Notes on Luke Wash Lidar Survey

Ramon Arrowsmith July 16, 2009 and updated on July 22, 2009 and July 28, 2009

Data background

These LiDAR data were gathered by the Wilson Company (http://www.wilsonco.com/projects/luke_wash.asp) for the Maricopa County Flood Control District and given to me by Jon Fuller of JE Fuller and Associates who had been working with them. We wanted to play with them to see how they looked and depicted the bajada landforms in west Phoenix.

In my earlier version, I was really confused by the projection and the location and ended up swapping x and y coordinates. That was wrong. Jon Fuller and Scott Ogden straightened me out and so now I know that this study area is just west of the Hassayampa River and just south of the I-10 (that is the highway along the north edge of the data). The data are in Arizona State Plane Central zone International Feet NAD83

Preparation

I took the cleaned individual files that had been processed by Jen Dischler.

The cleaned files look like this:

```
892380.40,417774.84,1050.38
892380.90,417790.81,1050.84
892383.06,417813.07,1049.72
892384.66,417830.71,1049.62
892389.35,417848.49,1049.78
```

The file names are: `new1099.pf1, new1099.pf2, new1099.pf3, new1099.pf4`

I first ran P2G (our local binning code—http://lidar.asu.edu) on the individual files with a grid resolution of 100 and the default 70.7 search radius.

When I imported the resulting files into ArcMap, I found that the individual files have a small overlap (Figures 1 and 2). So, next I loaded the files into Matlab using the script in appendix A and plotted things up (Figure 2). The colors show the different files (tiles). Finally, I concatenated the individual files into a single 1,964,929 line file on which all the subsequent processing was done. See http://lidar.asu.edu/KnowledgeBase/LukeWash/lukewash10_15grey.kmz for these data in a format viewable in Google Earth.

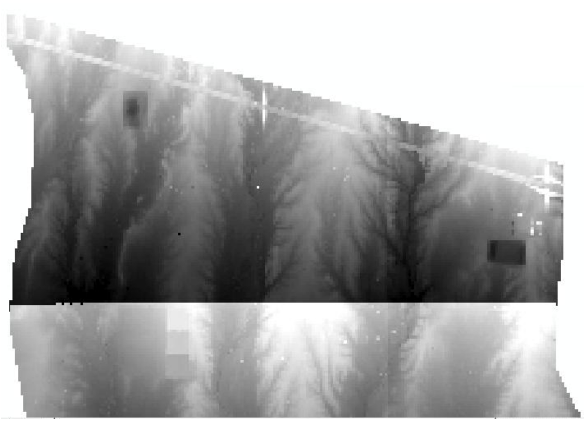


Figure 1. 100 foot grid resolution 70 foot search radius DEMs.

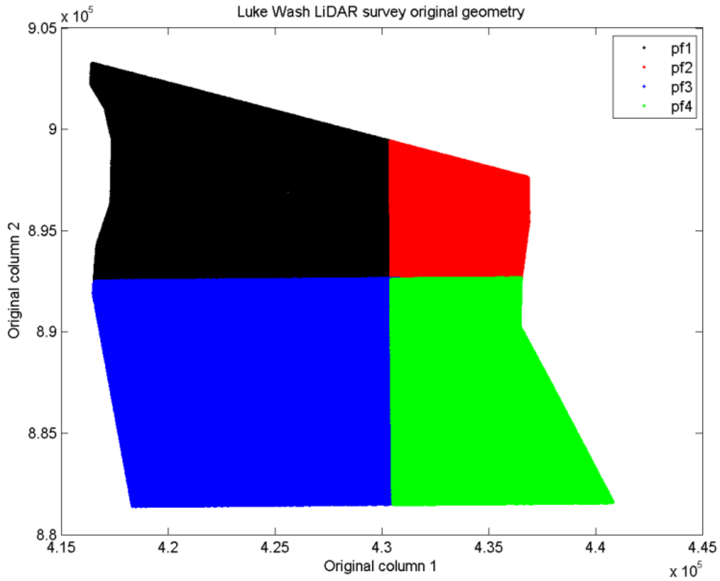


Figure 2. Tile geometry. Colors correspond to the 4 tiles.

Shot density investigation

So, what is the optimal gridding resolution and search radius? The Wilson Company who gathered the data say that the shots were on an 8 foot spacing (http://www.wilsonco.com/projects/luke_wash.asp). To make this investigation, I first wrote a Matlab script (appendix B) which let me take a hillshade of a DEM and overlay on it a subset of selected laser returns to visually assess the shot coverage. I also computed the average shot density for the entire selection and present it relative to a 100x100 foot box (Figures 3 and 4).

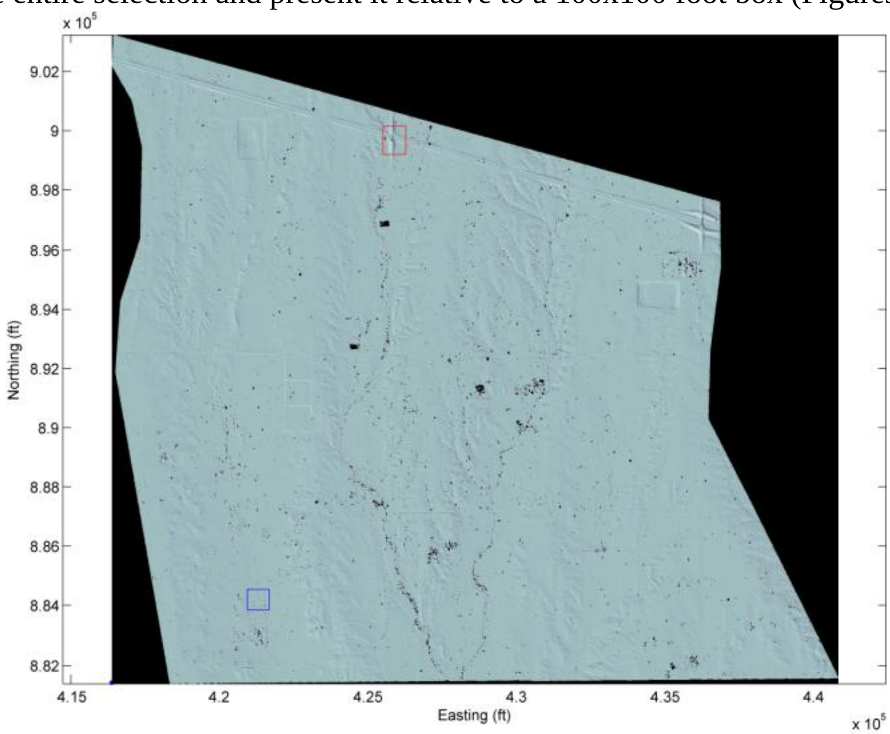


Figure 3. Overview hillshade of the dataset with two rectangles corresponding in color to the two zooms in figure 4.

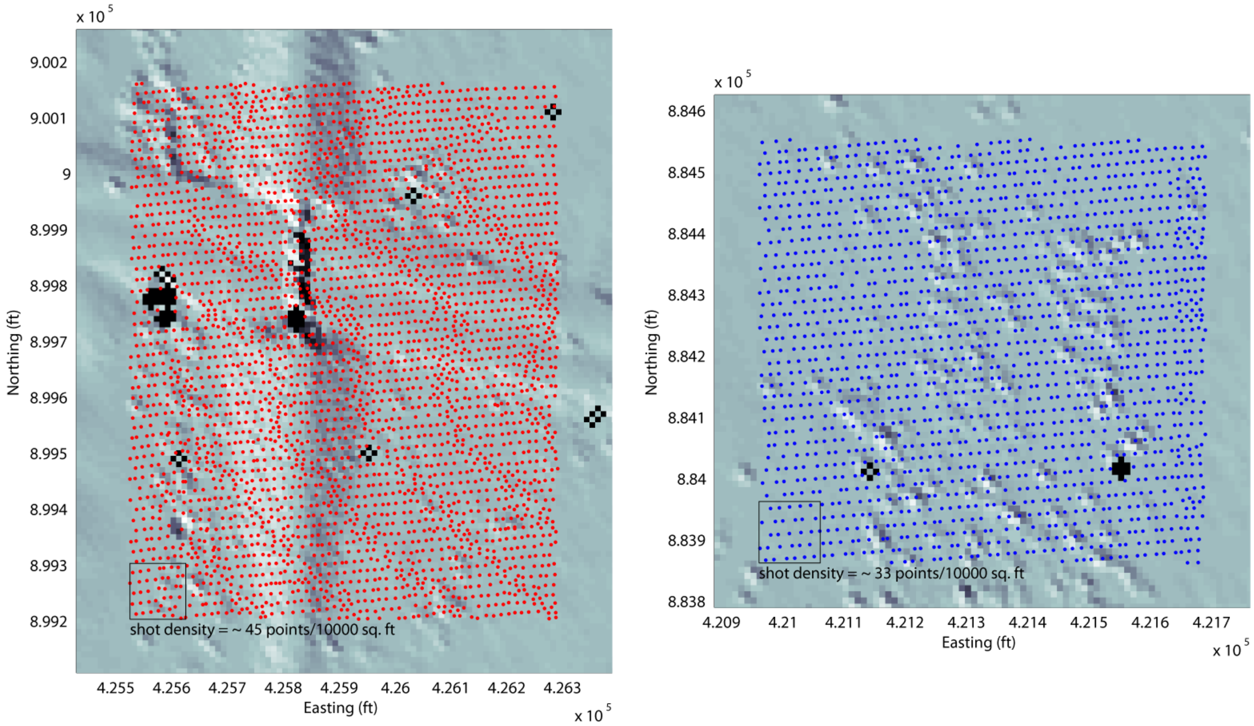


Figure 4. Zoom to laser returns overlain on hillshade (see figure 3 for locations). Right image covers riparian and bajada environment in southern portion of study area while left image covers highway interchange in north central portion of the study area. Both datasets have shot densities of around 40 points per 10,000 square feet. Clearly, the flightlines were north-south. The box at lower left of each image shows such a 100'x100' area even though the density is computed for the entire selection shown in each image (compare with Figure 4.5). This area is also discussed later in a comparion with USGS DEMs and Phoenix aerial photography.

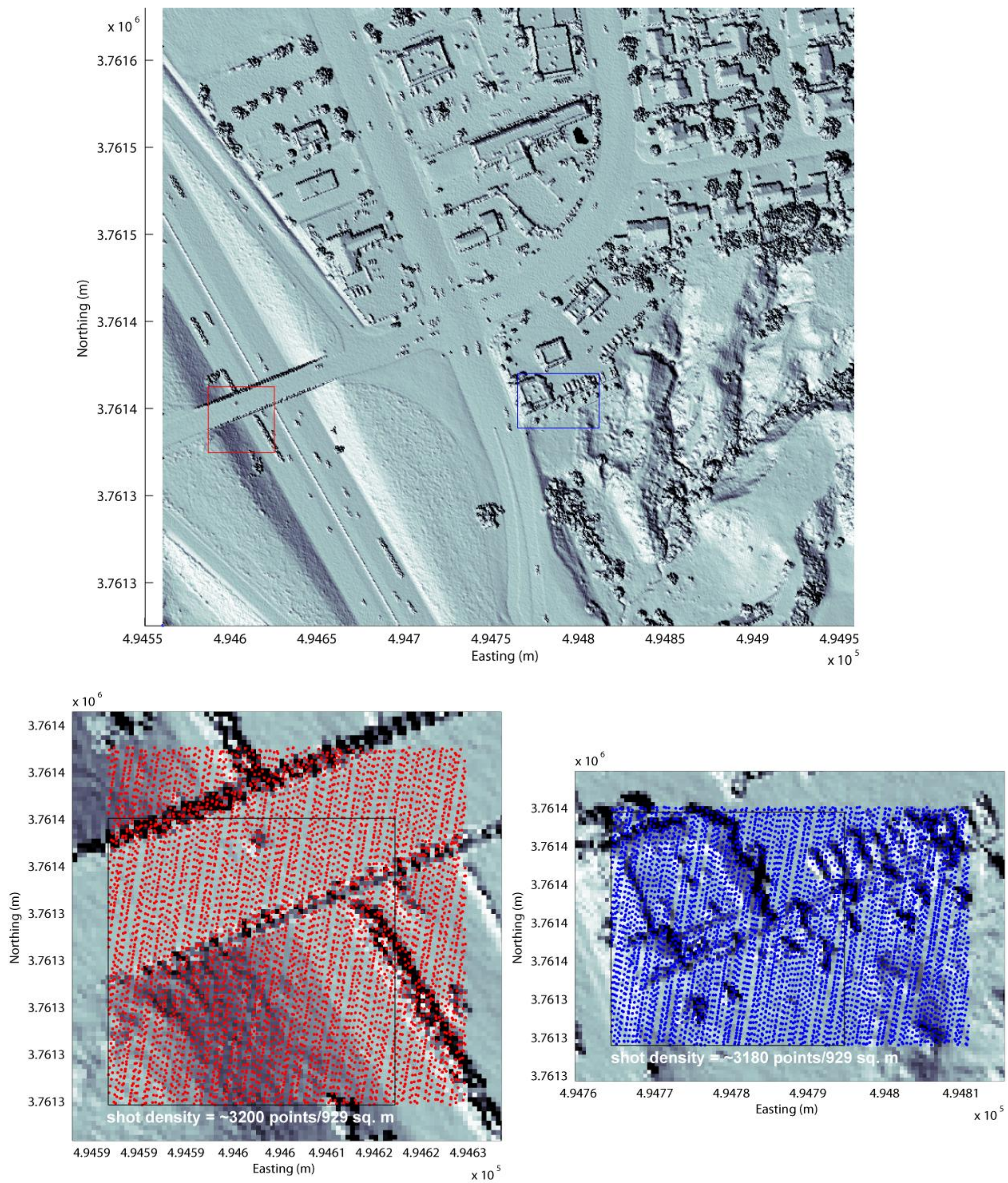


Figure 4.5. These data are a 0.5 m DEM from the area around San Bernardino California with research grade LiDAR shot densities about 100x the Luke Wash data (downloaded from http://www.opentopography.org). Note that the 929 m^2 boxes in the zoomed views below are the same area as the 10,000 ft^2 boxes in Figure 4. See this link http://lidar.asu.edu/KnowledgeBase/1248756159095 78.kmz for a more extensive KMZ file which can be browsed in GoogleEarth. This area is similar in landcover to that expected for greater Phoenix.

Secondly, after qualitatively exploring the DEMs generated using the local binning algorithm, I decided 10 feet was a good resolution and so I ran a number of different search radii (see shell script in Appendix C) and then loaded them and counted the nulls using the analyze_radius_luke.m Matlab script (appendix D). Figure 5 shows the result in which we can see clearly that as the search radius increases the number of nulls decreases. As is typical, there is an optimal search radius for a given shot density overall at the point when the null count change rate begins to flatten. In this case, that optimal number is somewhere around 10 to 15 feet. I used 15 feet for my visualizations.

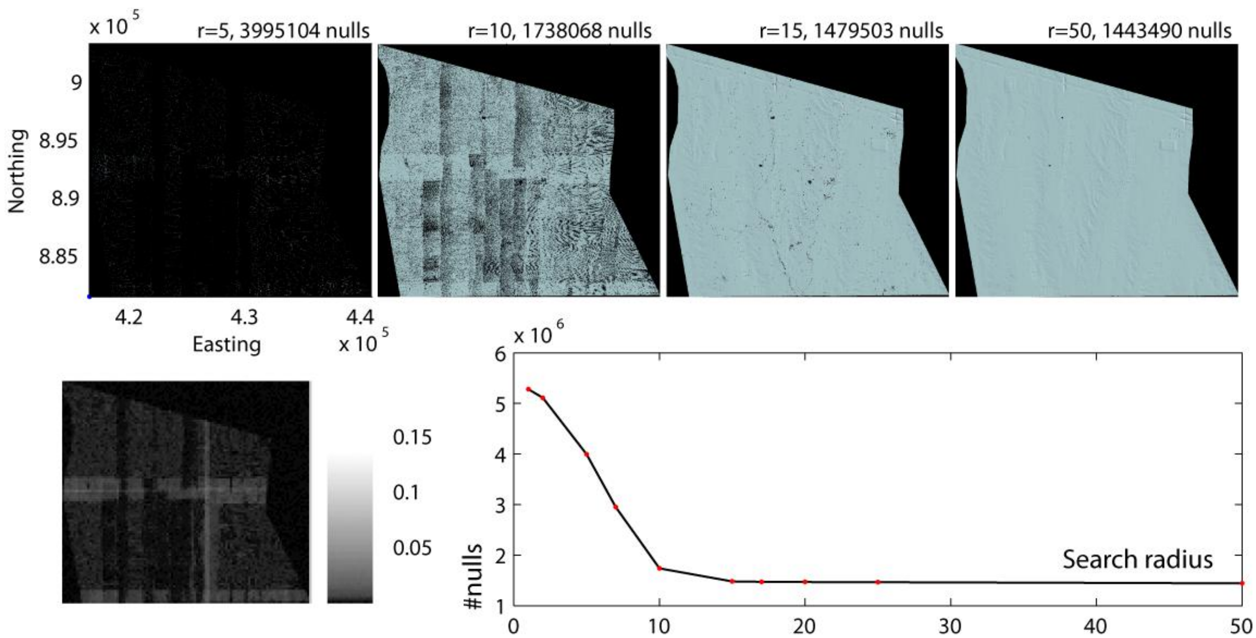
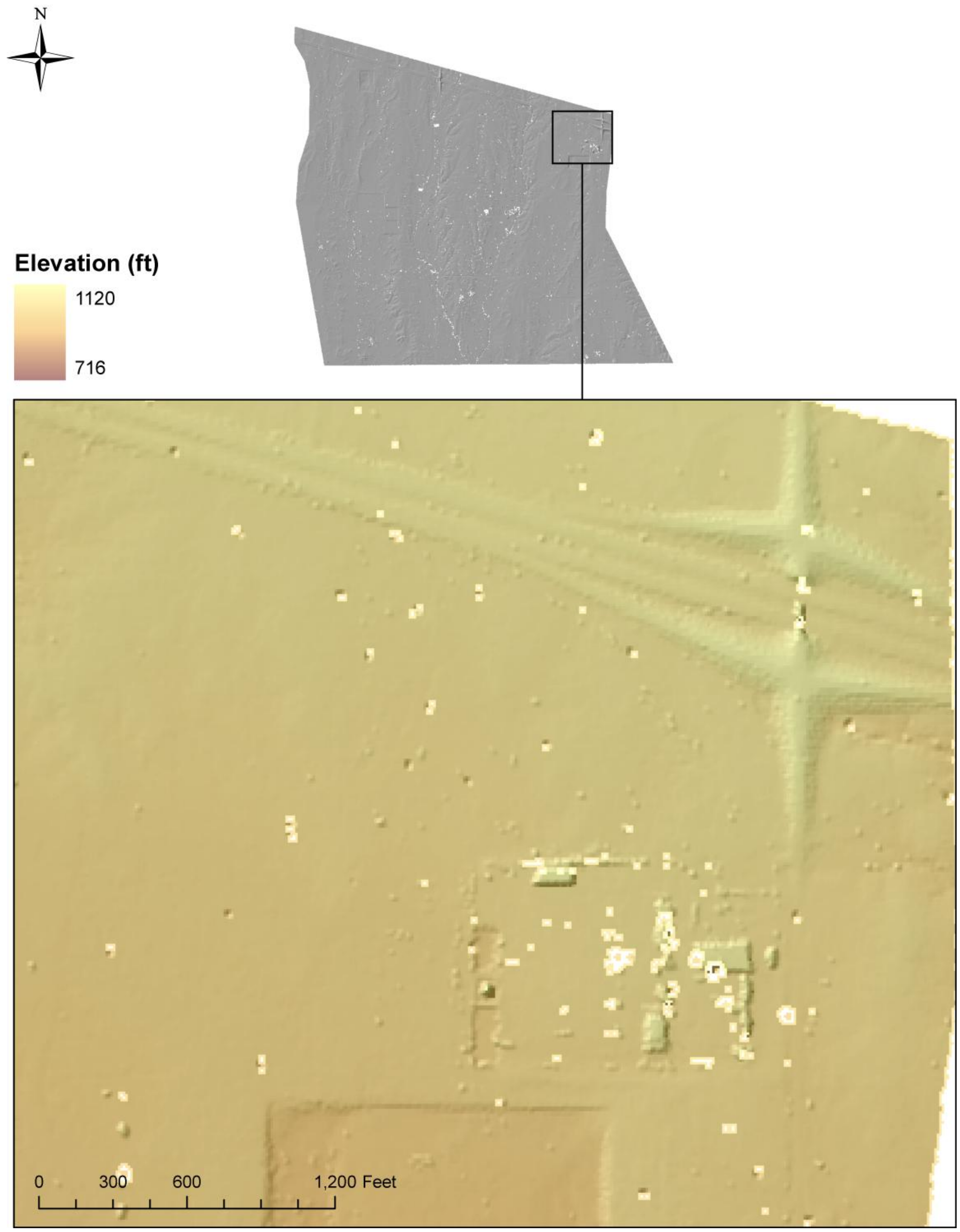
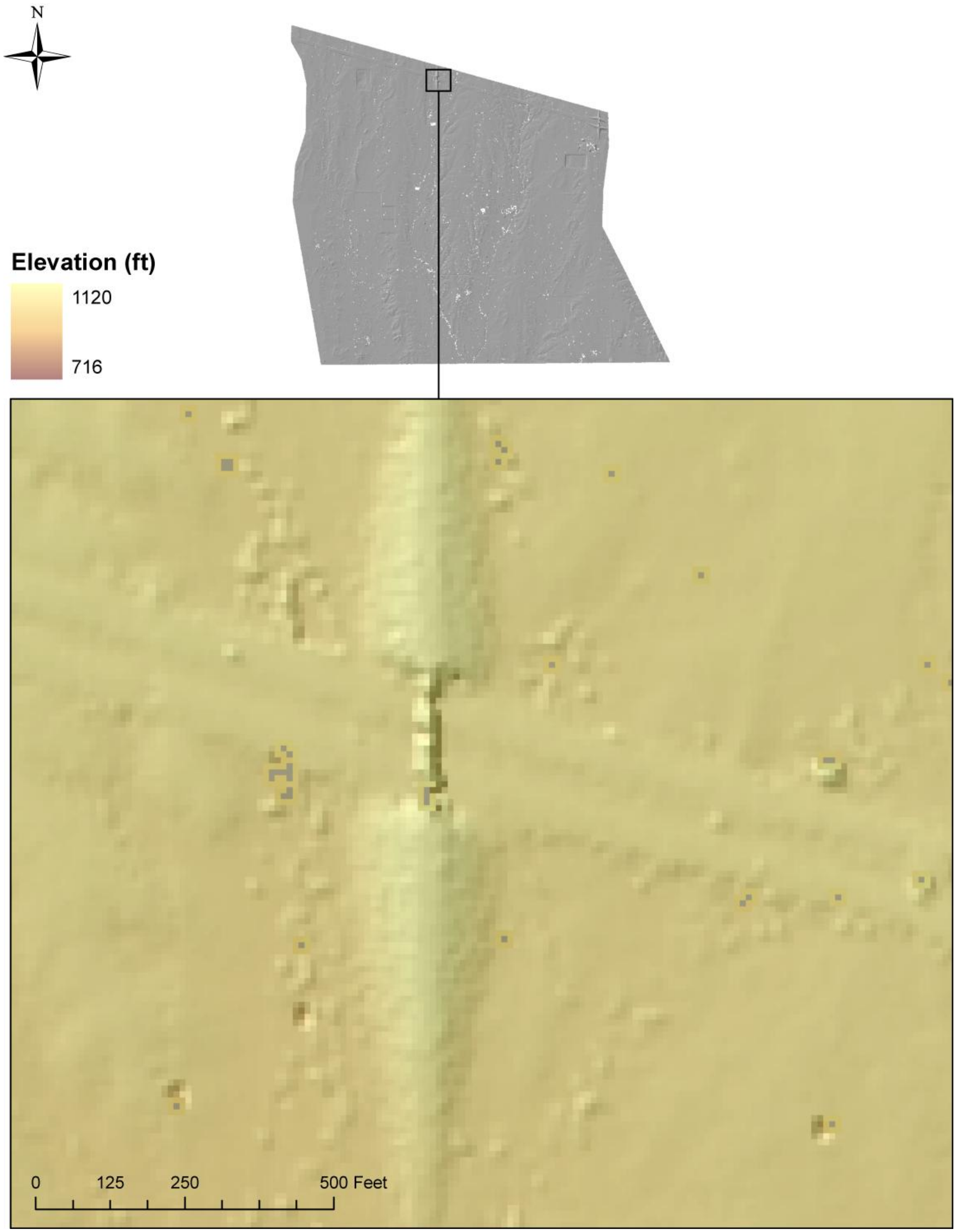
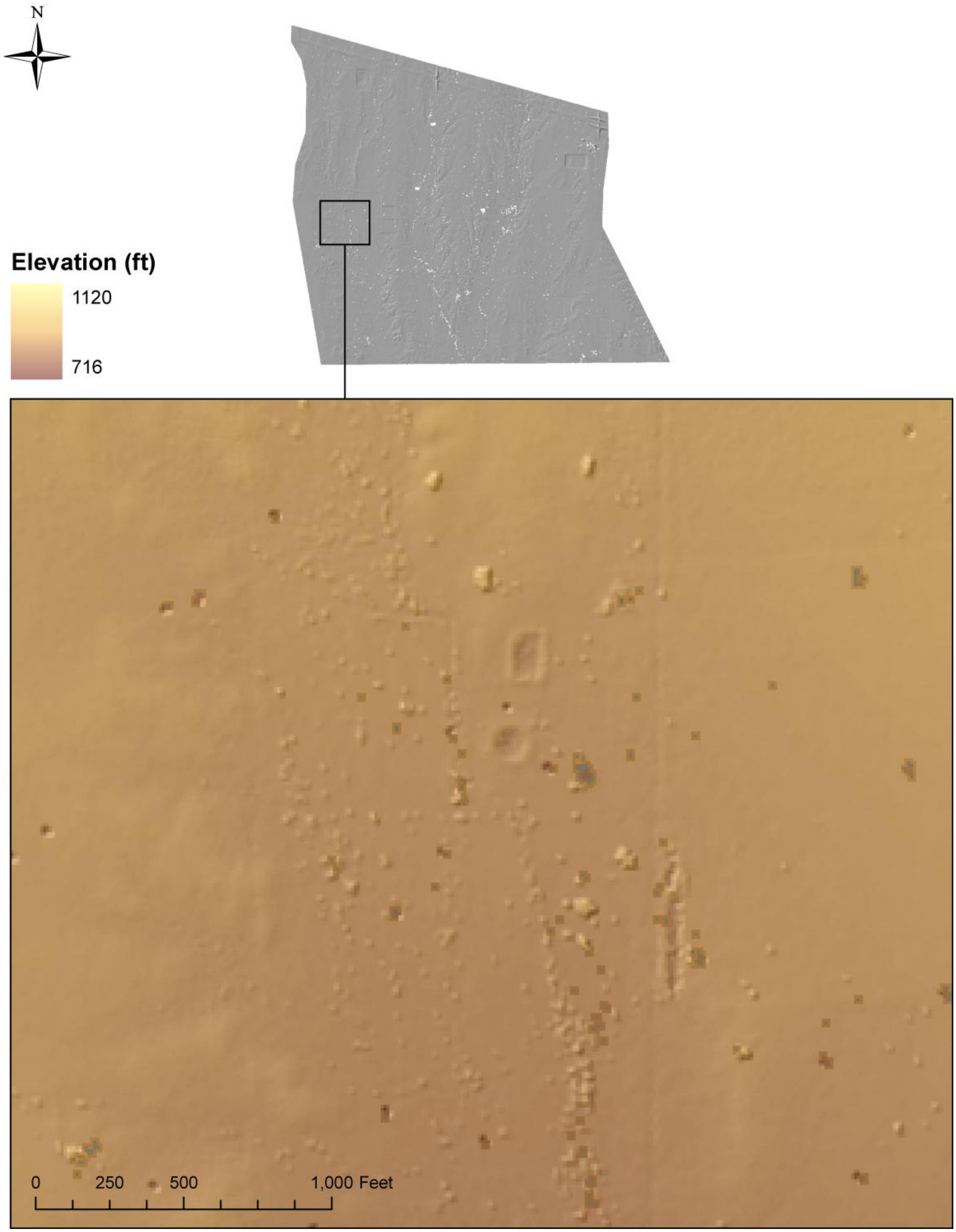


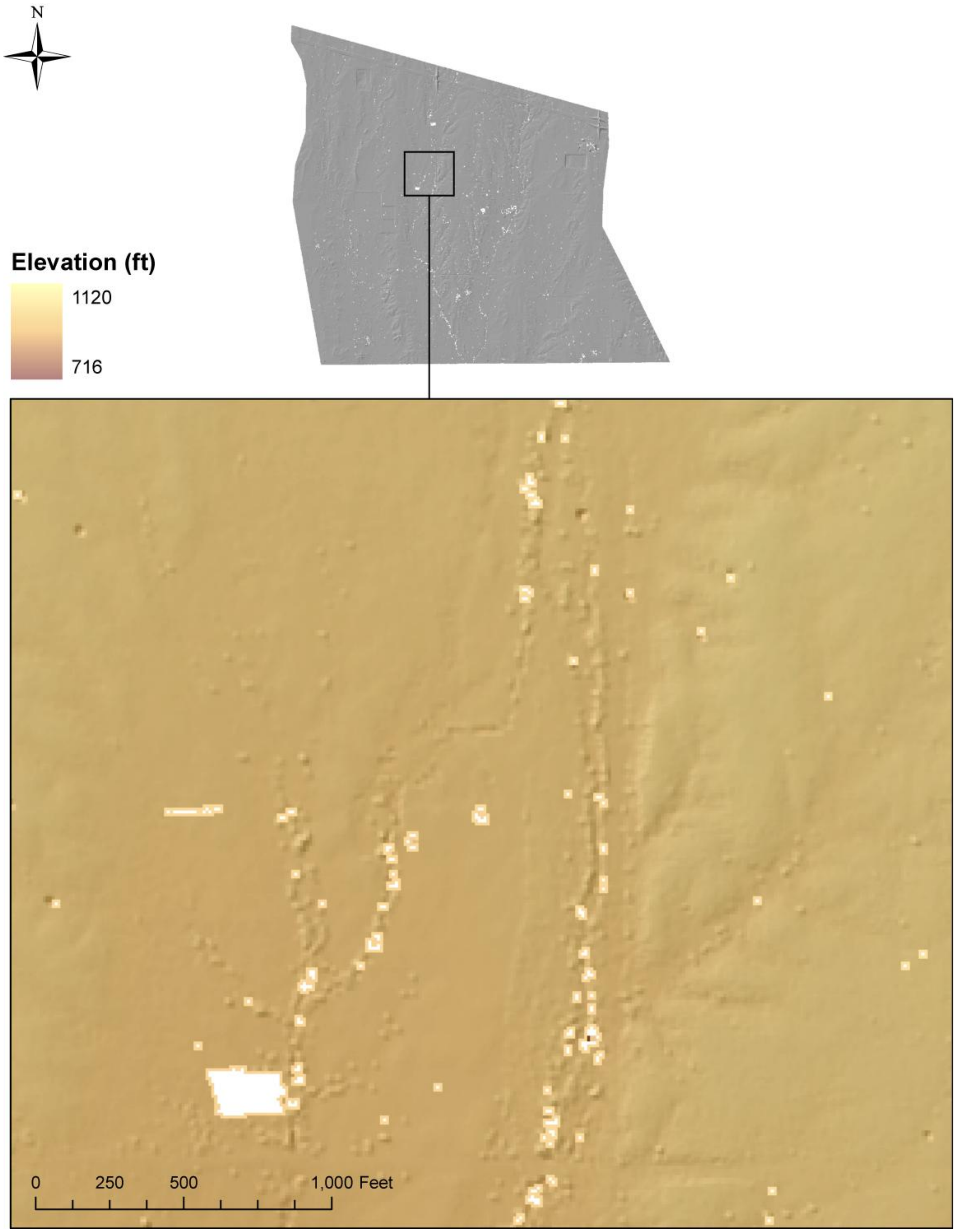
Figure 5. Search radius analysis shows that as search radius increases from 5 to 50 feet (row along top), the number of nulls decreases (plot on lower right). Lower left image shows shot densities per square foot and indicates that the highest densities come from the overlaps between the tiles (possibly duplicate points).

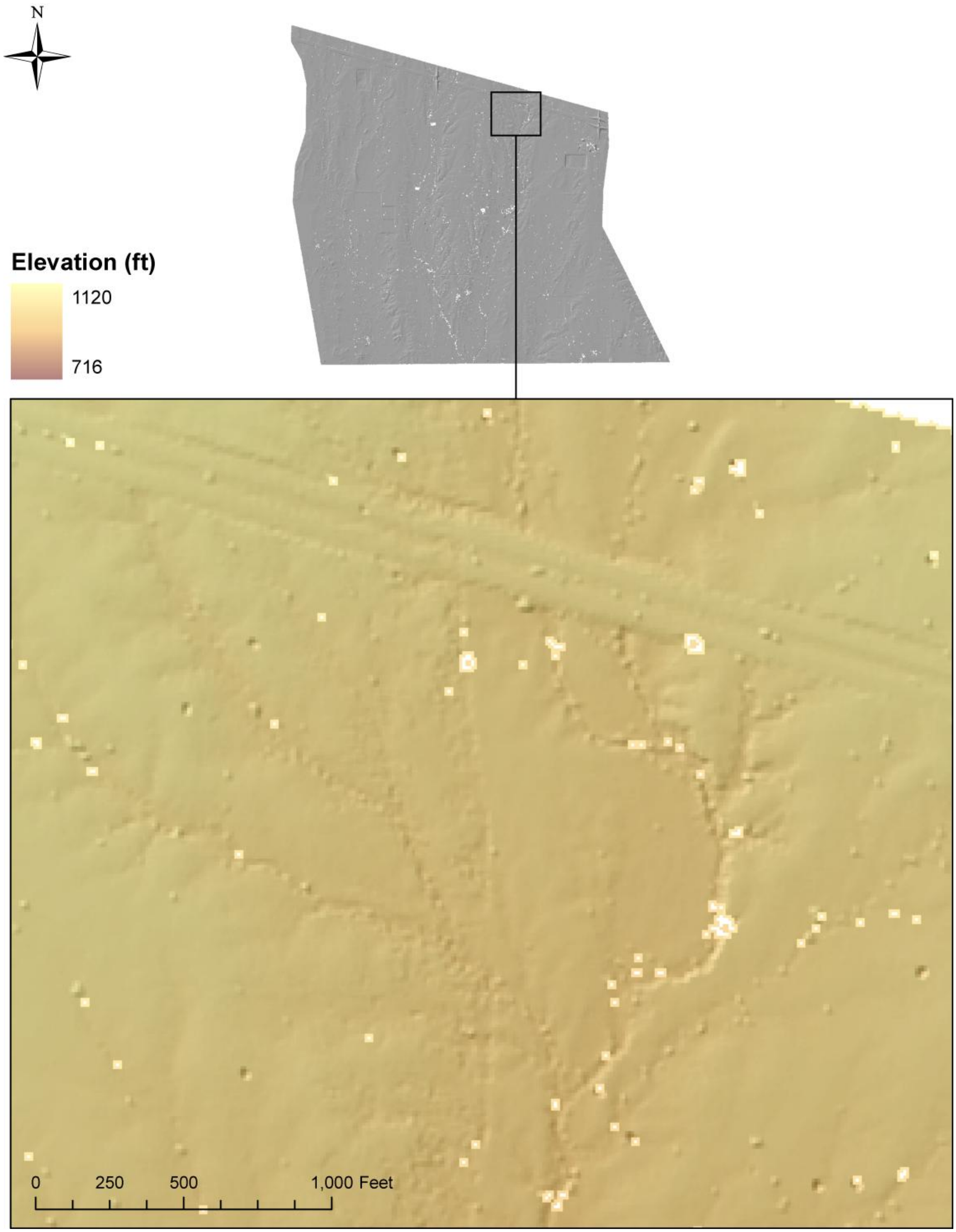
Data visualization. The following figures illustrate the data. While the data with which I am used to working along the San Andreas Fault in California is higher density (~1 shot/foot^2; whereas these data are about 100 times lower density), these data depict the bajada landforms reasonably well and are I think reasonably gridded (based on the above analysis about 10 times finer resolution (10 feet or 3 m versus 30 feet or 10 m) relative to the USGS 1/3' NED elevation data.

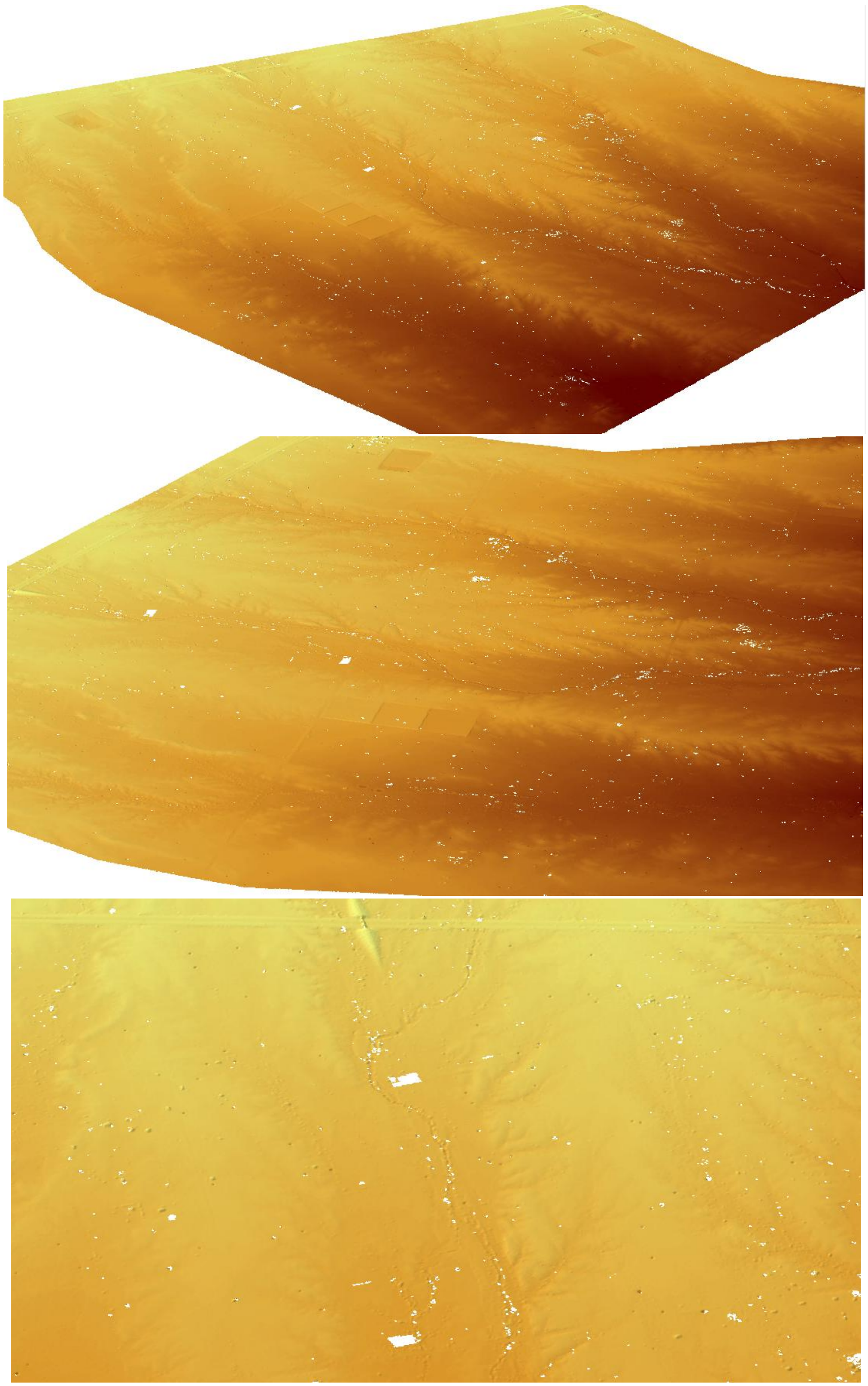



N
Elevation (ft)
1120
716
0
125
250
500 Feet


N
Elevation (ft)
1120
716
0
250
500
1,000 Feet


N
Elevation (ft)
1120
716
0
250
500
1,000 Feet

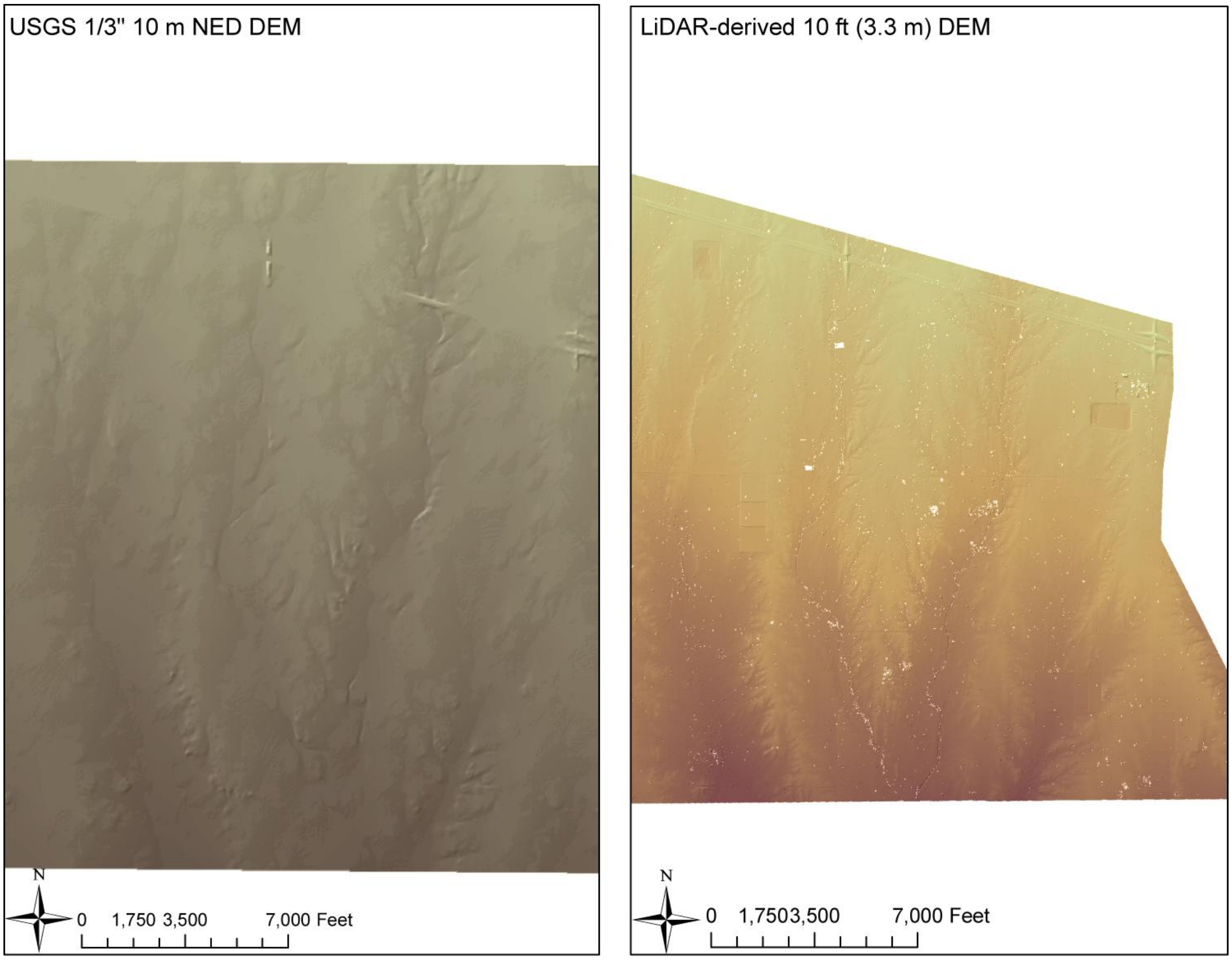

N
Elevation (ft)
1120
716
0
250
500
1,000 Feet



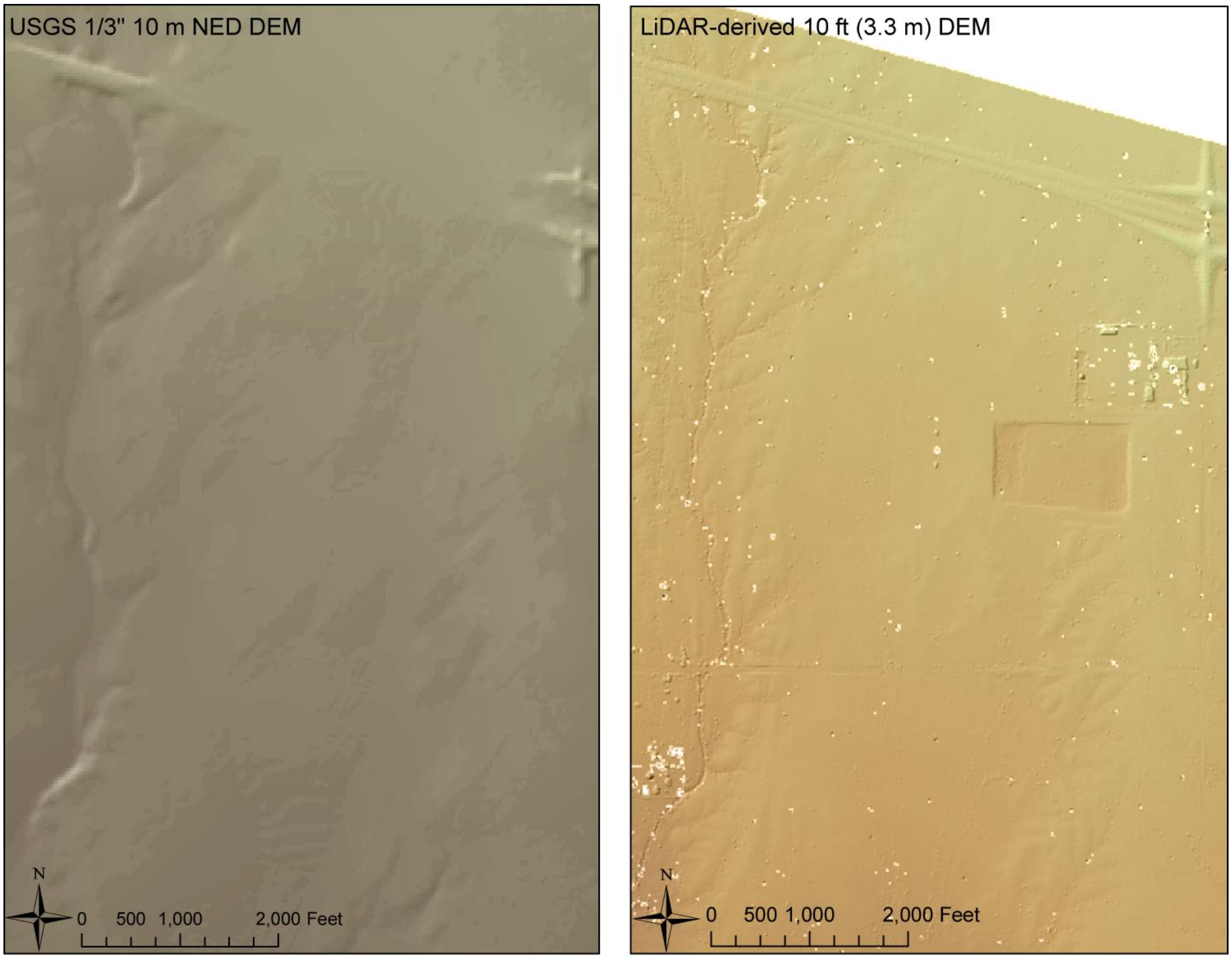
Perspective view around the area (rendered with no vertical exaggeration).

Comparisons with other data

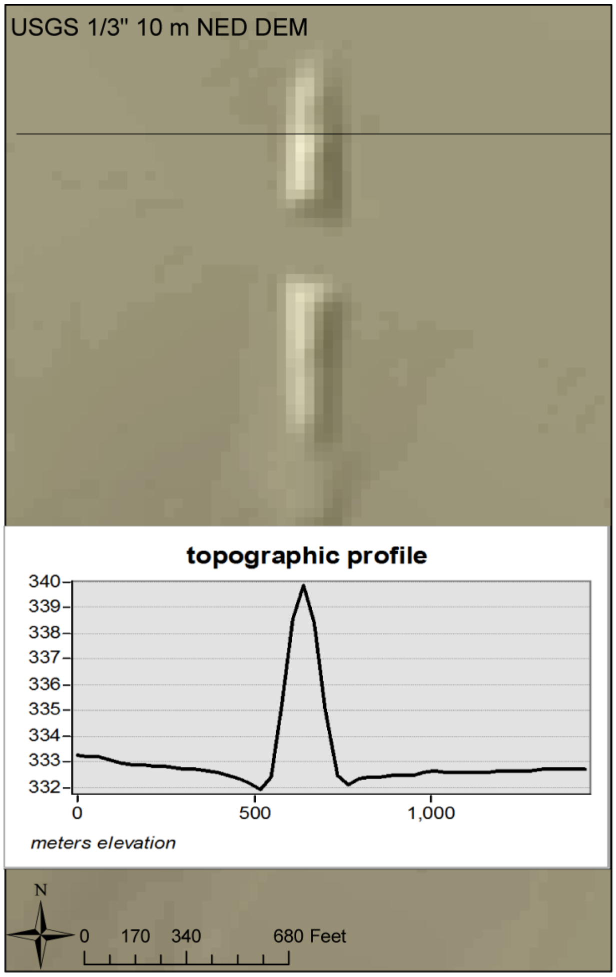
Finally, I compared the optimal LiDAR DEM (10 foot resolution computed with a 15 foot search radius) with the 10 meter resolution USGS 1/3" National Elevation Dataset (NED) DEM and with Phoenix 2006 0.8 ft aerial photography.

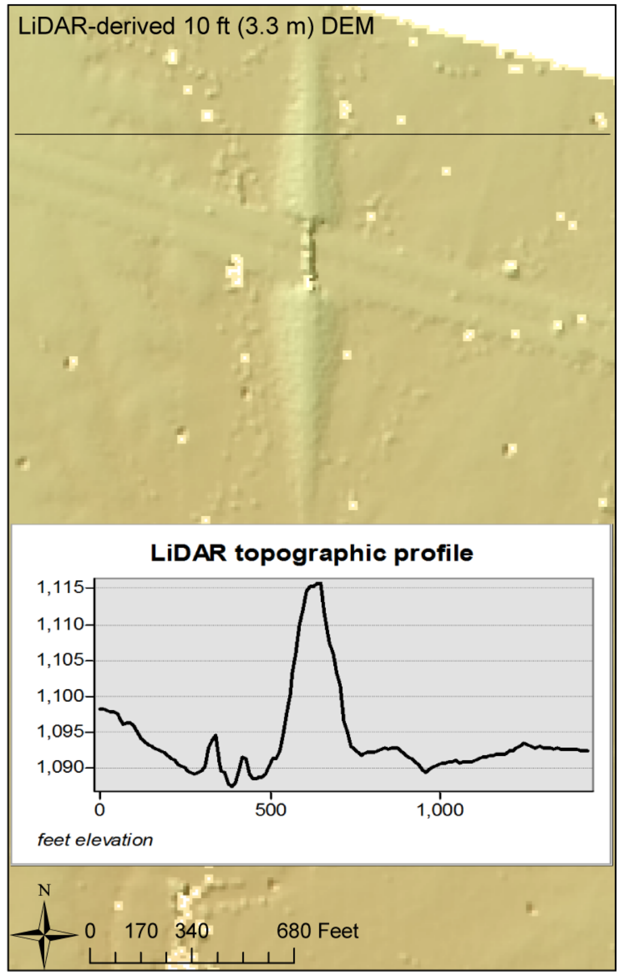


Dataset overview. Obviously the NED data are young enough that they include the I-10.



Zoom to northeastern portion of the dataset. Finer geomorphic features, riparian vegetation and what remains of cultural features are significantly degraded in the coarser NED data.





Comparison of topographic profiles from both datasets across the same transect (black line towards top of each map). Clearly the subtle geomorphology and some of the riparian vegetation is evident in the LiDAR-derived profile.

Comparison with high resolution (0.8 foot) Phoenix aerial photography. This quality and density of LiDAR data cannot compare well with the aerial photography. Higher shot densities would come closer to depicting the finer features in the landscape and biota.

Appendix A. Cleanme.m Maltab script for switching axes on Luke Wash LiDAR data.

```
%script to do some clean up on the Luke Wash LiDAR data
%JRA July 16, 2009
clear all
load new1099.pf1;
col1pf1 = new1099(:,1);
col2pf1 = new1099(:,2);
col3pf1 = new1099(:,3);

figure(1)
clf
plot(col1pf1, col2pf1, 'k.')
hold on

load new1099.pf2;
col1pf2 = new1099(:,1);
col2pf2 = new1099(:,2);
col3pf2 = new1099(:,3);
plot(col1pf2, col2pf2, 'r.')

load new1099.pf3;
col1pf3 = new1099(:,1);
col2pf3 = new1099(:,2);
col3pf3 = new1099(:,3);
plot(col1pf3, col2pf3, 'b.')

load new1099.pf4;
col1pf4 = new1099(:,1);
col2pf4 = new1099(:,2);
col3pf4 = new1099(:,3);
plot(col1pf4, col2pf4, 'g.')
legend('pf1', 'pf2', 'pf3', 'pf4')
title('Luke Wash LiDAR survey original geometry')
xlabel('Original column 1')
ylabel('Original column 2')
print -depsc original.eps

figure(2) %note that this switch is not necessary…
clf
plot(col2pf1, col1pf1, 'k.')
hold on
plot(col2pf2, col1pf2, 'r.')
plot(col2pf3, col1pf3, 'b.')
plot(col2pf4, col1pf4, 'g.')
legend('pf1', 'pf2', 'pf3', 'pf4')
title('Luke Wash LiDAR survey switched easting and northing')
ylabel('Original column 1')
xlabel('Original column 2')
print -depsc switched.eps

neweasting = [col2pf1; col2pf2; col2pf3; col2pf4];
size(neweasting)
newnorthing = [col1pf1; col1pf2; col1pf3; col1pf4];
size(newnorthing)
elevation = [col3pf1; col3pf2; col3pf3; col3pf4];
```

```
size(elevation)

%this is not quite working:
% fid = fopen('allswitched2', 'a+')
% fprintf(fid, '%.3f, %.3f, %.3f\n', neweasting, newnorthing, elevation)
% fclose(fid)
```

Appendix B. Analyze_points.m Matlab script to visualize Luke Wash Lidar data with selected points over hillshade

```
%script to plot the shots on a portion of the image
%JRA 7/16/09
%clear all
close all
figure(1)
clf
%load up the precomputed DEM and plot as a hillshade
Azimuth=315;
Zenith=45;
Zfact=1;
[X, Y, Easting, Northing, Elevation, dElev_dx, dElev_dy, nulls_count] =
LoadFile('10_15_all.idw.arc.asc');
Hillshds=hillshademe(Azimuth, Zenith, Zfact, dElev_dy, dElev_dx);
plothillshade(Easting, Northing, Hillshds)
xlabel('Easting (ft)')
ylabel('Northing (ft)')

%load all the points and concatenate them
load new1099.pf1;
col1pf1 = new1099(:,1);
col2pf1 = new1099(:,2);
col3pf1 = new1099(:,3);
load new1099.pf2;
col1pf2 = new1099(:,1);
col2pf2 = new1099(:,2);
col3pf2 = new1099(:,3);
load new1099.pf3;
col1pf3 = new1099(:,1);
col2pf3 = new1099(:,2);
col3pf3 = new1099(:,3);
load new1099.pf4;
col1pf4 = new1099(:,1);
col2pf4 = new1099(:,2);
col3pf4 = new1099(:,3);
newnorthing = [col2pf1; col2pf2; col2pf3; col2pf4];
size(newnorthing);
neweasting = [col1pf1; col1pf2; col1pf3; col1pf4];
size(neweasting);
 elevation = [col3pf1; col3pf2; col3pf3; col3pf4];
size(elevation);

%get the upper left and lower right of the area in which we want to plot
%the points

rangeulandlr = ginput;
xmin=rangeulandlr(1,1);
xmax=rangeulandlr(2,1);
ymin=rangeulandlr(2,2);
ymax=rangeulandlr(1,2);

plot([xmin xmax xmax xmin xmin], [ymin ymin ymax ymax ymin],'r-')

%let's zoom in to the red box
```

```
figure(2)
subplot(1,2,1)
expandaxis = 0.1;
widthofinterest=100;
plothillshade(Easting, Northing, Hillshds)

locs = find(neweasting > xmin & neweasting < xmax & newnorthing > ymin &
newnorthing < ymax);

epoints=neweasting(locs);
npoints=newnorthing(locs);

plot(epoints, npoints, 'r.')

%compute a little buffer for the range of map to display that is about 10%
(therange=0.1) wider than the selected data.

xrange = xmax-xmin;
yrange = ymax-ymin;

if xrange>yrange
    therange=xrange;
else
    therange=yrange;
end
xaxismin=xmin-(expandaxis.*therange);
xaxismax=xmax+(expandaxis.*therange);
yaxismin=ymin-(expandaxis.*therange);
yaxismax=ymax+(expandaxis.*therange);
axis([xaxismin xaxismax yaxismin yaxismax]);
xlabel('Easting (ft)')
ylabel('Northing (ft)')
%consider shot density
plot([xmin xmin+widthofinterest xmin+widthofinterest xmin xmin], [ymin ymin
ymin+widthofinterest ymin+widthofinterest ymin],'k-')
number_of_points = size(epoints);
areasampled = xrange.*yrange;
shotdensity=number_of_points(:,1)./areasampled;
s=sprintf('shot density = ~%3.0f points/%3.0f sq. ft\n', shotdensity.*10000,
widthofinterest.*widthofinterest);
text(xmin, ymin, s, 'VerticalAlignment','top')


%let's zoom in to the blue box
figure(1)
rangeulandlr = ginput;
xmin=rangeulandlr(1,1);
xmax=rangeulandlr(2,1);
ymin=rangeulandlr(2,2);
ymax=rangeulandlr(1,2);

plot([xmin xmax xmax xmin xmin], [ymin ymin ymax ymax ymin],'b-')

%let's zoom in
figure(2)
```

```
subplot(1,2,2)
expandaxis = 0.1;
widthofinterest=100;
plothillshade(Easting, Northing, Hillshds)

locs = find(neweasting > xmin & neweasting < xmax & newnorthing > ymin &
newnorthing < ymax);

epoints=neweasting(locs);
npoints=newnorthing(locs);

plot(epoints, npoints, 'b.')

xrange = xmax-xmin;
yrange = ymax-ymin;

if xrange>yrange
    therange=xrange;
else
    therange=yrange;
end
xaxismin=xmin-(expandaxis.*therange);
xaxismax=xmax+(expandaxis.*therange);
yaxismin=ymin-(expandaxis.*therange);
yaxismax=ymax+(expandaxis.*therange);
axis([xaxismin xaxismax yaxismin yaxismax]);
xlabel('Easting (ft)')
ylabel('Northing (ft)')
%consider shot density
plot([xmin xmin+widthofinterest xmin+widthofinterest xmin xmin], [ymin ymin
ymin+widthofinterest ymin+widthofinterest ymin],'k-')
number_of_points = size(epoints);
areasampled = xrange.*yrange;
shotdensity=number_of_points(:,1)./areasampled;
s=sprintf('shot density = ~%3.0f points/%3.0f sq. ft\n', shotdensity.*10000,
widthofinterest.*widthofinterest);
text(xmin, ymin, s, 'VerticalAlignment','top')
```

Appendix C. Bash script to concatenate files and run interp (linux version of P2G)

```
#!/bin/bash
#cat new1099.pf1 | awk 'BEGIN {FS = ","}; {print $1 "," $2 "," $3}'
>allswitched3
#cat new1099.pf2 | awk 'BEGIN {FS = ","}; {print $1 "," $2 "," $3}'
>>allswitched3
#cat new1099.pf3 | awk 'BEGIN {FS = ","}; {print $1 "," $2 "," $3}'
>>allswitched3
#cat new1099.pf4 | awk 'BEGIN {FS = ","}; {print $1 "," $2 "," $3}'
>>allswitched3

interp -i allswitched3 -o  10_1_all -r 1 --idwarc --resolution 10
interp -i allswitched3 -o  10_2_all -r 2 --idwarc --resolution 10
interp -i allswitched3 -o  10_5_all -r 5 --idwarc --resolution 10
interp -i allswitched3 -o  10_7_all -r 7 --idwarc --resolution 10
interp -i allswitched3 -o  10_10_all -r 10 --idwarc --resolution 10
interp -i allswitched3 -o  10_12_all -r 12 --idwarc --resolution 10
interp -i allswitched3 -o  10_15_all -r 15 --idwarc --resolution 10
interp -i allswitched3 -o  10_17_all -r 17 --idwarc --resolution 10
interp -i allswitched3 -o  10_20_all -r 20 --idwarc --resolution 10
interp -i allswitched3 -o  10_25_all -r 25 --idwarc --resolution 10
interp -i allswitched3 -o  10_50_all -r 50 --idwarc --resolution 10

tar cvf - *.arc | gzip -9c > run2.tar.gz
```

Appendix D. Matlab script analyze_radius_luke.m to count the number of nulls as a function of search radius for local binning. Needs functions LoadFile.m (appendix E), hillshademe.m (appendix F), and plothillshade.m (appendix G)

```
%Script to analyze effect of search radius variation
%7/16/2009
%JRA, some code borrowed from OZ

%Illumination parameters
Azimuth=315;
Zenith=45;
Zfact=1;
m=3;
n=2;
r=[];
nulls=[];
label_offset=100;
i=0;
figure(1)
clf
[X, Y, Easting, Northing, Elevation, dElev_dx, dElev_dy, nulls_count] =
LoadFile('10_1_all.idw.arc');
r=[r 1];
nulls=[nulls nulls_count];

[X, Y, Easting, Northing, Elevation, dElev_dx, dElev_dy, nulls_count] =
LoadFile('10_2_all.idw.arc');
r=[r 2];
nulls=[nulls nulls_count];

i=i+1
subplot(m,n,i)
[X, Y, Easting, Northing, Elevation, dElev_dx, dElev_dy, nulls_count] =
LoadFile('10_5_all.idw.arc');
Hillshds=hillshademe(Azimuth, Zenith, Zfact, dElev_dy, dElev_dx);
plothillshade(Easting, Northing, Hillshds)
text(Easting(1)-label_offset, Northing(1), ['r=5 ' num2str(nulls_count)])
r=[r 5];
nulls=[nulls nulls_count];
xlabel('Easting');
ylabel('Northing');
title('Hillshade map of topography');

[X, Y, Easting, Northing, Elevation, dElev_dx, dElev_dy, nulls_count] =
LoadFile('10_7_all.idw.arc');
r=[r 7];
nulls=[nulls nulls_count];

i=i+1
subplot(m,n,i)
[X, Y, Easting, Northing, Elevation, dElev_dx, dElev_dy, nulls_count] =
LoadFile('10_10_all.idw.arc');
Hillshds=hillshademe(Azimuth, Zenith, Zfact, dElev_dy, dElev_dx);
plothillshade(Easting, Northing, Hillshds)
text(Easting(1)-label_offset, Northing(1), ['r=10 ' num2str(nulls_count)])
r=[r 10];
```

```
nulls=[nulls nulls_count];

i=i+1
subplot(m,n,i)
[X, Y, Easting, Northing, Elevation, dElev_dx, dElev_dy, nulls_count] =
LoadFile('10_15_all.idw.arc.asc');
Hillshds=hillshademe(Azimuth, Zenith, Zfact, dElev_dy, dElev_dx);
plothillshade(Easting, Northing, Hillshds)
text(Easting(1)-label_offset, Northing(1), ['r=15 ' num2str(nulls_count)])
r=[r 15];
nulls=[nulls nulls_count];

[X, Y, Easting, Northing, Elevation, dElev_dx, dElev_dy, nulls_count] =
LoadFile('10_17_all.idw.arc');
r=[r 17];
nulls=[nulls nulls_count];

[X, Y, Easting, Northing, Elevation, dElev_dx, dElev_dy, nulls_count] =
LoadFile('10_20_all.idw.arc.asc');
r=[r 20];
nulls=[nulls nulls_count];

[X, Y, Easting, Northing, Elevation, dElev_dx, dElev_dy, nulls_count] =
LoadFile('10_25_all.idw.arc');
r=[r 25];
nulls=[nulls nulls_count];

i=i+1
subplot(m,n,i)
[X, Y, Easting, Northing, Elevation, dElev_dx, dElev_dy, nulls_count] =
LoadFile('10_50_all.idw.arc');
Hillshds=hillshademe(Azimuth, Zenith, Zfact, dElev_dy, dElev_dx);
plothillshade(Easting, Northing, Hillshds)
text(Easting(1)-label_offset, Northing(1), ['r=50 ' num2str(nulls_count)])
r=[r 50];
nulls=[nulls nulls_count];

i=i+1
subplot(m,n,i)
[X, Y, Easting, Northing, Elevation, dElev_dx, dElev_dy, nulls_count] =
LoadFile('10_10_den.den.arc.asc');
plothillshade(Easting, Northing, Elevation./314.159)
colorbar

i=i+1
subplot(m,n,i)
plot(r,nulls, 'k-', r,nulls,'r.')
```

Appendix E. Matlab function LoadFile.m reads an Arc Ascii grid and counts the nulls among other things.

```
function [X, Y, Easting, Northing, Elevation, dElev_dx, dElev_dy,
nulls_count] = LoadFile(file_name)

fid=fopen(file_name);
            firstline    = fgetl(fid);
            ncols        = str2double(firstline(7:length(firstline)));
            secondline   = fgetl(fid);
            nrows        = str2double(secondline(7:length(secondline)));
            thirddline   = fgetl(fid);
            xllcorner    = str2double(thirddline(11:length(thirddline)));
            fourthline   = fgetl(fid);
            yllcorner    = str2double(fourthline(11:length(fourthline)));
            fifthline    = fgetl(fid);
            cellsize     = str2double(fifthline(10:length(fifthline)));
            sixthline    = fgetl(fid);
            NODATA_value = str2double(sixthline(14:length(sixthline)));
            Elevation    = fscanf(fid, '%g', [ncols nrows]);
            size(Elevation);
fclose(fid);
            nulls_count = length(find(Elevation==NODATA_value)); %count nulls
            Elevation(find(Elevation==NODATA_value)) = NaN;
            Easting    = xllcorner:cellsize:(xllcorner+(ncols-1)*cellsize);
            Northing   = yllcorner:cellsize:(yllcorner+(nrows-1)*cellsize);
            Elevation  = flipud(Elevation');

[X, Y]=meshgrid(Easting,Northing);

        dx = abs(Easting(2) -Easting(1));  % get cell spacing in x and y
direction
        dy = abs(Northing(2)-Northing(1));  % from coordinate vectors

        [dElev_dx, dElev_dy]   = gradient(Elevation,dx,dy);
```

Appendix F. Matlab function hillshademe.m computes a hillshade as a function of illuminaton angle. Originally written by Olaf Zielke with some help from matlab central.

```
function Hillshds=hillshademe(CurrentAzimuth, CurrentZenith, CurrentZ_fact,
dElev_dy, dElev_dx)
      %Azimuth is 0 towards south and increases counterclockwise so 135 is
      %NE and 225 is NW

            CurrentAzimuth = 360.0-CurrentAzimuth+90; %convert to mathematic
unit
            CurrentAzimuth(CurrentAzimuth>=360)=CurrentAzimuth-360;
            CurrentAzimuth = CurrentAzimuth * (pi/180); %  convert to radians

            %lighting altitude
            CurrentZenith = (90-CurrentZenith) * (pi/180); % convert to
zenith angle in radians


            [asp,grad] = cart2pol(dElev_dy,dElev_dx); % convert to carthesian
coordinates
            grad       = atan(CurrentZ_fact*grad); %steepest slope
            % convert asp
            asp(asp<pi)=asp(asp<pi)+(pi/2);
            asp(asp<0)=asp(asp<0)+(2*pi);


            Hillshds =  255.0*( (cos(CurrentZenith).*cos(grad) ) + (
sin(CurrentZenith).*sin(grad).*cos(CurrentAzimuth-asp)) ); % ESRIs algorithm
            Hillshds(Hillshds<0)=0; % set hillshade values to min of 0.
```

Appendix G. Matlab function plothillshade.m simply plots a hillshade.

```
function plothillshade(Easting, Northing, Hillshds)

plot(Easting(1,1),Northing(1,1), 'b.');
hold on
imagesc(Easting,Northing,Hillshds);
colormap('bone');
%colormap('summer');
axis equal;
```