix} \tag{45}$$

The elements of the membership vector are given by Bayes' rule:

$$P(s_j|\zeta_j) = \frac{p(\zeta_j|s_j)P(s_j)}{\sum_s p(\zeta_j|s_j)P(s_j)} \tag{46}$$

The calculations in the E step are implicitly conditioned on the current estimate $\hat{\theta}$ of the state vector, and use the current values of prior probabilities $P(s_j)$, which are the elements of the weight vector:

$$w = \begin{bmatrix} w_{noise} \\ w_{signal} \end{bmatrix} = \begin{bmatrix} P(s_j = noise) \\ P(s_j = signal) \end{bmatrix} \tag{47}$$

If the source is noise, then the hit time is simply uniform over the integration time interval:

$$p(\zeta_j|s_j = noise) = \frac{1}{T} \tag{48}$$

If the source is signal, then the hit time is distributed as a Gaussian sum, assuming that pulse shape is Gaussian. For pulses that are narrow relative to the pulse spacing, taking only the nearest Gaussian is a very good approximation:

$$\begin{aligned} p(\zeta_j|s_j = signal) &= \sum_i u_i \exp\left(-\frac{1}{2}\left[\log(2\pi\sigma_{t\,i}^2) + \left(\frac{\zeta_j - \bar{t}_i}{\sigma_{t\,i}}\right)^2\right]\right) \\ &\approx u_{i(j)} \exp\left(-\frac{1}{2}\left[\log(2\pi\sigma_{t\,i(j)}^2) + \left(\frac{\zeta_j - \bar{t}_{i(j)}}{\sigma_{t\,i(j)}}\right)^2\right]\right) \end{aligned} \tag{49}$$

Another assumption here is that the height variation in the scene is significantly less than the pulse spacing.

The weights of the Gaussian sum are the normalized receive energies:

$$u_i = \frac{e_{RX\,i}}{\sum_i e_{RX\,i}} \tag{50}$$

If the range is not changing significantly over the integration time, and other things remain constant, then the weights would all be approximately equal to the reciprocal of the number of pulses within the integration time interval.

The mean of the Gaussian distribution is

$$\bar{t}_{i(j)} = t_{RX\,i(j)} - \frac{2}{c} h\left(\bar{x}_j, \bar{y}_j; \theta_h\right) \tag{51}$$

$\bar{x}_j$ and $\bar{y}_j$ are the mean coordinates of where the photon reflected off the surface (this time treating jitter as a known quantity):

$$\bar{x}_j = g_x\left(t_{RX\,i(j)}\right) + j_x\left(t_{RX\,i(j)}; \theta_x\right) + c_x\left(n_j\right) \tag{52}$$

$$\bar{y}_j = g_y\left(t_{RX\,i(j)}\right) + j_y\left(t_{RX\,i(j)}; \theta_y\right) + c_y\left(n_j\right) \tag{53}$$

The variance of the Gaussian distribution is the sum of the variance representing the Gaussian pulse shape and a term representing the broadening of the pulse due to the height variation over the lateral uncertainty of the measurement. The effect is that of convolving the pulse with the impulse response of the surface. The translation of lateral uncertainty into height uncertainty uses a local linear (planar) approximation to the surface, as illustrated in Figure 8. Accordingly, the lateral variances are multiplied by the squared gradients of the height function. The resulting variance of the hit time is

$$\sigma_{t\,i(j)}^2 \approx \sigma_{PW}^2 + \left(\frac{2}{c}\right)^2 \left(H_x^2\left(\bar{x}_j, \bar{y}_j; \theta_h\right)\sigma_x^2 + H_y^2\left(\bar{x}_j, \bar{y}_j; \theta_h\right)\sigma_y^2\right) \tag{54}$$

The lateral variances are the sum of the variance of the uniform location of arrival and the variance of the optical blur.

$$\sigma_x^2 = \frac{\Delta x_{pixel}^2}{12} + \sigma_{blur}^2 \tag{55}$$

$$\sigma_y^2 = \frac{\Delta y_{pixel}^2}{12} + \sigma_{blur}^2 \tag{56}$$

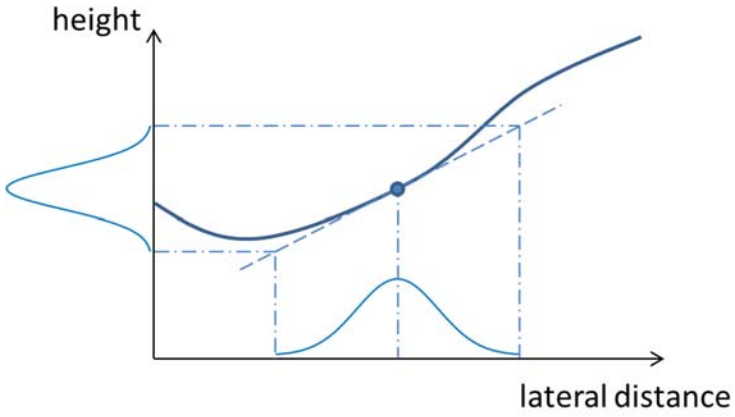

Figure 8. Translation of lateral uncertainty into height uncertainty via linear approximation of surface.

The pdf can alternately be expressed in terms of relative range rather than absolute time, by doing a change of variables:

$$p(\zeta_j|s_j = signal) \approx u_{i(j)}\frac{c}{2}\exp\left(-\frac{1}{2}\left(\log\left(2\pi\sigma_{r\,i(j)}^2\right)+\left(\frac{\Delta r_j}{\sigma_{r\,i(j)}}\right)^2\right)\right) \tag{57}$$

which is a function of the range difference and range uncertainty, given by:

$$\Delta r_j = \frac{c}{2}\left(\zeta_j - \bar{t}_{i(j)}\right) = \frac{c}{2}\left(\zeta_j - t_{RX\,i(j)}\right) + h\left(\bar{x}_j, \bar{y}_j; \theta_h\right) \tag{58}$$

$$\sigma_{r\,i(j)} = \frac{c}{2}\sigma_{t\,i(j)} = \sqrt{\left(\frac{c}{2}\sigma_{PW}\right)^2 + H_x^2\left(\bar{x}_j, \bar{y}_j; \theta_h\right)\sigma_x^2 + H_y^2\left(\bar{x}_j, \bar{y}_j; \theta_h\right)\sigma_y^2} \tag{59}$$

4.6 M STEP

The M step updates the state vector estimate and the weights that represent the prior probabilities of noise and signal. The weights are updated by taking the maximum likelihood estimate, which is found by simply averaging the membership across all measurements:

$$w = \frac{1}{N}\sum_j m_j \tag{60}$$

In EM, each measurement can come from one of multiple components of a mixture pdf. Generally, each component has a set of parameters that are updated in the M step. In this application, only the signal component has parameters (i.e., elements of the state vector) to update.

The state vector estimate is updated by using gradient descent to reduce a weighted sum cost function, where the weights are the membership for the signal component. There are three reasons for settling for a reduction in the cost function rather than a minimization. The first is that there is no known closed-form solution. The second is that the cost function is relatively expensive to evaluate. The third is that the cost function becomes less valid as the distance from the current state estimate increases. The cost function is refreshed by the E step. Therefore, it makes sense to take smaller steps and update the cost function frequently. Reducing rather than minimizing the cost function in the M step makes this algorithm Generalized Expectation Maximization (GEM).

The cost function, which is proportional to the negative log of the posterior probability density given that the source is signal, is

$$\Psi(\theta) = -\log\big(p(\theta)\big) - \sum_j m_{signal\ j} \log\Big(p\big(\zeta_j|\theta, s_j = signal\big)\Big) \tag{61}$$

$p(\theta)$ is the Gaussian prior pdf, as described previously. $p\big(\zeta_j|\theta, s_j = signal\big)$ was also given previously in equation (**57**), but the conditioning on the state vector was implicit. Gradient descent is used to take a step that reduces the cost function. This requires the partial derivatives of the cost function with respect to the state:

$$\frac{\partial \Psi}{\partial \theta} = \frac{\partial}{\partial \theta}\big(-\log\big(p(\theta)\big)\big) + \sum_j m_{signal\ j} \frac{\partial}{\partial \theta}\Big(-\log\Big(p\big(\zeta_j|\theta, s_j = signal\big)\Big)\Big) \tag{62}$$

The partial derivatives of the term representing the prior are

$$\begin{aligned} \frac{\partial}{\partial \theta}\big(-\log\big(p(\theta)\big)\big) &= (\theta - \bar{\theta})^T Q^{-1} \\ &= (\theta_x - \bar{\theta}_x)^T Q_x^{-1} \frac{\partial \theta_x}{\partial \theta} + \big(\theta_y - \bar{\theta}_y\big)^T Q_y^{-1} \frac{\partial \theta_y}{\partial \theta} + (\theta_h - \bar{\theta}_h)^T Q_h^{-1} \frac{\partial \theta_h}{\partial \theta} \end{aligned} \tag{63}$$

where $\bar{\theta}$ and Q are the mean and covariance of the prior, respectively. It should be noted that the prior and its partial derivatives can be evaluated without explicitly representing the covariance matrices or their inverses.

The partial derivatives in the sum are

$$\frac{\partial}{\partial\theta}\left(-\log\left(p\left(\zeta_j|\theta,s_j=signal\right)\right)\right)=-\frac{\partial}{\partial\theta}\log\left(u_{i(j)}\frac{c}{2}\right)+\frac{1}{2}\frac{\partial}{\partial\theta}\left(\log\left(2\pi\sigma_{r\,i(j)}^2\right)+\left(\frac{\Delta r_j}{\sigma_{r\,i(j)}}\right)^2\right)$$
$$=\frac{1}{\sigma_{r\,i(j)}}\left(\left(1-\left(\frac{\Delta r_j}{\sigma_{r\,i(j)}}\right)^2\right)\frac{\partial\sigma_{r\,i(j)}}{\partial\theta}+\frac{\Delta r_j}{\sigma_{r\,i(j)}}\frac{\partial\Delta r_j}{\partial\theta}\right) \tag{64}$$

where the partial derivatives of the range difference and range standard deviation are

$$\frac{\partial\Delta r_j}{\partial\theta}=\frac{\partial h}{\partial\theta} \tag{65}$$

$$\frac{\partial\sigma_{r\,i(j)}}{\partial\theta}=\frac{1}{\sigma_{r\,i(j)}}\left(\sigma_x^2H_x\frac{\partial H_x}{\partial\theta}+\sigma_y^2H_y\frac{\partial H_y}{\partial\theta}\right) \tag{66}$$

The height gradients and remaining partial derivatives are given in the appendix. This completes the derivation of the gradient of the cost function.

Gradient descent is used to improve the state estimate by trying various step sizes in the opposite direction of the gradient. A step size multiplier α determines the new state vector, at which the cost function is evaluated.

$$\hat{\theta}_{test}=\hat{\theta}_k-\alpha\frac{\partial\Psi}{\partial\theta} \tag{67}$$

If the cost is improved, then the previous state estimate is replaced with the new one, and the M step is complete. Otherwise, the next value of α in a descending logarithmically spaced list is tried.

$$\hat{\theta}_{k+1}=\hat{\theta}_{test}\ \ if\ \ \Psi\left(\hat{\theta}_{test}\right)<\Psi\left(\hat{\theta}_k\right) \tag{68}$$

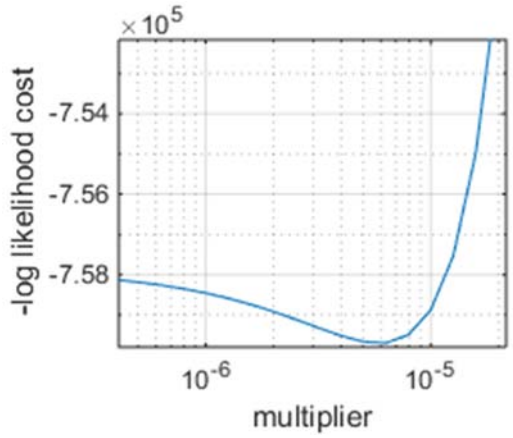

Figure 9. Example of cost function vs. step size multiplier.

Normally in EM, each component has its own cost function, and each cost function can be minimized or reduced independently. In this implementation, because only the signal component has parameters for the M step to update, it is unnecessary to evaluate the M step's cost function $\Psi(\theta)$. Instead, the negative log of the posterior pdf of the state (which is a by-product of the E step) is used in the M step to determine acceptance of a new state estimate. So, in the implementation, the E and M steps are effectively merged.

This page intentionally left blank.

5. RESULTS

The algorithm was tested with simulated data that was generated with the parameters listed in Table 1. The algorithm was written in vectorized MATLAB code, using its gpuArray functionality. To drastically improve computation time, when evaluating the height function or its gradients, only the kernels for the nearest 5 by 5 array of coefficients were evaluated. This is a good approximation because of the rapid decay of the 2D Gaussian kernel. The code was run on an LLGrid node with an Intel Xeon E5 processor and NVIDIA Tesla K80 GPU. The algorithm converged after approximately 80 minutes. This is slow, considering that it is only processing one FOV of the scene, but there are ways in which it can be sped up.

Figure 10 shows the output of a simple height image formation algorithm that compensates for jitter using the estimates from three different iterations representing initialization, partial convergence, and full convergence. The image formation algorithm is the same histogram approach described in the initialization section, except using equations (**52**) and (**53**) in place of (**38**) and (**39**). Plotted at the center of the height images is the residual jitter path over the integration time. The residual jitter path is roughly analogous to a blur function that gets convolved with the height image. Below the height images are height profiles that cut diagonally across the images from lower left to upper right. In the height profile images, both the estimate and the truth are shown.

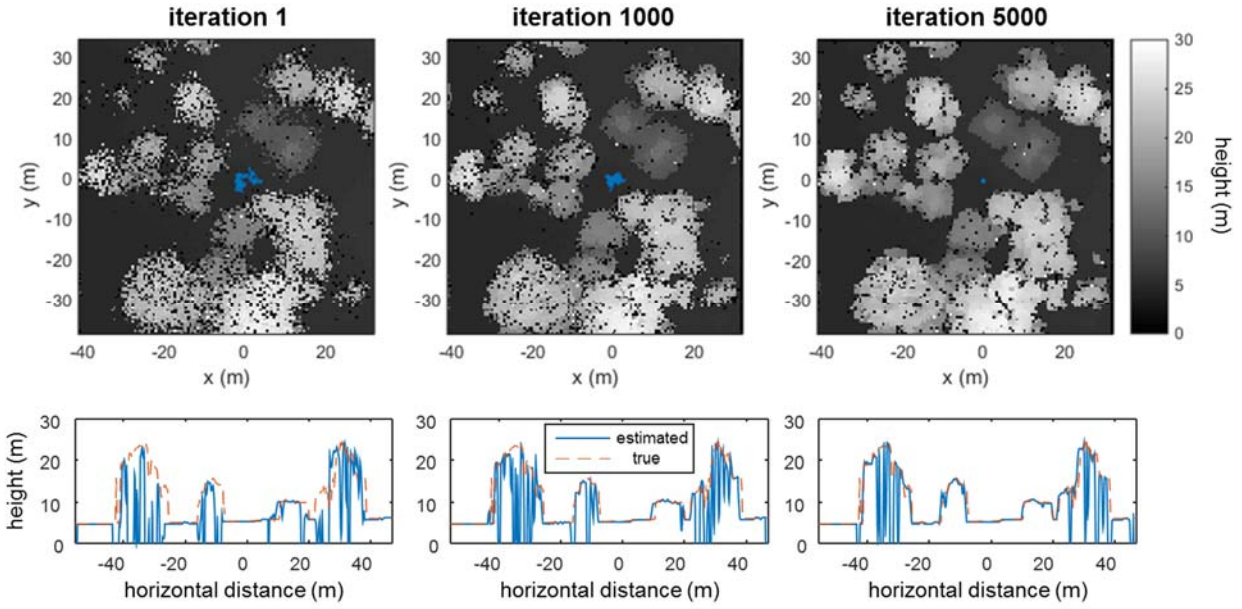

Figure 10. Surface estimation results.

In the first image, the residual jitter is the actual jitter (with a sign reversal), since the estimate is zero. The jitter spans multiple pixels, and the image is very blurred and does not have distinct edges. In the second image, representing partial convergence, the residual jitter has shrunk significantly, and the image is correspondingly less blurry. Some smaller objects appear that were not apparent before. In the third image, with full convergence, the residual jitter has shrunk to smaller than one GSD (pixel). Edges are distinct and the number of artificial holes has been reduced. The rectangular shapes and tapered roof of the building also stand out. The height profiles show the same progression of convergence to the true height.

Figure 11 shows the progression of the jitter estimates as the algorithm converges. In the upper left corner, the standard deviation of the residual x and y jitter is shown vs. iteration number. This metric is representative of the amount of blur that the residual jitter will cause. Vertical lines are drawn that correspond to the three height images from Figure 10. The standard deviation of the residual jitter can be seen to decrease by about an order of magnitude in the x and y dimensions.

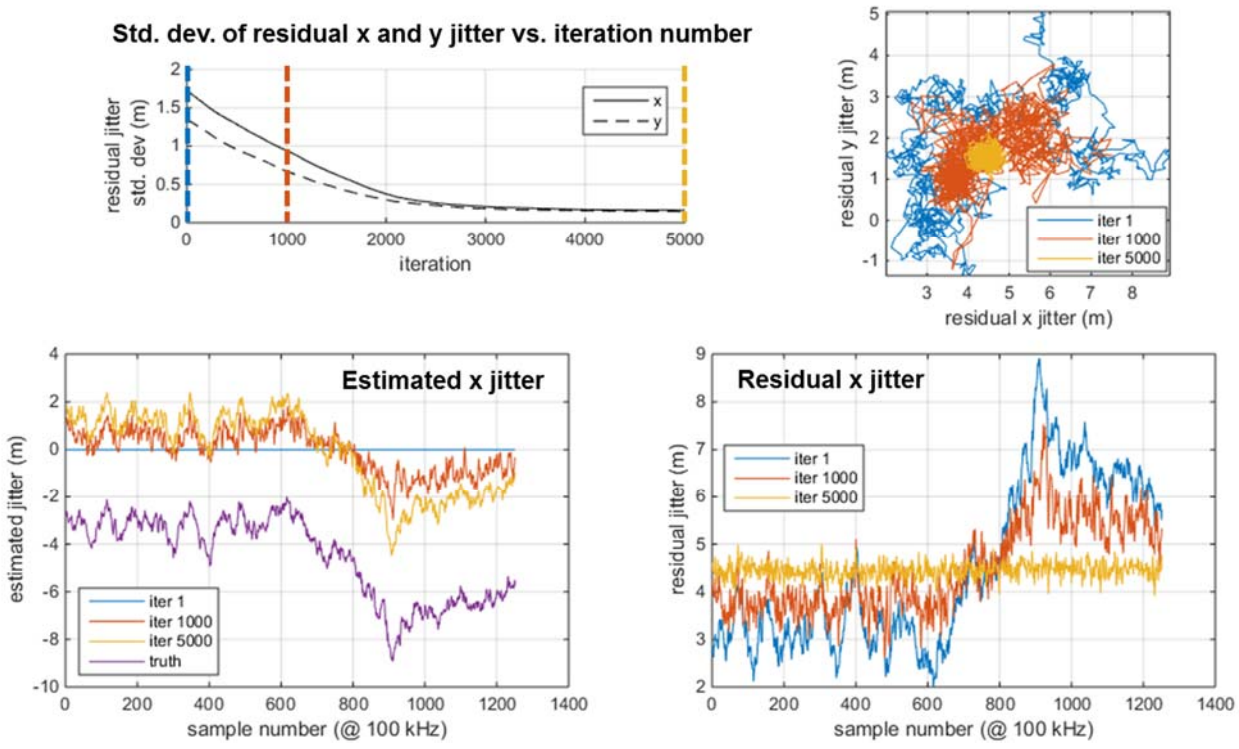

Figure 11. Jitter estimation results.

At the lower left is the estimated x jitter time sequence at the three different iterations, and the true x jitter. The jitter estimate starts as zero, and gradually converges to the shape of the true jitter. The truth has a DC offset, which is unobservable because of the limited observation time. The DC offset does not cause any image blurring, only an error in absolute lateral registration of the height image. It is relevant when trying to merge images or jitter estimates made from different integration time intervals, but it can be mitigated by image correlation.

In the lower right is the residual x jitter time sequence at the three different iterations. The residual jitter initially matches the negative of the truth. After convergence, only the DC offset and a small amount of high-frequency noise remain. At the upper right, the x and y residual jitter are plotted against each other, showing the "blur functions" shrinking as the algorithm converges.

This page intentionally left blank.

6. DISCUSSION

Initial tests of the jitter-estimation algorithm have been promising. Using the algorithm in a scenario in which the RMS jitter was about triple the pixel size significantly improved the quality of the image. The jitter-compensated image is much clearer and would be much more valuable for object recognition, whether performed by humans or computers. Future work is planned to more comprehensively characterize the performance of the registration algorithm as functions of jitter amplitude and spectral content, SNR, and ground surface characteristics.

The current implementation is still very slow; the utility would increase greatly if the algorithm could run faster. Future work is planned to speed the algorithm up by about an order of magnitude. Several possible approaches will be pursued. One is to develop an optimized implementation on a GPU. Another possibility is to find a better search algorithm than gradient descent or a better method for choosing step size. A third option is to iterate the grid sizes (and thus the state vector size) from coarse to fine, initializing each iteration with the result of the previous one.

This page intentionally left blank.

APPENDIX
GRADIENTS AND DERIVATIVES

The E and M steps require the gradient of the height function, and the M step also requires partial derivatives of the jitter and height functions with respect to the state. The jitter and height functions can be written in a compact form, as inner products between a coefficient vector and part of the state vector:

$$j_x = \Phi_j^T \theta_x \tag{69}$$

$$j_y = \Phi_j^T \theta_y \tag{70}$$

$$h = \left(\Phi_y \otimes \Phi_x\right)^T \theta_h \tag{71}$$

The terms in the first and second gradients of the height function are

$$H_x = \left(\Phi_y \otimes \Phi_x'\right)^T \theta_h \tag{72}$$

$$H_y = \left(\Phi_y' \otimes \Phi_x\right)^T \theta_h \tag{73}$$

$$H_{xx} = \left(\Phi_y \otimes \Phi_x''\right)^T \theta_h \tag{74}$$

$$H_{xy} = \left(\Phi_y' \otimes \Phi_x'\right)^T \theta_h \tag{75}$$

$$H_{yy} = \left(\Phi_y'' \otimes \Phi_x\right)^T \theta_h \tag{76}$$

For the jitter functions, if the jitter time grid has L samples, the coefficient vector is

$$\Phi_j = \begin{bmatrix} \varphi_{jitter}\left(\dfrac{t - t_{grid\,1}}{\Delta t_{grid}}\right) \\ \vdots \\ \varphi_{jitter}\left(\dfrac{t - t_{grid\,L}}{\Delta t_{grid}}\right) \end{bmatrix} \tag{77}$$

The coefficient vector for the height function is a Kronecker product of two vectors. If the grid of height coefficients is M by N, the two vectors and their first and second derivatives are

$$\Phi_x = \begin{bmatrix} \varphi_G\left(\frac{x - x_{grid\,1}}{\Delta x_{grid}}\right) \\ \vdots \\ \varphi_G\left(\frac{x - x_{grid\,M}}{\Delta x_{grid}}\right) \end{bmatrix} \tag{78}$$

$$\Phi'_x = \frac{1}{\Delta x_{grid}} \begin{bmatrix} \varphi'_G\left(\frac{x - x_{grid\,1}}{\Delta x_{grid}}\right) \\ \vdots \\ \varphi'_G\left(\frac{x - x_{grid\,M}}{\Delta x_{grid}}\right) \end{bmatrix} \tag{79}$$

$$\Phi''_x = \frac{1}{\Delta x_{grid}^2} \begin{bmatrix} \varphi''_G\left(\frac{x - x_{grid\,1}}{\Delta x_{grid}}\right) \\ \vdots \\ \varphi''_G\left(\frac{x - x_{grid\,M}}{\Delta x_{grid}}\right) \end{bmatrix} \tag{80}$$

$$\Phi_y = \begin{bmatrix} \varphi_G\left(\frac{y - y_{grid\,1}}{\Delta y_{grid}}\right) \\ \vdots \\ \varphi_G\left(\frac{y - y_{grid\,N}}{\Delta y_{grid}}\right) \end{bmatrix} \tag{81}$$

$$\Phi'_y = \frac{1}{\Delta y_{grid}} \begin{bmatrix} \varphi'_G\left(\frac{y - y_{grid\,1}}{\Delta y_{grid}}\right) \\ \vdots \\ \varphi'_G\left(\frac{y - y_{grid\,N}}{\Delta y_{grid}}\right) \end{bmatrix} \tag{82}$$

$$\Phi_y'' = \frac{1}{\Delta y_{grid}^2}\begin{bmatrix} \varphi_G''\left(\frac{y - y_{grid\,1}}{\Delta y_{grid}}\right) \\ \vdots \\ \varphi_G''\left(\frac{y - y_{grid\,N}}{\Delta y_{grid}}\right) \end{bmatrix} \tag{83}$$

The 1D Gaussian kernel and its first and second derivatives are

$$\varphi_G(w) = \exp\left(-\frac{1}{2}\left(\frac{w}{\sigma}\right)^2\right) \tag{84}$$

$$\varphi_G'(w) = -\frac{w}{\sigma^2}\exp\left(-\frac{1}{2}\left(\frac{w}{\sigma}\right)^2\right) \tag{85}$$

$$\varphi_G''(w) = \frac{w^2 - \sigma^2}{\sigma^4}\exp\left(-\frac{1}{2}\left(\frac{w}{\sigma}\right)^2\right) \tag{86}$$

The width of the Gaussian kernel is set by

$$\sigma = \frac{1}{2} \tag{87}$$

The required partial derivatives of the jitter functions and the height function and its gradient are

$$\frac{\partial j_x}{\partial \theta} = \Phi_j^T \frac{\partial \theta_x}{\partial \theta} \tag{88}$$

$$\frac{\partial j_y}{\partial \theta} = \Phi_j^T \frac{\partial \theta_y}{\partial \theta} \tag{89}$$

$$\frac{\partial h}{\partial \theta} = H_x \frac{\partial j_x}{\partial \theta} + H_y \frac{\partial j_y}{\partial \theta} + \left(\Phi_y \otimes \Phi_x\right)^T \frac{\partial \theta_h}{\partial \theta} \tag{90}$$

$$\frac{\partial H_x}{\partial \theta} = H_{xx} \frac{\partial j_x}{\partial \theta} + H_{xy} \frac{\partial j_y}{\partial \theta} + \left(\Phi_y \otimes \Phi_x'\right)^T \frac{\partial \theta_h}{\partial \theta} \tag{91}$$

$$\frac{\partial H_y}{\partial \theta} = H_{xy} \frac{\partial j_x}{\partial \theta} + H_{yy} \frac{\partial j_y}{\partial \theta} + \left(\Phi_y' \otimes \Phi_x\right)^T \frac{\partial \theta_h}{\partial \theta} \tag{92}$$

This page intentionally left blank.

BIBLIOGRAPHY

"The Expectation Maximization Algorithm," Frank Dellaert, GVU Center; College of Computing; Georgia Tech, GIT-GVU-02-20, 2002

Dempster, A., Laird, N., and Rubin, D. (1977). "Maximum likelihood from incomplete data via the EM algorithm." *Journal of the Royal Statistical Society*, Series B, 39(1):1–38.

Hartley, H. (1958). "Maximum likelihood estimation from incomplete data." *Biometrics*, 14:174–194.

McLachlan, G. and Krishnan, T. (1997). "The EM algorithm and extensions." Wiley series in probability and statistics. John Wiley & Sons.

This page intentionally left blank.