

Lennokki-LAS-datan hyödyntäminen metsänhoitotöissä

Mikko Kemppinen

Opinnäytetyö
Toukokuu 2016
Tekniikan ja liikenteen ala
Ohjelmistotekniikan insinööri

Tekijä(t) Kempainen Mikko	Julkaisun laji Opinnäytetyö, AMK	Päivämäärä Toukokuu 2016
	Sivumäärä 41	Julkaisun kieli Suomi
		Verkojulkaisulupa myönnetty: x
Työn nimi Lennokki-LAS-datan hyödyntäminen metsänhoitotoissa		
Tutkinto-ohjelma Ohjelmistotekniikka		
Työn ohjaaja(t) Hannu Luostarinen, Raija Hämäläinen		
Toimeksiantaja(t) AMK IT-instituutti / Maaseutu 2.0 projekti		
<p>Tiivistelmä</p> <p>Aiheena on ohjelmoida graafinen käyttöliittymätyökalu metsien inventointiin. Työkalun tarkoitus on hyödyntää LAS-paikkatietotiedostoja analysoiden ja kartoittaen alueet joustavasti metsänhoito- ja inventointitarpeiden mukaisesti. Sovellus pohjautuu kansallisiin metsätietostandardeihin. Kehitettyä työkalua käytettiin Visulahden ja Tuusulan LAS-paikkatietotiedostojen analysointiin, käsittelyyn ja metsän kehitysluokkien kartoitukseen eri testileikkauksille. Sovellus käyttää Python-ohjelmointikieltä. Työkalun kehityksessä keskitytään sen nopeuden ja muistinkäytön parantamiseen, myös analysoitavan datan tuloksien tarkkuuteen ja ymmärrettävyyteen. Tuusulan ja Visulahden LAS-tiedostojen eroavan tietorakenteen sovittaminen ohjelmaan. Keilausdatan visualisoinnissa käytettiin FugroViewer työkalua.</p> <p>Aihe on osa isompaa projektia nimeltä Maaseutu 2.0, mikä tukee metsäpalveluyrittäjyyttä, metsäomaisuuden hallinnointia ja maataloustoimintaa.</p> <p>Opinnäytetyötyön yhteydessä tutustuttiin lennokkien rakennusprosessiin ja kasattiin yksinkertainen lennokki. Toteutuksen onnistuminen vaati osien yhteensopivuuden määrittämisen ja tilaamisen kansainvälisiltä markkinoilta. Kokoaminen ja testaus toteutettiin internetistä löytyneitä ohjeita hyödyntäen.</p> <p>Ohjelman ajettaessa lopputuotoksena syntyy analyysikuva käsitellystä alueesta, mistä voidaan värikoodien avulla tunnistaa kehitysluokat ja kiireellisyys.</p>		
Avainsanat (asiasanat) Laserkeilaus, lennokki, metsien inventointi, metsien kehitysluokat		
Muut tiedot		

Author(s) Kempainen Mikko	Type of publication Bachelor's thesis	Date Month Year Language of publication:
	Number of pages 41	Permission for web publication: x
Title of publication Utilizing UAV LAS Data in Forestry		
Degree programme Software Engineer		
Supervisor(s) Hannu Luostarinen, Raija Hämäläinen		
Assigned by AMK IT-institute / Maaseutu 2.0 project		
Abstract <p>The subject is to program graphic interface tool to invent forests. The meaning of this tool is to exploit LAS-files by analyzing and surveying areas for the needs of forestry and invent. The application is based on forest data standards. Developed application is used to analyze, development class mapping and handling of the location data(LAS) of Visulahti and Tuusula for different test cuts. The program uses Python-programming language. The development of the program focuses on the use of memory, time usage, preciseness and understanding of the analyzed information. Also implementing the handling of different data structures with Tuusula and Visulahti . For visualizing the data we used "FugroViewer".</p> <p>The program is a part of bigger Project called "Maaseutu 2.0", which supports forest service entrepreneur, forest property management and agricultural.</p> <p>The thesis also included getting to know the building process of quadcopter/plane. To achieve a working quadcopter, we acquired a lot of information about the compatibility of the parts and ordering them from the global market. Executive and testing was performed with the information found online from different instructions.</p> <p>When the program is run, the final product is an analysis picture of the processed area with color codes for development class and urgency.</p>		
Keywords/tags (subjects) laserscanning, plane, forest, agriculturals, quadcopter, plane		
Miscellaneous		

Sisältö

1	Lähtökohdat	5
1.1	Työn taustat ja tehtävä kuvaus	5
1.2	Toimeksiantaja	6
2	Metsänmittausmenetelmät	6
2.1	Perinteiset menetelmät.....	6
2.2	Uudet menetelmät	7
3	Laserkeilaus	7
3.1	Yleistä	7
3.2	Laserkeilauksen periaate.....	9
3.3	LAS-tiedosto	9
3.4	LAZ-tiedosto	9
3.5	Avoin LAS-data.....	10
3.6	Laserkeilausdatan käsittely Lastools-työkalulla	10
4	Droonit	11
4.1	UAV	11
4.2	Kauko-ohjatun ilma-aluksen ja lennokin säädökset.....	12
4.3	Laserkeilauslennokki	13
5	Ohjelmointiympäristö	14
5.1	Python-ohjelmointikieli	14
5.2	Käytetyt kirjastot	15
5.3	PyCharm IDE	16
6	Mallinnus.....	17
6.1	Kehitysluokkien määrittäminen	17
6.2	Kyyt- ja aaltoalgoritmi	18
6.3	Puulajin määrittäminen ilmakuvista	18
7	Ohjelman kulku	18
7.1	Ohjelman kulkukaavio	18

	2
7.2 LAZ-tiedoston esikäsittely	19
7.3 Alueen rajausta	20
7.4 Pisteet listoihin	21
7.5 Kehitys- ja hoitotarveluokan määrittäminen	23
7.6 Kehitys- ja hoitotarveluokan tallennus tiedostoon	23
7.7 Piirto tiedostosta	24
7.8 CLM-tiedosto	24
8 Tulosten käsittely Visulahti ja Tuusula	25
8.1 Visulahti	25
8.2 Tuusula	27
9 Oman droonilennokin rakentaminen	29
9.1 Dronin valitseminen ja hankinta	29
9.2 Kokoaminen	30
9.3 Mitta- ja toimilaitteet	31
9.4 Lennonohjausjärjestelmän ohjelmisto	31
9.5 Tärkein toimilaite on moottorit	32
9.6 Lennokin testaus	33
10 Pohdinta	34
10.1 Maapintatason määrittäminen	34
10.2 Puuston kehitysluokan vääristyminen	35
10.3 Jatkosuunnitelma	36
Lähteet	37
Liitteet	39
Liite 1. Tuusula info-tiedosto	39
Liite 2. Visulahti info-tiedosto	40
Liite 3. Siikalatva aaltoalgoritmi esimerkki T2 varttunut taimikko	42

Kuviot

Kuvio 1. Laserkeilausdrooni.....	5
Kuvio 2. Relaskoopin käyttö metsässä	6
Kuvio 3. Relaskoopin toimintaperiaate.....	7
Kuvio 4. Laserkeilauksella tuotettu 3D-malli.....	8
Kuvio 5. Visulahden data Avoiminen aineistojen tiedostopalvelussa.....	10
Kuvio 6. Fugroviewer-ohjelmalla luotu Visulahden 3D-malli.....	11
Kuvio 7. General Atomics MQ-1 Predator.....	12
Kuvio 8. Laserkeilauslennokin toimintaperiaate.....	14
Kuvio 9. Import Pythonilla.....	15
Kuvio 10. Muuttujat C#:ssa ja Python:ssa	15
Kuvio 11. Matplotlib:llä luotu diagrammi	16
Kuvio 12. PyCharm kehitysympäristö.....	17
Kuvio 13. Puiden ominaisspektri	18
Kuvio 14. Ohjelman kulkukaavio.....	19
Kuvio 15. Alueen rajaus.....	20
Kuvio 16. LaserPoint-luokka	21
Kuvio 17. Pisteiden sijoitus listoihin.....	22
Kuvio 18. Luettavan txt-tiedoston rakenne-esimerkki.....	23
Kuvio 19. CLM-tiedostorakenne.....	24
Kuvio 20. Visulahden ilmakehän kuva.....	25
Kuvio 21. Visulahden analyysikuva.	26
Kuvio 22. Visulahdesta leikkauskuva.....	27
Kuvio 23. Tuusula ilmakehän kuva.....	28
Kuvio 24. Tuusula analyysikuva.....	29
Kuvio 25. Itserakennettu drooni.	30
Kuvio 26. OpenPilot CC3D –lennonohjausjärjestelmä	31
Kuvio 27. LibrePilot.....	32
Kuvio 28. Moottorien pyörimissuunnat.....	33
Kuvio 29. Lennokin testaus Kuopion Puijossa.....	34
Kuvio 30. Leikkausten vertailukuviot.	35

Kuvio 31. Visulahti kehitysluokka virhe.....	36
Kuvio 32. Uusi tekniikka skannaa metsästä oksanreiätkin.....	36

Taulukot

Termit ja lyhenteet

Drone Termi viittaa usein sotilaskäytössä oleviin miehittämättömiin ilma-aluksiin. Muun muassa tästä syystä siviilikäytössä olevista miehittämättömistä ilma-alusjärjestelmistä käytetään yleensä mieluummin lyhennettä RPAS.

Lennokki lentämään tarkoitettu laite, jonka mukana ei ole ohjaajaa ja jota käytetään harraste- tai urheilutarkoitukseen.

LAS-paikkatietotiedosto Laserkeilauksessa syntynyt paikkatietotiedosto.

RPA (Remotely Piloted Aircraft): kauko-ohjattu ilma-alus; miehittämätön ilma-alus, jota ohjataan kauko-ohjauspaikasta ja käytetään lentotyöhön.

RPAS (Remotely Piloted Aircraft System): kauko-ohjatun ilma-aluksen käytön kokonaisjärjestelmä, johon sisältyy: kauko-ohjattu ilma-alus, sen kauko-ohjauspaikat, tarvittavat ohjaus- ja seurantayhteydet ja muut erikseen määrätyt käytön edellyttämät järjestelmän osat.

UA (Unmanned Aircraft): miehittämätön ilma-alus; ilma-alus, joka on tarkoitettu lentämään ilman ilma-aluksessa mukana olevaa ohjaajaa; tällä ei tarkoiteta lennokkia.

UAS (Unmanned Aircraft System): miehittämättömän ilma-aluksen käytön kokonaisjärjestelmä; miehittämätön ilma-alus ja sen käytön edellyttämät järjestelmän osat.

UAV (Unmanned Aerial Vehicle): vanhentunut lyhenne/termi, vastaava nykyinen on UA (ks. edellä).

1 Lähtökohdat

1.1 Työn taustat ja tehtäväkuvaus

Työn tarkoitus on helpottaa metsäalueiden hoitotarpeiden kartoittamista ja inventointia UAV-aluksen (ks. kuvio 1.) laserkeilauksessa syntyneitä LAS-paikkatietotiedostoja hyödyntäen. Opinnäytetyö keskittyi aiemmissa opinnäytetöissä lentokonelaserkeilausaineiston käsittelyyn kehitetyn ohjelman muokkaamiseen lennokkimateriaalin kirjoittamiseen paremmin soveltuvaksi. Ohjelmistoa jatkokehittäessä toteutettiin uusia ominaisuuksia ja tutustuttiin laserkeilausdatan käsittelyyn, lukemiseen ja kirjottamiseen. Ohjelmaa kehitettiin myös helpommaksi käyttää, erilaisten datojen analysointiin ja sovelluksella voidaan luoda CLM-dataa, mitä voidaan käyttää muissa LAS-datan analysointi ohjelmissa kuten ArcGIS. Pää tarkoituksena on vähentää metsän mittaukseen vaadittavia resursseja ja nopeuttaa prosessia, mutta kuitenkin tuottaa vertailukelpoista dataa halvalla ja nopeasti. Ohjelman lopputuotoksena on raportti, millä voidaan arvioida metsän hoitotarpeita ja saantoja. Kalevi Pietikäinen on ollut mukana opinnäytetyössä metsäpuolen ja algoritmien asiantuntijana Marjatta ja Eino Kollin säätiön apurahalla.

”Maastossa toteutettava koelaperusteinen metsänmittaus on sekä kallista, että asiantuntemusta vaativaa työtä”. (Uusia menetelmiä metsävaratietojen hyödyntämiseen 2014.)



Kuvio 1. Laserkeilausdrone

1.2 Toimeksiantaja

Opinnäytetyön toimeksiantajana toimi Jyväskylän Ammattikorkeakoulun projektialue. Työ tehtiin osana Maaseutu 2.0 -projektia. Maaseutu 2.0 -projektin tarkoitus on käyttää luonnonvaratietojärjestelmiä yhteistyökonsepteja hyödyntäen ja parantaen alan yritystoiminnan kilpailukykyä ja osaamista. Hanke tukee metsäpalveluyrittäjyyttä, metsäomaisuuden hallinnointia ja maataloustoimintaa työkohteiden kartoitusmenetelmiä verkko- ja mobiilipalveluilla.

2 Metsänmittausmenetelmät

2.1 Perinteiset menetelmät

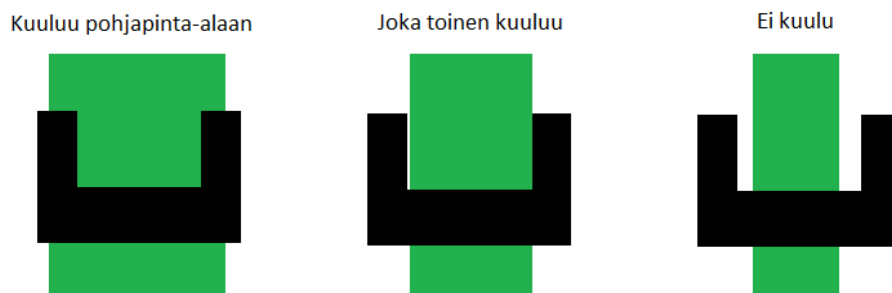
Suomen metsiä on inventoitu jo 1920-luvulta alkaen. Perinteisenä menetelmänä on käytetty relaskoopia metsien silmämääräisen arvioinnin apuvälineenä kuten kuviossa 2 näkyy. Pohjapinta-alan ja keskipituuden avulla voidaan taulukosta saada suoraan puustontilavuus.



Kuvio 2. Relaskoopin käyttö metsässä

Relaskooppi asetetaan rinnan korkeudella ja lasketaan kaikki puut mitkä ovat hahloa isompia 360 asteen ympärillä itsestäsi. Jos puu on hahlon kanssa saman paksuinen,

niin lasketaan joka toinen puu (ks. Kuvio 3.). (Puustontilavuuden määrittäminen relaskoopin ja kepin avulla 2013.)



Kuvio 3. Relaskoopin toimintaperiaate.

2.2 Uudet menetelmät

Uusia menetelmiä metsänmittauksissa ovat esimerkiksi erilaiset puhelinsovellukset, jotka mittaavat kuvasta puiden halkaisijaa, etäisyyttä toisistaan ja niin edelleen.

Melkein sadassa vuodessa metsäinventointi on kehittynyt huimasti ja uusimpana teknologiana käytetään laserkeilausta. Lennokkilaserkeilauksen ja lentokonelaserkeilauksen vahvuus on, että voidaan saada kokonaisvaltaista tietoa isoistakin alueista.

3 Laserkeilaus

3.1 Yleistä

Laserkeilauksen tarkoituksena on luoda lasersäteiden avulla tarkkaa kolmiulotteista tietoa koskematta kohteeseen. Laserkeilausta on käytetty maaston korkeuskartoituksessa, rakennusmittauksissa sekä nyt myös metsien huoltotarpeiden ja saantomittaamisen apuna. Kuviossa 4 havainnollistetaan rakennusmittauksien apuna käytettyä laserkeilausta.

Laserkeilaus täydentää koordinaattimittausmaailman uudella korkean yksityiskohtaisella mittausmenetelmällä, jolla pystytään keräämään ympäröivästä maailmasta tietoa monipuolisemmin ja nopeammin.

(Laserkeilauksen perusteita ja mittauksen suunnittelu 2006.)

Valtakunnallista lentokonelaserkeilausta on jo käytetty Suomessa vuodesta 2008. Pää tarkoituksena on luoda koko Suomesta 15 cm korkeustarkkuudella oleva korkeusmalli. Tätä tehtävää suorittavat lentokoneet laserkeilaten joka paikan Suomen maasta ja mittauksen on suunniteltu olevan valmis vuoteen 2019 mennessä. (Maanmittauslaitoksen uusi valtakunnallinen korkeusmalli laserkeilaamalla 2009)

Erilaisia laserkeilaimia on kolmea eri tyyppiä;

Keilain	Etäisyys	Tarkkuus
Kaukokeilain	100 m – 100 km	<10 cm
Maalaserkeilain	1 – 300 m	<2 cm
Teollisuuslaserkeilain	< 30 m	<1 mm



Kuvio 4. Laserkeilauksella tuotettu 3D-malli

3.2 Laserkeilauksen periaate

Mittakeilain lähettää lasersäteitä tiheinä rastereina ja säteiden tiheyttä säädetään etäisyydestä ja mittauskohteesta riippuen. Lasersäteen törmätessä pintaan keilain mittaa säteen etäisyyden, intensiteetin ja laskee kimmoamispisteen koordinaatit. Kohteet keilataan yleensä monesta suunnasta katvealueiden vähentämiseksi. Laserkeilauksen tulokseksi saadaan pistepilvi. Tästä pistepilvestä pystytään luomaan tietokoneella 3D-malli ja analysoimaan sitä. Pisteitä pistepilvessä voi olla satoja miljoonia tai jopa miljardeja. (Laserkeilauksen perusteita ja mittauksen suunnittelu 2006.)

3.3 LAS-tiedosto

LAS-aineisto koostuu tavuista ja sillä on teollisuusstandardi. Se on julkinen tietotyyppi, joka on tarkoitettu isojen kolmiulotteisten koordinaatistojen tallentamiseen. Standardin käyttämä tavujärjestys on "little endian" (Las specification version 1.4 - R13, 2013, 3).

LAS-tiedosto koostuu eri lohkoista, yleensä näitä lohkoja on 2 – 4. Pakollisia tietoja tiedostorakenteessa ovat julkinen otsikkolohko ja pistetietotallenteet, nämä lohkot sisältävät itse pisteosumat. Lisäksi tiedostossa voi olla muuttuvapituiset tietueet ja laajennetut muuttuvapituiset tietueet. Alkuperäisten lohkojen järjestys ja nimet englanniksi ovat Public header block, variable length records (VLR), Point data records ja Extended variable length records (EVLRL).

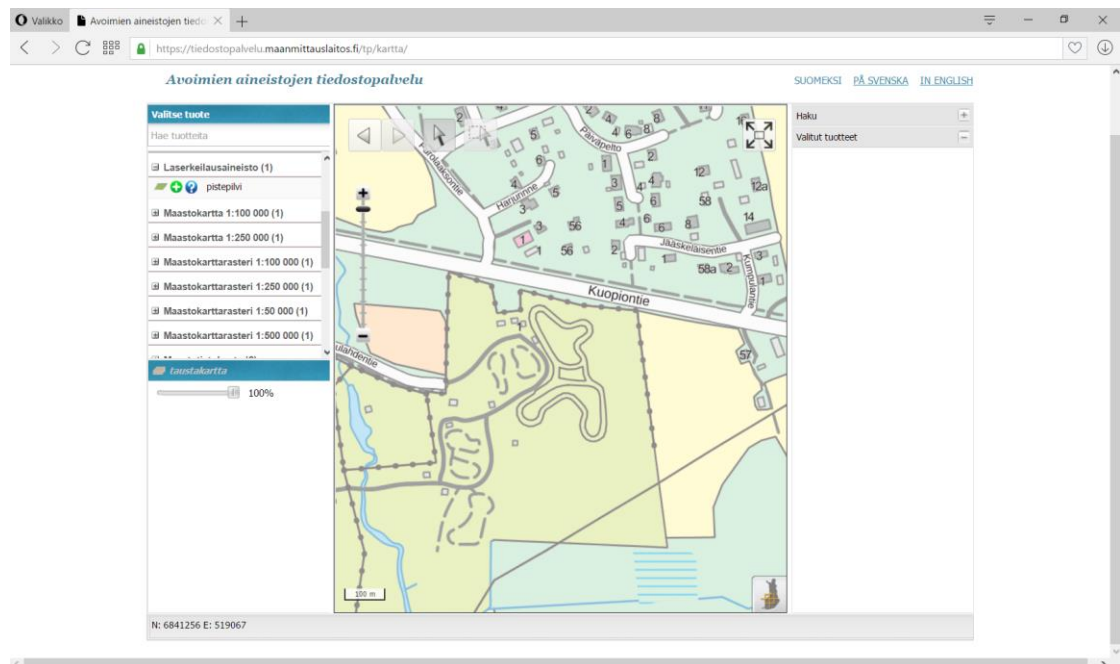
3.4 LAZ-tiedosto

LAZ-data on LAS-datasta kompressoitu avoin tietomuoto. LAZ-dataa voidaan luoda LASzip-nimisellä ohjelmalla LAS-datasta ja se pienentää tiedoston kokoa 5–13 kertaa pienemmäksi alkuperäisestä menettämättä mitään dataa. Siitä on myös tullut de-facto teollisuusstandardi kompressoidulle LiDAR:lle. LASzip-ohjelmaa käytetään todella laajasti maailmalla, sitä käyttää esimerkiksi OpenTopography (USA), Maanmittauslaitos ja NOAA (Pohjois-Amerikka). Monet ohjelmistopaketit tukevat kompressoitua LAZ-standardia, kuten FME, QT Modeler, LAStools, Scanopy, TopoDOT, CloudCompare, RiProcess, OPALS, ja GlobalMapper. LAZ:lle löytyy laajat

dokumentaatiot, että kommentoinnit C ja C#-ohjelmointikielille. (Five Myths about LAS, LAZ, and “Optimized LAS”. 2015)

3.5 Avoin LAS-data

Suomen Maanmittauslaitos tuottaa kaikille ilmaista laserkeilausdataa yhteiskunnan eri tarpeisiin. Jokainen voi käydä etsimässä haluamansa alueen ja ladata tiedoston tietokoneella. Aineistoa voidaan käsitellä monella eri ohjelmalla (ks. luku 3.5). Koko Suomea ei palvelusta vielä löydy, joten alueet on jaettu kiireellisyyden perusteella mahdollisimman nopeasti keilattaviksi. Opinnäytetyössä käsiteltävä Visulahden alue sattui löytymään palvelusta (ks. Kuvio 5). Maailmalta löytyy myös monia muita keilausdatapankkeja kuten OpenTopography ja NOAA.

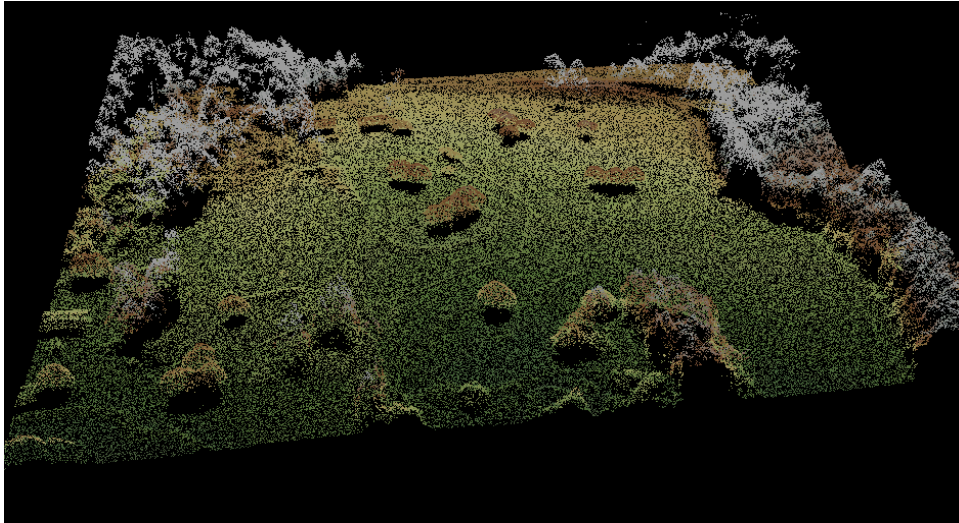


Kuvio 5. Visulahden data Avoimien aineistojen tiedostopalvelussa

3.6 Laserkeilausdatan käsittely Lastools-työkalulla

Laserkeilausdatan käsittelyyn on markkinoilla monta erilaista sovellusta. Yleisillä sovelluksilla yleensä luodaan 3D-malleja. Kuviossa 6 näkyvä kuva on luotu FugroViewer nimisellä ohjelmalla. Kuva on Visulahden datasta tehty.

Laserkeilausdatan käsittelyssä suosittuja sovelluksia ovat esimerkiksi LasTools ja FugroViewer.



Kuvio 6. Fugroviewer-ohjelmalla luotu Visulahden 3D-malli

4 Dronit

4.1 UAV

UAV:lla tarkoitetaan miehittämätöntä lentoalusta, jota voidaan lentää kauko-ohjauksella, valmiiksi ohjelmoidulla ohjelmalla tai monimutkaisempaa dynaamista automatisointia käyttäen. Miehittämättömiä lentoaluksia käytetään moniin eri tehtäviin, kuten tiedustelu, hyökkäys, logistiikka, mainos, tutkimus ja kehitys. UAV-aluksia lähetetään yleensä tehtäville, jotka ovat liian vaarallisia, likaisia tai tylsiä miehitetyle alukselle.

UAV-aluksen rakenne voi muistuttaa helikopteria tai lentokonetta riippuen aluksen tehtävästä ja tarkoituksesta. Lentokoneen mallinen UAV pystyy kantamaan isompia kuormia ja lentämään energiatehokkaammin kuin kopterin mallinen. Kopterimallinen UAV taas toimii paremmin ahtaissa paikoissa ja pystyy lentämään ilmassa paikallaan. Tosin taas akkujen kesto on moninkertaisesti huonompi kuin lentokonemallisilla.

Yhdysvallat käytti UAV-aluksia ensimmäistä kertaa ensimmäisen maailmansodan aikana. Sotilaskäytössä UAV-alukset ovat yleensä täysikokoisia lentokoneita ja niitä ohjataan etäohjauksen avulla tiedustelutarpeisiin(ks. Kuvio 7). Myös 1980-luvulla käytettyjä ristelyohjuksia on 2000-luvun UAV-innostuksesta johtuen kutsuttu UAV:eiksi. (Unmanned aerial vehicle 2016.)



Kuvio 7. General Atomics MQ-1 Predator

4.2 Kauko-ohjatun ilma-aluksen ja lennokin säädökset

Liikenteen turvallisuusvirasto Trafi on antanut 9.10.2015 määräyksen OPS M1-32. Määräyksen tarkoitus on luoda yhteisen pelisäännöt turvalliselle ilmailuharrastamiselle ja ammattitoimijoille. Kauko-ohjatulla ilma-aluksella tarkoitetaan yritystoimintaan perustuvaa ilmailua ja lennokilla harrastetoimintaan.

Tärkeimpiä säädöksiä työtoimintaan:

- Kauko-ohjatun ilma-aluksen käyttämiseen ilmailuun tämän määräyksen mukaisesti ei vaadita lentotyölupaa eikä kauko-ohjatun ilma-aluksen käyttämiseen ilmailuun sovelleta lentotyöstä annettua määräystä OPS M1-23.

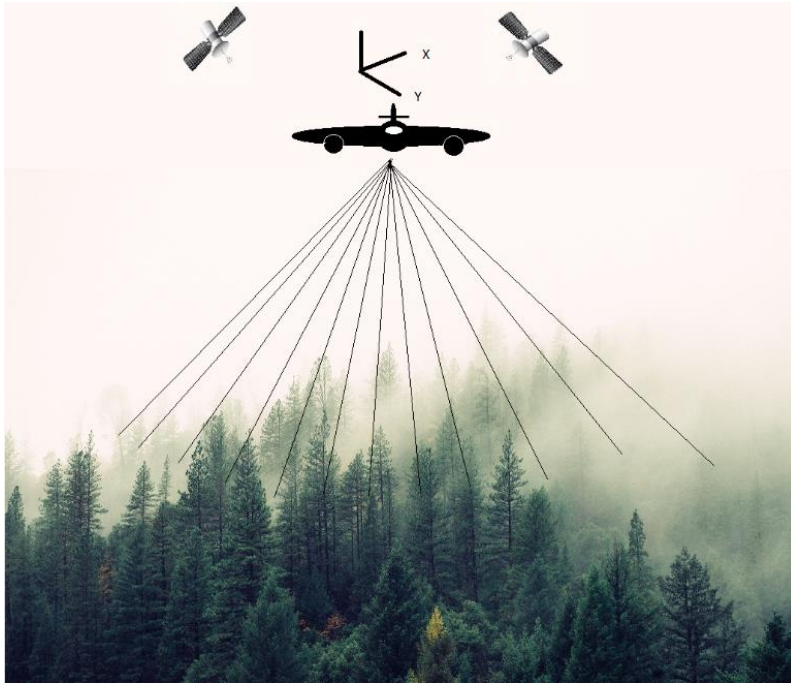
- Kauko-ohjatun ilma-aluksen käyttäjän on ilmoitettava seuraavat tiedot Trafille: tiedot käyttäjästä, tekniset perustiedot ilma-aluksesta, aiotun toiminnan laajuus ja laatu, tieto siitä missä toimintaa aiotaan harjoittaa (tiheä asutus tai väkijoukon yllä).
- Lennätykset on suoritettava siten, että niistä aiheutuu mahdollisimman vähän vaaraa ulkopuolisille ihmisille ja meluhaitat ovat mahdollisimman pienet.
- Lennätyskorkeus ei saa ylittää 150 metriä maan tai veden pinnasta.

Lennokkiharrastetoiminnan säädökset poikkeavat hieman työtoimintaan perustuvista. Lennokin lennättäminen ei ole sallittua väkijoukon yläpuolella ja maksimi paino lennokilla on 3 kg. Lennokista on käytävä ilmi sen omistajan nimi ja yhteystiedot. Lennättäminen lentoaseman lähietäisyydessä saa tapahtua 5 kilometrin päässä ja maksimissaan 50 m korkeudessa. Lennokkien on myös väistettävä kaikkea muuta lentoliikennettä. Lennokkien ja ilmailu-alusten sääntelyyn kuuluu myös yksityisyyden suoja, kotirauha ja tietosuoja. (Ilmailusäädös 2016.)

4.3 Laserkeilauslennokki

Lennokki kerää tietoja sijainnista GPS-avulla(ks. kuvio 8). Laserkeilain lennokissa tallentaa osumat ja sijaintitiedot. Näillä tiedoilla voidaan luoda kolmiulotteinen kuva. Lennokin työalue on yleensä määritelty etukäteen, joten mitattava alue saadaan keilattua mahdollisimman tasaisesti ja katvealueita jää mahdollisimman vähän. Lennokin lentoreitti on siis ennalta määritelty ja itse käyttäjän ei tarvitse itse lentää konetta.

Suomen valtakunnallisessa laserkeilaustyössä käytetään lentokonelaserkeilausta, koska mitattavat pinta-alat ovat todella isoja. Lentokonelaserkeilatassa LAS-datassa on normaalisti noin yksi osuma neliömetrille, kun taas opinnäytetyössä analysoitavassa datassa osumia neliölle tulee noin 200-500. Tämä vaikuttaa ohjelmarakenteeseen ja algoritmeihin todella paljon.



Kuvio 8. Laserkeilauslennokin toimintaperiaate

5 Ohjelmointiympäristö

5.1 Python-ohjelmointikieli

Python on laajasti käytetty korkeantason tulkattu dynaaminen ohjelmointikieli, joka on syntynyt vuonna 1989. Sen suunnittelu painottuu koodin helppoon luettavuuteen ja vähäiseen koodirivien määrään. Uusin versio kirjoitushetkellä Pythonista oli 3.5.1, mutta myös 2.7 kehitetään vielä. Python-kieltä voidaan käyttää monella eri käyttöjärjestelmällä ja se soveltuu hyvin pieniin, sekä suuriin projekteihin. Kolmannen osapuolen ohjelmilla voidaan joustavasti pakata ja suorittaa Python-sovelluksia ilman Python-tulkin asentamista. Python tukee myös useita ohjelmointiparadigmoja, kuten olio-ohjelmointia.

Kielen ydinfilosofiana on ollut viisi eri lausetta:

1. Kaunis on rumaa parempaa
2. Eksplisiittinen on parempi kuin implisiittinen
3. Yksinkertaisuus on monimutkaista parempaa
4. Kompleksi on parempi kuin monimutkainen
5. Luettavuus merkitsee

Pythonin laaja standardikirjasto tarjoaa työkaluja moneen eri työhön. Se on Pythonin yksi suurimmista vahvuuksista. Näitä kirjastoja kutsutaan moduuleiksi. Käyttäjän halutessa käyttää moduulia, sen liittäminen tapahtuu samalla tavalla kuin useimmissa ohjelmointikielissä eli ”import”. Eri moduuleista voidaan myös käyttää vain tiettyä osaa. Kuviossa 9 esitetty matplotlib-moduulin import komento, joka on suosittu kaksiulotteisten kuvien piirtämiseen soveltuva kirjasto.

```
6 import matplotlib
```

Kuvio 9. Import Pythonilla

```
import matplotlib
```

Suurin eroavaisuus Pythonista muihin kieliin on lauseita erottavien puolipisteiden puuttuminen ja muuttujien puuttuvat tyypit. Pythonin dynaaminen tyyppitys määrää ajon aikana tyypit muuttujille niille annettavien arvojen perustella, joten erikseen muuttujan tyyppiä ei tarvitse mainita, niin kuin esimerkiksi C#-ohjelmointikielellä kuten Kuviossa 10. Koodilohkot erotellaan joko välilyönneillä tai sisennyksillä. (Python - Programming language. 2016)

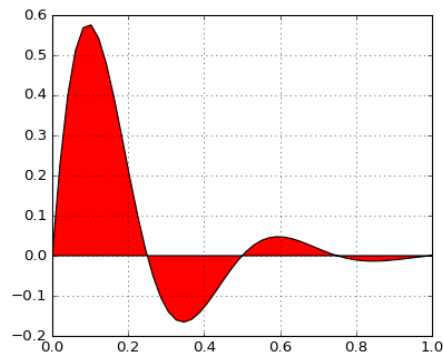
```
1 //C#
2 int x = 1;
3 string y = "abc";
4
5 //Python
6 x_int = 1
7 x_string = "abc"
```

Kuvio 10. Muuttujat C#:ssa ja Pythonissa

5.2 Käytetyt kirjastot

Tärkein käytetty kirjasto projektissa oli matplotlib. Tätä kirjastoa käytetään pisteiden piirtämiseen karttalehdelle. Matplotlib on 2D python kirjasto, jolla voidaan luoda julkaisukelpoista materiaalia. Kirjasto yrittää tehdä vaikeista asioista mahdollisia, sillä voidaan esimerkiksi luoda muutamalla koodirivillä erilaisia diagrammeja tai

parvikuvioita kuten kuviossa 11. Sitä voidaan myös käyttää esimerkiksi Python-skripteissä ja webpalvelimissa. (matplotlib 2016.)

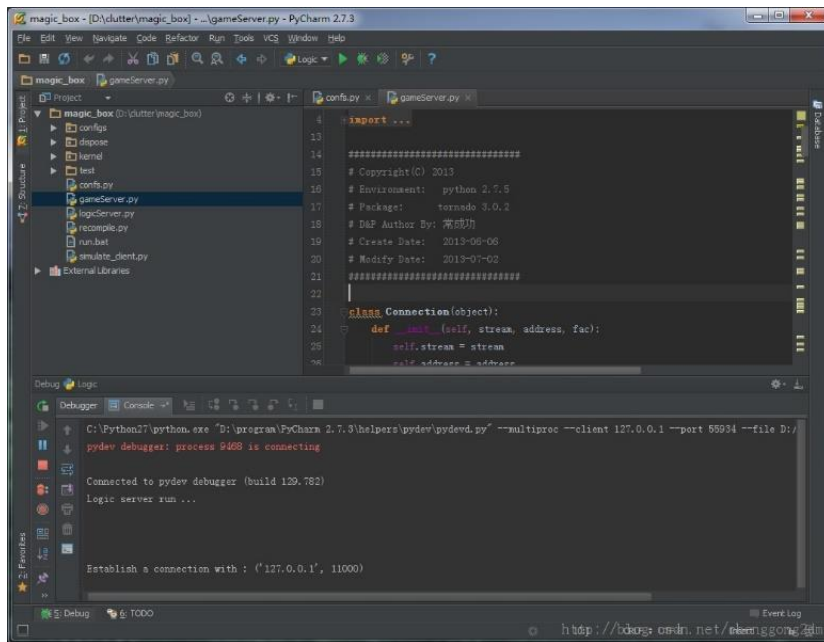


Kuvio 11. Matplotlibillä luotu diagrammi

5.3 PyCharm IDE

PyCharm IDE on integroitu kehitysympäristö (ks. Kuvio 12), jota käytetään Python-ohjelmointikielen koodaamiseen. Se pitää sisällään debuggerin, yksikkötestaajan ja tuen webkehitykseen Djangoilla, web2py:llä ja Flaskilla. PyCharmin omistaa tsekkiläinen yritys nimeltä JetBrains. PyCharmia voidaan käyttää monella eri käyttöjärjestelmällä kuten Windows, Linux ja Mac OS X. Kehitysympäristön betaversion on julkaistu kesällä 2010 ja syksyllä saapui ensimmäinen versio 1.0.

PyCharm on ilmainen vapaan lähdekoodin projekteissa ja opetuksellisissa tarkoituksissa. Lähdekoodi on vapaata lähdekoodia ja koodin voi käydä katsomassa GitHub:ssa. (PyCharm 2016.)



Kuvio 12. PyCharm-kehitysympäristö

6 Mallinnus

6.1 Kehitysluokkien määrittäminen

Kehitysluokalla tarkoitetaan jonkun metsäpalstan metsänhoito- ja hakkuutarvetta kyseisellä mittaushetkellä. Kehitysluokan määrittää metsän puuston korkeus ja tiheys. Tärkeimpiä kehitysluokkia ovat pieni taimikko T1, varttunut taimikko T2 ja nuori metsikkö O2. Alemmassa taulukossa kehitysluokkien keskipituudet puille.

Ohjelmalla tuotetussa kehityskuvassa eri värit viittaavat eri kehitysluokkaan. Värien avulla kuvasta pystytään helposti löytämään alueet, mitkä tarvitset esimerkiksi harvennusta. Kehitysluokan ja kiireellisyyden määrittäminen metsäalueille helpottaa metsänomistajan hoitotöiden budjetointia ja aikatauluttamista.

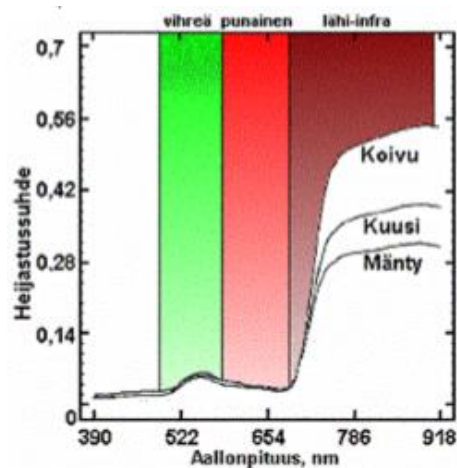
Kehitysluokka	T1 Pieni taimikko	T2 Varttunut taimikko	O2 Nuori metsä
Puiden keskipituus	< 1,3 m	> 1,3m ja < 7m	> 7m

6.2 Kyyt- ja aaltoalgoritmi

Ohjelma käyttää kyyt- ja aaltoalgoritmeja kasvillisuuden kehitysluokkien, että hoitotarpeiden kiireellisyyden määrittämiseen. Ohjelma kutsuu jokaiselle listalle metodia nimeltä laske_ruutu_kyyt ja palauttaa LaserPoint-olion lopuksi. Tämän metodin tehtävä käytännössä on laskea, millä korkeudella sijaitsee suurin osa osumista ja määrittää kuinka paljon listassa on maaosumia. Metodi siis vertailee osumien korkeuseroja. Maaosumien määrä määrittää kiireellisyyden eli mitä vähemmän maaosumia on, niin sitä kiireellisempi alue on.

6.3 Puulajin määrittäminen ilmakuvista

Puulajit voidaan havaita yksinkertaisimmin erottelemalla ne lehti- ja havupuihin. Lehti- ja havupuiden lähi-infra-alueen säteilyssä on suurimmat erot, joten metsänkuvauksessa on hyvä käyttää infrapunavalolle herkkää sensoria (ks. Kuvio 13). Kasvillisuuden ominaissäteilyyn vaikuttaa myös vuodenaika, ilmansaasteet, että kasvin kasvuvaihe. (Opinnäytetyö Hannu Sirén)



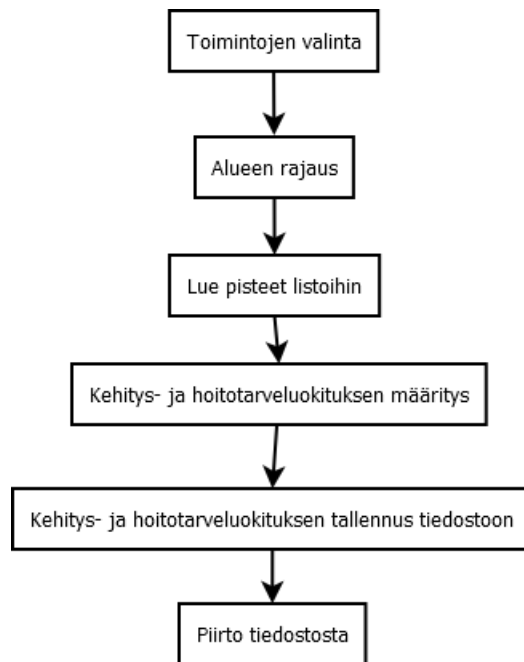
Kuvio 13. Puiden ominaisspektri

7 Ohjelman kulku

7.1 Ohjelman kulkukaavio

Alla olevasta kuviosta 14 on helppo ymmärtää ohjelman peruskulkukaavio. Ohjelman kulku voi myös muuttua, jos halutaan esimerkiksi piirtää vanhasta max-tiedostosta.

Tällöin pisteitä ei tarvitse uudestaan lukea listoihin, vaan voidaan lukea suoraan vanha max-tiedosto ja piirto tapahtuu muutamassa sekunnissa.



Kuvio 14. Ohjelman kulkukaavio.

7.2 LAZ-tiedoston esikäsittely

Ohjelmalla ei käsitellä suoraan LAS- tai LAZ-tiedostomuotoja vaan ne ensin muunnetaan txt-muotoon käyttäen LAS2TXT-sovellusta, minkä on kehittänyt GmbH. Kirjastoja LAZ-käsittelyyn löytyy ainakin C, C# ja JavaScript ohjelmointikielille.

Ohjelmaa kutsutaan komentorivillä, mille annetaan parametrejä, että ohjelma tietää mitä sen pitää tehdä. Komentorivikäsky on selitetty tarkemmin alempana taulukossa.

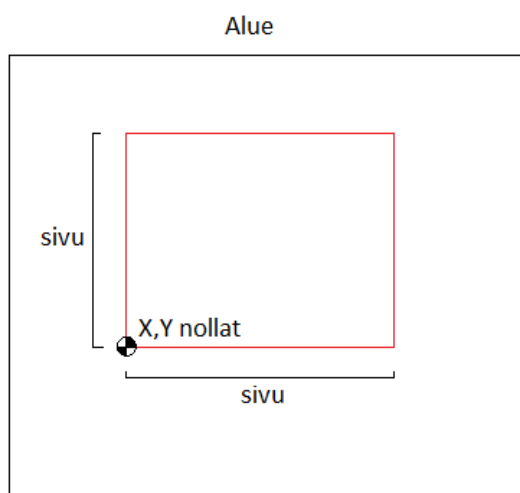
```
C:\LASools\bin\las2txt.exe -i "C:\laz-datat\pointspieni.laz" -parse xyzirndecaup -o "C:\laz-datat\pointspieni.txt"
```

1	Määritetään käytettävän .exe:n sijainti.	C:\LASools\bin\las2txt.exe
2	Input, määrittää mistä tietoa haetaan.	-i

3	Inputille tiedostopolku eli minkä tiedoston ohjelma hakee.	"C:\laz-datat\pointspieni.laz"
4	Määritetään mitä tiedostolle tehdään.	-parse
5	Määritetään mitä tietoja tiedostosta halutaan.	xyzirndecapt
6	Output, minne tallennetaan.	-o
7	Tiedostopolku, minne tallennetaan.	"C:\laz-datat\pointspieni.txt"

7.3 Alueen rajaus

Analysoitava alue rajataan sen koon mukaan ja sille määritellään nollapisteet $x:n$ ja $y:n$ mukaan. Nämä x ja y ovat kaikista pienimmät arvot, mitä alueelta voi löytyä eli vasen alanurkka karttalehdeltä halutulta alueelta. Halutulle alueelle määritetään sivunpituus, joten ohjelma tietää miltä alueelta se kuvan luo. (ks. Kuvio 15)



Kuvio 15. Alueen rajaus

Jos koko alue halutaan kuvata, niin nollapisteiden haku onnistuu myös meta-tiedostosta, jos sellainen sattuu löytymään LAS-tiedoston yhteydestä. (ks. Liitteet 1-2). Tämä metatiedosto voidaan luoda myös LAS2TXT-sovelluksella.

7.4 Pisteet listoihin

Ohjelma luo listoja sen verran kuin analysoitava alue niitä vaatii ja niihin sijoitetaan laserosumapisteitä. Luotujen listojen määrä riippuu analysoitavan alueen kokonaispinta-alasta ja yksittäisen analysoitavan alueen koosta. Sen jälkeen ohjelma käy läpi tekstitiedostoa kuten kuviossa 17. Jokainen piste sijoitetaan LaserPoint-olioon. Olio rakenne esitetään kuviossa 16.

```
class LaserPoint(object):
    def __init__(self, line):
        line = line.split(' ')
        self.x = float(line[0]) # Ruudun X GPS
        self.y = float(line[1]) # Ruudun Y GPS
        self.z = float(line[2]) # Ruudun Z
        self.intensity = int(line[3])
        self.ret_num = int(line[4])
        self.results = int(line[5])
        self.scan_direction = int(line[6])
        self.flight_line_edge = int(line[7])
        self.classification = int(line[8]) # Kehitysluokka
        self.angle = int(line[9])
        self.user_data = int(line[10])
        self.point_src_id = int(line[11])
        self.gps_time = float(line[12])

    def get_txt_row(self):
        miono_lista = map(str, [self.x, self.y, self.z, self.intensity, self.ret_num,
                               self.results, self.scan_direction, self.flight_line_edge,
                               self.classification, self.angle, self.user_data,
                               self.point_src_id, self.gps_time])
        return ' '.join(miono_lista) + '\n'
```

Kuvio 16. LaserPoint-luokka

Kun tiedostoa käydään for-loopilla (ks. Kuvio 17), niin LaserPoint-olio sijoitetaan juuri tiettyyn listaan riippuen kyseisen pisteen sijainnista valitulla alueella. Jos LaserPoint sijaitsee valitun alueen ulkopuolella, ei lisätä ja vaan printataan teksti ”alueen ulkopuolella”.


```

with open("../Conf/TXT/Visulahti.txt") as fin: # Tämä täytyy aina muuttaa

for line in fin:
    p = LaserPoint(line)
    xRivi = int(math.fabs((p.x - xORI)) / RK)
    ySara = int(math.fabs((p.y - yORI)) / RK)

    if xRivi <= RKlkm and ySara <= RKlkm and xRivi >= 0 and ySara >= 0:

        saraR[RKlkm * xRivi + ySara].append(p)
        if lask % 10000 == 0:
            # if lask <= 100:
                print("lask, p.x, p.y, RKlkm, xRivi, ySara ", lask, p.x, p.y, RKlkm, xRivi, ySara)
            lask += 1
        else:
            print('alueen ulkopuolella')
print('len(saraR), RKlkm, xRivi, ySara ', len(saraR), RKlkm, xRivi, ySara)

```

Kuvio 17. Pisteiden sijoitus listoihin.

Kuviossa 18 näkyy listoihin lisättävien LaserPoint-pisteiden tietuerakenne. Listoja syntyisi 320 m x 320 m alueelle jo noin 100 000 kpl 1 m x 1 m listoina ja 8 m x 8 m listoina 1600 kappaletta. Ruutujako ja osumien määrä listoissa vaikuttavat laskentanopeuteen ja muistin käyttöön todella paljon. Ohjelmassa voidaan myös valita analysoitavan ruudun koko ja tämä valinta yleensä perustuu yleensä osumatiheyteen. Maanmittauslaitoksen sivuilta saatavassa LAS-datassa on yleensä noin 0,8 osumaa per neliömetri, kun taas datassa mitä opinnäytetyössä analysoitiin, osumia oli noin 200 per neliömetri. Jos osumia on esimerkiksi yksi per neliömetri niin 1x1 m listoihin tulisi vain yksi osuma per lista, näin tuloksista tulisi todella epävarmoja. Näissä tapauksissa valitaan isompi alue analysoitavaksi, joten saadaan kokonaisvaltaisempaa tietoa alueesta. Osumatiheyserot materiaaleissa johtuvat korkeudesta ja laitteesta, millä ne on kuvattu. Opinnäytetyössä käytetty materiaali on lennokkikeilattu ja Maanmittauslaitoksen materiaali on lentokonekeilattu; tästä syntyvät isot osumistiheyserot.

x	y	z	i	c
518764.99	6843806.54	0.05	26728	1 1 1 0 0 0 0 1 0.000000
518764.49	6843806.01	0.00	28527	1 1 1 0 0 0 0 1 0.000000
518768.54	6843807.32	0.05	30326	1 1 1 0 0 0 0 1 0.000000
518765.82	6843807.56	0.09	30326	1 1 1 0 0 0 0 1 0.000000
518768.52	6843805.85	-0.02	29555	1 1 1 0 0 0 0 1 0.000000
518768.59	6843809.13	0.22	30069	1 1 1 0 0 0 0 1 0.000000
518765.53	6843808.06	0.20	44718	1 1 1 0 0 0 0 1 0.000000
518764.24	6843806.34	0.06	31097	1 1 1 0 0 0 0 1 0.000000
518766.96	6843805.48	-0.02	32382	1 1 1 0 0 0 0 1 0.000000
518766.60	6843809.80	0.67	31868	1 1 1 0 0 0 0 1 0.000000
518766.99	6843806.80	0.06	32382	1 1 1 0 0 0 0 1 0.000000
518768.50	6843806.15	0.00	32382	1 1 1 0 0 0 0 1 0.000000
518767.33	6843805.59	-0.03	28784	1 1 1 0 0 0 0 1 0.000000
518767.77	6843808.77	0.22	33153	1 1 1 0 0 0 0 1 0.000000
518766.18	6843809.85	0.70	37008	1 1 1 0 0 0 0 1 0.000000
518768.36	6843807.98	0.09	41634	1 1 1 0 0 0 0 1 0.000000
518767.77	6843807.15	0.06	31354	1 1 1 0 0 0 0 1 0.000000
518768.60	6843807.19	0.03	26985	1 1 1 0 0 0 0 1 0.000000
518764.95	6843809.78	0.65	38807	1 1 1 0 0 0 0 1 0.000000
518766.73	6843809.79	0.65	37265	1 1 1 0 0 0 0 1 0.000000

Kuvio 18. Luettavan txt-tiedoston rakenne-esimerkki.

7.5 Kehitys- ja hoitotarveluokan määrittäminen

Ruudukolle eli jokaiselle listalle määritetään kehitysluokka ja hoitotöiden kiireellisyys käyttäen Kyyt- ja Aallot-algoritmia. Algoritmit laskevat osuinen tiheyttä ja osumakorkeuden painottavuutta. Ruudun laskettu lopputulos edustaa yhtä LaserPoint-tietuetta, mikä sisältää ruudun koordinaatit, korkeuden ja kehitysluokan. Katso esimerkki liitteestä 4.

7.6 Kehitys- ja hoitotarveluokan tallennus tiedostoon

Lopulta LaserPoint olion tiedot tallennetaan max-tiedostoon käyttäen get_txt_row metodia LaserPoint luokasta. Tässä jalostetun tiedoston koko tipahtaa alkuperäisestä noin 1/200 osaan, mikä esimerkiksi nopeuttaa piirtämistä huomattavasti. Alla esimerkki tiedoston rakenteesta.

```
0.0 260.0 0.0 0 0 0 0 2 0 0 0 -5.08
1.0 260.0 11.0 0 0 0 0 25 0 117 0 -5.16
2.0 260.0 16.24 0 0 0 0 30 0 179 304 -5.79
3.0 260.0 17.07 0 0 0 0 30 0 0 457 -6.0
4.0 260.0 18.93 0 0 0 0 30 0 0 667 -5.47
5.0 260.0 18.0 0 0 0 0 30 0 22 210 -5.54
6.0 260.0 16.25 0 0 0 0 30 59 330 389 -5.73
7.0 260.0 12.57 0 0 0 0 25 180 170 64 -6.16
```

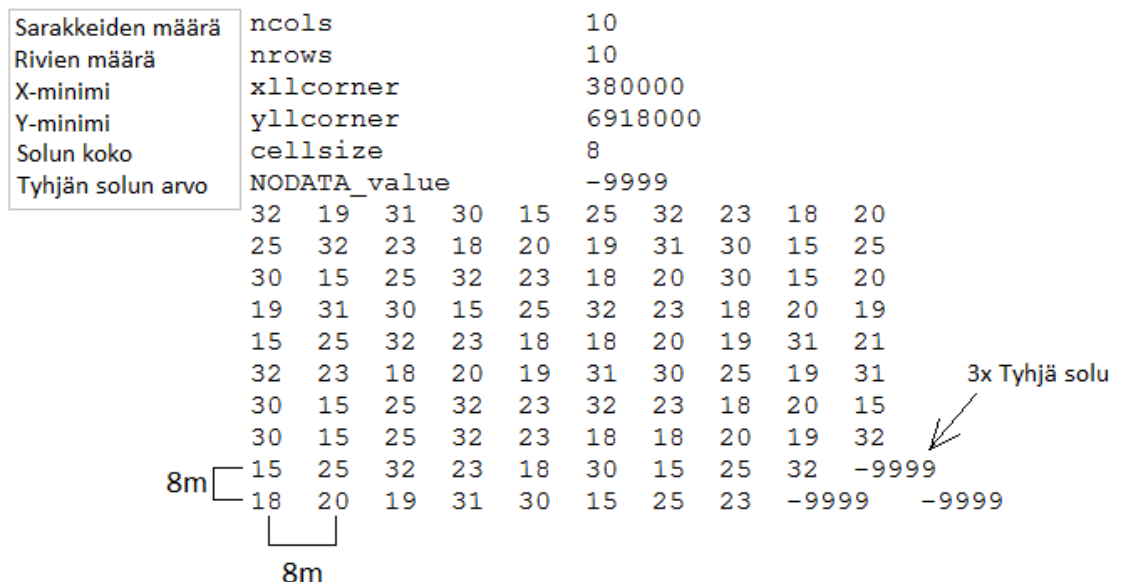
7.7 Piirto tiedostosta

Kun max-tiedoston kirjoitus ja tallennus ovat valmiit, ohjelma aloittaa piirtämisen. Se käy max-tiedoston pisteet läpi listaan ja piirtää pisteet listasta matplotlib-kirjastoa hyväksi käyttäen.

7.8 CLM-tiedosto

Ohjelmalla voidaan myös luoda CLM-tiedosto max-tiedostosta. CLM-tiedostomuotoa voidaan käyttää esimerkiksi Qgis ja ArcGIS-ohjelmissa. Näin voimme tuottaa helposti luettavaa dataa myös ihmisille, jotka eivät käytä kehittämäämme ohjelmaa.

CLM-tiedoston tiedostorakenne perustuu tasaisiin välimatkoihin lukujen välissä ja jokainen luku kertoo sen sijainnin kehitysluokan. Se on kuin karttalehti, missä jokaisen luvun välillä on tietty välimatka. Tämä välimatka on ilmoitettu tiedoston alussa meta-tiedoissa (*cellsize*), joten ohjelma pystyy piirtämään siitä karttalehden. Meta-tiedoissa ilmoitetaan rivien ja sarakkeiden määrä, että x ja y minimiarvot. Kuviossa 19 yritetään selittää yksinkertaisesti, miten CLM-tiedostorakenne toimii.



Kuvio 19. CLM-tiedostorakenne.

8 Tulosten käsittely Visulahti ja Tuusula

8.1 Visulahti

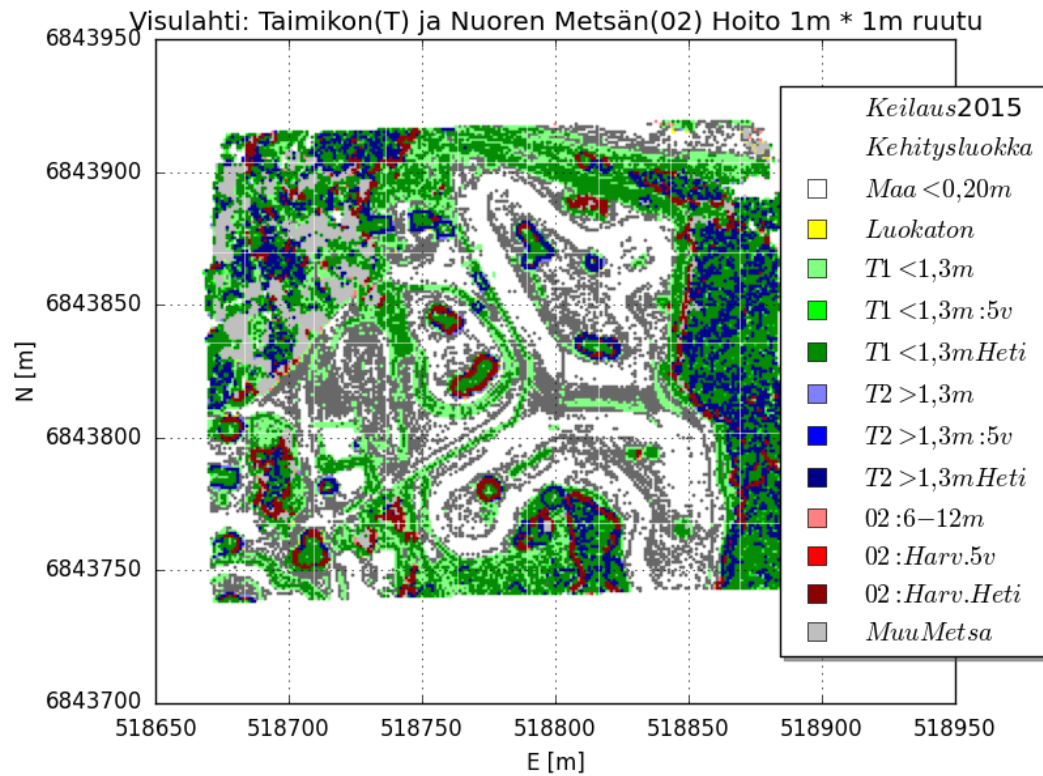
Yksi käsiteltävä alue oli Mikkelin Visulahden karting-rata. Alue ei ole paras mahdollinen metsien inventoinnin arviointiin, koska alueesta vain murto-osa on metsäaluetta. Ilmakuvan avulla voidaan havainnoida tuloksia eri näkökulmasta ja nähdään, mitä alue oikeasti pitää sisällään (ks. Kuvio 20). Kuitenkin metsiä pystyttiin havainnoimaan keilatun alueen vasemmasta ja oikeasta reunasta. Alue soveltui myös hyvin maapisteiden tunnistamiseen ja virhearvioiden korjausanalysoimiseen.



Kuvio 20. Visulahden ilmakuva

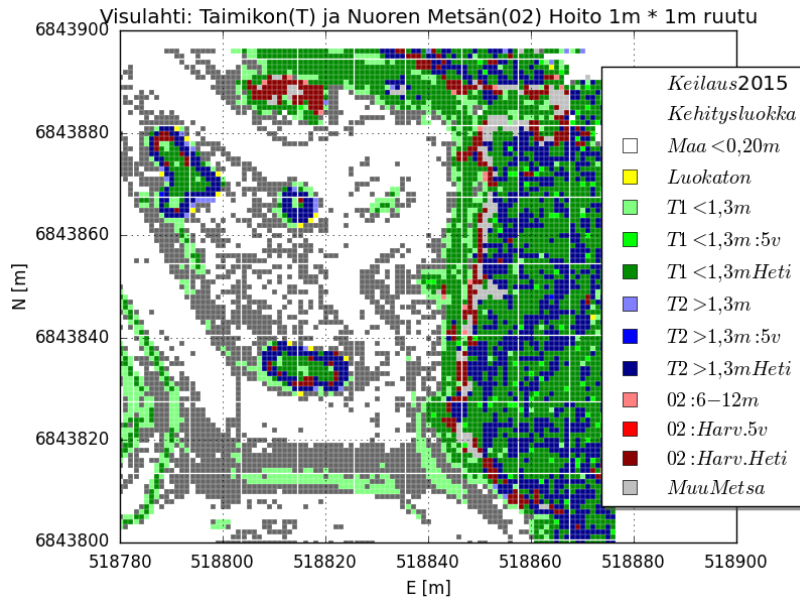
Kuviossa 21 nähdään Visulahdesta luotu analyysikuva. Kun verrataan tätä analyysikuvaa ilmakuvaan, voidaan havainnoida, missä puita sijaitsee ja missä on vain maata. Verratessa kuvia huomataan heti, että ohjelma on merkinnyt T1 eli pieneksi taimikoksi suuria osia alueista, missä puita ei sijaitse. Tämä johtuu maan kaltevuudesta ja täten ohjelma on asettanut maapisteen hieman väärälle korkeudelle. Kuitenkin suurin osa radan muodoista on merkitty maaksi ja ohjelma on onnistunut määrittelemään alueita myös oikein. Puustoalueet vasemmassa

yläkulmassa ja oikealla on kuitenkin hyvin luokiteltu T2 tai O2 alueiksi ja niissä mitään virhettä ei tapahdu.



Kuvio 21. Visulahden analyysikuva.

Ohjelmalla pystytään myös analysoimaan pienempiä leikkauksia alueista. Kuviossa 22 nähdään Visulahden alueesta analysoitu leikkauksikuva. Leikkauksesta voi helposti havaita virheen, mitä on käsitelty enemmän luvussa 10.2.



Kuvio 22. Visulahdesta leikkauskuva.

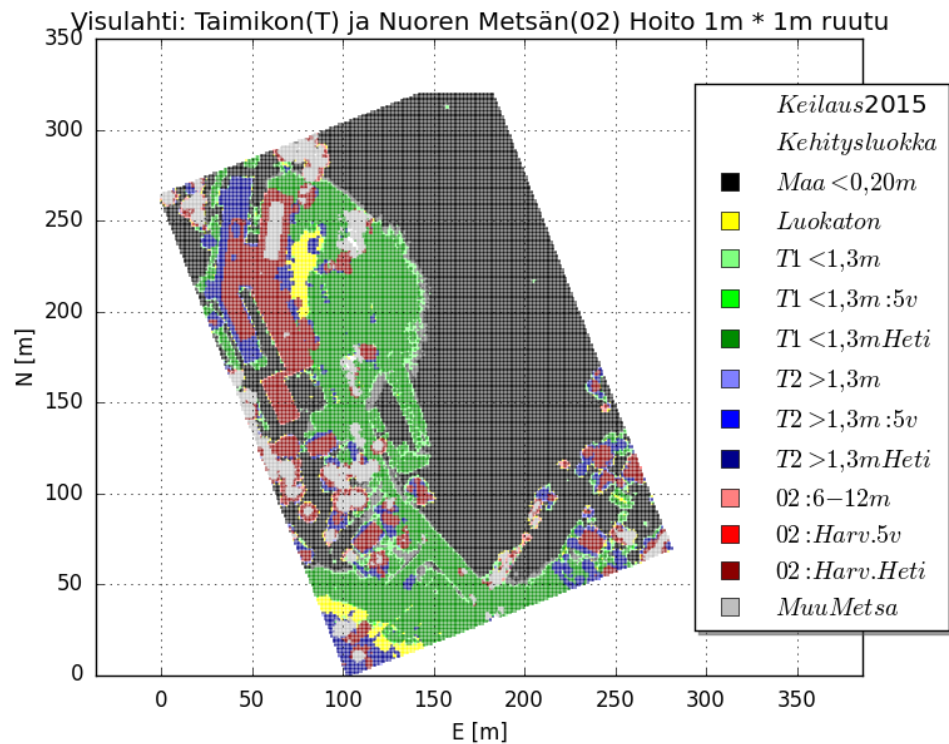
8.2 Tuusula

Tuusulan analyysikuvasta ja ilmakuvaista voidaan huomata suoraan, että rakennuksien luokituksia ei ohjelmalla tehdä (ks. Kuvio 23 - 24). Täten ohjelma luulee rakennuksia myös metsäksi ja merkkää niille kehitysluokat. Tätä virhettä yritetään korjata harjoittelun aikana ja se voisi olla mahdollista korjata tutkimalla intensiteettiä tai rpg-arvoja ja sitä kautta määrittää, että kyseessä ei ole elävää materiaalia. Tässä analyysikuvassa maa on merkitty mustaksi, mikä helpottaa maa alueiden tunnistusta.

Näissäkin tuloksissa ohjelma on merkinnyt paljon maa-alueita T1 kehitysluokaksi. Se johtuu taas maan kaltevuuden vaihteluista, joka sekoittaa ohjelmaa.



Kuvio 23. Tuusula ilmakekuva



Kuvio 24. Tuusula analyysikuva.

9 Oman droonilennokin rakentaminen

9.1 Droonin valitseminen ja hankinta

Rakennettavan lennokin hankintaprosessi oli aikaa vievä prosessi, koska valittavia osia on todella paljon. Lennokin hankintapäätöstä tehdessä oli hyvä miettiä sen käyttötarkoituksia kuten esimerkiksi valokuvaaminen, laserskannaus tai kisaaminen FPV:n avulla. Kaupallisia valmiita lennokkipaketteja on paljon markkinoilla, monet niistä ovatkin todella laadukkaita ja soveltuvia sekä aloittelijoille että ammattilaisille.

Päätin kasata oman droonin eri valmistajien osista, että pääsin kurkistamaan, mitä oikeasti tapahtuu lennokin sisällä ja mikä voi mennä vikaan. Osien valitseminen vaikuttaa lentoaikaan, nosteseen ja kantavuuteen, mutta myös moneen muuhun asiaan.

Tilasin valitsemani osat Aliexpressistä (www.Aliexpress.com). Tämä jälleenmyyjä toimii Hong Kongissa ja osoittanut itsensä luotettavaksi toimittajaksi. Toimitusajat vaihtelevat yleensä kahdesta viikosta kuuteen viikkoon.

Lennokkiin kuuluvia tärkeitä ja välttämättömiä osia ovat moottorit, lennonohjausjärjestelmä, runko, akku, propellit ja nopeudensäätimet.

9.2 Kokoaminen

Ennen kuin osat olivat edes saapuneet postiin, olin YouTubesta opiskellut lennokin kasausvideoita. Näistä videoista oli todella paljon apua kokoamisprosessin edetessä ja sain arvokasta tietoa kokoamisprosessin eri työvaiheista. Työvaiheiden oikea järjestys oli onnistuneen lennokin kasaamisen salaisuus. Tärkein syy oikealla työjärjestykselle oli se, että näin pystyin sijoittamaan jokaisen virranjakajan ympärille kutistesukan, joka suojaa hyvin ja pitää paikallaan niitä (ks. Kuvio 25). Virranjakajia kopterissa oli neljä eli jokaiselle moottorille oma. Jokaisessa kopterin ”jalassa” on yksi virranjakaja kutistesukan alla suojassa.



Kuvio 25. Itserakennettu drooni.

9.3 Mitta- ja toimilaitteet

Lennonohjausjärjestelmä on lennokin aivot ja tässä lennokissa käytin OpenPilot CC3D-lennonohjausjärjestelmää (ks. Kuvio 26). CC3D tukee myös GPS-paikannusta ja sille on mahdollista määrittää esimerkiksi lentoreittejä tai kotiinpaluu. Tosin CC3D ei sisällä GPS-moduulia vaan se pitää erikseen ostaa ja liittää.



Kuvio 26. OpenPilot CC3D –lennonohjausjärjestelmä

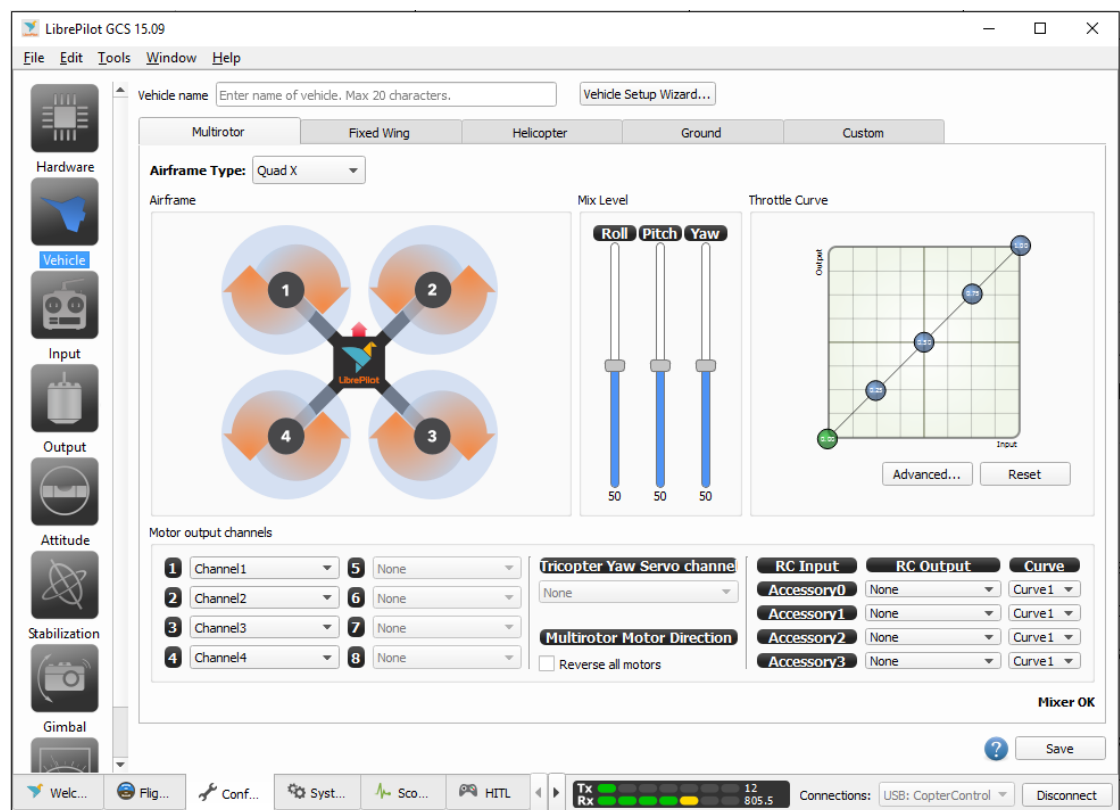
Mittalaitteina lennonohjausjärjestelmässä voivat toimia kiihtyvyy-, ilmanpaine-, magnetometri- ja nopeusanturi, GPS-paikannin ja gyroskooppi, riippuen lennokin käyttötarkoituksesta. Lennonohjausjärjestelmä voi sisältää myös autopilotin ja muita lentoa avustavia ohjelmistoja. Gyron tehtävänä on pitää silmällä lennokin asentoa ilmassa ja tarvittaessa suoristaa lennokki, jos esimerkiksi kova tuuli puhaltaa sitä pois reitiltä. Magnetometri toimii taas kompassina ja sitä voidaan käyttää esimerkiksi, jos GPS signaalia häviää satunnaisesti. GPS:n avulla ohjataan yleensä lennokkia tiettyä reittiä tai sillä voidaan pitää lennokki ilmassa tietyssä halutussa pisteessä.

9.4 Lennonohjausjärjestelmän ohjelmisto

OpenPilot CC3D:tä käytetään avoimen lähdekoodin ohjelmistolla nimeltä *LibrePilot* (ks. kuvio 27), koska OpenPilot on konkurssiin mennyt yritys ja näin ollen heidän sovelluksellaan ei ole enää ylläpitoa. LibrePilot on OpenPilot-ohjelmiston kopio ja sen käyttöliittymä on täysin samanlainen kuin OpenPilot:lla oli. LibrePilotin perustajajäsenet ovat olleet pitkään mukana OpenPilot:n kehittämisessä. Ohjelmisto on todella helppo käyttää aloittelijallekin, lennokin määrittäminen onnistuu helposti

”Wizardin” avulla, joka opastaa käyttäjän prosessin läpi hyvin yksinkertaisesti sanasta sanaan. (LibrePilot. 2016.)

Kaikkiin LibrePilotin asetuksiin ja toimintoihin löytyy dokumentaatio internetistä helposti ymmärrettävässä muodossa. Ohjelmistolla määritetään lennokille sen tyyppi, moottoreille starttivirrat ja monia muita asioita. Oletusasetukset kuitenkin toimivat mainiosti, joten aloittelijan ei tarvitse heti osata määrittää kaikkia asetuksia itse ja ohjelmisto ei tunnu monimutkaiselta. Laajoilla asetuksilla jokaisen on helppo löytää itselleen oikeat asetukset ja parantaa lentokokemusta.

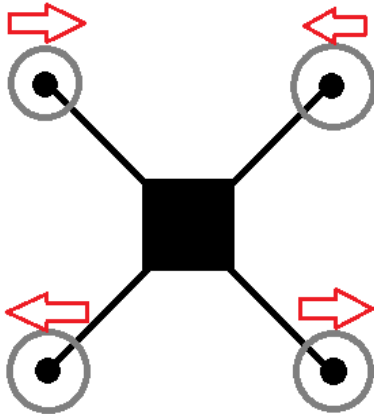


Kuvio 27. LibrePilot.

9.5 Tärkein toimilaite on moottorit

Lennonohjausjärjestelmää voi ohjata moottoreita mittalaitteiden kuten GPS:stä saatavan tiedon perusteella tai ohjaimesta lähtevien radiosignaalien avulla riippuen käyttötarkoituksesta (ks. Kuvio 28). Moottorien pyörimissuunnat ovat aina vastakkaiset seuraavan moottorin mentäessä myötä päivään kiertäessä. Näin kopteri

ei lähde pyörimään itsensä ympäri vaan vastakkaiset voimat kumoavat toisensa ja kopteri nousee ylöspäin.



Kuvio 28. Moottorien pyörimissuunnat.

9.6 Lennokin testaus

Lennokin testaaminen sujui moitteettoman hyvin (ks. Kuvio 29). Mikään ei ollut väärin asetettu tai muutenkaan epäkunnossa. Lennokin hallitseminen ilmassa tosin oli alussa todella vaikeaa, mutta jo muutamien lentotuntien jälkeen itsevarmuus kasvoi nopeasti ja lentäminen helpottui. Lennokin oma gyro-anturi ja automaattinen tasapainotus lennonohjausjärjestelmässä auttoi lentämistä hyvin paljon. GPS-paikannin olisi auttanut todella paljon etenkin tuulessa lentämistä.

Lentovirheiden sattuessa ja maahan törmätessä runko kesti uskomattoman hyvin iskuja ja mitkään osat eivät menneet rikki, propelleja lukuun ottamatta.

Hiilikuiturungon ansiosta lennokka kesti hyvin kovempiakin iskuja ja tärkeimmät osat olivat hyvin suojassa rungon sisällä. LiPo-akun sijoittaminen oli mahdollista vain kopterin päälle sen suuren koon takia. Tämän takia kopteri on hieman räjähdysherkkä, jos törmäys tapahtuu ja osuminen tapahtuu suoraan akkuun. LiPo-akut ovat tunnetusti herkkiä erilaisille kolhuille ja hankaumille.


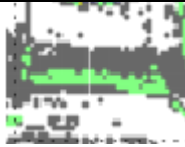




Kuvio 29. Lennokin testaus Kuopion Puijossa.

10 Pohdinta

10.1 Maapintatason määrittäminen

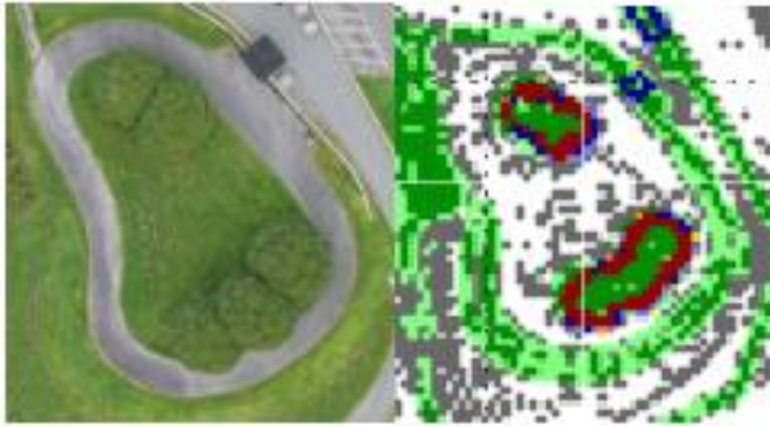
Maapintatason määrittämiseen liittyvät virheet ovat vaikeita ratkaista. Esimerkiksi kun maa on kaltevaa, on vaikeaa määrittää, miten paljon maa voi nousta analysoitavalla koe-alueella ennen kuin se luokitellaan kasvustoksi. Tämä voi helposti johtaa kasvuston korkeuden virhearvioon ja tätä kautta kehitysluokan vääristymiseen. Tämä virhe voidaan havaita esimerkiksi Visulahden tai Tuusulan materiaaleista. Kuviossa 30 alempana on molemmista materiaaleista leikkausesimerkit pieniltä alueilta ja nähdään kuinka alueella ei ole puita, mutta silti ohjelma merkkää alueelle kehitysluokan, koska maanpinta on sen verran kalteva. Ohjelma siis löytää yli 20cm eroavia korkeusarvoja ja antaa alueelle tällöin kehitysluokan T01 eli pieni taimikko.

	Ilmakuva	Analyysikuva
Visulahti Tässä maataso maalattu valkoisella		
Tuusula Tässä maataso maalattu mustalla		

Kuvio 30. Leikkausten vertailukuvat.

10.2 Puuston kehitysluokan vääristyminen

Kun ohjelma analysoi alueita ja kohdalle sattuu tulemaan puu tai puita niin on mahdollisuus, että puiden kohdalla ei löydy yhtään "todellista" maapistettä eli maapistettä mikä osuisi todella maahan. Tällaisessa tapauksessa kehitysluokka vääristyy ja tippuu alaspäin. Tämän pystyy helposti havainnoimaan Visulahden materiaalista (ks. Kuvio 31). Puiden ympärillä nähdään kehitysluokkaa O2 eli nuorta metsää. Puiden keskellä taas kehitysluokkaa onkin T1 eli pientä taimikkoa. Tämä on virhe ja se johtuu siitä, että "todellisia" maaosumia ei puiden keskellä ole ja täten maapiste on korkeampi kuin sen pitäisi olla. Olisi ehkä mahdollista ennalta analysoida koko kartta-alueen maapisteet esimerkiksi 8 x 8 metrin alueissa ja tämän jälkeen antaa jokaiselle 1 x 1 m alueelle oma maapiste. Jos jokin 1 x 1 m alueen maapisteistä olisi todella paljon isompi kuin kaikista matalin maapiste koko 8 x 8 m alueella niin tällöin, kyseiselle 1 x 1 m maapisteelle annettaisiin koko 8 x 8 m alueen pienin maapiste. Tämän tyyllisellä ennalta analysoinnilla voisi olla ehkä mahdollista korjata tämä virhe.



Kuvio 31. Visulahden aineistoanalyysissä havaittu kehitysluokkavirhe.

10.3 Jatkosuunnitelma

Kesällä suoritettavan harjoittelun aikana on luvassa uutta lennokkikeilattua materiaalia Konnevedeltä. Tämä todennäköisesti tehdään lennokkikeilaimella, jolla saadaan parempaa ja tarkempaa dataa kuin aikaisemmin (ks. Kuvio 32). Alueella on kehitysluokkia T1, T2 ja O2 sekä myrskutuhoalue. Tämän materiaalin pohjalta on hyvä lähteä kehittämään ohjelmistoa eteenpäin.



Kuvio 32. Uusi tekniikka skannaa metsästä oksanreiätkin.

Lähteet

General Atomics MQ-1 Predator. 2016. Wikipedia – vapaa tietosanakirja. Viitattu 17.5.2016. https://fi.wikipedia.org/wiki/General_Atomics_MQ-1_Predator

Ilmailusäädös. 2016. Liikenneturvallisuusvirasto Trafin verkkosivut. Viitattu 21.3.2016.
http://www.trafi.fi/filebank/a/1444309294/3965f696e8f5d3879ef280bd1c0e6b5f/18719-OPS_M1-32_fi.pdf

LASzip Compression Details Published in PE&RS Journal. 2013. Rapidlasso verkkosivut. Viitattu 12.5.2016. <https://rapidlasso.com/2013/02/25/laszip-published-in-asprs-pears/>

LibrePilot. 2016. LibrePilot verkkosivut. Viitattu 16.5.2016. <http://www.librepilot.org/>

L. Markelin. 2007. Digitaalisten ilmakuvien radiometrinen laatu ja sen mittaaminen. Diplomityö. Viitattu 5.2.2016.

Maanmittauslaitoksen uusi valtakunnallinen korkeusmalli laserkeilaamalla. 2009. Maanmittauslaitos. Viitattu 11.5.2016. <http://docplayer.fi/330616-Maanmittauslaitoksen-uusi-valtakunnallinen-korkeusmalli-laserkeilaamalla.html>

matplotlib. 2016. matplotlib verkkosivut. Viitattu 12.5.2016. <http://matplotlib.org/>

M. Isenburg. Five Myths about LAS, LAZ, and “Optimized LAS”. 2015. Rapidlasso verkkosivut. Viitattu 12.5.2016. <https://rapidlasso.com/2015/05/12/five-myths-about-las-laz-and-optimized-las/>

PyCharm. 2016. Wikipedia - vapaa tietosanakirja. Viitattu 14.4.2016. <https://en.wikipedia.org/wiki/PyCharm>

Python(Programming language). 2016. Wikipedia - vapaa tietosanakirja. Viitattu 14.4.2016. [https://en.wikipedia.org/wiki/Python_\(programming_language\)](https://en.wikipedia.org/wiki/Python_(programming_language))

Puustontilavuuden määrittäminen relaskoopin ja kepin avulla. 2013. Puuntuottaja verkkosivut metsässä. Viitattu 14.4.2016.
<http://www.puuntuottaja.com/puustontilavuuden-maarittaminen-relaskoopin-ja-kepin-avulla/>

Sirén, H. 2015. Metsän kehitysluokkien tunnistus lentokonekeilausaineistojen perusteella Python-ohjelmointina. Opinnäytetyö. Jyväskylän ammattikorkeakoulu, ohjelmistotekniikan koulutusohjelma. Viitattu 11.4.2016. <https://www.theseus.fi/handle/10024/98022>

Uusi tekniikka skannaa metsästä oksanreiätkin. 2016. Kauppalehden artikkeli. Viitattu 21.5.2016. <http://www.kauppalehti.fi/uutiset/uusi-tekniikka-skannaa-metsasta-oksanreiatkin/9yn9QU8Q>

Unmanned aerial vehicle. 2016. Wikipedia - vapaa tietosanakirja. Viitattu 12.5.2016.
https://en.wikipedia.org/wiki/Unmanned_aerial_vehicle

Uusia menetelmiä metsävaratietojen hyödyntämiseen 2014, p. Weckström.
Jyväskylän ammattikorkeakoulun julkaisuja 180. Viitattu 5.4.2016.
<https://www.theseus.fi/handle/10024/79627>

Puustontilavuuden määrittäminen relaskoopin ja kepin avulla. 2013. Viitattu 3.5.2016. <http://www.puuntuottaja.com/puustontilavuuden-maarittaminen-relaskoopin-ja-kepin-avulla/>

V. Joala. 2006. Laserkeilauksen perusteita ja mittauksen suunnittelu. Viitattu 26.4.2016. <https://drive.google.com/file/d/0B3MfAg-wXowIN2Q4MzJIYjktZTA5Ni00ZGM5LTlkOWUtNTQzMdIwZTI3NDVm/view?pref=2&pli=1>

What is a LAS dataset? N.d. ArcGIS verkkosivut. Viitattu 11.4.2016
<https://desktop.arcgis.com/en/arcmap/latest/manage-data/las-dataset/what-is-a-las-dataset-.htm>

Liitteet

Liite 1. Tuusula info-tiedosto

lasinfo report for Tuusula1_75m.las

reporting all LAS header entries:

```
file signature:      'LASF'
file source ID:     1
global_encoding:    0
project ID GUID data 1-4: 00000000-0000-0000-0000-000000000000
version major.minor: 1.2
system identifier:   'Agisoft PhotoScan'
generating software: 'Agisoft PhotoScan'
file creation day/year: 245/2015
header size:        227
offset to point data: 321
number var. length records: 1
point data format:   2
point data record length: 26
number of point records: 21587562
number of points by return: 21587562 0 0 0 0
scale factor x y z: 0.01 0.01 0.01
offset x y z:        -1000 -1000 -100
min x y z:           -156.67 -79.48 -16.91
max x y z:           124.62 254.39 27.05
```

variable length header record 1 of 1:

```
reserved           0
user ID            'LASF_Projection'
record ID          34735
length after header 40
description        "
```

GeoKeyDirectoryTag version 1.1.0 number of keys 4

key 1024 tiff_tag_location 0 count 1 value_offset 0 - GTModelTypeGeoKey: ModelTypeUndefined

key 1025 tiff_tag_location 0 count 1 value_offset 1 - GTRasterTypeGeoKey: RasterPixellsArea

key 2048 tiff_tag_location 0 count 1 value_offset 0 - GeographicTypeGeoKey: look-up for 0 not implemented

key 2054 tiff_tag_location 0 count 1 value_offset 9001 - GeogAngularUnitsGeoKey: look-up for 9001 not implemented

reporting minimum and maximum for all LAS point record entries ...

```
X      84333  112462
Y      92052  125439
Z      8309   12705
```

```
intensity  514  65535
return_number  1  1
number_of_returns  1  1
edge_of_flight_line  0  0
scan_direction_flag  1  1
classification  0  0
scan_angle_rank  0  0
user_data  0  0
point_source_ID  1  1
```

Color R 0 65535

G 0 65535

B 0 65535

number of first returns: 21587562

number of intermediate returns: 0

number of last returns: 21587562

number of single returns: 21587562
 overview over number of returns of given pulse: 21587562 0 0 0 0 0
 histogram of classification of points: 9273318 never classified (0)

Liite 2. Visulahti info-tiedosto.

lasinfo report for Visulahti.las

reporting all LAS header entries:

file signature: 'LASF'
 file source ID: 1
 global_encoding: 0
 project ID GUID data 1-4: 00000000-0000-0000-0000-000000000000
 version major.minor: 1.2
 system identifier: 'Agisoft PhotoScan'
 generating software: 'Agisoft PhotoScan'
 file creation day/year: 246/2015
 header size: 227
 offset to point data: 413
 number var. length records: 2
 point data format: 2
 point data record length: 26
 number of point records: 9273318
 number of points by return: 9273318 0 0 0 0
 scale factor x y z: 0.01 0.01 0.01
 offset x y z: 518000 6843000 -100
 min x y z: 518668.76 6843738.43 -2.96
 max x y z: 518887.95 6843919.98 31.93

variable length header record 1 of 2:

reserved 0
 user ID 'LASF_Projection'
 record ID 34735
 length after header 56
 description ""
 GeoKeyDirectoryTag version 1.1.0 number of keys 6
 key 1024 tiff_tag_location 0 count 1 value_offset 1 - GTModelTypeGeoKey: ModelTypeProjected
 key 1025 tiff_tag_location 0 count 1 value_offset 1 - GTRasterTypeGeoKey: RasterPixellsArea
 key 3072 tiff_tag_location 0 count 1 value_offset 3134 - set_epsg_code: look-up for 3134 not implemented
 set_ProjectedCSTypeGeoKey: look-up for 3134 not implemented
 ProjectedCSTypeGeoKey: look-up for 3134 not implemented
 key 3073 tiff_tag_location 34737 count 22 value_offset 0 - PCSCitationGeoKey: ETRS89 / ETRS-GK27FIN
 key 3076 tiff_tag_location 0 count 1 value_offset 9001 - ProjLinearUnitsGeoKey: Linear_Meter
 key 4099 tiff_tag_location 0 count 1 value_offset 9001 - VerticalUnitsGeoKey: Linear_Meter

variable length header record 2 of 2:

reserved 0
 user ID 'LASF_Projection'
 record ID 34737
 length after header 22
 description ""
 GeoAsciiParamsTag (number of characters 22)
 ETRS89 / ETRS-GK27FIN|

reporting minimum and maximum for all LAS point record entries ...

X	66876	88795
Y	73843	91998
Z	9704	13193
intensity	1285	65535
return_number	1	1

```
number_of_returns 1 1
edge_of_flight_line 0 0
scan_direction_flag 1 1
classification 0 0
scan_angle_rank 0 0
user_data 0 0
point_source_ID 1 1
Color R 0 65535
      G 771 65535
      B 0 65535
number of first returns: 9273318
number of intermediate returns: 0
number of last returns: 9273318
number of single returns: 9273318
overview over number of returns of given pulse: 9273318 0 0 0 0 0
histogram of classification of points:
  9273318 never classified (0)
```

Liite 3. Siikalatva aaltoalgoritmi esimerkki T2 varttunut taimikko

