Contact Information


```
Darryl Granger
Department of Earth & Atmospheric Sciences
Purdue University
550 Stadium Mall Drive
West Lafayette, IN 47907-2051

Tel: 765-494-3258
Fax: 765-496-1210
dgranger@purdue.edu
http://www.eas.purdue.edu/~dgranger
```


Project Name: Detection of Tectonically Deformed Shorelines in Southern Illinois

1. Survey Area

The survey area contains seven polygons located 10 km west and 7 km north of Lawrenceville, Illinois (Figure 1). This area was flown and completed on October 1, 2005 (Day 274). The survey was completed using an Optech 1233 Airborne Laser Terrain Mapper (http://www.optech.ca/) mounted in a twin engine Piper Chieftain (N931SA).

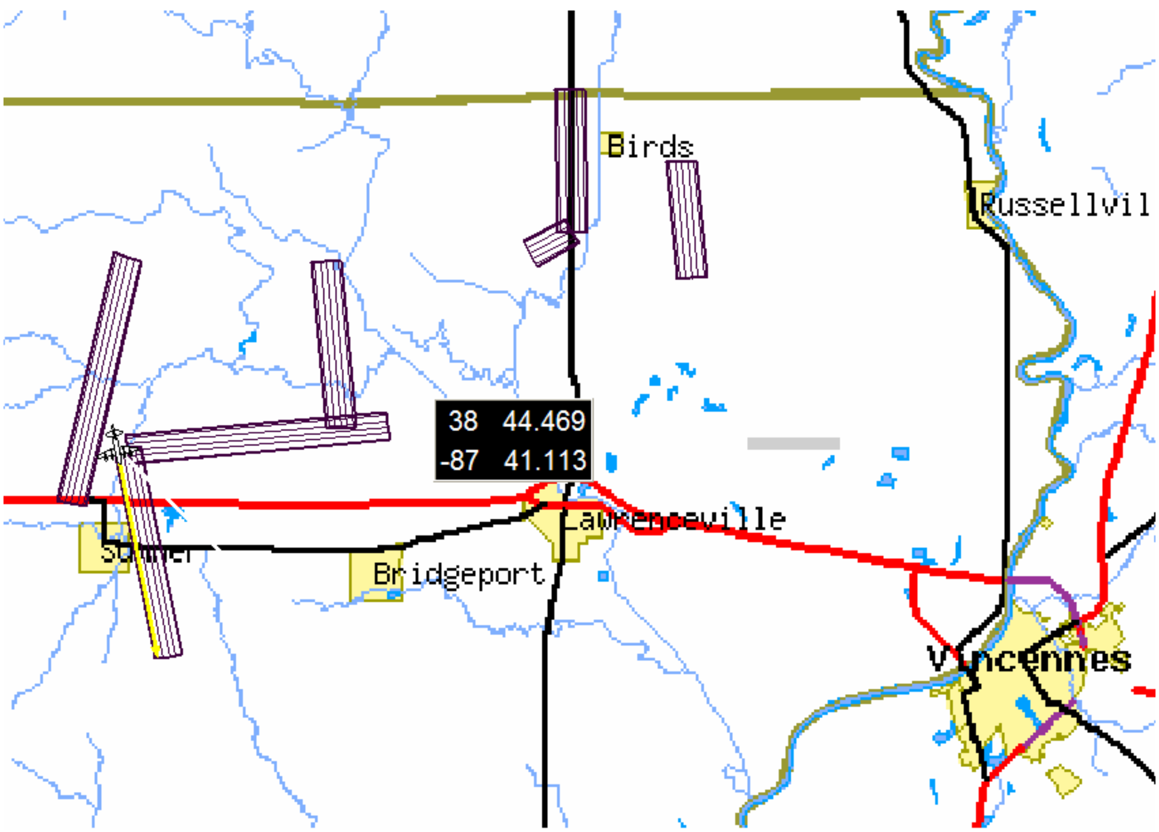


Figure 1. Project boundary and locations over Lawrenceville, Illinois.

2. Survey Parameters

The survey areas were flown with 35 total flight lines. Five boxes are oriented north-south while two boxes are oriented east-west. The laser range was targeted at 600 m above ground level (AGL). Aircraft speed was targeted at 125 kts. Figure 2 shows additional parameters.

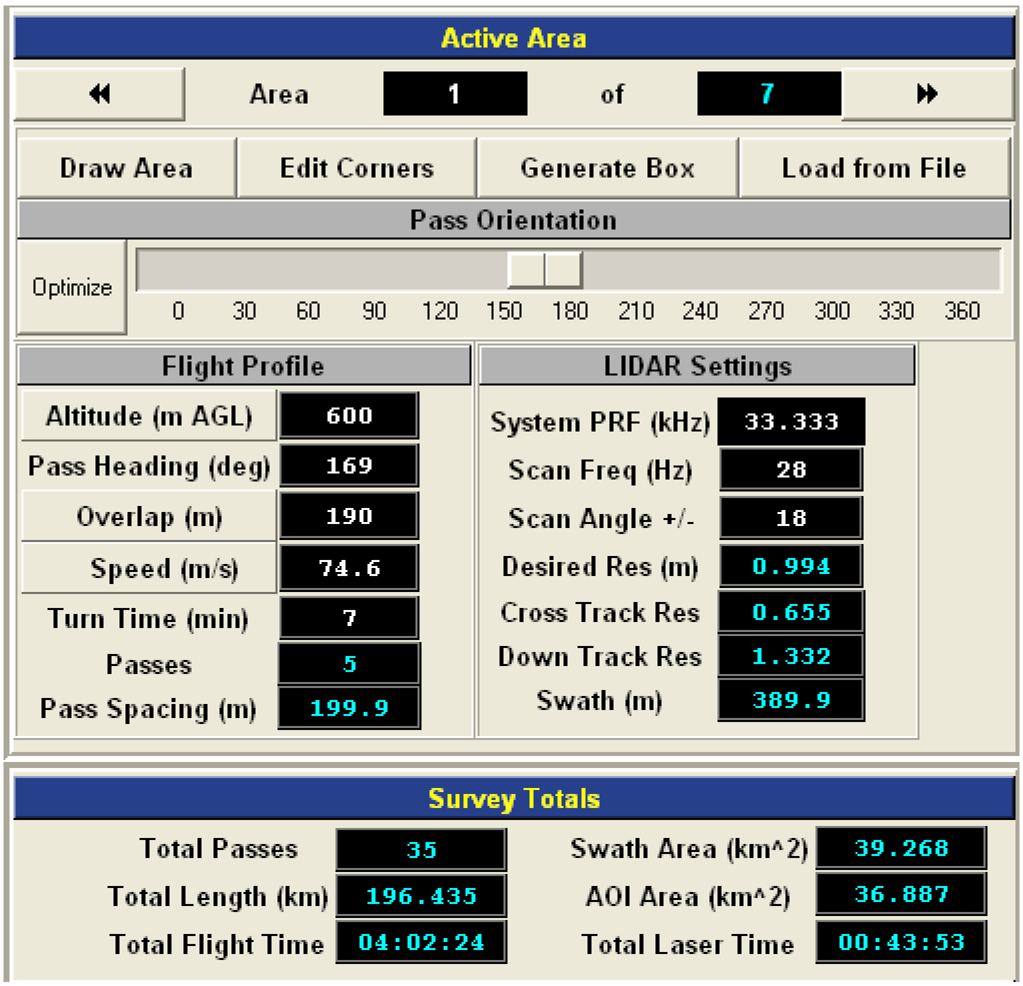


Figure 2. Flight parameters for Lawrenceville, Illinois.

3. GPS Reference Stations

Two GPS reference stations were used during the survey. Both receivers were placed on newly set marks at the Lawrenceville-Vincennes International Airport. The stations were observed for 4 hours during the survey on October 1, 2005 (Day 274). All GPS observations were logged at a 1-second rate and were submitted to the NGS online processor OPUS. Final coordinates for the reference stations were based on these OPUS solutions (http://www.ngs.noaa.gov/OPUS/). For more information on the CORS network, refer to http://www.ngs.noaa.gav/CORS/. Ground equipment at both stations included ASHTECH Z-Extreme receivers and choke ring antennas (Part #700936.D) mounted on 1.5 m conventional tripods.

4. Navigation Processing

The airplane trajectory for this survey was processed using KARS software (Kinematic and Rapid Static) by Dr. Gerry Mader of the NGS Research Laboratory.

After GPS processing, the trajectory and the (Inertial Measurement Unit) data collected during the flight were input into APPLANIX software POSPROC which uses a Kalman Filter to produce a final navigation solution (aircraft position and orientation) at 50 Hz, in SBET format (Smoothed Best Estimated Trajectory).

5. Calibration and Laser Point Processing

The SBET and the raw laser range data were combined using Optech's REALM processing suite to generate the laser point dataset. A few small test sites containing crossing flight-lines were initially extracted and used for relative calibration with TerraSolid's TerraMatch software. This application measures the differences between laser surfaces from overlapping flightlines and translates them into correction values for the system orientation -- easting, northing, elevation, heading, roll and/or pitch. . After obtaining adjustments to calibration values using TerraMatch, laser point processing was re-done and the calibration rechecked.

Results are shown below in TerraMatch output format:

```
Used loaded points
Trajectories: D:\Projects\Granger_05\TerraScan\trajectory\274_x1\
No known points
Observe every 1th point
Intensity not used
Solution for whole data set

Starting average dz:    0.1609
Final average dz:       0.0557

Standard error of unit  0.0249

Execution time: 1417.2 sec
Number of iterations: 25
```

```
Points        2278354
H shift       -0.0030    Std dev  0.0020
R shift       -0.0267    Std dev  0.0006
P shift       +0.0292    Std dev  0.0017
Scale        -0.00637
```

No absolute ground calibration was performed on these data, so a bias may be present with respect to ellipsoid heights obtained by GPS during ground surveys. If this bias exists it should be in the range of +/- 0.15 meters.

The laser point data was extracted by flight strip, separately for each one of the 7 project boxes (corridors)

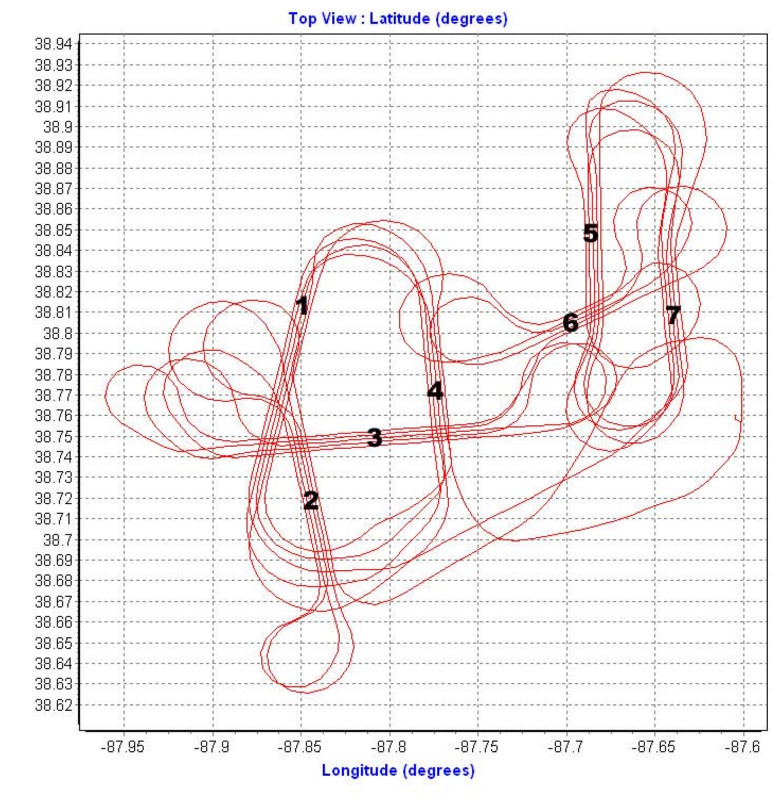


Figure 3. Project boxes (corridors) index.

All coordinates were processed with respect to NAD83 and referenced to the national CORS96 network. The projection for the 9 column output is UTM Zone 16, with ellipsoid heights, and units in meters. The last return data was extracted from the 9-column format and the heights reprojected to orthometric heights in NAVD88, computed using NGS GEOID03 model with the Corpscon v6.0 software (Corps of Engineers Coordinate Conversion).

The most complete output format is nine-column ASCII (space delimited), one file per flight strip. The nine columns are as follows:

```
1. GPS time (seconds of week)
2. Easting last return
3. Northing last return
4. Height last return
5. Intensity last return
6. Easting first return
7. Northing first return
8. Height first return
9. Intensity first return
```

Note that in these 9-column files no geoid model has been applied - height values are ellipsoid heights and these height values will NOT match orthometric heights (elevations) found in the 3-column files or in the 1-meter DEM grid nodes.

During processing, a scan cutoff angle of 1.5 degrees was used to eliminate points at the edge of the scan lines. This was done to improve the overall DEM accuracy (points farthest from the scan nadir are the most affected by small errors in pitch, roll and scanner mirror angle measurements). Points with very low intensity values were also filtered out (intensity values less than 7), because these points also tend to be the least accurate. This is due to the fact that very weak return pulses yield the noisiest range measurements. These points represent a very small percentage of the total number of points, usually in the neighborhood of a few hundredths of one percent.

6. Filtering and DEM Production

Terrasolid's TerraScan (http://terrasolid.fi) software was used to classify the last return LIDAR points and generate the "bare-earth" dataset. Each of the 7 corridors was processed individually.

The classification routine consists of three algorithms:

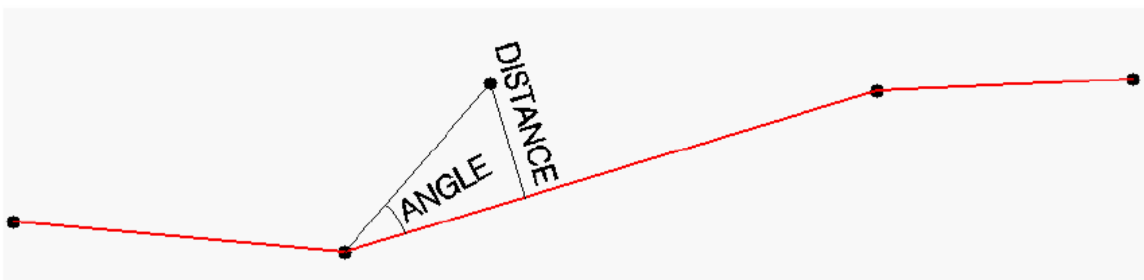
1) Removal of "Low Points". This routine was used to search for possible error points which are clearly below the ground surface. The elevation of each point (=center) is compared with every other point within a given neighborhood and if the center point is clearly lower then any other point it will be classified as a "low point". This routine can also search for groups of low points where the whole group is lower than other points in the vicinity. The parameters used on this dataset were:

```
Search for: Groups of Points
Max Count (maximum size of a group of low points): 6
More than (minimum height difference): 0.5 m
Within (xy search range):  10.0 m
```

2) Ground Classification. This routine classifies ground points by iteratively building a triangulated surface model. The algorithm starts by selecting some local low points assumed as sure hits on the ground, within a specified windows size. This

makes the algorithm particularly sensitive to low outliers in the initial dataset, hence the requirement of removing as many erroneous low points as possible in the first step.

The routine builds an initial model from selected low points. Triangles in this initial model are mostly below the ground with only the vertices touching ground. The routine then starts molding the model upwards by iteratively adding new laser points to it. Each added point makes the model follow ground surface more closely. Iteration parameters determine how close a point must be to a triangle plane so that the point can be accepted to the model. **Iteration angle** is the maximum angle between point, its projection on triangle plane and closest triangle vertex. The smaller the Iteration angle, the less eager the routine is to follow changes in the point cloud. **Iteration distance** parameter makes sure that the iteration does not make big jumps upwards when triangles are large. This helps to keep low buildings out of the model. The routine can also help avoiding adding unnecessary point density into the ground model by reducing the eagerness to add new points to ground inside a triangle with all edges shorter than a specified length.



Ground classification parameters used:

```
Max Building Size (window size): 40.0 m
Max Terrain Angle: 80.0
Iteration Angle: 6.0
Iteration Distance: 1.2 m
Reduce iteration angle when edge length < : 5.0 m
```

3) Below Surface removal. This routine classifies points which are lower than other neighboring points and it is run after ground classification to locate points which are below the true ground surface. For each point in the source class, the algorithm finds up to 25 closest neighboring source points and fits a plane equation through them. If the initially selected point is above the plane or less than “Z tolerance”, it will not be classified. Then it computes the standard deviation of the elevation differences from the neighboring points to the fitted plane and if the central point is more than “Limit” times standard deviation below the plane, the algorithm it will classify it into the target class.

Below Surface classification parameters used:

```
Source Class: Ground
Target Class: Low Point
Limit: 8.00 * standard deviation
Z tolerance: 0.10 m
```

After classification the ground points were outputted in 2km x 2km overlapping tiles (60m overlap), ASCII format (XYZI), and gridded at 1m cell size using Golden Software's SURFER ver. 8.01. The tiles need to overlap in order to obtain consistent transitions from one tile to the adjacent ones.

Gridding parameters:

```
Gridding Algorithm: Kriging
Variogram: Linear
Nugget Variance: 0.07 m
MicroVariance: 0.00 m
SearchDataPerSector: 10
SearchMinData: 5
SearchMaxEmpty: 1
SearchRadius: 40m
```

The resulted Surfer grid tile set was exported to ESRI ArcInfo floting point binary format and using an in-house C++ application the overlap was trimmed from each tile. The trimmed tiles were exported to ESRI ArcInfo GRID format and merged into one seamless raster dataset.

A similar process was used to generate the unfiltered seamless grids.