INTEGRATING EXTERNAL GROUND POINTS IN FORESTS TO CREATE DTMS FROM DENSE-MATCHING PHOTOGRAMMETRY

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

KEY WORDS: point,clouds, photogrammetry, dense-matching, forestry, external ground points

ABSTRACT: The biggest problem of generating a Digital Terrain Model (DTM) from the point clouds produced from aerial imagery with photogrammetry software is dense vegetation: when plants completely cover the terrain not a single point is generated on the ground. The complete lack of ground points in larger vegetated areas such as closed forests or plantations means that the many processing workflows for vegetation analysis that have been developed for LiDAR cannot be used for photogrammetric point clouds unless we are getting those missing ground points some other way. In the following we describe how to integrate external ground information with dense matching point clouds such that a reasonable bare-earth terrain model can be created and tree heights can be measured.

Our dense-matching example input has 35,338,368 points covering 3.4 square kilometer with an average point spacing of 31 centimeter. Attempts to ground-classify this point cloud directly are futile as there are no ground points under the canopy in the forested area. Therefore 558 ground points were manually surveyed in the forest of interest. They are spaced around 50 to 120 meters apart from another.

We first "densify" the manually collected ground points by interpolating them onto a two meter raster that is then clipped against a polygon delineating the forest of interest. We merge the result with the dense matching points and mark the lowest point per one meter grid cell. The marked points are then classified into ground and non-ground. Then the height of each point above the triangulation (TIN) of ground points can be computed. In all areas where external ground information was available we can now compute a reasonable Canopy Height Model (CHM) and measure tree heights.

All the data as well as the software modules will be available to the attendees via this Web link [1] so that they may reproduce the presented methodology after the conference.

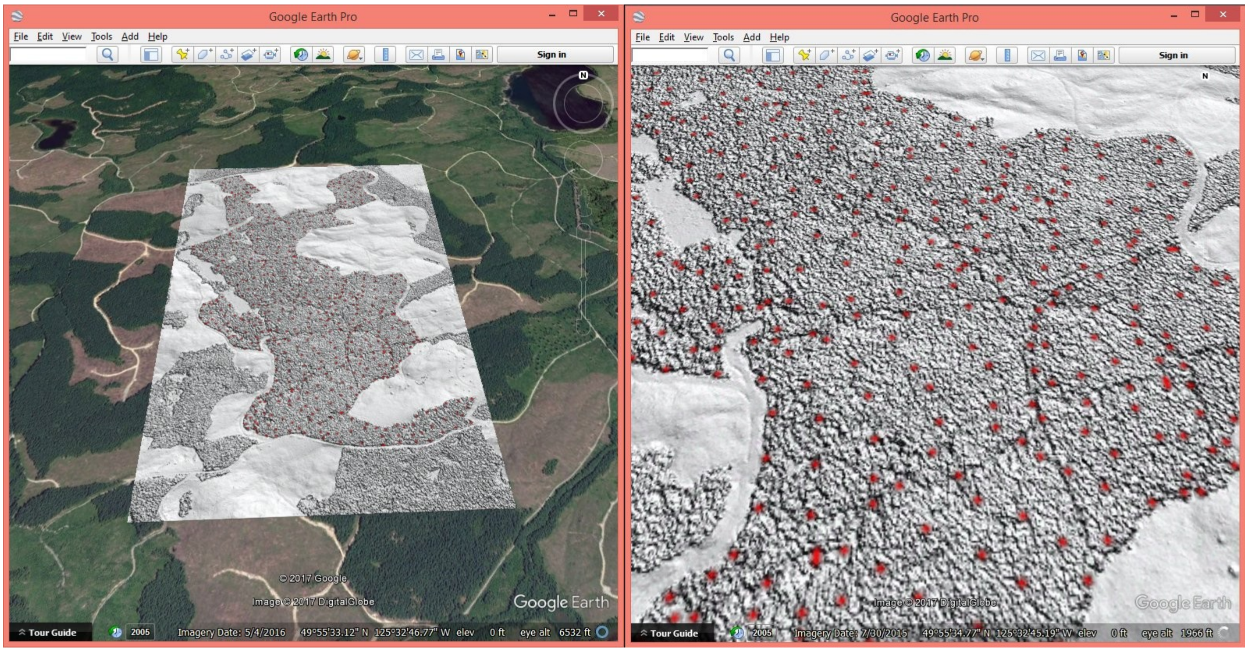


Figure 1: Hillshade of dense matching points and location of manual ground control points (red)

1. INTRODUCTION

The biggest problem of generating a Digital Terrain Model (DTM) from the photogrammetric point clouds that are produced from aerial imagery with dense-matching software such as SURE, Pix4D, or Photoscan is dense vegetation: when plants completely cover the terrain not a single point is generated on the ground. This is different for LiDAR point clouds as the laser can even penetrate dense multi-level tropical forests. The complete lack of ground points in larger vegetated areas such as closed forests or dense plantations means that the many processing workflows for vegetation analysis that have been developed for LiDAR cannot be used for photogrammetric point clouds … unless … well unless we are getting those missing ground points some other way. In the following we see how to integrate external ground points to generate a reasonable DTM under a dense forest with LAStools.

From this web site [1] you can download the dense matching point cloud, the manually collected ground points, and the forest stand delineating polygon that we are using in the following example workflow. In Figure 1 you can see a hillshading of the dense matching points created with las2dem together with the location of the manually collected ground control points (red) created with lasgrid with the commen lines shown below.

```
las2dem -i DenseMatching.laz ^
        -thin_with_grid 1.0 ^
        -extra_pass ^
        -step 2.0 ^
        -hillshade ^
        -odix _hill_2m -opng

lasgrid -i ManualGround.laz ^
        -set_RGB 255 0 0 ^
        -step 10 -rgb ^
        -odix _grid_10m -opng
```

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

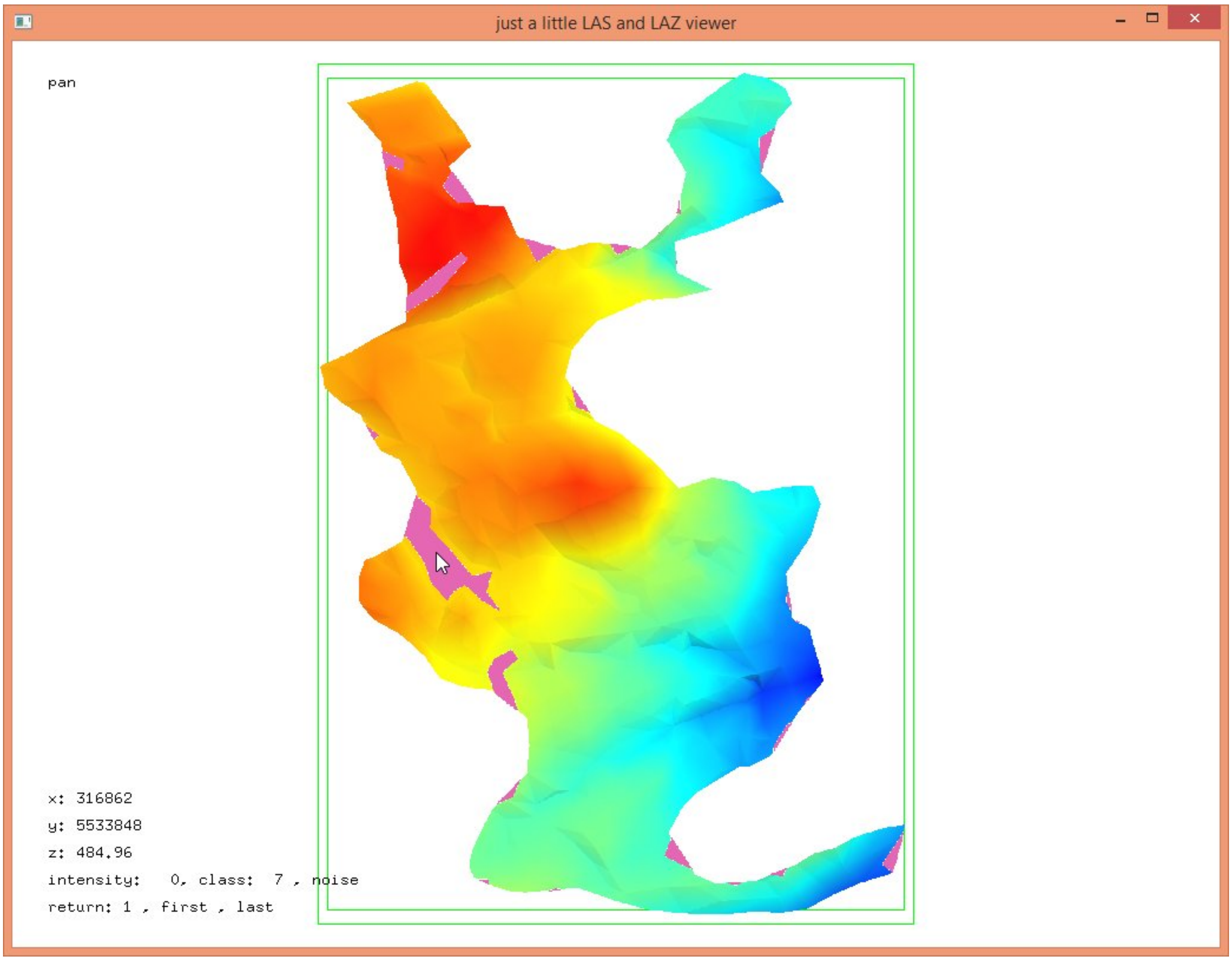
Attempts to ground-classify the dense matching point cloud directly are futile as there are no ground points under the canopy in the heavily forested area. Therefore 558 ground points were manually surveyed in the forest of interest that are around 50 to 120 meters apart from another. We show how to integrate these points into the dense matching point cloud such that we can successfully extract bare-earth information from the data.

In the first step we "densify" the manually collected ground points by interpolating them with triangles onto a raster of 2 meter resolution that we store as LAZ points with las2dem. You could consider other interpolation schemes to "densify" the ground points, here we use simple linear interpolation to prove the concept. Due to the varying distance between the manually surveyed ground points we allow interpolating triangles with edge lengths of up to 125 meters. These triangles then also cover narrow open areas next to the forest, so we clip the interpolated ground points against the forest stand delineating polygon with lasclip to classify those points that are really in the forest as "key points" (class 8) and all others as "noise" (class 7).

```
las2dem -i ManualGround.laz ^
        -step 2 ^
        -kill 125 ^
        -odix _2m -olaz

lasclip -i ManualGround_2m.laz ^
        -set_classification 7 ^
        -poly forest.shp ^
        -classify_as 8 -interior ^
        -odix _forest -olaz
```

In Figure 2 we show the resulting densified ground points colored by elevation that survive the clipping against the forest stand delineating polygon and were classified as "key points" (class 8). The interpolated ground points in narrow open areas next to the forest that fall outside this polygon were classified as “noise” (class 7) and are shown in violet. They will be dropped in the next step.



```
x: 316862
y: 5533848
z: 484.96
intensity:   0, class:  7 , noise
return: 1 , first , last
```

Figure 2: Densified ground points colored by elevation, points eliminated by clipping against forest stand in violet.

We then merge the dense matching points with the densified manual ground points (while dropping all the violet points marked as noise) as input to lasthin and reclassify the lowest point per 1 meter by 1 meter with a temporary code (here we use class 9 that usually refers to “water”). Only the subset of lowest points that receives the temporary classification code 9 will be used for ground classification later.

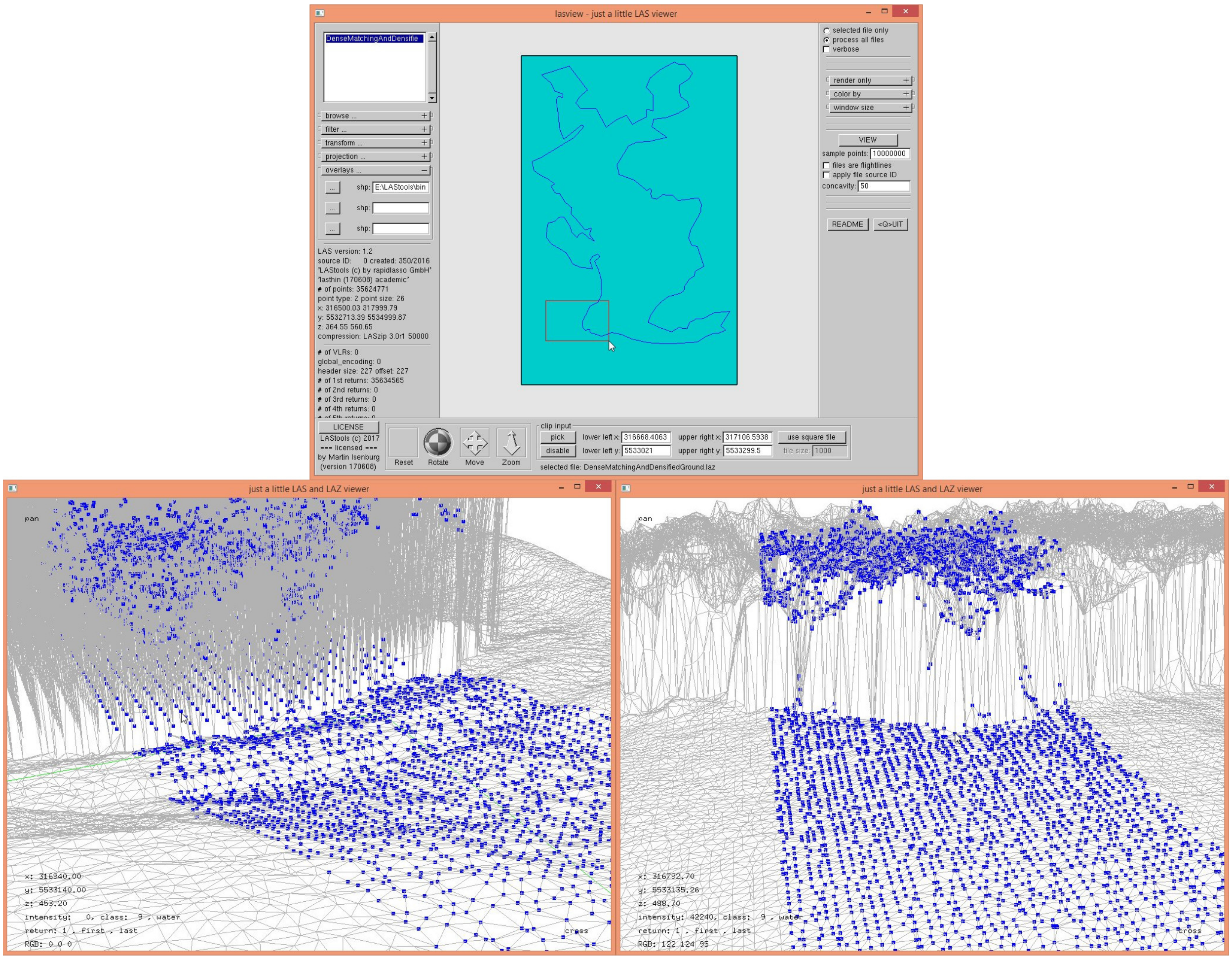
```
lasthin -i DenseMatching.laz ^
        -i ManualGround_2m_forest.laz ^
        -drop_class 7 ^
        -merged ^
        -lowest -step 1 -classify_as 9 ^
        -o DenseMatchingAndDensifiedGround.laz
```

We use the GUI of lasview to pick several interesting areas for visual inspection. The selected points load much faster when the LAZ file is spatially indexed and therefore we first run lasindex. For better orientation we also load the forest stand delineating polygon as an overlay into the GUI.

```
lasindex -i DenseMatchingAndDensifiedGround.laz
lasview -i DenseMatchingAndDensifiedGround.laz -gui
```

We pick the area shown in Figure 3 (top) that contains the target forest with manually collected and densified ground points (bottom left) and a forested area with only dense matching points (bottom right). The difference between the two in terms of number of ground points could not be more drastic as the visualizations clearly show.

We overlay the polygon for orientation and sample 10 million points in the selected area. The wireframe view of triangulating the lowest point in each square meter (class 9). The densified ground points are clearly visible as a grid under the canopy in the target forest. Other forested areas do not have their lowest points anywhere near the ground.



shp: E:\LAStools\bin

shp:

shp:

```
LAS version: 1.2
source ID:     0 created: 350/2016
'LAStools (c) by rapidlasso GmbH'
'lasthin (170608) academic'
# of points: 35624771
point type: 2 point size: 26
x: 316500.03 317999.79
y: 5532713.39 5534999.87
z: 364.55 560.65
compression: LASzip 3.0r1 50000
# of VLRs: 0
global_encoding: 0
header size: 227 offset: 227
# of 1st returns: 35634565
# of 2nd returns: 0
# of 3rd returns: 0
# of 4th returns: 0
```

LAStools (c) 2017
=== licensed ===
by Martin Isenburg
(version 170608)

○ selected file only
● process all files
☐ verbose

sample points: 10000000
☐ files are flightlines
☐ apply file source ID
concavity: 50

clip input
lower left x: 316668.4063 upper right x: 317106.5938
lower left y: 5533021 upper right y: 5533299.5
tile size: 1000
selected file: DenseMatchingAndDensifiedGround.laz

```
x: 316940.00
y: 5533140.00
z: 453.20
intensity:    0, class:  9 , water
return: 1 , first , last
RGB: 0 0 0
```

```
x: 316792.70
y: 5533135.26
z: 488.70
intensity: 42240, class:  9 , water
return: 1 , first , last
RGB: 122 124 95
```

Figure 3: Clipped area (top),target forest with manually collected and densified ground points (bottom left). forested area with only dense matching points (bottom right).

Now we run ground classification using lasground with option '-town' using only the points with the temporary code 9 by ignoring all other classifications 0 and 8 in the file. We leave the temporary classification code 9 unchanged for all the points that were not classified with "ground" code 2 so we can visualize them later.

```
lasground -i DenseMatchingAndDensifiedGround.laz ^
          -ignore_class 0 8 ^
          -town ^
          -non_ground_unchanged ^
          -o GroundClassified.laz
```

We use the GUI of lasview to pick several interesting areas after running lasindex and again load the forest stand delineating polygon as an overlay into the GUI. We pick an area that contains all three scenarios: the target forest with manually collected and densified ground points, an open area with only dense matching points, and a forested area with only dense matching points. The result shown in Figure 4 is as expected: in the target forest the manually collected ground points are used as ground and in the open area the dense-matching points are used as ground. But there is no useful ground in the other forested area.

```
lasindex -i GroundClassified.laz
lasview -i GroundClassified.laz -gui
```

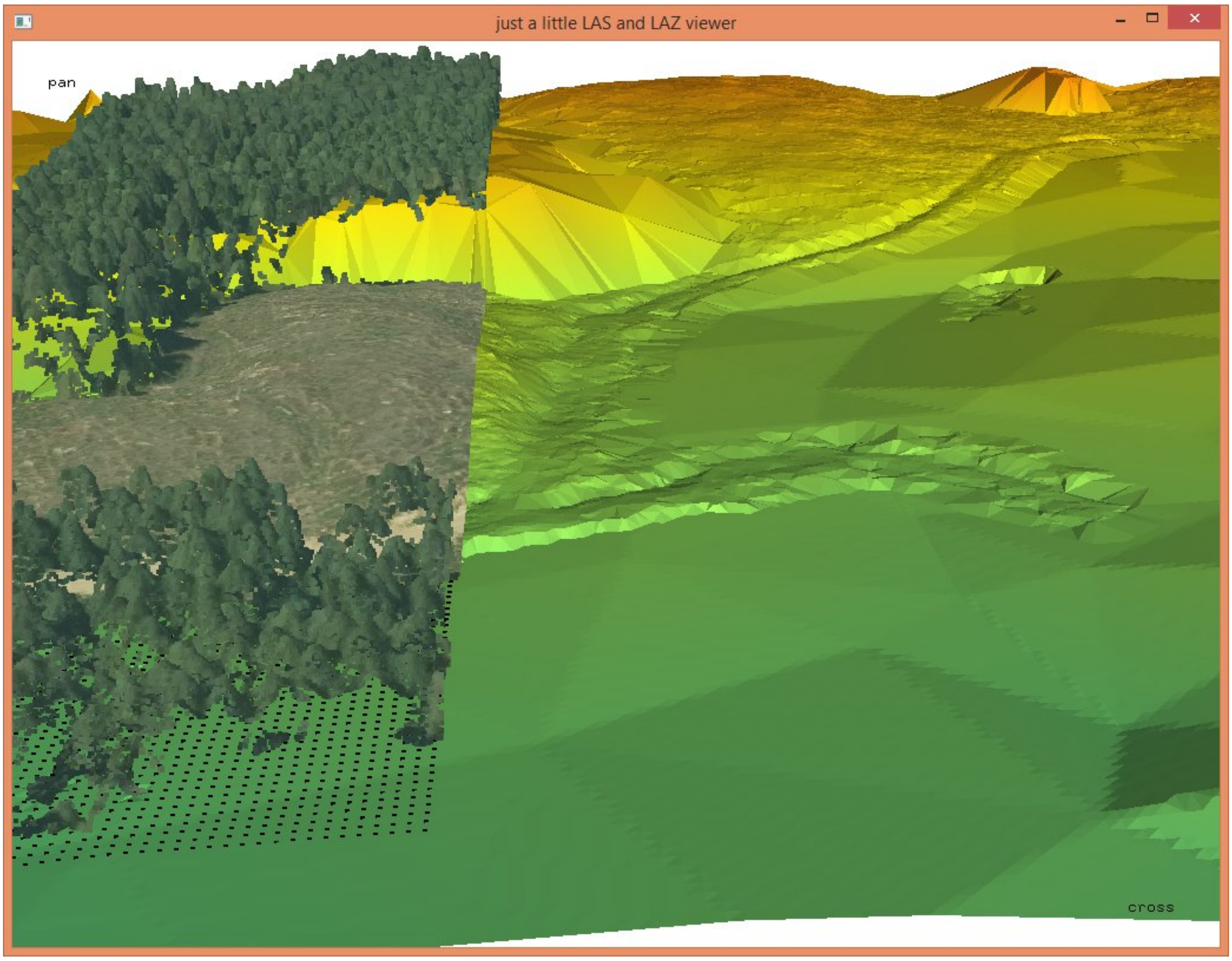


Figure 4: Ground points triangulated into a shaded TIN with a strip of all points rendered on top. Only for the forest using external information (front) the terrain makes sense. Elsewhere the TIN latches onto tree tops (back).

Now we can compute the heights of the points above ground for our target forest with lasheight and either replace the z elevations in the file of store them separately as “extra bytes”. Then we can compute, for example, a Canopy Height Model (CHM) that color codes the height of the vegetation above the ground with lasgrid. Of course this will only be correct in the target forest where we have “good” ground but not in the other forested areas. We also compute a hillshaded DTM to be able to visually inspect the topography of the generated terrain model.

```
lasheight -i GroundClassified.laz ^
          -store_as_extra_bytes ^
          -o GroundClassifiedWithHeights.laz

lasgrid -i GroundClassifiedWithHeights.laz ^
        -step 2 ^
        -highest -attribute 0 ^
        -false -set_min_max 0 25 ^
        -o chm.png

las2dem -i GroundClassified.laz ^
        -keep_class 2 -extra_pass ^
        -step 2 ^
        -hillshade ^
        -o dtm.png
```

For forests on complex and steep terrain the number of ground points that needs to be manually collected may make such an approach infeasible in practice. However, maybe there are other sources of elevation, such as a low-resolution DTM of 10 meter or 25 meter provided by a local government. Or maybe even a high resolution DTM of 1 or 2 meter from a LiDAR survey that was carried out several years ago. While the forest may have grown a lot in the past years, the ground under the forest will probably not have changed much …

3. RESULTS

Clearly the resulting CHM shown on the left in Figure 5 is only meaningful in the target forest where we used the manually collected ground points to create a reasonable DTM. In the other forested areas the ground is only correct near the forest edges and gets worse with increasing distance from open areas. The DTM shown on the right in Figure 5 exhibits some interesting looking bumps in the middle of areas with manually collected ground point. Those are a result of using the dense-matching points as ground whenever their elevation is lower than that of the manually collected points (which is decided in the lasthin step). Whether those bumps represent true elevations of are artifacts of low erroneous elevation from dense-matching remains to be investigated.

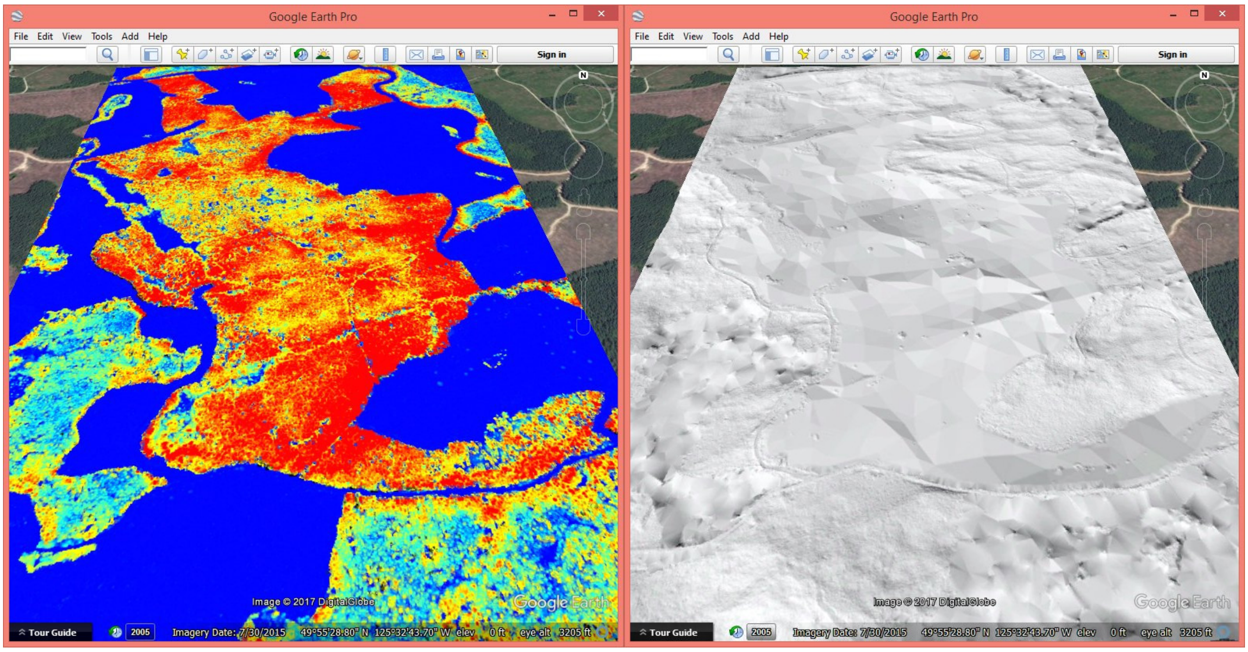


Figure 5: The CHM (left) color-codes vegetation height from blue (0 meter) to red (25 meter or higher). Heights for the target forest where external ground information was used look plausible, however they are wrong for the other surrounding forested aras. The DTM (right) in those areas is not true ground which falsifies the CHM heights.

4. CONCLUSION

Our method has successfully integrated external ground information in form of 558 ground points were manually surveyed with a dense matching point cloud of 35,338,368 points such that a bare-earth terrain model can be created and a Canopy Height Model could be created. In all areas of the input that was covering 3.4 square kilometers where external ground information was available we were able to compute a reasonable Canopy Height Model (CHM) which would then, for example, allow us to measure the height of the trees or use it as an input to a biomass prediction model.

ADDITIONAL MATERIAL:

[1] http://rapidlasso.com/2017/06/13/integrating-external-ground-points-in-forests-to-improve-dtm-from-dense-matching-photogrammetry/