Adaptive Signal Processing Laboratory (ASPL)
Electrical and Computer Engineering Department
University of Florida

Initial Simulator Code for ALSM

ASPL Report No. Rep_2004-12-002

Puskal Pokharel, K. Clint Slatton

December 12, 2004

Point of Contact:
Prof. K. Clint Slatton
University of Florida; PO Box 116130; Gainesville, FL 32611
Tel: 352.392.0634, Fax: 352.392.0044, E-mail: slatton@ece.ufl.edu

Problem statement:

We need the ability to simulate basic ALSM functions so that we can investigate possible distribution of points under different terrain and landcover scenarios. It will also help us investigate artifacts in the ALSM data, such as the “corduroy pattern”.

This work was started by Puskal Pokharel puskal@ufl.edu in the spring of 2004. It was briefly picked up in the summer by "Kuei-Tsung Shih (Philips)" ktshih@ufl.edu, but he quickly moved to spectral analysis of the corduroy pattern, so he didn’t make any changes to the simulator.

The write up in this report is the final summary and Matlab code listing provided by Puskal in the summer of 2004.

A Summary of My Work

This small summary is to introduce the work I did in the ASPL to anyone interested. For those who are not familiar with the basic mechanics of the scanning process in ALSM, I first present a small introduction:

The initial objectives for sensor modeling in ALSM have been the following:

- Explain ALSM working in simple terms
- Identify effects of various parameters like mirror sampling
- Relate the simulation results to what is seen in actual ALSM data
- Identify the causes for artifacts in the data
- Look for remedies for the artifacts

The following figures illustrate the scanning geometry of the ASLM. They also give the meanings of the symbols that shall be used later.

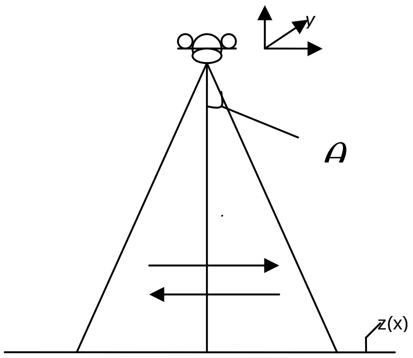

Figure 1: the plane moves in the y-direction and mirror shoots laser shots to the ground at an angle θ as it oscillates in the x-direction, scanning the

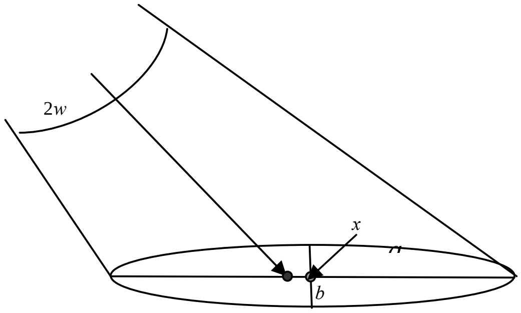

Figure 2: the laser spreads out with a divergence of 2w and makes an elliptical footprint with semi-axes a

The following equations quantify the geometry:

$$a = \tfrac{1}{2} \cdot h \cdot [(\tan(\theta + w) - \tan(\theta - w)]$$

$$x = \tfrac{1}{2} \cdot h \cdot [(\tan(\theta + w) + \tan(\theta - w)]$$

$$x + \Delta x = \tfrac{1}{2} \cdot h \cdot [\tan(\theta + \Delta\theta + w) + \tan(\theta + \Delta\theta - w)]$$

$$b = \frac{a \cdot h \cdot \tan(w)}{\cos(\theta)[\sqrt{a^2 - (h \cdot \tan\theta - x)^2}\,]}$$

$$A = \pi \cdot a \cdot b$$

$$\Delta\theta = \frac{4 \times ScanR \times ScanA}{PulseR}$$

$$x = h \tan\theta$$

$$x + \Delta x = h \tan(\theta + \Delta\theta)$$

Here *x, a, b,* θ and *h* have the meanings shown by the previous diagrams. $\Delta\theta$ is the angular separation between two laser shots, Δx is the corresponding linear separation of the footprints on the ground *A* is the area of a footprint. These equations have been derived using basic geometry and simple projection principles.

Having given a succinct review of the ALSM geometry, I'll now provide a list of programs developed during my work in the Lab.

- ✓ Simple test for the idea of modeling lidar scatter from a rough surface as the convolution of the transmitted waveform and a surface reflectivity function. (Lidar Scatter.m)

- ✓ Simulation of the working of ASLM using simple geometry to show the variation in size and patterns. This is included in the codes alsm_geometry.m, alsm_geometry2.m, alsm_geometry3.m, alsm_geometry4.m, alsm_geometry5.m and alsm_geometry6.m. The first one is the most simplest and features were added and refined in each successive version. Future direction:
    - ▪ Use the same ideas of projection and trigonometry to simulate patterns for uneven grounds.

- ✓ Simulation showing the distribution of the laser shots on the ground. The codes for this are: simulatorV5p5.m, simulatorV5p6.m, simulatorV5p61.m, and simulatorV5p62.m. As before the latter ones are more sophisticated, with the ability to work for randomly generated 1-D terrain models. Future direction:

    - ▪ Try to incorporate the effects of shadowing. This can be done effectively by mapping the location of each shot to the corresponding scan angle that results in the shot and by keeping track of the largest angle before each shot position. Its more effective to draw the shots from the middle towards the edges, because the middle portion has virtually no shadows.

    - ▪ Use the ideas for 2-D terrain models.

- ✓ Simulation to compare the point density plots for both the simulator data and the real data. The basic idea was to grid the laser in square bins (3m by 3m) for both the simulated results and real data (Flagler and Vallejo). To simulate the periodicity in the plots, the y-position of the airplane was modulated by a sinusoid. The amplitude and the frequency of such a sinusoid were adjusted to match the patterns for the real density plots as closely as possible. The codes for this are: Point_dens.m (get the point density plot for the simulated data), att_SLmar22_ymodul.m (to show the modulated-like patterns of the laser shots on the ground in 2-D), alsm1 (to grid the Flagler data), alsm2 (to grid the Vallejo data), Time_Vs_XYZ.m (study the coupling of the x and y dimensions of the real data, and a method of regression to average out such a coupling). Future direction:

    - ▪ I believe that the extent of periodicity seen in the point density plots can be lowered by using the regressed y-data. One can easily demonstrate this just by using this y-data instead of the original data. If the pretense that there is modulation-type error in the estimate of the coordinates of the laser shots, from which we started this work in the first place, is correct, then this apparent modulation in y-data whose frequency matches the scanning frequency of the mirror might be a contributing (though not the only) factor to the artifacts.

- ✓ Simulation to demonstrate the effects of sampling the scan angle. The programs for this are: Sampling1.m and Sampling2.m. The two programs are for two approaches of estimating the scan angle- the first by just sampling it at regular intervals and the other by using an averaging method to estimate the scan angle from the sampled values. As can be seen, a simple averaging method reduces the error significantly (from +-5m in elevation of the ground to +-1m). Future direction:
    - ▪ Look for better and more efficient methods to alleviate sampling errors.

- ✓ Trial to remove or at least lower the periodic defects that can be seen using dithering. The idea here is to blend or hide in easily noticeable and systematic errors by adding some noise and quantization, and even a little smoothing. So far I have just touched upon the idea and demonstrated some results using simple simulation. The codes for this are: Dither1.m and Dither2.m. The first one simply shows how the periodic patterns can be camouflaged by adding small amounts of noise with a simple 3-d model. The second one demonstrates in 1-d the use of quantizing and smoothing after adding in the noise. Future direction:
    - ▪ Look to improve upon the idea and possibly explore other image processing methods

These are the final and 'cleaned up' versions of many codes I have written. I haven't documented all the codes because some of them are too trivial (like generating a movie file) and some of them have been combined and refined to get the above mentioned versions. All these files, the presentation files and this document are stored in ASPL computers at ***C:\ALSM_Geometry****. I have tried to make this presentation brief but clear. Still there might be questions about the programming specifics of the codes and other general questions. I would be more than happy to expatiate on my ideas and would welcome the person to contact me by email or in person.*

Puskal P Pokharel
puskal@ufl.edu
Saturday, June 05, 2004

Code Description:

Summary

The following figures illustrate the scanning geometry of the ASLM. They also give the meanings of the symbols that shall be used later.

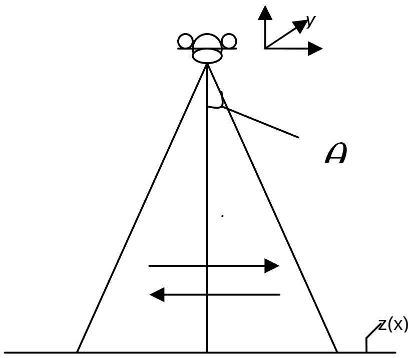

Figure 1: the plane moves in the y-direction and mirror shoots laser shots to the ground at an angle θ as it oscillates in the x-direction, scanning the

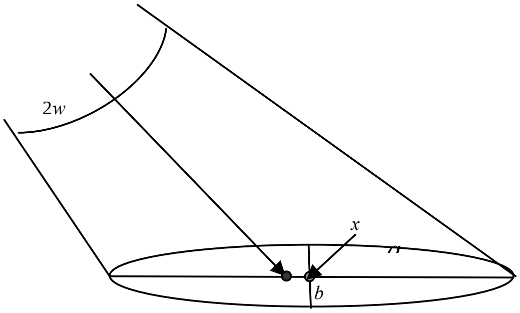

Figure 2: the laser spreads out with a divergence of $2w$ and makes an elliptical footprint with semi-axes a

The following equations quantify the geometry:

$$a = \tfrac{1}{2} \cdot h \cdot [(\tan(\theta + w) - \tan(\theta - w)]$$
$$x = \tfrac{1}{2} \cdot h \cdot [(\tan(\theta + w) + \tan(\theta - w)]$$
$$x + \Delta x = \tfrac{1}{2} \cdot h \cdot [\tan(\theta + \Delta\theta + w) + \tan(\theta + \Delta\theta - w)]$$
$$b = \frac{a \cdot h \cdot \tan(w)}{\cos(\theta)[\sqrt{a^2 - (h \cdot \tan\theta - x)^2}\,]}$$
$$A = \pi \cdot a \cdot b$$

$$\Delta\theta = \frac{4 \times ScanR \times ScanA}{PulseR}$$

$$x = h \tan \theta$$
$$x + \Delta x = h \tan(\theta + \Delta\theta)$$

Here x, a, b, θ and h have the meanings shown by the previous diagrams. $\Delta\theta$ is the angular separation between two laser shots, Δx is the corresponding linear separation of the footprints on the ground A is the area of a footprint. These equations have been derived using basic geometry and simple projection principles.

The summary of the major matlab codes:

a. Simple test for the idea of modeling lidar scatter from a rough surface as the convolution of the transmitted waveform and a surface reflectivity function. (**Lidar Scatter.m**)

b. Simulation of the working of ASLM using simple geometry to show the variation in size and patterns. This is included in the codes **alsm_geometry.m**, **alsm_geometry2.m**, **alsm_geometry3.m**, **alsm_geometry4.m**, **alsm_geometry5.m** and **alsm_geometry6.m**. The first one is the most simplest and features were added and refined in each successive version. alsm_geometry2.m incorporates scan angle noise and alsm_geometry3 incorporates quantization and the later ones plot multiple scan lines.

c. Simulation showing the distribution of the laser shots on the ground. The codes for this are: **simulatorV5p5.m**, **simulatorV5p6.m**, **simulatorV5p61.m**, and **simulatorV5p62.m**. As before the latter ones are more sophisticated, with the ability to work for randomly generated 1-D terrain models. They show the position of the laser shots.

d. Simulation to compare the point density plots for both the simulator data and the real data. The basic idea was to grid the laser in square bins (3m by 3m) for both the simulated results and real data (Flagler and Vallejo). To simulate the periodicity in the plots, the y-position of the airplane was modulated by a sinusoid. The amplitude and the frequency of such a sinusoid were adjusted to match the patterns for the real density plots as closely as possible. The codes for this are: **Point_dens.m** (get the point density plot for the simulated data),

att_SLmar22_ymodul.m (to show the modulated-like patterns of the laser shots on the ground in 2-D), **alsm1** (to grid the Flagler data), **alsm2** (to grid the Vallejo data), **Time_Vs_XYZ.m** (study the coupling of the x and y dimensions of the real data, and a method of regression to average out such a coupling).

e. Simulation to demonstrate the effects of sampling the scan angle. The programs for this are: **Sampling1.m** and **Sampling2.m**. The two programs are for two approaches of estimating the scan angle- the first by just sampling it at regular intervals and the other by using an averaging method to estimate the scan angle from the sampled values. As can be seen, a simple averaging method reduces the error significantly (from +-5m in elevation of the ground to +-1m).

f. Trial to remove or at least lower the periodic defects that can be seen using dithering. The idea here is to blend or hide in easily noticeable and systematic errors by adding some noise and quantization, and even a little smoothing. So far I have just touched upon the idea and demonstrated some results using simple simulation. The codes for this are: **Dither1.m** and **Dither2.m**. The first one simply shows how the periodic patterns can be camouflaged by adding small amounts of noise with a simple 3-d model. The second one demonstrates in 1-d the use of quantizing and smoothing after adding in the noise.

LIST OF COMMON INPUT VARIABLES (these are the major variables, all codes have variables explained)

For programs listed in b. and c. above

h:	*height in cms.*
w:	*half the beamwidth of the pulse in radians*
scanR:	*scan rate Hz*
pulseR:	*pulse rate Hz*
thetaR:	*max of mirror angle in rad*
anglePpulse:	*scan angle change per pulse: rad/pulse*
a:	*the semi major axis of the ellipse*
d:	*the center of the ellipse, which is different from the center of the laser beam*

xa: *distace between the center of the ellipse and the center of the laser beam on the ground*

ya: *the length of the perpendicular to the center of the beam from the edge*

bsq: *the square of semiminor axis of the ellipse*

d1: *'d' in meters*

x,y1,y2: *the x and y coordinates in the equation of the ellipse, for each x there are two values of y-y1,y2.*

theta1: *angle without noise*

noise: *scan angle noise*

theta: *noisy angle*

dd: *'d' for noisy scan angle*

dd1: *'dd' in meters*

xx: *'x' for the noisy angle*

rho: *path length*

For programs listed in d. and e. above

fprf: *PRF freq*

fscan: *scan cycle freq*

Tprf: *PRF period [sec]*

Tmr: *mirror reading sampling period*

Thetamr: *incremental angle for scan sampling*

Tscan: *scan period [seconds per scan cycle]*

thetaMx: *scan angle bound in radian*

Nscan: *number of scan periods we wish to simulate* simulatorV5p62.m

Time_Vs_XYZ.m

Alsm1.m

Alsm2.m

ALSM_geometry2.m

ALSM_geometry3.m

alsm_geometry4.m

alsm_geometry5.m

alsm_geometry6.m

ALSM_geometry.m

att_SLmar22_ymodul.m

Dither1.m

Dither2.m

Lidar Scatter.m

Point_dens.m

Sampling1.m

Sampling2.m

simulatorV5p5.m

simulatorV5p6.m

simulatorV5p61.m