Karttapullautin vegetation mapping guide

Guide version 20131027 / Jarkko R.

To do this you will need area where laser data overlaps with existing map with green mapping or are you have green mapped by yourself or imagery you can use to verify green mapping is about right.

1. Open pullauta.ini and set greenshades parameter to wide range to get enough green for balancing it. Something like:
greenshades=0.4|0.8|1.3|1.9|2.6

2. Make map as usual. open pullautus.png and look at the green mapping. If you have green "stripes", linear parst of the area greener than orhers, you will have to balance it by using pointvolumefactor value in in. change for example from 0.15 to 0.35 and make map again. And change value egain until green looks balanced. It mey be usefull to do it once wti a too low and with once with too high value and then compare whinch is closer and estimate value between them. Usually pointvolumefactor should be somewhere between 0 and 0.5

To make iteration faster you can make vegetation mapping part only if you have made full map once. With command line command "pullauta makevegenew" makes vegetation mappinga and then command "pullauta" renders png maps again with this new vegetation.

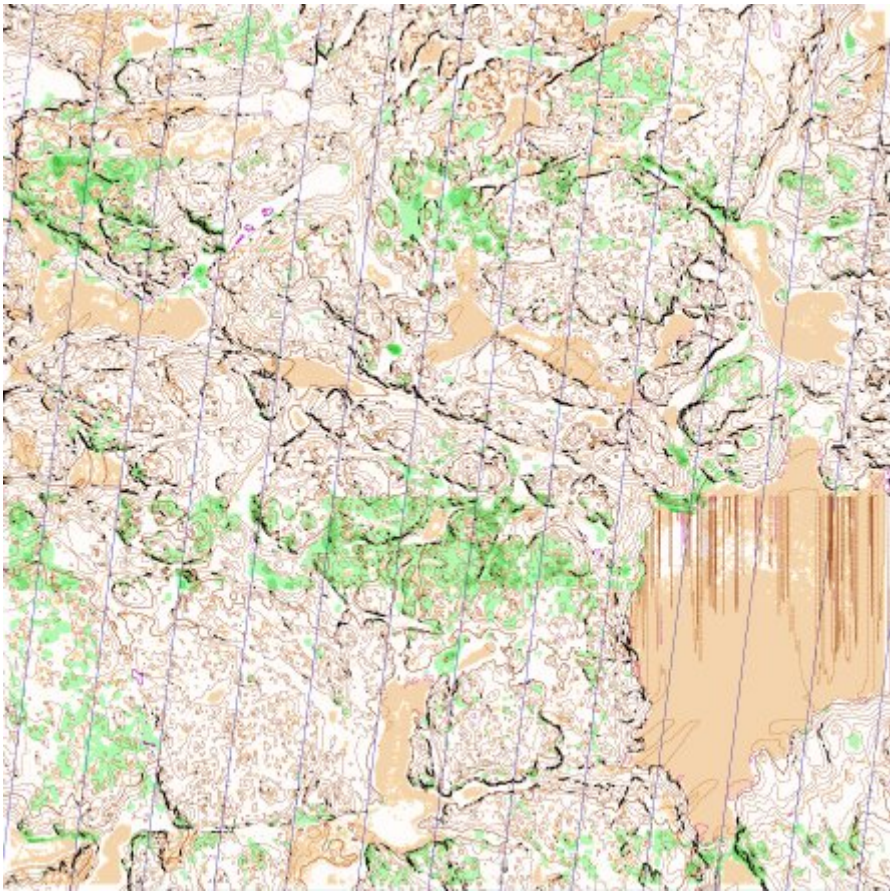

Image 1. Typical green stripes, areas where canning lines overlaps and point density is higher.

3. When balance is right, clip small part of the data (area with various green shades mapped on teal O map) with lastools command
las2las -i in.laz -o out.laz -clip 630250 4834500 630500 4834750
Coordinates you must figure out by yourself. You can also do this with the whole tile, it just takes a bit more time. Then make map again by dragging laz file on pullauta.exe

Objective here is finding good values for greenshades parameter.
greenshades=0.4|0.8|1.3|1.9|2.6
The first values is thresold fors lightest green, last one is darkest (fifth) green.

5. Then compare png map's green mapping against green with the real O map. For example if O maps light green is somewhere between lightes and second lightest green (0.4 and 0.8), set greeshades to
greenshades=0.4|0.5|0.6|0.7|0.8
and make map again. And use new png map version to figure out the right value for light green, like 0.68

6. Repeat the same greenshades iteration to get darker green right.

7. Then use these values as two first greens and set rest to 99 (to get only two greens).
greenshades=0.68|1.13|99|99|99
You can repeat it thord time to get third green if you like.

Basic configureation is done, but you may not be happy with it yet. Low forest may look too green compared to tall forest, or the opposite. Or some parst that should be dark green is white, even if some other part of the areas is mapped just fine.

8. Balancing tall forest and low forest. There is series of values in ini to do it:

thresold1=1|2.5|0.39
thresold2=2.5|6|0.43
thresold3=6|10|0.44
thresold4=10|20|0.41
thresold5=20|99|0.38

First two numbers is forest roof height. As you can see all heigt ranges between 1 and 99 meters are defined. Third one is balancing value. Lower the value is, easier forest with that roof height becomes green. You can start with same factor for them all, and then start tuning it right.

You can have as many rows as you like and set roof zones (thresolds) as you like, just number them 1...n

9. fixing "about 8m tall dense bush gets not mapped as green at all" problem. If bush is dense there may not be much points at all between roof and ground, because the lidar systems usually expects big enough gap between points. If it is really dense there may never be big enough gap, and it often ends up getting just first return at 5..8 meters and ground. And the space between end up looking empty. That is a problem, because we are intresed mostly points about 0.5 .. 5 meters above ground, because hits there indicates there is something and it may come to way when one tries to run there. This is why we usually set pulautin to count points about between 0.8 and 5 meters. Pullautin compares that count against the count of points below "greenground" parameter, for example below 0.8 meters. Bigger ratio of "green hits" means it is more green. To issue this empty

zone problem pullautin has series of parameters to define how green hits are calculated:

zone1=0.8|5|99|1
zone2=5|9|11.0|0.75

Explanation:
- zone1: hits between 0.8 and 5 meters abowe gound are calculated as 1 hit, if forest roof is lower than 99 meters.
- zone2: if point wasn't calculated in yet, then if hit is between 5 and 9 meters above ground and if forest roof is lower than 11 meters then it is counted in, but not as one it but just as a 0.75 hit.
Like this you can gather data from various heights and estimate what it most likely for runner's point of view.

So, you need to know hat you forest is like. If you have mostly 8m tall pine trees with no low branches (~ Barbate pine forest) you should not count hits at 7 meters in at all. But if your 8m tall trees are dense all the way down you should most likely count higher hits in at least partly if roof is below ~ 12 meters.

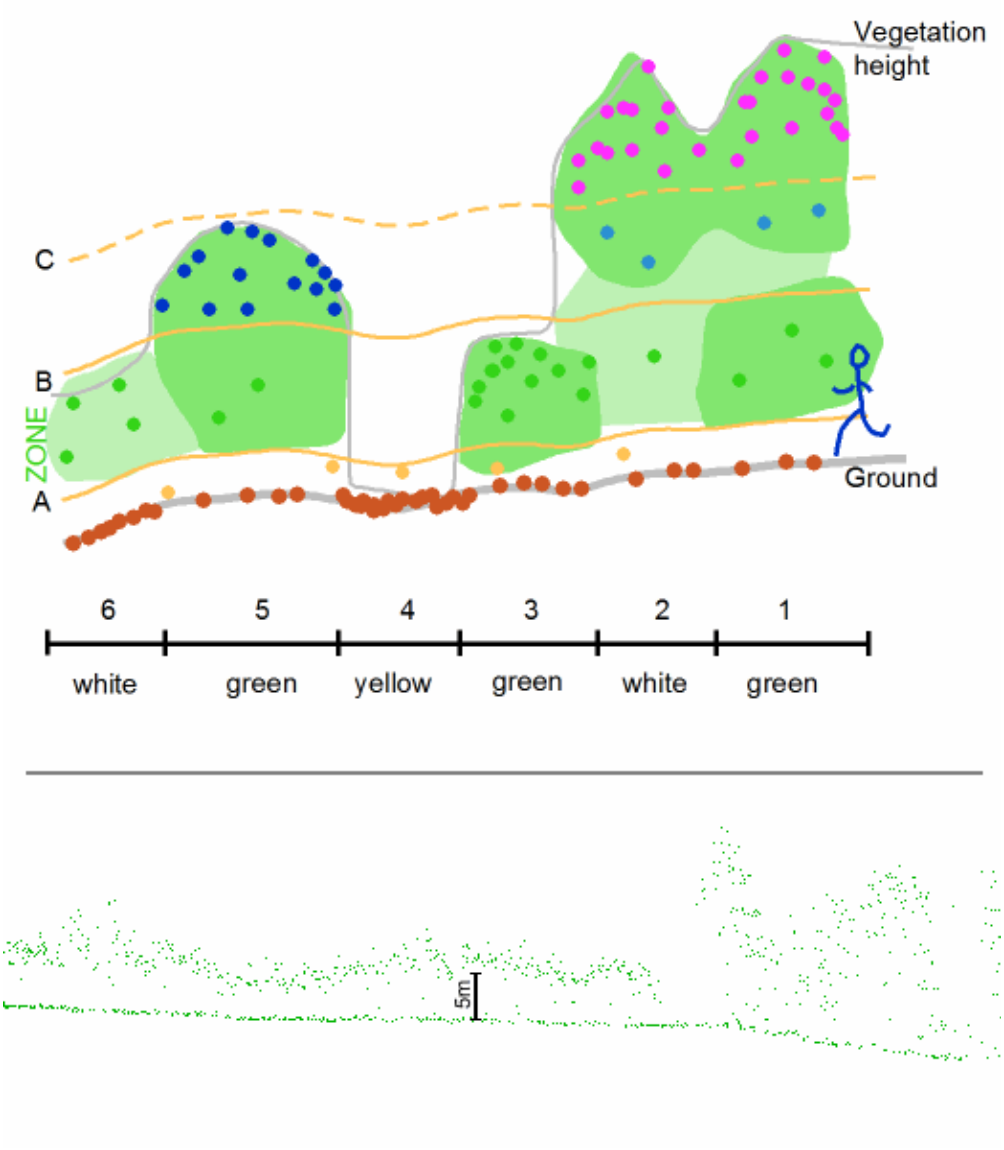

Image 2.

Image 2 is slightly out dated version of the pullautin green mapping, but this still may give some idea how calculation gets done. Brown and yellow dots are calculated as ground and that figure is compared against green hit count. Green dots are counted in as green, dark green dots also partially, liht blue and purple dots will not.

After changing the way points are counted in you will most likely have to iterate and determine greenshades parameter values again.

Other parameters:

- "**greenround**". With this parameter you set the thresold used to divide hits to these two groups (green and below green) pullautin compares and calclulates greeness value, value that gets balanced with "pointvolumefactor", "thresold" values and finally compared against "greenshades" to set the green tone for the area.

- "**blocksize**". area used for calculation is done "blocksize" * 3mx3m. Smaller blocksize gives more scattered (but more accurate location of edges) green, larger gives smoother and more reliable color but location of edges may not be as accurate.

- "**dotsize**" is diameter, how large dot is drawn on map for each "blocksize"

- "**yellowheight**". Lower than this is seen as open/yellow.

- "**yellowthresold**". How big part of hits must be below "yellowheight" to get it mapped with yellow. Usually something like 0.88 (over 88% of points should be below "yellowheight").

Deciduous vs. evergreen problem. Usually scanning is done when there is no leaves. If may make evergreen parst of the forest look more green than the rest of the area. Pullautin has no way to fix this issue, other than making two versions of green mapings and meerging them manually wtih photo editor using aerial imagery to determine what it most likey is like. But usually this is not that much of a problem really.