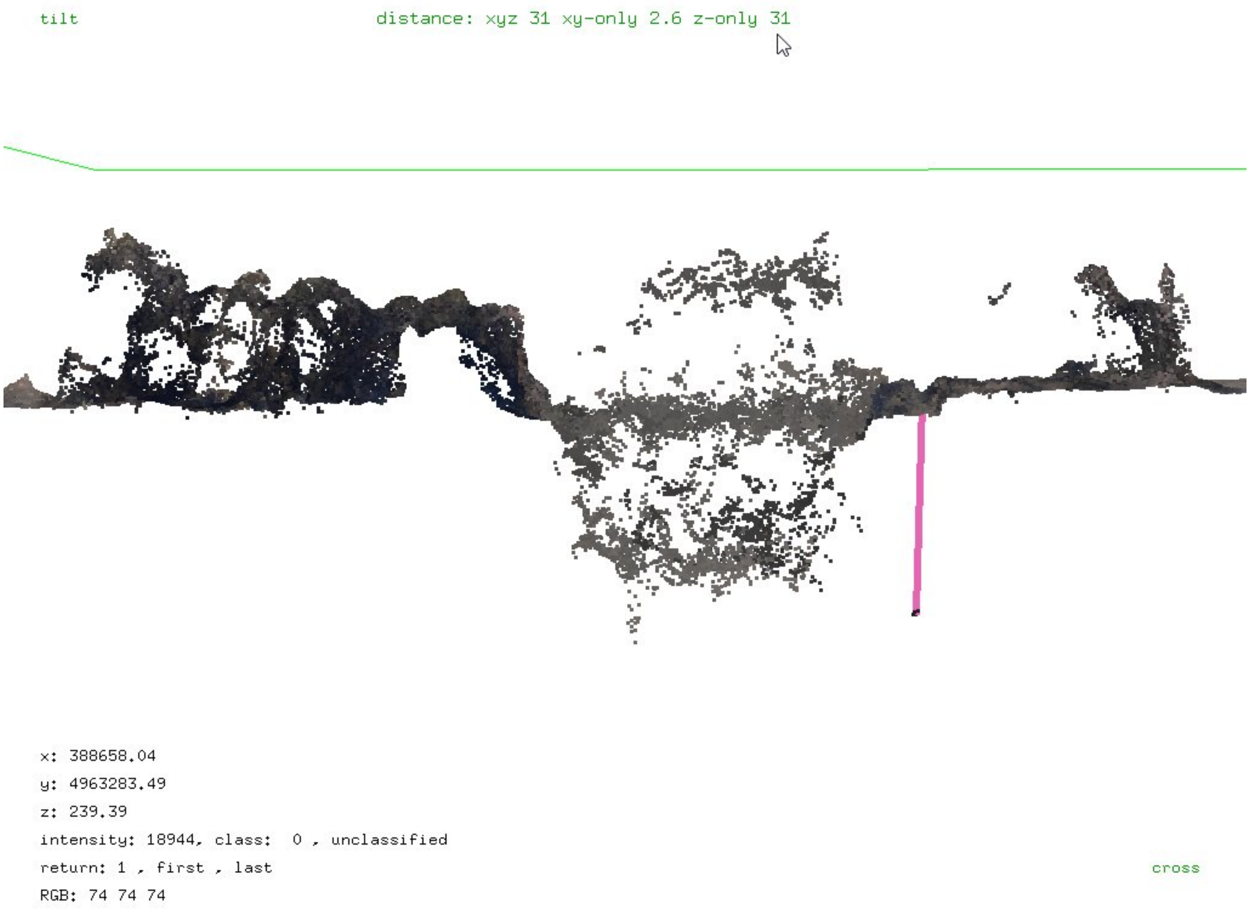
REMOVING EXCESSIVE LOW NOISE FROM DENSE-MATCHING POINT CLOUDS

Martin Isenburg
[1]rapidlasso GmbH, Casparigasse 16, 97296 Sommerhausen, GERMANY
Email: martin@rapidlasso.com

KEY WORDS: point,clouds, photogrammetry, dense-matching, aerial imagery, noise removal

ABSTRACT: Point clouds produced from overlapping aerial imagery using dense-matching such as implemented by the photogrammetry software suites from SURE, Pix4D, or Photoscan can include a fair amount of the kind of "low noise", namely error points well below the true terrain. Such low noise causes trouble when attempting to construct a Digital Terrain Model (DTM) from these photogrammetric points as common algorithm for classifying points into ground and non-ground points tend to “latch onto” those low points, thereby producing a very poor representation of the terrain. In this article we describes one possible workflow for eliminating excessive low noise on one particular example data set from a region in Italy that was generated it with the Agisoft Photoscan software.
The original file contains 87,261,083 dense-matching points and covers an area of roughly 12.8 square kilometers with an average spacing of 38 cm between the points. We describe a pipeline that generously marks all low points as noise by first breaking the points into smaller 500 by 500 meter tiles, then selecting the highest point of every 2.5 by 2.5 meter grid, then removing isolated points from those, then temporarily classifying the remaining points into ground and non-ground, then constructing a temporary ground surface that is only used to mark all points that are 0.5 meters of lower as noise. These points are then ignored and a standard ground classification algorithm can be used to create a reasonable DTM that is not impacted by the excessive low noise points.
Both the data set as well as the software modules used to remove the low noise will be available to the attendees so that they may reproduce this processing pipeline on their own after the conference.



```
distance: xyz 31 xy-only 2.6 z-only 31

x: 388658.04
y: 4963283.49
z: 239.39
intensity: 18944, class:  0 , unclassified
return: 1 , first , last
RGB: 74 74 74
```

Figure 1: There is a lot of low noise in tile ‘panca_388500_4963000.laz’. The vertical distance shown is 31 meters.

1. INTRODUCTION

Point clouds produced from overlapping aerial imagery using dense-matching such as implemented by the photogrammetry software suites from SURE, Pix4D, or Photoscan can include a fair amount of the kind of "low noise", namely error points well below the true terrain. Such low noise causes trouble when attempting to construct a Digital Terrain Model (DTM) from these photogrammetric points as common algorithm for classifying points into ground and non-ground points tend to "latch onto" those low points, thereby producing a very poor representation of the terrain. In this article we describes one possible workflow for eliminating excessive low noise on one particular example data set from a region in Italy that was provided by Miss Muriel Lavy who generated it with the Agisoft Photoscan software and of which a typical example can be seen Figure 1. Both the data set as well as the software modules used to remove the low noise will be available to the attendees so that they may reproduce this processing pipeline on their own after the conference.

We leave the usual inspection of the content with lasinfo, lasview, and lasvalidate that we always recommend on newly obtained data as an exercise to the reader. Note that a check for proper alignment of flightlines with lasoverlap that we consider mandatory for LiDAR data is not applicable for dense-matching points as the concept of independent strips of elevation samples collected in separate flights over the terrain does not exist. Instead the elevations samples are computed from the (necessarily) heavy overlap between the aerial images.

2. DATA PROCESSING

With lastile we turn the original file with 87,261,083 points into many smaller 500 by 500 meter tiles for efficient multi-core processing. Each tile is given a 25 meter buffer to avoid edge artifacts. The buffer points are marked as withheld for easier on-the-fly removal. We add a (terser) description of the WGS84 UTM zone 32N to each tile via the corresponding EPSG code 32632

```
lastile -i muriel\20161127_Pancalieri_UTM.laz ^
        -set_classification 0 ^
        -tile_size 500 -buffer 25 -flag_as_withheld ^
        -epsg 32632 ^
        -odir muriel\tiles_raw -o panca.laz
```

Because dense-matching points often have a poor point order in the files they get delivered in we use first rearrange them into a space-filling curve order as this will speed up most following processing steps.

```
lassort -i muriel\tiles_raw\panca*.laz ^
        -odir muriel\tiles_sorted -olaz ^
        -cores 7
```

We then reclassify the highest point of every 2.5 by 2.5 meter grid cell with classification code 8. As the spacing of the dense-matched points is around 40 cm in both x and y, around 40 points will fall into each such grid cell from which the highest is then classified as 8:

```
lasthin -i muriel\tiles_sorted\panca*.laz ^
        -step 2.5 ^
        -highest -classify_as 8 ^
        -odir muriel\tiles_thinned -olaz ^
        -cores 7
```

Considering only those points classified as 8 in the last step we then run lasnoise to find points that are highly isolated in wide and flat neighborhoods that are then reclassified as 7. See the README file of lasnoise for a detailed explanation of the different parameters:

```
lasnoise -i muriel\tiles_thinned\panca*.laz ^
         -ignore_class 0 ^
         -step_xy 5 -step_z 0.1 -isolated 4 ^
         -classify_as 7 ^
         -odir muriel\tiles_isolated -olaz ^
         -cores 7
```

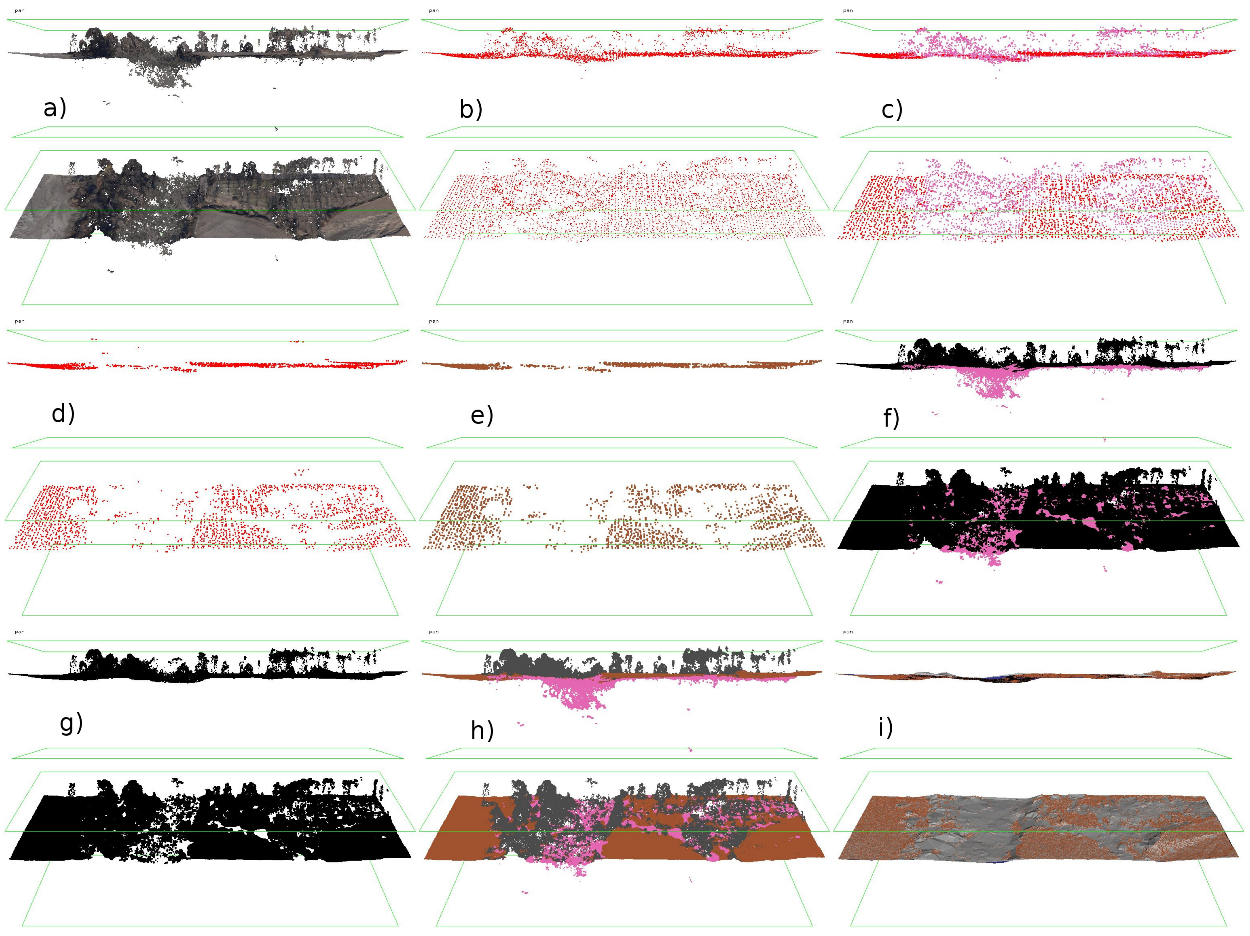


Figure 2: Results of the individual processing steps: (a) raw and noisy input points, (b) lasthin classifies highest point per 2.5 by 2.5 meters as 8 (red), (c) lasnoise classifies isolated points (pink) as noise, (d) remaining thinned points that are not classified as noise, (e) lasground classified temporary ground points (brown) within this subset, (f) lasheight marks all points below the temporay ground as noise, (g) the resulting denoised points only, (h) another run of lasground classifies the final ground points (brown) working only on those denoised points but the noise points (pink) are kept, (i) a TIN (triangular irregular network) that was constructed from the final ground points.

Now we run a temporary ground classification of only (!!!) on those points that are still classified as 8 using the default parameters of lasground. Hence we only use the points that were the highest points on the 2.5 by 2.5 meter grid and that were not classified as noise in the previous step. See the README file of lasground for a detailed explanation of the different parameters:

```
lasground -i muriel\tiles_isolated\panca*.laz ^
          -city -ultra_fine -ignore_class 0 7 ^
          -odir muriel\tiles_temp_ground -olaz ^
          -cores 7
```

The result of this temporary ground filtering is then merely used to mark all points that are 0.5 meter below the triangulated TIN of these temporary ground points with classification code 12 using lasheight. See the README file of lasheight for a detailed explanation of the different parameters:

```
lasheight -i muriel\tiles_temp_ground\panca*.laz ^

          -do_not_store_in_user_data ^

          -classify_below -0.5 12 ^
          -odir muriel\tiles_temp_denoised -olaz ^
          -cores 7
```

In the resulting tiles the low noise (but also many points above the ground) are now marked and in a final step we produce properly classified denoised tiles by re-mapping the temporary classification codes to conventions that are more consistent with the ASPRS LAS specification using las2las:

```
las2las -i muriel\tiles_temp_denoised\panca*.laz ^
        -change_classification_from_to 1 0 ^
        -change_classification_from_to 2 0 ^
        -change_classification_from_to 7 0 ^
        -change_classification_from_to 12 7 ^
        -odir muriel\tiles_denoised -olaz ^
        -cores 7
```

The final classification of all points that are not already classified as noise (7) into ground (2) or non-ground (1) was done with a final run of lasground. See the README file of lasground for a detailed explanation of the different parameters:

```
lasground -i muriel\tiles_denoised\panca*.laz ^
          -ignore_class 7 ^
          -city -ultra_fine ^
          -odir muriel\tiles_ground -olaz ^
          -cores 7
```

Then we create a seamless hill-shaded DTM tiles by triangulating all the points classified as ground into a temporary TIN (including those in the 25 meter buffer) and then rasterizing only the inner 500 meter by 500 meter of each tile with option '-use_tile_bb' of las2dem. For more details on the importance of buffers in tile-based processing see this blog post [2] here.

```
las2dem -i muriel\tiles_ground\panca*.laz ^
        -keep_class 2 ^
        -step 1 -hillshade ^
        -use_tile_bb ^
        -odir muriel\tiles_dtm -opng ^
        -cores 7
```

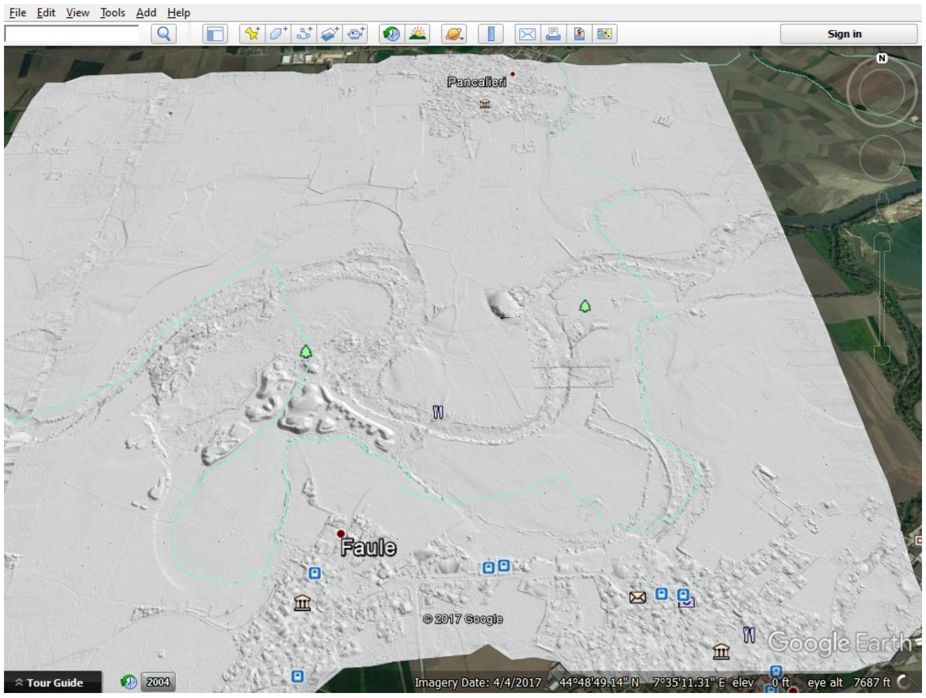
3. RESULTS

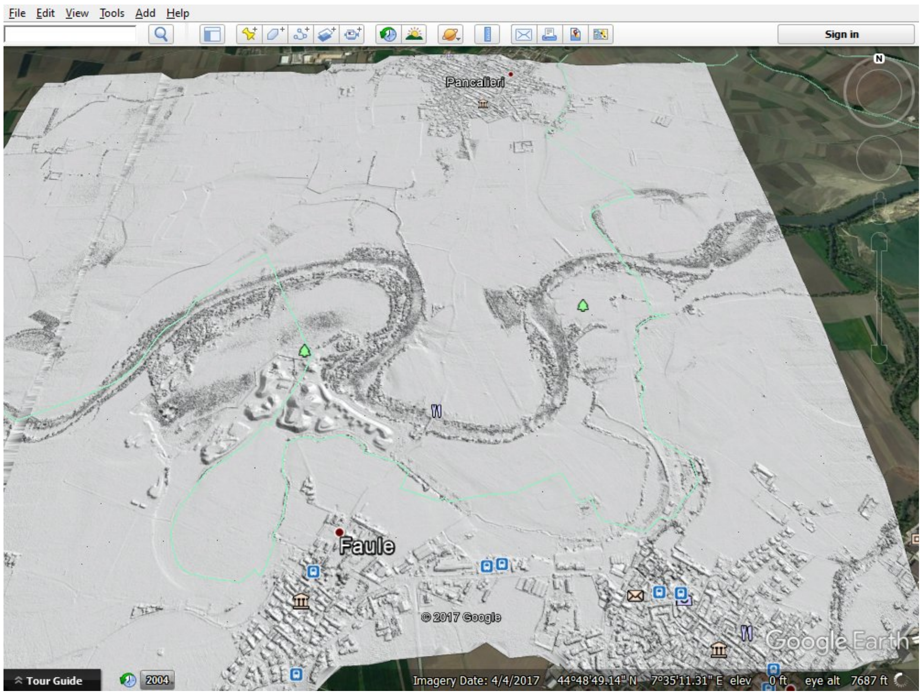
Let us visually check what each of the above steps has produced by zooming in on a 300 meter by 100 meter strip of points with the bounding box (388500,4963125) to (388800,4963225) in tile 'panca_388500_4963000.laz':

In Figure 2 you see (a) the raw colored point cloud., (b) the highest point per 2.5 by 2.5 meter grid cell classified as 8 (red), (c) isolated points as noise marked with classification 7 (pink), (d) point surviving thinning and noise removal are input for temporary ground classification, (e) temporary ground points (brown) used for low noise removal, (f) all raw points falling 0.5 meter under TIN of temporary ground points become low noise (pink), (g) points that are not low noise are used as input for final ground classification, (h) final ground classification (brown) and low noise points (pink), (i) resulting ground TIN via triangulation of final ground points.

4. CONCLUSION

Our method is successful in removing excessive noise typical in photogrammetric points that are well below the true terrain. Such low noise causes trouble when attempting to construct a Digital Terrain Model (DTM) from the points. Here you see the original DSM side-by-side with resulting DTM after low noise removal. One dense forested area near the center of the data was not entirely removed due to the lack of ground points in this area. Integrating external ground points [3] or manual editing with lasview are two possible way to rectify these few remaining errors …


File Edit View Tools Add Help
Sign in
Pancalieri
Faule
© 2017 Google
Google Earth
Tour Guide
2004
Imagery Date: 4/4/2017 44°48'49.14" N 7°35'11.31" E elev 0 ft eye alt 7687 ft


File Edit View Tools Add Help
Sign in
Pancalieri
Faule
© 2017 Google
Google Earth
Tour Guide
2004
Imagery Date: 4/4/2017 44°48'49.14" N 7°35'11.31" E elev 0 ft eye alt 7687 ft

ADDITIONAL MATERIAL:
[1] http://rapidlasso.com/2017/07/04/removing-excessive-low-noise-from-dense-matching-point-clouds/
[2] http://rapidlasso.com/2015/08/07/use-buffers-when-processing-lidar-in-tiles/
[3] http://rapidlasso.com/2017/06/13/integrating-external-ground-points-in-forests-to-improve-dtm-from-dense-matching-photogrammetry/