

# **Lentokeilausdatan käsittely ja analysointi metsän inventoinnissa.**

Antti Ruotsalainen

Opinnäytetyö

Toukokuu 2016

Tekniikan ja liikenteen ala

Insinööri (AMK), Ohjelmistotekniikan tutkinto-ohjelma

Tekijä(t) Ruotsalainen, Antti	Julkaisun laji Opinnäytetyö, AMK	Päivämäärä 19.5.2016
	Sivumäärä 39	Julkaisun kieli Suomi
		Verkojulkaisulupa myönnetty: x
Työn nimi <b>Lentokeilausdatan käsittely ja analysointi metsän inventoinnissa</b>		
Tutkinto-ohjelma Ohjelmistotekniikan Koulutusohjelma		
Työn ohjaaja(t) Hannu Luostarinen ja Raija Hämäläinen		
Toimeksiantaja(t) Maaseutu 2.0, Petteri Weckström		
Tiivistelmä <p>Tekniikka on tuonut metsänelinkeinon avuksi lentokeilauksen. Vaikka lentokeilaus on hyödyllinen kartoitustyökalu, sen tuottama datamäärä on liian suuri säilöttäväksi helposti.</p> <p>Opinnäytetyössä kehitettiin ohjelma, joka kykenee tiivistämään lentokeilauksen datan helpommin käsiteltävään ja säilöttävään muotoon. Näin pystytään jatkokehityksen myötä tiivistämään ulos työmäärä- ja kustannusarvioita.</p> <p>Projektin pohjana toimi aiemmissa opinnäytetöissä luotu ja eteenpäin kehitetty python-sovellus, joka on nyt käännetty C#-kielelle. Ohjelmaan on tehty graafinen käyttöliittymä ja LAS-datan muuntaminen CLM-tiedostoksi. Ohjelmaan myös kirjoitettu uusi tapa määritellä metsäalueiden kehitysluokkia ja kiireellisyyksiä. Ohjelmaan lisättiin myös avoimen lähdekoodin avulla python-esiversiosta tuttu karttalehtien piirto graafiseen muotoon. Karttalehtiä voidaan tarkastella joko ohjelman sisällä tai kirjoittaa ulos kuvatiedostomuotoon. Piirtämisessä hyödynnettiin avoimen lähdekoodin Oxyplot-rajapintaa.</p> <p>Lentokeilausdatan mahdollisuudet metsävaraintojen hyödyntämisessä ovat suuret. Datan hyödyntäminen vaatii kuitenkin paremmat resurssit ja erityisesti yksityiskohtaista tietotaitoa metsän hoidosta ja kasvatuksesta.</p> <p>Projektin tavoitteet jäivät valitettavasti vajaavaisiksi, sillä siitä jäi puuttumaan useita suunniteltuja osia. Ohjelma ei kykene lukemaan LAS-tiedostotyyppiä eikä siinä ole prototyyppiä työmäärä ja kustannusarvioiden laskemisesta. Ohjelman päätavoite, LAS-txt datan muuttamiseen CLM-tiedostoksi, onnistui ongelmitta. Projektin yhteydessä myös kehitettiin uusia menetelmiä kasvuluokan ja kiireellisyyden analysointiin pistepilvidatasta.</p>		
Avainsanat ( <a href="#">asiasanat</a> )  CLM, LAS, LAZ, Lentokeilaus, Oxyplot, C#, Metsän kehitysluokat		
Muut tiedot		

Author(s) Ruotsalainen, Antti	Type of publication Bachelor's thesis	Date 19.5.2016 Language of publication: Finnish
	Number of pages 39	Permission for web publication: x
Title of publication Airborne laser scanning data using in forest industry.		
Degree programme Software Engineering		
Supervisor(s) Hannu Luostarinen ja Raija Hämäläinen		
Assigned by Maaseutu 2.0, Petteri Weckström		
<p>Technology has introduced airborne laser scanning for the forest industry. Even though airborne laser scanning is a useful tool, the data it produces is too extensive to store easily.</p> <p>The goal of the project was to create a program that can compress data from airborne laser scanning so that it can be stored and still be used for workload and cost estimates of the project. The program also needed a user interface.</p> <p>The project is based on a python application that has been improved in the previous theses on the subject, and it has been under continuous development since. The goal of the project was to translate the program into C# language and create a graphic user interface as well as to make it possible to compress LAS data to CLM datatype. During the development process the program was also equipped with new ways to calculate growth and urgency classifications, and it also has a feature to draw the calculated areas to a visual map on the screen or to a file in the same way as in the python pre-version. The visualization of data was made possible with the use of the open-source library Oxyplot.</p> <p>Airborne laser scanning is a useful tool for forest industry, however, the analysis of the data needs more resources and specific expertise about forestry.</p> <p>Unfortunately, the project ran out of time and not all of the planned parts were implemented. The application does not have a reader for the original LAS-datatype and it cannot do calculations for workload or projections for cost estimates. The main goal, compression of LAS-txt data to the CLM-datatype was achieved. Also, new ways to calculate growth and urgency classifications from point cloud data were experimented.</p>		
Keywords/tags ( <a href="#">subjects</a> ) CLM, LAS, LAZ, airborne laser scanning, C#, Forest Growth Categorizing		
Miscellaneous		

## Sisältö

### Lyhenteet ja termit

1	Työn Lähtökohdat .....	6
2	Projektiin liittyvät organisaatiot.....	6
2.1	Maanmittauslaitoksen keilaukset .....	6
2.2	Suomen Metsäkeskus SMK .....	7
2.3	Metsäntutkimuslaitos METLA .....	7
2.4	Luonnonvarakeskus LUKE.....	7
2.5	Metsaan.fi.....	8
3	Tietoperusta projektin kehitykselle. ....	8
3.1	Suomen metsävarannot .....	8
3.2	Metsien kehitysluokat .....	8
3.3	Tietoformaatit .....	12
3.3.1	Karttalehdet, LAS ja LAZ-data .....	12
3.3.2	CLM-tietotyyppi .....	13
3.3.3	DTM, DEM & DSM .....	14
3.3.4	OxyPlot piirtokirjasto.....	15
3.4	Python version teoria. ....	15
3.4.1	Kyyt-algoritmi .....	15
3.4.2	Aalto-algoritmi.....	15
3.5	BackgroundWorker .....	16
4	Käytännön toteutus.....	16
4.1	Johdanto .....	16
4.2	MainWindow – Päänäkymä.....	19
4.2.1	PartialAreaHandler- luokka ja sen aliluokat. ....	20
4.2.2	LAS-tiedoston lukeminen ja käsittely PartialAreaHandlerilla.....	22

4.2.3	LAS-tiedoston ja DTM-maanpintamallin käsittely PartialAreaHandlerilla. ....	24
4.2.4	Kehitysluokat ja kiireellisyyden määrittäminen. ....	25
4.3	MapHandlingWindow ja MaplotWindow .....	28
4.3.1	Luetun LAS-datan muuttamien CLM-tiedostoksi. ....	30
4.3.2	Karttojen piirtäminen .....	31
5	Pohdinta .....	32
5.1	Käyttöliittymän mahdollinen jatkokehitys .....	33
5.2	Muistinvaraus .....	33
5.2.1	GrowthClass-luokka.....	34
5.2.2	Oxyplotin ongelmat. ....	36
5.3	Tarkkuus .....	36
5.4	Muistinhallinnan kokeilut.....	37
5.5	LAS-datan lukeminen.....	38
5.6	Työmääräarviot ja tulevaisuus. ....	38
	Lähteet.....	40

## Kuviot

Kuvio 1. T1 Metsä ennen ja jälkeen hoidon.....	10
Kuvio 2. T1 metsä hoidon jälkeen .....	11
Kuvio 3. T2 metsä, kiireellinen .....	11
Kuvio 4. DSM ja DTM ero (Wikipedia DTM ja DSM. 2015.).....	14
Kuvio 5. Aktiivisuuskaavio ohjelman kulusta ja käyttötarkoituksista. ....	18
Kuvio 6. Ohjelman kulkulogiikka. ....	19
Kuvio 7. MainWindow-näkymä ajon aikana.....	20
Kuvio 8. Luokkapiirros LAS-txt ja DTM-datan käsittelevästä PartialareaHandler- luokasta .....	21
Kuvio 9. PartialAreaHandler – Päämetodi.....	22
Kuvio 10. Sekvenssidiagrammi LAS-txt lukemisesta .....	23
Kuvio 11. Sekvenssidiagrammi LAS ja DTM tiedon käsittelystä .....	24
Kuvio 12. DTM-indeksin laskutapa .....	25
Kuvio 13. Teoreettinen T1 kasvuluokan pistehajonta.....	26
Kuvio 14. Teoreettinen T2 Kasvuluokka.....	26
Kuvio 15. Teoreettinen huono T2 kasvuluokka.....	27
Kuvio 16. MapHandlingWindow – hallintaikkuna.....	29
Kuvio 17. MapHandlingWindow ja sen aliluokat .....	29
Kuvio 18. CLM tiedoston kirjoitus. ....	30
Kuvio 19. Esimerkkikuva 1000 * 1000 metriä käsitellystä alueesta.....	31
Kuvio 20. LinQ-lauseke pisteiden noutamiseen kokoelmalistasta.....	32
Kuvio 21. MVVM-malli. 2012. ....	33
Kuvio 22. Growthclass-luokan muistinvaraus ennen .....	34
Kuvio 23. muistinvaraus Growthclass-luokan poistamisen jälkeen jälkeen .....	34
Kuvio 24. GrowthClass luokan objektimäärävertailu .....	35
Kuvio 25. Growthclass luokan muistinvarausvertailu .....	35
Kuvio 26. Datan nouto SQLite tietokannasta ja DTM -tiedostosta .....	37

**Taulukot**

Taulukko 1. Metsän kasvu- ja kiireellisyysluokat. ....	9
Taulukko 2. Puuston kehitysluokat. 2013. ....	10
Taulukko 3. CLM-luokan otsikkotiedot. ....	13
Taulukko 4. Teoreettinen esimerkkipistejakauma Aalto -algoritmin taulukossa. ....	16

## Lyhenteet ja termit

Avoim Standardi	Avoimesti saatavilla oleva määrittely, esimerkiksi vakiojärjestys jossa tieto on tallennettu.
Binääritiedosto	Tiedostomuoto, jossa data on esitettyä binäärimuodossa.
Double-muuttuja	Muuttuja johon voidaan säilöä numeroarvoja.
Implementointi	Luoda suunniteltu ominaisuus.
LAS-data	Avoimen standardin datamuoto, joka on pakattuna binäärimuotoon. Lentokeilauksen tulokset on tallennettuna LAS-data muotoon.
Lentokeilaus	Lentokoneella suoritettava maaston kartoitusmenetelmä.
Metsätalous	Metsänhoitoon ja kasvatukseen painottuva elinkeinotoiminta.
Nuget-paketinhallinta	Vapaan lähdekoodin paketinhallintaohjelma Visual Studio-kehitysympäristölle. Helpottaa kolmannen osapuolten pakettien etsimisen ja asentamisen.
Pistepilvidata	Lentokeilauksen tuottama data on pistepilvidataa, eli se kuvastaa mittauspisteitä 3-ulotteisessa tilassa.
Ratkaisuarkisto	Visual studio -ohjelmistoympäristön tietorakenne mihin voidaan koota useampia projekteja joilla voi olla riippuvuuksia tai muita yhteyksiä toisiinsa.
String-muuttuja	Muuttuja johon voidaan säilöä tekstiä.
Säikeistämisen	Säikeistäminen mahdollistaa useamman ohjelmakoodin ajamisen yhtäaikaan. Esimerkiksi säikeistämällä voidaan ajaa koodia taustalla pitäen samalla käyttöliittymä responsiivisena.
Txt-tiedosto	tiedosto, joka sisältää vain tekstiä.
Responsiivinen	Kykenee ottamaan käskyjä vastaan.

## 1 Työn Lähtökohdat

Metsien lentokeilaus on uuden ajan työkalu, joka mahdollistaa suomen metsävarantojen kartoittamisen aiempaa tehokkaammin. Lentokeilauksen tuottaman datan esittämiseen on kehitetty useampikin työkaluja, mutta ohjelmistolle, joka pystyisi analysoimalla ja tiivistämään lentokeilauksen tuottaman LAS-datan tiivimpään muotoon, löytyy kysyntää. Tämän opinnäytetyön tarkoituksena oli luoda työkalu jolla kyetään lukemaan, analysoimaan ja kääntämään LAS-dataa helpommin käsiteltävään muotoon. Projektin metsäasiantuntijana toimi Kalevi Pietikäinen Marjatta ja Eino Kollin säätiön apurahalla.

Projektin päätoimeksiantajana toimi Maaseutu 2.0. Maaseutu 2.0 ohjelma pyrkii uudistamaan metsänhoitotyötä. Metsien ajantasaiseen kartoitukseen tarvittavat työkalut ovat olleet olemassa jo vuosikausia, mutta keskittäminen yhtenäiseksi kokonaisuudeksi on jäänyt toteuttamatta. Maaseutu 2.0 tarkoitus on ”siirtää omaisuuden hallinta digiaikaan” (Yrittävä maaseutu. 2016.).

Tarkoituksena on avoimen standardin ja järjestelmäriippumattomuuden avulla helpottaa tiedon, työtarjousten ja työvoiman liikkumista maa- ja metsätalousaloilla.

## 2 Projektiin liittyvät organisaatiot

Suomen maapinta-alasta 86 % on metsää. Näin ollen ei ole yllätys että maassamme on useita metsänhoitoon ja asiantuntijapalveluihin erikoistuneita organisaatiota. Nämä organisaatiot ovat enimmäkseen valtiohallinnon alaisia organisaatiota.

### 2.1 Maanmittauslaitoksen keilaukset

Suomessa Maanmittauslaitos suorittaa muiden tehtäviensä ohessa myös maastokeilausta, ja organisaatiolla on Suomen Metsäkeskuksen kanssa ”Suomi loppuun” -keilaussuunnitelma. Keilaussuunnitelman tarkoitus on saada katettua koko

Suomen maasto ja luoda tarkka valtakunnallinen 2 metrin korkeusmalli (Maanmittauslaitoksen laserkeilausaineisto. N.d.).

”Laserkeilausaineistoja hyödynnetään myös korkeuskäyrien luomisessa, kaavoituksessa ja metsien inventoinneissa sekä muissa luonnonympäristön muutosta seuraavissa tutkimuksissa ja seurannoissa.” (Maanmittauslaitoksen laserkeilausaineisto. N.d.).

Maanmittauslaitoksen suoritti vuonna 2014 laserkeilausta 71 702 neliökilometrin alueelta (Maanmittauslaitoksen vuositilastoja. 2014).

Maanmittauslaitoksen mittaukset ovat avointa aineistoa.

## 2.2 Suomen Metsäkeskus SMK

Suomen Metsäkeskus on Maa- ja Metsätalousministeriön alainen organisaatio, joka pyrkii edistämään suomalaisten metsänomistajien asemaa neuvomalla, valvomalla lainsäädännön noudattamista sekä keräämällä tietoa. Suomen Metsäkeskus suorittaa metsien kartoitusta laserkeilauksen lisäksi myös käyttämällä ilmakuvausta, koealamittausta ja maastoinventointia (Suomen Metsäkeskus. N.d)

## 2.3 Metsäntutkimuslaitos METLA

Metsätutkimuslaitos on Maa- ja Metsätalousministeriön tulosohtauksen alainen organisaatio, joka pyrkii kehittämään metsään liittyviä palveluja, tuotteita sekä metsänhoitoratkaisuja. METLA pyrkii keräämään ja tuottamaan tietoa metsä- ja puutaloudesta tieteellisellä tutkimuksella ja asiantuntijatyöllä.

Metla on perustanut erillisiä tutkimusmetsiä jo 1920-luvulta, ja organisaatiolta löytyy tietoa jo 1800-luvulla suoritetuista kokeista. Tutkimusmetsäverkon laajuus on noin 30 000 hehtaaria maa- ja vesialueita. (Metla. N.d)

## 2.4 Luonnonvarakeskus LUKE

Luonnonvarakeskus on Maa- ja Elintarviketalouden Tutkimuskeskuksen (MTT), Metsätutkimuslaitoksen (Metla), Riista- ja Kalatalouden Tutkimuslaitoksen (RKTL) sekä Maa- ja Metsätalousministeriön Tietopalvelukeskuksen Tilastointipalveluiden (Tike) yhdistymisessä vuonna 2015 muodostettu organisaatio.

Luonnonvarakeskuksen lakisääteisiin viranomaistehtäviin kuuluu taimituotannon

varmentaminen, luonnonvarojen seuranta, geenivarojen säilytys sekä kasvihuonekaasujen laskenta (Luke N.d.)

## 2.5 Metsaan.fi

Metsaan.fi on metsänhoitoon erikoistunut ilmainen palvelu, jonka tarkoitus on yhdistää metsänomistajat, metsäalan palveluntarjoajat ja Suomen Metsäkeskus. (Metsaan. N.d.)

# 3 Tietoperusta projektin kehitykselle.

## 3.1 Suomen metsävarannot

Suomen metsätalous elää ja voi hyvin. Maapinta-alasta noin 86 prosenttia on metsää (26.2 miljoonaa hehtaaria), josta 20.3 miljoonaa on puuntuotannon käytössä. Kokonaistilavuus puustolle 2.4 miljardia joka lisääntyy kasvun myötä 105.5 miljoonaa kuutiometriä. Yksityisten metsätilojen omistajuus jakantuu usein useammalle henkilölle, joten yksityisiä vähintään yhden hehtaariin kokoisia alueita löytyi vuoden 2013 lopussa 376 000, joiden omistajina toimii 685 000 henkilöä. (Metsänomistajat Haapavesi-Kärsämäki ja Siikalakeus-lehti. 2016. 26)

## 3.2 Metsien kehitysluokat

”Kehitysluokka kuvaa puuston metsänhoidollista ja puuntuotannollista kehitysvaihetta tietyllä hetkellä monimuotoisuudesta tai muista arvoista riippumatta.” (Suomen Metsäurheiluliitto. N.d)  
Metsän kehitysluokilla pyritään erottelemaan kasvistot kasvu- ja metsähoitotarpeen perusteella. Näin metsänhoito saadaan ajoitettua tehokkaammin ja yksilöllisemmin alueittain.

T1 vaiheessa kyseessä on taimikkometsä, johon osoitetaan enemmän harvennus ja perkaus tyyppisiä töitä. T1 taimikkometsä on vanhastaan rajattu 1,3 metrin korkeuteen koska se on helppo ollut katsoa ”noin vyötärönkorkeudelta”.

T2 vaiheessa on kyseessä varttunut taimikko, jolle pitää osoittaa harvennushakkuita. Alarajana toimii 1,3 metriä ja ylärajana kuusella alle 7 metriä ja koivulle alle 9 metriä. Keskiläpimitta rinnankorkeudella alle 8 senttimetriä. (Puuston kehitysluokat. N.d)

O2 vaiheessa kyseessä on nuori kasvatusmetsikkö. pituus on väliltä 6-12 metriä ja keskiläpimitta rinnankorkeudelta noin 8-16 senttimetriä.

Kehitysluokille löytyy enemmänkin määritelmiä mutta tämän projektin kannalta nämä kolme ovat oleellisimpia.

Taulukossa 1 on eritelty puuston kiireellisyysluokat vaiheittain. Kasvuluokat vaihtuvat 5-numeron välein eli 15, 20, 25 ja 30 ovat ”pohja-arvoja”. Näistä arvoista alkaa 4-asteinen kiireysmäärittely. Kiireysmäärittelyn on tarkoitus kertoa tarvitseeko alue välitöntä hoitoa ja jos ei, niin millä aikavälillä se tulisi hoitaa. Harvennus ja hoitotarve pyritään määrittämään 3-7 vuoden tarkkuudella.

Taulukko 1. Metsän kasvu- ja kiireellisyysluokat.

14	A0 – aukea
15	T1 - uudistamis- /täydennystarve
16	T1 - harvennustarve 6-10v
17	T1 - harvennustarve 1-5 v
18	T1 - perkaustarve KIIREELLINEN
19	T1 - perkaustarve HETI (myöhässä)
20	T2 - uudistamis-/täydennystarve
21	T2 - harvennustarve 7-10 v
22	T2 - harvennustarve 3-6 v
23	T2 - hoitotarve KIIREELLINEN (1-2 v)
24	T2 - hoitotarve HETI (myöhässä)
25	O2 - ei hoitotarvetta
26	O2 - hoitotarve 7-10 v
27	O2 - hoitotarve 3-6 v
28	O2 - hoitotarve KIIREELLINEN (1-2 v)
29	O2 - hoitotarve HETI (myöhässä)
30	muu metsä (varttunut kasvatusmetsä)

Pelkästään kehitysluokista puhuttaessa voidaan metsä luokitella vastaavasti taulukon 2 kuvaamalla tavalla.

Taulukko 2. Puuston kehitysluokat. 2013.

Uudistusala ja taimikko	A0 S0,T1,T2,Y1,05
Nuori kasvatusmetsä	02
Varttunut kasvatusmetsä	03
Uudistuskypsä metsä	05

Uudistusala ja taimikko on vielä pelkästään kuluerä metsänkasvattajalle. Täten näitten metsien hoitotarpeiden kartoitus on erityisen tärkeä, jotta metsänomistaja kykenee maksimoimaan hyödyn metsäpalstasta hyvällä hoidolla. Siksi tässäkin työssä keskitytään näitten kasvuluokkien analysointiin.

Vaikka metsä olisi vasta taimivaiheessa se tarvitsee silti järjestelmällistä harvennusta. (ks. kuvio 1)



Kuvio 1. T1 Metsä ennen ja jälkeen hoidon.

Nuorta taimikkoa tulee harventaa tasaisesti että puuston kasvu saadaan maksimoitua (ks. kuvio 2). On hyvä ottaa huomioon selkeä järjestelmällisyys ja kaiken kasvamista haittaavan poisto.



Kuvio 2. T1 metsä hoidon jälkeen

Metsän hoitamatta jättäminen haittaa metsän kasvua huomattavasti (ks. kuvio 3).

Tämä metsä määriteltäisiin jo erittäin kiireelliseksi koska se täysin umpeenkasvanut.



Kuvio 3. T2 metsä, kiireellinen

### 3.3 Tietoformaatit

#### 3.3.1 Karttalehdet, LAS ja LAZ-data

LAS-data on American Society for Photogrammetry and Remote Sensing-organisaation (ASPRS) kehittämä avoimen lähdekoodin binääritiedostomuoto. LAS-dataa käytetään tallentamaan lentokeilausaineiston tuottamaa 3-ulotteista pistepilvidataa (Digital Preservation. 2015.). Pistepilvidatasta voidaan piirtää kolmiulotteinen malli alueesta, jossa lentokeilaus on suoritettu.

LAZ-datamuoto on LAS-dataan pakattu tiedostomuoto, jolla saadaan häviöttömästi tiivistettyä lentokeilausdata helpommin säilöttävään muotoon. Pakkaukseen sopivia työkaluja on kehittänyt muunmuassa rapidlasso GmbH, jonka LASzip ohjelmisto on myös avointa lähdekoodia. LASzip on osa LAStools ohjelmistopakettia. (LASzip documentation. N.d).

Projektin kehitystyössä käytettiin Maanmittauslaitokselta saatua lentokeilausaineistoa, joka oli muutettu las2txt-ohjelmistolla txt-muotoon. Txt-muotoiset karttalehdet ovat keilausmittaustuloksia jotka on otettu noin 6000\*6000 metrin alueelta. Lentokoneella tehtäessä tämänkokoiselle alueelle osuu noin 31 miljoonaa osumaa (keskimäärin 0,8 / neliometri). Tiedoston kooksi tulee muutoksen jälkeen noin 2 gigatavua.

LAS-dataa käsitellään kolmessa muodossa.

1. Luokittelematon LAS-data (alla) johon ei ole laskettu mitään tietoja.
2. LAS data johon on laskettu kasvuston korkeus delta-z, joka korvaa korkeustiedon z luokittelemattomassa LAS tiedostossa.
3. LAS datan mukana toimitetaan DTM tiedosto joka sisältää maastonkorkeudet. Tämän avulla voidaan tarkentaa korkeuslaskelmia kun Z-datasta lasketaan kasvuston korkeutta. Tämä vastaa DTM:ää joka esitellään luvussa 3.3.3.

Mittauspisteiden data on karttalehdillä alla olevassa muodossa.

382999.97 6919478.97 146.46 6 1 1 0 0 1 -2 0 27 57791572.329861
---

(Ensimmäinen datarivi karttalehdeltä N4234A)

Tämän projektin kannalta tarpeellisia arvoja on vain kolme ensimmäistä:

1. Piste X koordinaatti.
2. Piste Y koordinaatti.

### 3. Piste Z koordinaatti (Etäisyys mittauspisteestä maassa).

Tämän opinnäytetyön lopputuloksena kehitettävän ohjelman on tarkoitus laskea ja muuntaa tämänkoinen ja muotoinen tiedosto CLM-tietomuotoon. Tiedoston koko putoaa noin 1,70 megatavuun, jolloin koko suomen tämänhetkiset karttalehdet saataisiin mahtumaan yhdelle muistitikulle.

#### 3.3.2 CLM-tietotyyppi

CLM -luokka on projektin yhteydessä kehitetty tietorakenne johon ohjelma tallentaa syötetyn alueen kasvuluokat.

```
ncols 5999
nrows 5999
xllcorner 380000
yllcorner 6918000
cellsize 8
NODATA_value -9999
31 31 31 31 31 31 31 ...
```

CLM:än tarkoitus on saada puristettua karttalehden data mahdollisimman pieneen tilaan. Tämä saadaan aikaiseksi jakamalla alueen pisteet 8\*8 metriä kokosiin ruutuihin ja laskemalla näille ruuduille kehitys- ja kiireellisyysluokka.

Taulukko 3. CLM-luokan otsikkotiedot.

Ncols	rivimäärä
Nrows	sarakemäärän
Xllcorner	X-piste alueen lounaiskulmassa
yllCorner	y-piste alueen lounaiskulmassa
Cellsize	Ruudun koko
NODATA_value	Tällä arvolla merkitystä ohjelmasta ei ole dataa.

X- ja Y pisteen jälkeen olevat kirjaimet kertovat minkä kulman kordinaatteja ne kuvaavat. "ll" tarkoittaa "lower-left" joten kyseessä on kuvatun alueen lounaiskulma. Tämän kulman avulla sekä ruudun koon ja rivi sekä sarakemäärän avulla voidaan alueen pisteille antaa oikeat sijainnit.

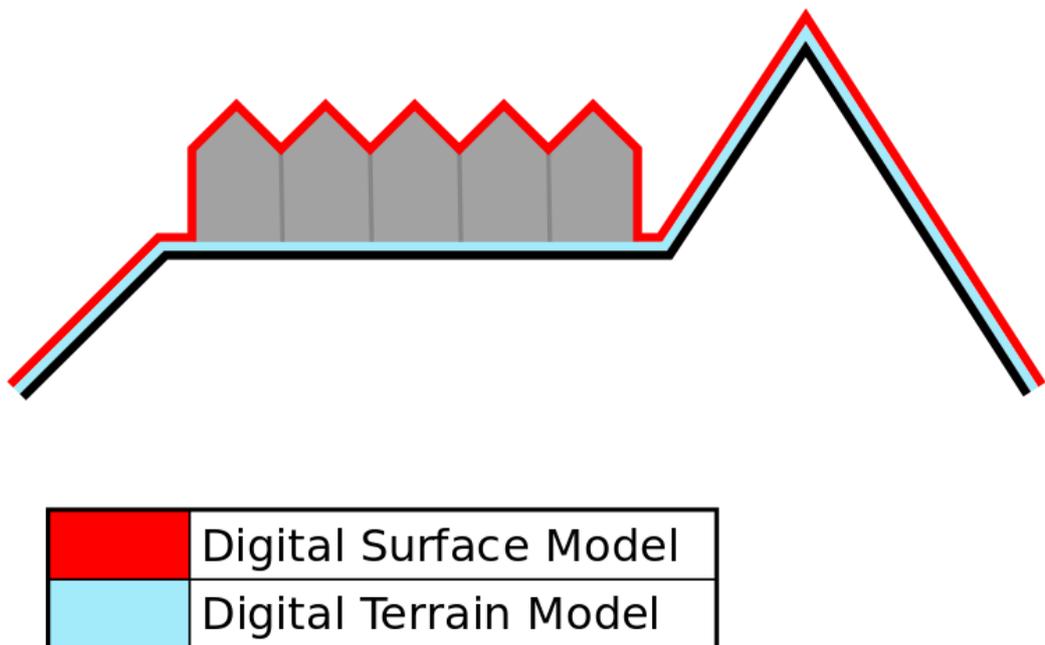
Näitten arvojen jälkeen alkaa numerosarja jonka jokainen arvot kuvaavat kehitysluokkaa ja kiireellisyyttä (esitelty kappaleessa 2.1.1.)

CLM mahdollistaa 2 gigatavua vievän tiedoston puristamisen noin 1,7 megatavuun. CLM- numeroarvoista voidaan laskea saantoja eli hyötypuun määriä alueelta sekä tehdä työmääräarvioita. Tämä vaatii kuitenkin lisätietoa siitä kuinka paljon keskimäärin menee tietyssä kiireellisyyssvaiheessa olevan alueen siistimiseen. CLM-tiedosto voidaan myös lukea ArcGIS- ja QGIS-ohjelmilla kartan piirtämistä varten.

### 3.3.3 DTM, DEM & DSM

DTM (Digital Terrain Model), DEM (Digital Elevation Model) sekä DSM (Digital Surface Model) ovat erityyppisiä maanpintamalleja. Digital Surface Model pyrkii mallintamaan kaikki havaitut korkeuspinnat (esim. talojen katot.) ja DTM mallintaa vain maatasen (Digital terrain model. N.d.) (ks. kuvio 4.).

LAS datan avuksi voidaan projektin luomassa ohjelmassa lisätä DTM tiedosto jolloin maastopisteet on varmemmin löydettävissä.



Kuvio 4. DSM ja DTM ero (Wikipedia DTM ja DSM. 2015.)

### 3.3.4 OxyPlot piirtokirjasto.

Oxyplot on avoimen lähdekoodin kirjasto, jolla voidaan piirtää kaavakuvia eri alustoilla. Oxyplot valittiin hyvän dokumentaation sekä useiden tuettujen alustojen takia. Tuettuja alustoja on muunmuassa GTK#, Windows Phone, Android, WPF, Silverlight (Oxyplot. N.d.). Oxyplotin lisenssityyppi, MIT, mahdollistaa hyödyntämisen myös suljetun lähdekoodin ohjelmistoissa (Oxyplot license. N.d.).

## 3.4 Python version teoria.

Taisto Kuikan ja Kalevi pietikäisen alkuperäisversiossa käytettiin kahta eri laskukaavaa kun pyrittiin löytämään kasvu- ja kiireellisyysluokitus karttaan jaetulle ruudukolle.

### 3.4.1 Kyyt-algoritmi

Kun karttalehden pisteet oli jaettu halutun kokoiisiin alueisiin käytettiin kyyt-algoritmia laskemaan niille kasvuluokka ja kiireellisyys. Algoritmi käy läpi laserpisteet ja antaa jokaiselle kehitysluokan. Algoritmi seuraa myös kuinka moni osuma osuu peräkkäin samaan kehitysluokkaan. Eniten laserpisteitä saanut kehitysluokka määrittää ruudun kehitysluokan ja peräkkäisten osumien summa antoi ruudulle kiireellisyyden.

### 3.4.2 Aalto-algoritmi

Aalto-algoritmi oli toinen kokeellinen menetelmä laserpisteiden analysointiin. Algoritmi ajettiin samalla lailla kuin kyyt-algoritmissa, eli vasta sen jälkeen kun karttalehti oli ruudutettu ja pisteet jaettu ruutuihin. Ruudun pisteet käytiin läpi yksi kerrallaan ja katsottiin mihin kasvuluokkaan piste kuului. Riippuen kasvuluokasta pisteen pituus sai arvon jolla pisteen korkeusarvo jaetaan. Tulokset kerättiin taulukkoon, josta kaikkien pisteiden läpikäynnin jälkeen katsottiin eniten pisteitä saanut ruutu määrittä kehitysluokan ja kiireyden.

Esim Pisteen korkeus on 0,4 metriä, joten se kuuluu T1 kasvuluokkaan. Jaetaan T1 "kertomella" 0.1, tulos 4. Piste T1 rivin sarakkeeseen 4.

Allaoleva ruudukko täytetään tällaisella laskukaavalla saaduilla pisteillä kunnes karttaruudun sisällä olevat pisteet on käyty läpi.

Taulukko 4. Teoreettinen esimerkkipistejakauma Aalto -algoritmin taulukossa.

maa	4	0	0	0	0	0	0	0	0
T1	1	1	2	4	0	0	0	0	0
T2	0	2	0	0	0	0	0	0	0
02	0	0	0	0	0	0	0	0	0

Koska kyseessä on pisteyden jakautumisella toimiva algoritmi, ruutukoon pitää olla vähintään 8\*8 m jotta alueen pistemäärä on vähintään 50 paikkeilla. Vähemmällä pistemäärällä tarkkuus kärsii huomattavasti. Ruutukoon kasvaessa ruudun arviointitarkkuus paranee, mutta ruutukoon kasvaessa ruudut ovat liian suuria rajatakseen vaihtuvat metsätyypit ja maastonmuodot.

### 3.5 BackgroundWorker

BackgroundWorker-luokka on .Net ominaisuus mikä mahdollistaa ohjelmakoodin säikeistämisen niin, että laskutoimitukset voidaan suorittaa taustalla ja käyttöliittymä pysyy responsiivisena ( .Net Dokumentaatio. BackgroundWorker, N.d.)

## 4 Käytännön toteutus

### 4.1 Johdanto

Kehitystyön pohjana toimi python-ohjelmistokielelle kehitetty ohjelma, jonka ovat kehittäneet Taisto Kuikka ja Kalevi Pietikäinen. Ohjelmaa luki sisäänsä LAS-dataa txt-muodossa, ja datasta laskettiin KYYT- tai Aalto-algoritmeilla karttalehden kuvaamalle alueelle kasvu- ja kiireellisyysluokka. Ohjelmaa säädettiin yksinkertaisella kytkinmenetelmällä vaihtamalla boolean-muuttujia True tai False asentoon. Ohjelma ongelmana oli että sisälsi myös paljon kovakoodattuja muuttujia eikä sillä ollut käyttöliittymän. Ohjelmalla kyettiin kuitenkin piirtämään hyviä karttakuvia Matplotlib-

kirjaston avulla ja ne saatiin esitettyä graafisesti näytöllä sekä tallennettua kuvatiedostoksi.

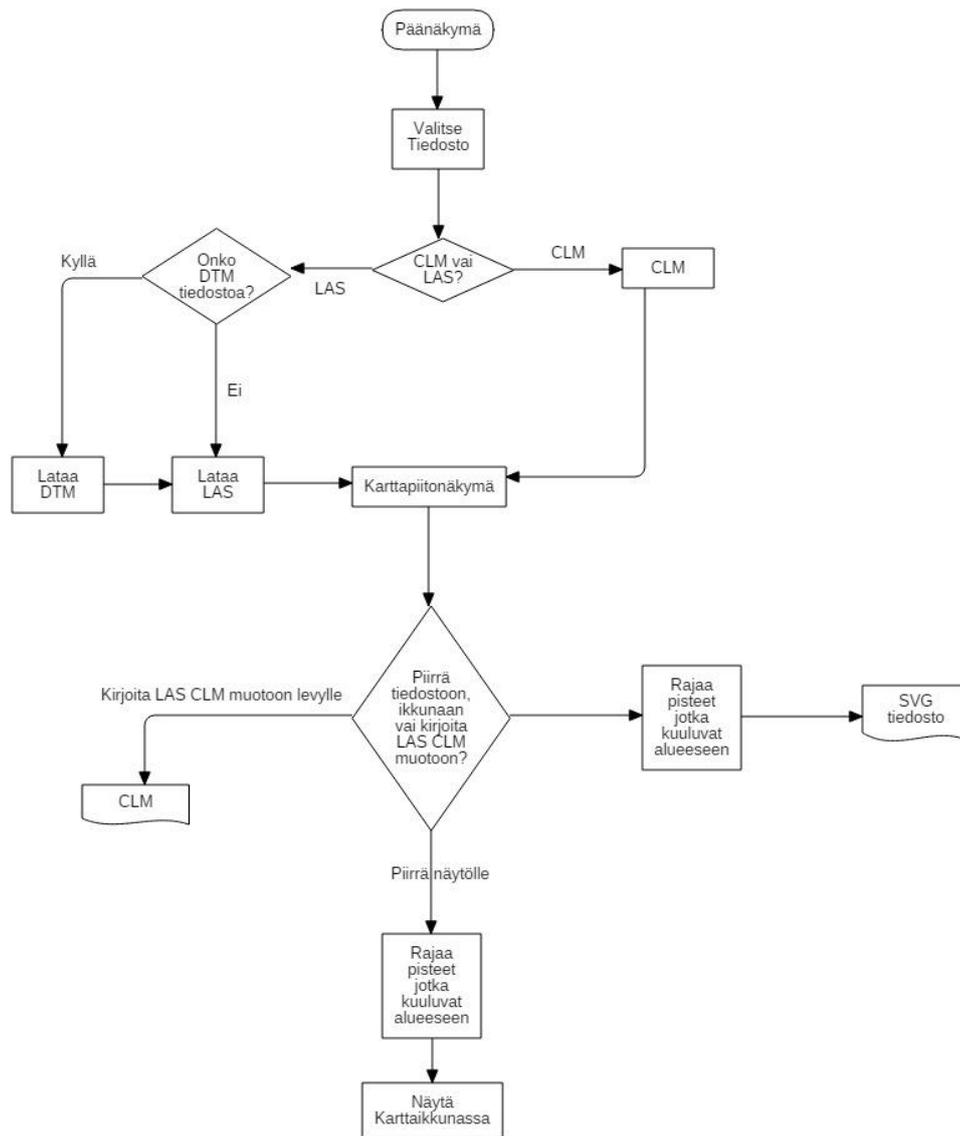
Uuden projektin tavoitteeksi muodostui:

- LAS-txt datan tiivistäminen algoritmilla CLM-tietomuotoon.
- Mahdollisuus DTM-txt datan lisäämisestä LAS-txt datan käsittelyyn.
- Yksinkertaisen käyttöliittymän kehittäminen.
- Python-versiossa mukana ollut kartan piirtomahdollisuus LAS-txt ja CLM datasta.

Ohjelmaa harkittiin alkuun alustariippumattomaksi mutta teknisten vaikeuksien ja ajankäytön säästämiseksi päätettiin tehdä WPF-pohjainen windows -sovellus. Lopputulokseksi muotoitui ohjelma, joka vaatii pyöriäkseen tietokoneen jossa on vähintään 64-bittinen Windows käyttöjärjestelmä ja vähintään 4 gigatavua keskusmuistia.

Karttapiirron toteuttamiseen valittiin kevyen tutkimustyön jälkeen Oxyplot-kirjasto. Valintaan vaikutti selkeä ja hyvin saatavilla oleva dokumentaatio, avoimen lähdekoodin MIT-lisenssi sekä jatkokehityksessä mahdollisesti hyödynnettävä useiden alustojen tuki.

Käyttöliittymä on jaettu kolmeen osaan käyttötarkoituksen mukaan. Mainwindow eli päänäkymä suorittaa tiedostolatauksen ja laskutoimitukset, MapHandlingView eli karttapiirtonäkymä hallitsee mitä ladatulle tiedostolle tehdään, MapPlotWindow piirtää halutun alueen kuvatiedostoon tai esittää sen näytöllä. LAS-txt:n avuksi voidaan syöttää DTM-dataa sisältävän txt tiedoston, joten sisäänlataus ja kiireellisyysluokat täytyy käsitellä kahdella eri tavalla riippuen oliko DTM-data vai ei. CLM-tiedosto voidaan suoraviivaisesti lukea sisään ja siirtyä seuraavaan vaiheeseen (ks. kuvio 5)



Kuvio 5. Aktiivisuuskaavio ohjelman kulusta ja käyttötarkoituksista.

Ohjelmalla on yhtenäinen kulkujärjestys ja logiikka, päänäkömä suorittaa datan sisäänluvun ja laskutoimitukset, kartankäsittelynäkymässä käyttäjä voi päättää mitä datalle tehdään (ks. kuvio 6).



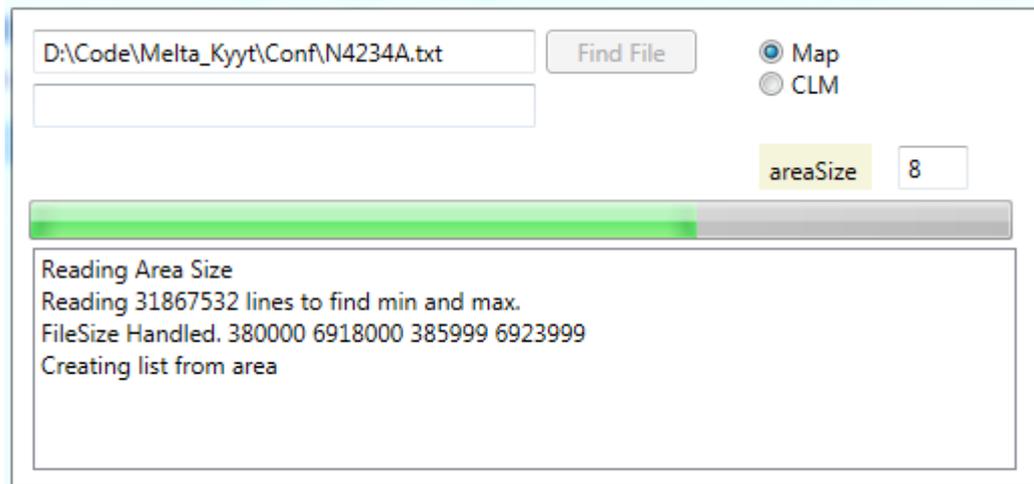
Kuvio 6. Ohjelman kulkulogiikka.

Sisäänluettavan datan tyyppin tai tyyppien perusteella ohjelma ottaa käyttöön datalle sopivan käsittelijän (PartialAreaHandler, CLMHandler), joka vuorostaan alustaa tekstinlukijan ja kutsuu tarvittavaa tekstinlukumetodia. Tekstinlukija lukee ja parsii tekstin ja alustaa näillä arvoilla käsittelijän hallussa olevan kokoelmaluokan. Kokoelmaluokkaan on kerätty tekstin päädata ja kokoelman sisältävään listaan on järjestelty yksilöitävä data (Laserpisteiden osumat ja CLM tiedot ruuduille jaettuna).

Riippumatta mitä dataa on ladattu, päänäköymä siirtää ladatun kokoelman karttahalintanäkymälle, jossa käyttäjä tekee päätöksen mitä ladatulle datalle tehdään.

## 4.2 MainWindow – Päänäkymä

Päänäkymän tarkoitus on ottaa vastaan tiedostot joita halutaan tarkastella tai käsitellä ohjelmalla. LAS- ja DTM-tiedostoa käytettäessä voidaan säätää haluttu ruutukoko jolle alueen pisteet jaetaan areaSize-valikolla (ks. kuvio 7).



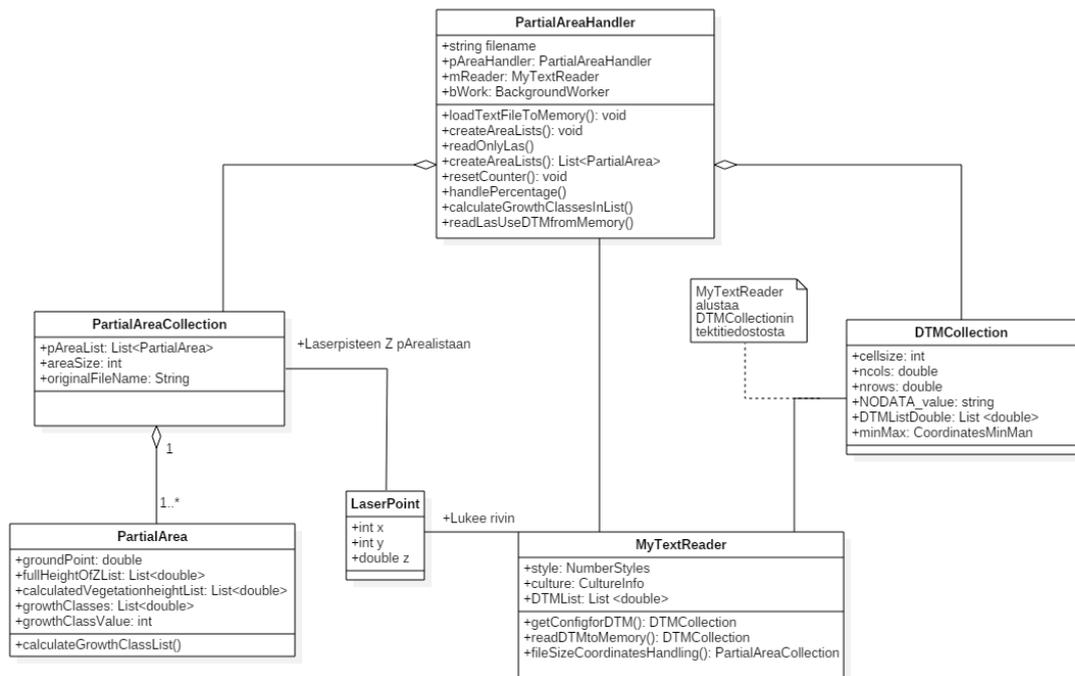
Kuvio 7. MainWindow-näkymä ajon aikana.

Varsinaisen ajon alkaessa tiedostojen sisältö tarkastetaan että kyseessä on LAS-, DTM- tai vaihtoehtoisesti CLM-tiedosto. Tämän jälkeen tiedostojen sijainnit annetaan backgroundworker-luokalle, joka suorittaa laskutoimitukset erillisessä säikeessä samalla päivittäen tilannetietoja päänäkymään.

#### 4.2.1 PartialAreaHandler- luokka ja sen aliluokat.

PartialAreaHandler-luokka suorittaa suurimman osan ohjelman toiminnallisuuksista. Luokan päätehtävä on hoitaa LAS-txt ja DTM-txt datan käsittelyn.

Kokoelmaluokka PartialAreaCollection sisältää listan PartialArea olioita. Nämä oliot kuvastavat ruutuja joihin sisäänluettava LAS-data jaetaan, ja ne on luotu jakamalla alueen koko jaettuna käynnistyksen yhteydessä käyttäjän antamalla arvolla (ks. kuvio 8).



Kuvio 8. Luokkapiirros LAS-txt ja DTM-datan käsittelevästä PartialAreaHandler-luokasta

Oletusarvona käytetään 8\*8m ruutuja. Tämä koko on aiemmissa laskutoimituksissa (python kehitys) todettu sisältävän tarpeeksi dataa jotta kasvuluokka ja kiireellisyys kyetään määrittämään mutta ei liian suuri että kasvuluokka tai metsätyyppi pääsisi muuttumaan alueen sisällä eikä maastossa pääse tapahtumaan suuria muutoksia.

PartialArea:n calculateGrowthClassList-metodit suorittaa laskutoimitukset millä määritetään alueen sama kasvuluokka ja kiireellisyys.

DTMCollection on kokoelmaluokka mahdollisesti käytettävälle DTM-datalle. DTM-datan sisältämiä maapistetietoja käyteen selvittämään LAS-datan pisteistä alueen kasvuston korkeutta.

Tärkein huomioitava ohjelman toiminnassa on että jokainen ruudukko arvoidaan itsenäisenä alueena. Vierekkäisiä alueita ei vertailla keskenään vaan jokainen ruutu saa arvonsa vaan ruudun sisäisten pisteiden perustella.

Ohjelman eri työvaiheet on jaettu PartialAreaHandler-luokan muihin metodeihin. Päämetodi kutsuu vuoronperään eri työvaiheeseen tarvittavaa metodia (ks. kuvio 9).

```

public void loadTextFileToMemory(string lasFile,int areaSize, string dtmfilepath = "",int handlingvalue=0)
{
    Stopwatch watch = new Stopwatch();
    watch.Start();
    MyTextReader mReader = new MyTextReader();
    handleReporting(0, "Creating list from area");
    pAreaCollection.minMax=mReader.fileSizeCoordinatesHandling(lasFile,handleReporting);
    pAreaCollection.pAreaList= createAreaLists(areaSize,pAreaCollection.minMax);
    if (dtmfilepath != "") {
        this.dtmCollection = mReader.readDTMtoMemory(dtmfilepath);
        mReader.readLasUseDTMfromMemory(lasFile, ref pAreaCollection.pAreaList, ref dtmCollection, pAreaCollection.minMax, pAreaCollection.areaSize, handleReporting);
    }
    else
    {
        mReader.readOnlyLas(lasFile,ref pAreaCollection, handleReporting);
    }

    calculateGrowthClassesInList();
    watch.Stop();
    string time= MyTextReader.kelloaika(watch.Elapsed);
    handleReporting(0," area handled in " +time);
}

```

## Kuvio 9. PartialAreaHandler – Päämetodi

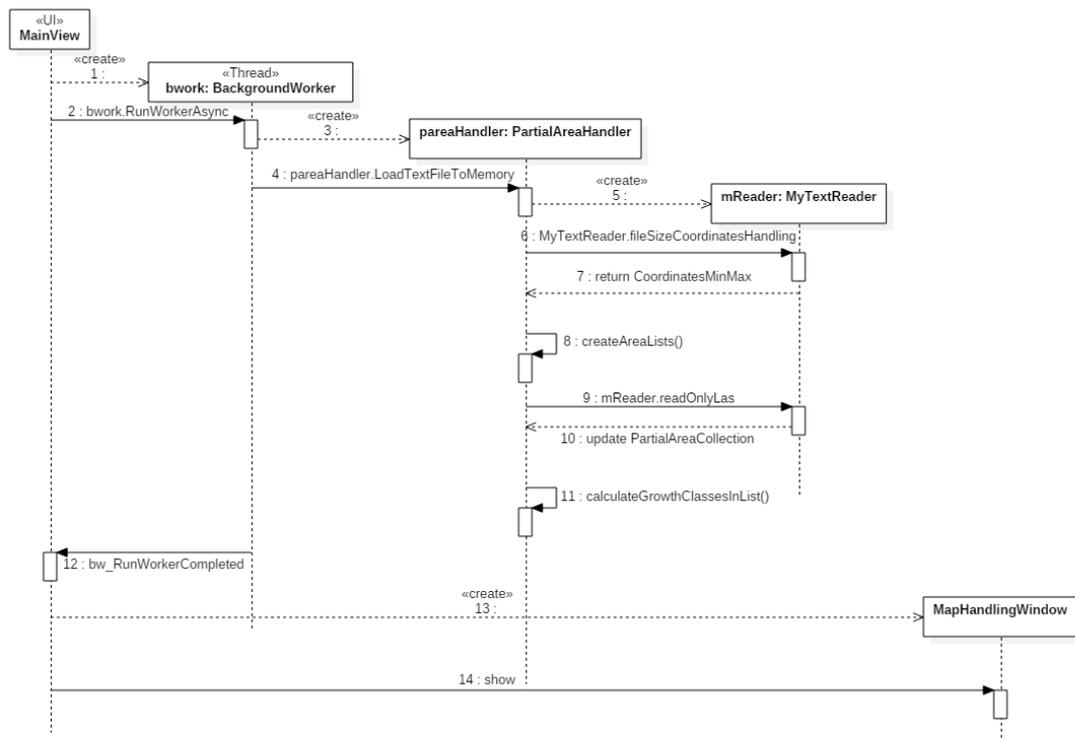
Päämetodi paloittelu alametodeihin helpottaa muutoksien hallintaa ja erillaisia kokeiluja.

### 4.2.2 LAS-tiedoston lukeminen ja käsittely PartialAreaHandlerilla.

Suunnitelma LAS tiedoston käsittelylle oli seuraava:

1. LAS data luetaan läpi ja etsitään pienin X- ja Y-koordinaatti sekä suurin X- ja Y-koordinaatti.
2. Pienimpien ja suurimpien kordinaattien avulla luodaan ruudukkomallinnus jossa ruudun pituus ja leveys on ohjelman alussa annettu arvo.
3. LAS data luetaan rivi kerrallaan, etsitään pisteelle paikka ruudukosta ja asetetaan korkeustieto ruudukkoon.

Kun tiedosto on valittu ohjelma luo erillisen backgroundworker-säikeen. Säikeen käyttö mahdollistaa käyttöliittymän säilyttämisen responsiivisena silloin kuin ohjelma työstää pidempiä laskutoimituksia (ks. kuvio 10.).



Kuvio 10. Sekvenssidiagrammi LAS-txt lukemisesta

Backgroundworker alustaa kappaleessa 3.1.1 esitellyn PartialAreaHandler-luokan joka hoitaa LAS-txt datan käsittelyn.

PartialAreaHandler alustaa tekstinlukijaluokan ja suorittaa:

- Pienimpien ja suurimpien X ja Y arvojen etsimisen
- Ruutulistan luomisen pienimpien ja suurimpien arvojen perusteella.
- Mittauspisteiden jaon ruutuihin.
- Ruutujen sisäisen datan analysoinnin ja kasvuluokan sekä kiirreellisyyden määrittämisen jokaiselle ruudulle.

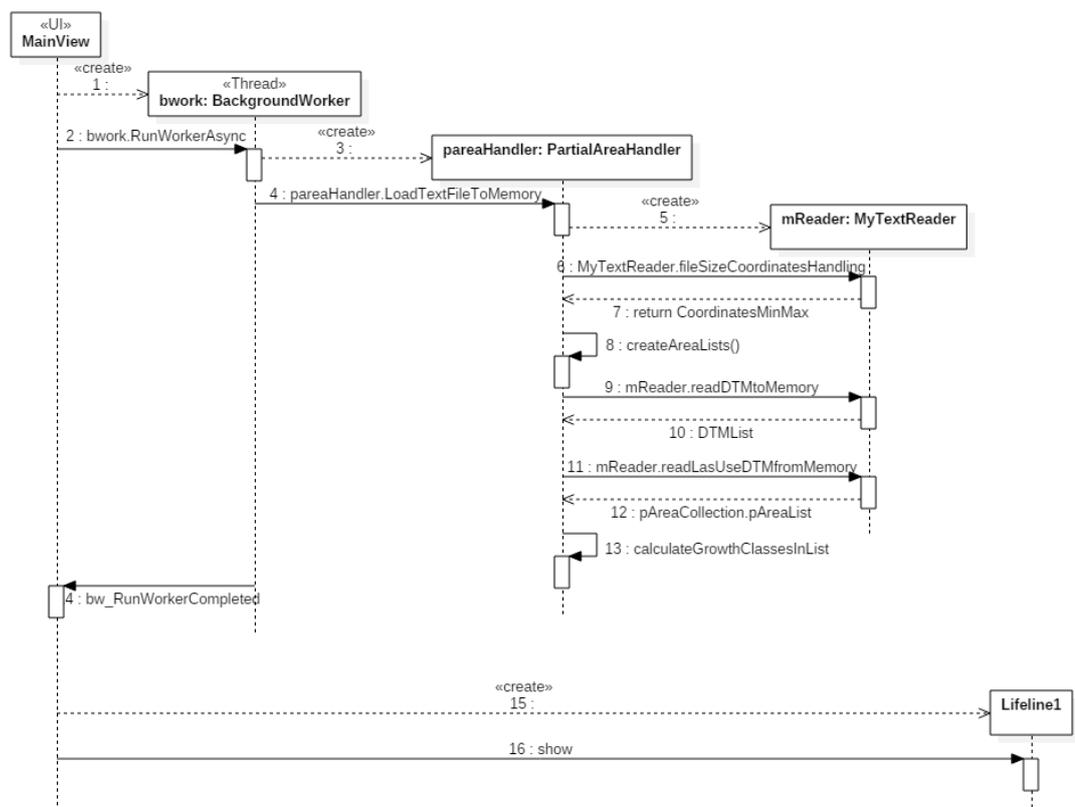
Kun työ on suoritettu, taustasäie palauttaa arvot pääikkunalle joka lähettää arvot eteenpäin seuraavaksi kartanhallintaikkunalle.

Jokaiselle LAS-txt tiedoston riville lasketaan oikea indeksi, joka kertoo mihinkä ruutuun ruutulistassa kyseinen laseripiste kuuluu. Tämän mahdollistaa listan luomiseen käytetyn logiikan säännöllisyys. Menetelmä poistaa tarpeen käydä ruutulistaa läpi jokaisen laseripsteen kohdalla jolloin rivien lisäämiseen kuluva aika nousisi huomattavasti. Varmuuden vuoksi vielä ennen pisteen lisäämistä ruutuun tarkistetaan että pisteen X- ja Y-kordinaatit ovat ruudun minimi ja maksimiarvojen sisällä.

### 4.2.3 LAS-tiedoston ja DTM-maanpintamallin käsittely PartialAreaHandlerilla.

Käytettäessä DTM-dataa joudutaan työskentelyjärjestystä hieman muuttamaan jotta tarvittavat laskutoimitukset saadaan suoritettua LAS- ja DTM-dataa käyttäen.

Erytisesti datan muistiinlataus johti ongelmiin jolloin ohjelmiston käyttövaatimuksia päädyttiin nostamaan. DTM-tiedostoa tiedostoa käytettäessä käytetään samoja luokkia kuin pelkän LAS-datan latauksessa, joskin käyttöön otetaan uusia metodeja ja LAS-lataus suoritetaan eri metodilla (ks. kuvio 11.)



Kuvio 11. Sekvenssidiagrammi LAS ja DTM tiedon käsittelystä

DTM-tiedosto luetaan kuin sisään samaan tyyliin kuin CLM-tiedosto, mutta jokaiselle pisteelle ei tehdä omaa oliota joka säilyttäisi koordinaatitiedot pisteelle. Tämä tehtiin koska jokaisen pisteen koordinaatin lisääminen lisäisi muistinvarausmäärää kolminkertaiseksi ja DTM-pisteiden parittaminen LAS mittauspisteiden kanssa voidaan muutenkin määritellä lyhyellä laskutoimituksella (ks. kuvio 12.).

```

private int getDTMIndex(LaserPoint newpoint,CoordinatesMinMax minMax)
{
    int y = (int)Math.Floor( (newpoint.y - minMax.minY));
    double dy = y * minMax.maxX - minMax.minY;

    int dx = (int)Math.Floor((newpoint.x - minMax.minX));
    int index =Convert.ToInt32(dx + dy);
    return index;
}

```

Kuvio 12. DTM-indeksin laskutapa

Koska DTM-tiedoston lukujärjestys tiedetään, voidaan alueen maksimitietoja vertaamalla kulloisenkin laserpisteen tietoihin laskea mikä arvo DTM-listasta tulee laskea Laserpisteen kanssa. Tämän jälkeen laserpisteen Z-arvosta vähennetään DTM-arvo ja lisätään se ruutulistan oikean ruudun `calculatedVegetationheightList`-listaan. Huomioitavaa on että pelkkää LAS-dataa käytettäessä käyteeään eri listaa siltä varalta, että kaikille laserpisteille ei löytyisikään tarvittavaa maapistetietoa. Maapisteettömät pisteet käsitellään erikseen samalla menetelmällä kuin käsitellessä LAS-dataa ilman DTM-tietoja eli maapisteeksi valitaan alin piste LAS-datan joukosta ja sitä verrataan muihin pisteisiin.

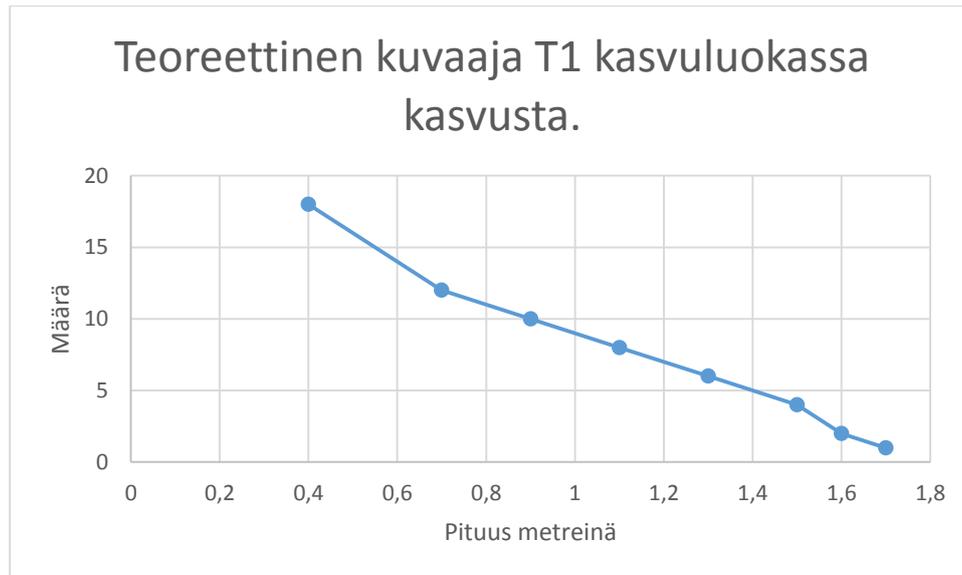
#### 4.2.4 Kehitysluokat ja kiireellisyyden määrittäminen.

Kun kaikki LAS-data ja mahdollinen DTM-data maapistetietoineen on ladattu muistiin, alue ruudutettu halutun kokosiin ruutuihin sekä laserpisteet jaettu ruutuihin, pitää selvittää ruuduille kehitysluokat ja kiireellisyys ruuduille (kehitysluokkien teoria on käyty luvussa 3.2.2.)

Kun käytössä on lentokoneella tehty lentokeilaus ja ruudun kokona käytetään oletusarvoa 8\*8 metriä, jokaiselle ruudulle osuu noin 50 laserpistettä. Nämä laserpisteet kuvastavat kasvuston korkeutta kyseisessä pisteessä, ja näiden pisteiden perusteella olisi kyettävä mallintamaan kehitysluokka ja hoidon kiireellisyyden tarve. Koska pisteitä on rajallinen määrä ja niitten osumisesta alueen kasvustoon ei ole mitään varmuutta on tehtävä tiettyjä oletuksia. Pelkästään LAS-txt dataa käytettäessä maapisteeksi määritetään jokaisen ruudun alin piste z-koordinaatti, jolla lasketaan muille ruudun pisteille delta-z, eli ”kasvupituus”. Tällöinen yhdistelmä jättää paljon mahdollisuuksia huonon arvion tekemiseen.

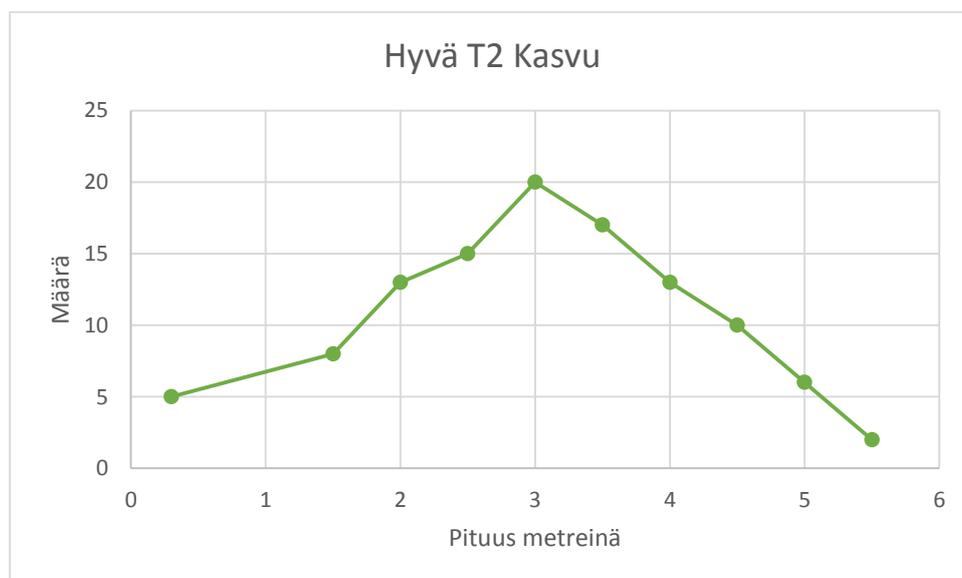
Ensimmäisessä raakaversiossa pisteet jaettiin kasvuluokkiin ja katsottiin mikä luokka saa eniten pisteitä. Jos luokan pistemäärä on yli 50 % ruudun pisteistä se luokiteltiin kiireelliseksi. Tämän tyyppinen arviointi jätti täysin huomiotta rajatapaukset joissa ruudun sisällä olevan kasvuston kasvu on rajavaiheessa kasvuluokkien välillä.

T1 kasvuluokassa metsän kasvu on niin aikaisessa vaiheessa että laserpisteiden muodostama käytä on selkeästi laskeva (ks. kuvio 13.)



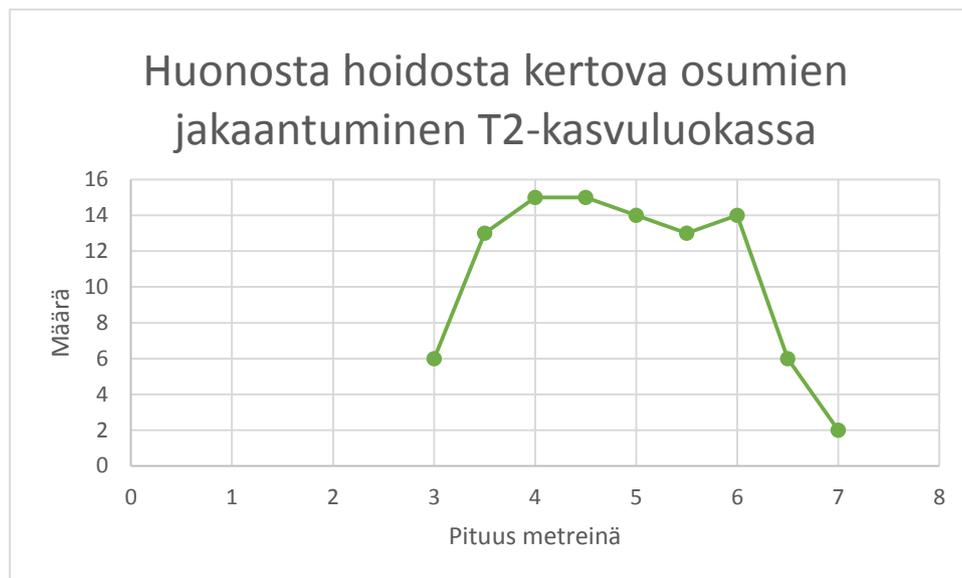
Kuvio 13. Teoreettinen T1 kasvuluokan pistehajonta.

Tämä esimerkkitapaus ei ole vielä hirveästi lähellä 1.6 metrin rajapyykkiä joten harvennuksella ei vielä ole kiire.



Kuvio 14. Teoreettinen T2 Kasvuluokka.

T2 osumat näyttävät hyvin tasaisesti jakaantuneilta. T1 vaiheen jälkeen kasvun optimitapauksissa kuvastaa terävää aaltoa. 0,3m korkeudella olevat osumat voidaan olettaa olevan maapisteosumia jolloin alueen on selvästi hoidettua.



Kuvio 15. Teoreettinen huono T2 kasvuluokka.

Ylläolevasta kuvasta nähdään kuinka mittauspisteet ovat pakkaantuneet T2 kasvuluokan yläpäähän ja osumien määrän jakaantuminen on liian tasaista. Tämä viittaa umpeen kasvamiseen, ja erityisesti maapisteiden olemattomuus viittaa huonoon hoitoon.

Oletetaan että:

1. Metsän ollessa lähellä senhetkisen kasvuluokan ylärajaa sen tulisi olla harvennettu.
2. Useat maapisteet ruudun sisällä viittaavat että alue on hoidettu.
3. Jos ruudun pisteistä 50% on pääkasvuluokassa alue on liian tiheään kasvanut.

Ensimmäiselle vaatimukselle annettiin suurempi painoarvo, ja se nostaa kiireellisyyttä 2 astetta.

Kun LAS-pisteet on jaoteltu ruutuihin, PartialAreaHandler ajaa jokaisen ruudun calculateGrowthClassList-metodin läpi.

Ensin kasvupisteet jaetaan kehitysluokkiin ruudun pääkehitysluokan selvittämiseksi. Pääkehitysluokan arvo tulee pohja-arvoksi muille laskutoimituksille (kasvuluokkien teoria ja pohja-arvot on esitelty kappaleessa 3.2).

Kasvupisteistä valikoidaan kaikki pisteet jotka ovat alle 0.3 metriä korkeita. Tälle alueelle voidaan olettaa osuvan pelkästään maapisteitä. Jos maapisteiden määrä kaikista pisteistä on alle 30 % kasvaa kasvuluokan kiireellisyys yhden.

Seuraavaksi verrataan pääkasvuluokan pisteiden määrää kaikkien ruudun pisteiden määrään. Jos ruudun pisteistä 50 % on pääkasvuluokassa, nousee kiireellisyys jälleen yhdellä.

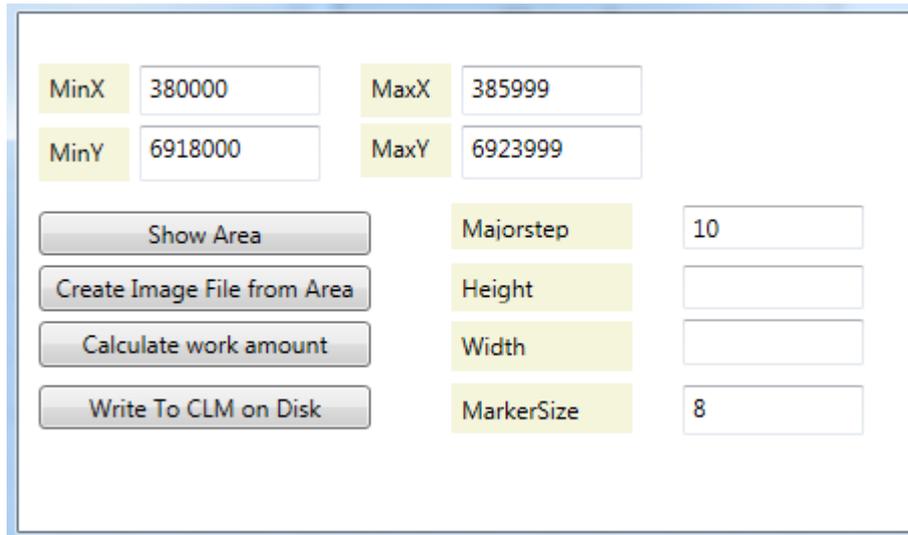
Viimeisenä tarkistetaan kuinka suuri osa pisteistä on pääkasvuluokaksi valitun luokan yläpäässä, eli paljon pisteitä on +- 30 % pääkasvuluokan maksimiarvon molemmin puolin. Tässäkin vaaditaan yli puolet pisteistä.

Esimerkkinä: jos pääkasvuluokaksi on valikoitunut nuori taimikko T1, on pohja-arvo 15. Alueen pisteistä yli 30 % on maapisteitä (+0) mutta pisteistä yli 50 % on pääkasvuluokassa (+1). Pääkasvuluokan yläraja on 1.6 metriä jolloin tarkistetaan kuinka paljon pisteitä on 1.6 +- 0.48 alueella, jolloin nähdään että pisteistä yli 50 % on tällä alueella (+2). Lopullinen arvo ruudulle on siis  $15 + 0 + 1 + 2 = 18$  eli "T1 - perkaustarve KIIREELLINEN". Tämä alue tulisi käsitellä mahdollisimman pian.

Lisäpohdintaa ja kehitysideoita kasvuluokan ja kiireellisyyden määrittelystä löytyy kappaleesta 5.

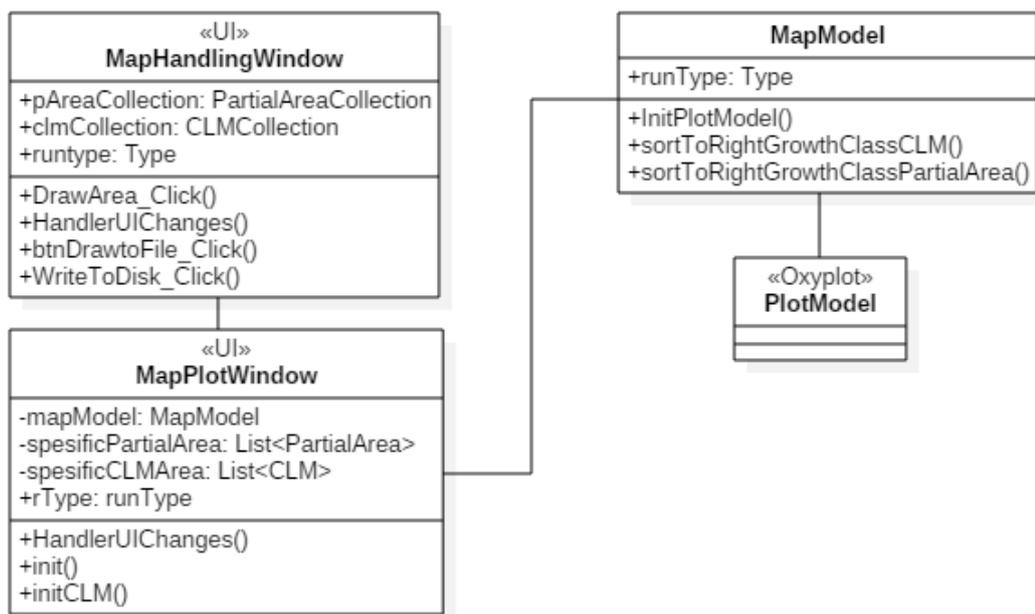
### 4.3 MapHandlingWindow ja MaplotWindow

MapHandlingWindow-hallintaikkunassa käyttäjän on mahdollista esittää laskettu alue näytöllä, piirtää se SVG-tiedostoksi tai luoda lasketuista arvoista CLM-tiedosto myöhempää käyttöä varten. Yläreunassa olevat kordinaattiruudut mahdollistavat alueen rajaamisen, oikeassa reunassa olevat säätöruudut ovat Oxyplot-kirjastolle annettavia arvoja (ks. kuvio 13.).



Kuvio 16. MapHandlingWindow – hallintaikkuna

Koska ohjelmassa ei ole luotuna automaattista skaalausta, on käyttäjälle annettu mahdollisuus määrittää kartan ruutuja kuvaavan merkin koko, kuvan reunassa olevien kordinaatti-akselin askelväli sekä piirrettäessä kuvatiedostoon halutun kuvan koko. MapHandlingWindow-näkymä hallinnoi MapPlotWindow-näkymää joka alustetaan MapModel-luokalla joka sisältää Oxyplot-kirjaston Plotmodel-luokan (ks. kuvio 14.).



Kuvio 17. MapHandlingWindow ja sen aliluokat

### 4.3.1 Luetun LAS-datan muuttamien CLM-tiedostoksi.

Mainwindow-hallintapaneelin suoritettua kaikki tarpeelliset laskutoimitukset ja siirrettyä arvot MaphandlingWindow-näkymälle ei lisämuutoksia tarvita.

PartialAreaCollection lähetetään sellaisenaan MyTextWriter-luokalle joka suorittaa writeCLM-metodin (ks. kuvio 15.).

```

try {
    using (System.IO.StreamWriter xfile =
        new System.IO.StreamWriter(filename, true))
    {
        xfile.WriteLine("ncols " + width);
        xfile.WriteLine("nrows " + height);
        xfile.WriteLine("xllcorner " + pAreaCollection.minMax.minX);
        xfile.WriteLine("yllcorner " + pAreaCollection.minMax.minY);
        xfile.WriteLine("cellsize " + pAreaCollection.areaSize);
        xfile.WriteLine("NODATA_value -9999");
        int i = 0;
        int counter = 0;
        foreach (PartialArea parea in pAreaCollection.pAreaList)
        {
            counter++;
            if (lasty == 0)
            {
                inttoStringlist.Add(parea.growthClassValue);
                lasty = parea.minmax.minY;
            }
            else if (lasty < parea.minmax.minY)
            {
                inttoStringlist.Add(parea.growthClassValue);
                lasty = parea.minmax.minY;
            }
            else
            {
                //write to file, clear list and string.
                string numberString = string.Join(" ", inttoStringlist.ToArray());
                xfile.WriteLine(numberString);
                inttoStringlist.Clear();

                inttoStringlist.Add(parea.growthClassValue);
                lasty = parea.minmax.minY;
            }
        }
        if (inttoStringlist.Count!=0) {
            string numberString = string.Join(" ", inttoStringlist.ToArray());
            xfile.WriteLine(numberString);
        }
    }
}

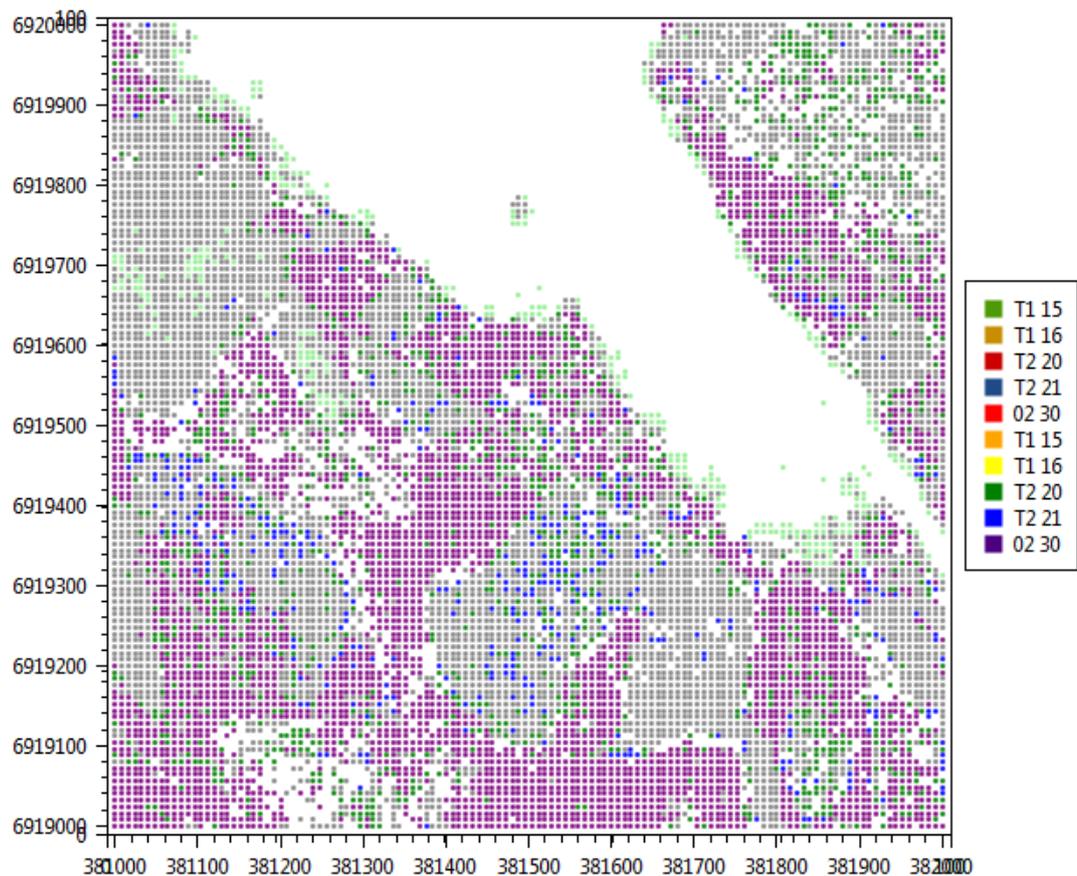
```

Kuvio 18. CLM tiedoston kirjoitus.

Metodi luo uuden tekstitiedoston, joka saa nimensä alkuperäisen LAS-tiedoston pohjalta. Tiedostoon kirjataan CLM-tyyliin pohjatiedot alueesta jolta data on otettu, ja sen jälkeen ruvetaan rivi kerrallaan täyttämään tiedostoa lasketulla kasvu- ja kiireellisyysluokitus datalla.

### 4.3.2 Karttojen piirtäminen

Ohjelmaan haluttiin lisätä karttakuvien esittäminen ja piirto, niinkuin pohjana toimineessa python-sovelluksessa. Tähän käyttöön löytyi kätevä Oxyplot-kirjasto. Oxyplot mahdollistaa WPF-ohjelman sisäisen karttapiirron (ks. kuvio 16) sekä piirtämisen SVG-vektorikuvatiedostoon.



Kuvio 19. Esimerkkikuva 1000 \* 1000 metriä käsitellystä alueesta.

Riippumatta siitä onko kyseessä tiedostoksi tallentaminen vai kartan esittäminen ohjelman sisällä, toimintaperiaate on sama.

Käyttäjä rajaa halutessaan alueen joka piirretään MapHandlingDialog-ikkunassa.

Rajauksen parametrit tarkastetaan että ne ovat maksimissaan karttalehden alueella ja syötetään SharedMethods-luokan getSpesifiArea()-metodiin. Metodi suorittaa LinQ-lausekkeen (ks. kuvio 17.) jolla se hakee kokoelmaluokasta kaikki rajatun alueen sisällä olevat pisteet.

```

public static List<PartialArea> GetSpecificArea(CoordinatesMinMax minMax, ref PartialAreaCollection pareaCollection)
{
    List<PartialArea> resultlist = pareaCollection.pAreaList.Where(c => c.minmax.minX >= minMax.minX &&
        c.minmax.minX <= minMax.maxX &&
        c.minmax.minY >= minMax.minY &&
        c.minmax.minY <= minMax.maxY
    ).ToList();

    return resultlist;
}

```

Kuvio 20. LinQ-lauseke pisteiden noutamiseen kokoelmalistasta.

MaplotWindow luo MapModel-luokan joka vuorostaan alustaa Oxyplotin vaatiman PlotModel-luokan.

Oxyplotilla on omat objektit joita alustamalla ja lisäämällä Plotmodel-luokkaan saadaan muokattua näytettävää kaaviota. Koska projektissa tarvittiin esittää yksittäisiä pisteitä päädyttiin käyttämään ScatterSeries-merkkipistekokoelmaa. Koska kaikki mittapisteet jotka lisätään samaan merkkipistekokoelmaan saavat saman merkausvärin jouduttan luomaan yhtä monta merkkipistekokoelmaa kuin on kasvu- ja kiireellisyysluokkia jotta jokainen saataisiin värikoodattua oikein.

## 5 Pohdinta

Projekti kärsi aikatauluongelmista johtuen huonoista ratkaisuista kehitysvaiheessa. Erityisesti muistinhallintaongelmiin käytetty aika olisi voitu hyödyntää paremmin jos oltaisiin samantien nostettu ohjelman käyttövaatimuksia sen sijaan että olisi yritetty erillaisia kiertotieratkaisuja.

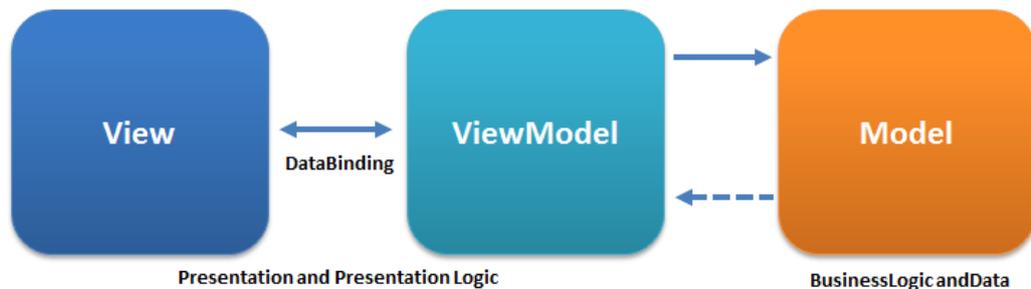
Projekti saavutti kuitenkin vähimmäistavoitteet:

- Käyttöliittymä saatiin kehitettyä.
- Kasvu- ja kiireellisyysluokkien laskemiseen saatiin testattua uusia metodeja.
- LAS data saatiin tiivistettua CLM muotoon.
- Karttapiirto saatiin toteutettua.

Projektissa on kuitenkin jatkokehityspotentiaalia, ja tässä kappaleessa on tarkoitus selittää ongelmat mihin törmättiin sekä miten ne voidaan poistaa jatkossa.

## 5.1 Käyttöliittymän mahdollinen jatkokehitys

Käyttöliittymä ei varsinaisesti ole oikeaoppisen Model-View-View-Model-mallin (ks. kuvio 21.) mukainen.



Kuvio 21. MVVM-malli. 2012.

MVVM:än tarkoitus on erottaa käyttöliittymä varsinaisesta suorittavasta koodista, eli käyttöliittymän puolella ei olisi minkäläistä koodia vaan se lähettäisi käskyn taustakoodille, jossa kaikki suoritukset tehtäisiin (MVVM-artikkeli. N.d.) Vaikka näkymille tehtiin alkeellinen viewmodel-luokka joka säilyttää arvot ja arvot päivittyvät molemminsuuntaisesti näkymiltä viewmodel -luokille ja takaisin, jäi ratkaisu silti vajaavaseksi. Käyttöliittymällä on vielä erikseen nappien painalluksia vastaanottavat koodiosat sekä siellä jonkinverran datatarkistuksia. Näitten osien siirtoa model-luokkaan harkittiin mutta se olisi joko vaatinut monimutkaisen ”patenttiratkaisun” käyttämistä tai kolmannen osapuolen kirjaston käyttöä. Koska ongelma itsessään ei ollut projektin tärkeysjärjestyksessä korkealla kumpikin ratkaisu jätettiin toteuttamatta.

## 5.2 Muistinvaraus

Muistinvaraus osoittautui yhdeksi suurimmista ongelmista projektin aikana. Johtuen LAS- ja DTM-datan muistiinlatausratkaisusta ohjelman piti pystyä lataamaan kerralla iso ja yhtenäinen osa keskusmuistia käyttöön, mikä osoittautui 32-bittisten järjestelmien kohdalla mahdottomaksi johtuen arkkitehtuuriratkaisuista. Samanlaisiin ongelmiin törmäsi samaan aiheeseen liityvässä opinnäytetyössään Hannu Siren. Hän ratkaisi paloittelemalla datan useampaan osaan, toisin kuin tässä työssä tyydyttiin nostamaan laitevaatimuksia ohjelman ajamiseen (Siren. 2015. 35.).

### 5.2.1 GrowthClass-luokka.

GrowthClass-luokan tarkoitus oli pitää tiedossa mittauspisteiden jakaantuminen kasvuluokkiin jokaisen karttaruudun sisällä. Koska luokka sisälsi string-muuttujan, 4 kappaletta double-muuttujia ja listan kaikista ruudun sisälle tulevista mittauspisteistä doublena sen aiheuttama muistinvaraus kasvoi naurettaviin mittasuhteisiin sen tarpeellisuuteen nähden (ks. kuvio 19. ja 20.).

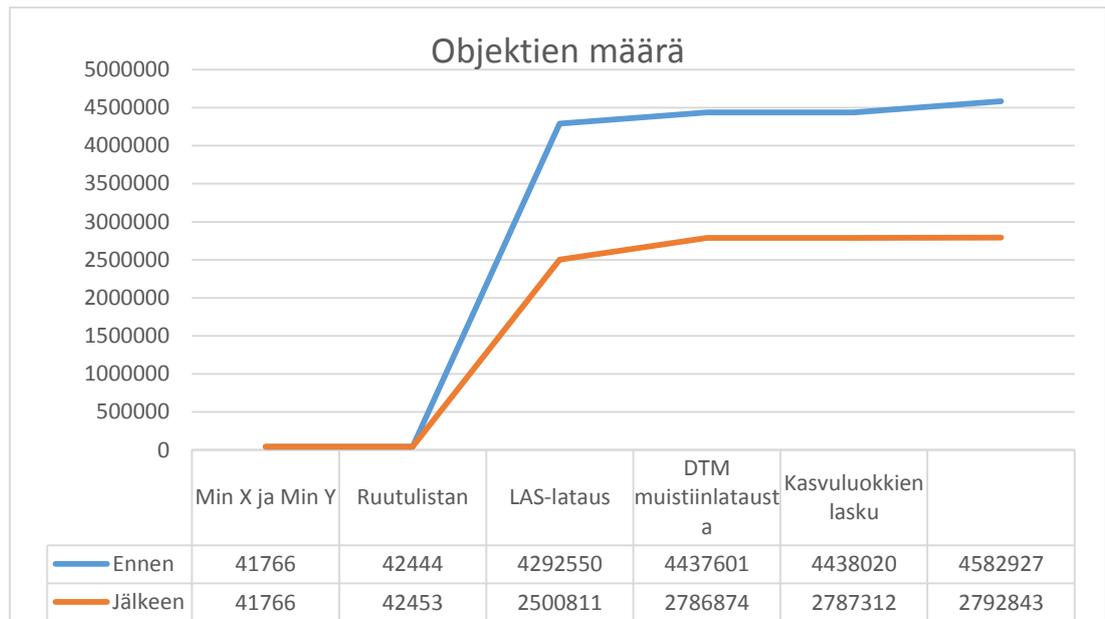
Object Type	Count ▼	Size (Bytes)	Inclusive Size (Bytes)
MyLibrary.GrowthClass	999,999	47,999,952	63,999,960
List<Double>	999,999	225,276,732	225,276,732
MyLibrary.CoordinatesMinMax	562,501	22,500,040	22,500,040
List<MyLibrary.GrowthClass>	562,500	29,250,000	93,249,960
MyLibrary.PartialArea	562,500	22,500,000	347,526,684
RuntimeType	7,457	209,672	209,672

### Kuvio 22. Growthclass-luokan muistinvaraus ennen

Object Type	Count	Size (Bytes) ▼	Inclusive Size (Bytes)
List<Double>	999,999	243,589,908	243,589,908
MyLibrary.CoordinatesMinMax	562,501	22,500,040	22,500,040
MyLibrary.PartialArea	562,500	22,500,000	288,589,908
List<MyLibrary.PartialArea>	1	4,194,344	292,784,252
RuntimeType+RuntimeTypeCache	50	310,324	317,912

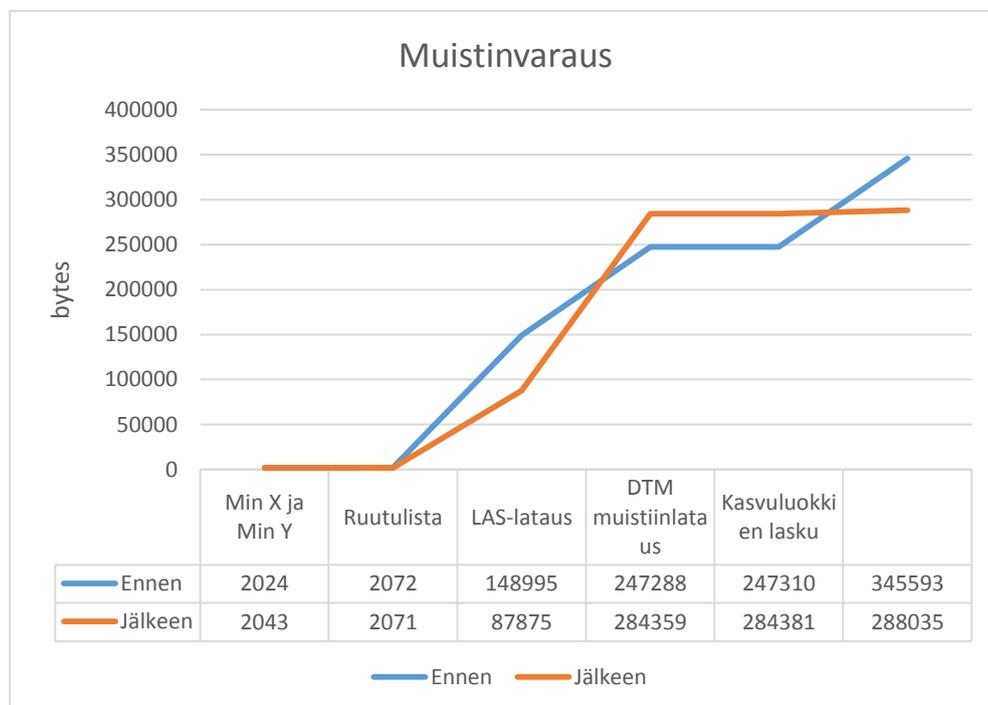
### Kuvio 23. muistinvaraus Growthclass-luokan poistamisen jälkeen jälkeen

Ruutulistan luominen aiheuttaa piikin, koska GrowthClass-luokkia oli 4 kappaletta jokaisessa ruudussa. Tämä lisäsi objektien määrää tuntuvasti (ks. kuvio 21.).



Kuvio 24. GrowthClass luokan objektimäärävertailu

Kun kasvuluokkia ruvettiin laskemaan, GrowthClass-luokan monistaminen aiheutti muistinvarauspiikin ajon lopussa (ks. kuvio 22.).



Kuvio 25. Growthclass luokan muistinvarausvertailu

### 5.2.2 Oxyplotin ongelmat.

Oxyplotin käyttö varsinkin isompien alueiden piirtämisessä vaatii paljon keskusmuistia. Keskumuistiin käyttöön vaikuttavat muunmuassa kuvan ruudutuksen määrä, reuna-akselien väliviivojen määrä, onko merkkikuvaaja ”täytetty” (ympyrä, neliö, kolmio tai timantti) vai ei (plus-merkki, risti, tähti). Näistä syistä alunperin merkinä käytety ympyrä vaihdettiin neliöksi.

Keskumuistin varaus nousi pelkästään piirrosta 960 megatavuun, ja poistamalla ruudutus putosi muistinkulutus 619 megatavuun.

Vaikka Oxyplot on hyvä avoimen lähdekoodin kirjastoksi, siitä löytyi yllättäviä puutteita. Kirjaston asentaminen Nuget-pakentinhallinan kautta ja toimintaan saattaminen yhteistyössä WPF-alustan kanssa aiheutti liian paljon hukkaan heitettyä aikaa, mikä taas kostautui ajan loppumisena loppukehityksestä. Myöskään manuaalin lupaama PNG-kuvatiedostoon piirtäminen ei saatu toimimaan.

## 5.3 Tarkkuus

Lasermittaus ei anna täsmällisiä osunapisteitä maastosta. Täten ei ole mitään takeita siitä että pisteet ovat osuneet tasaisesti puustoon, joten kaikki laskelmat kehitysluokista ja kiireellisyydestä ovat arvioita. Erityisesti kuusipuiden muoto lisää osumispinta-alaa mutta jättää realistisen korkeuden arvailun varaan.

Metsä ei jätä vapaata tilaa käyttämättä, joten myöskään maahan osuminen ei ole varmaa. Tämä tekee maapisteen valinnasta epävarmaa, jos DTM-maastopintamallitiedostoa ei ole käytössä.

Tarkemman datan saamiseksi maastoon voidaan viedä GPS-asema. GPS asemaa säilytetään useampi maastossa vähintään 30 minuuttia, mutta yleensä jopa puoli päivää, että se kerkeää ottaa yhteyden mahdollisimman moneen satelliittiin. Näin asema kerkeää ottaa yhteyden useampaan satelliittiin ja laskea tarkan sijaintinsa. Tämän datan yhdistäminen keilausmittauksen dataan saadaan tarkempia mittaustuloksia.

Lentokeilauksen mittauspisteen tallentaminen on monivaiheinen. Lentokone tallentaa gps-sijaintinsa, lähettää lasersignaalin, tarkistaa sijainnin uudelleen signaalin palattua,

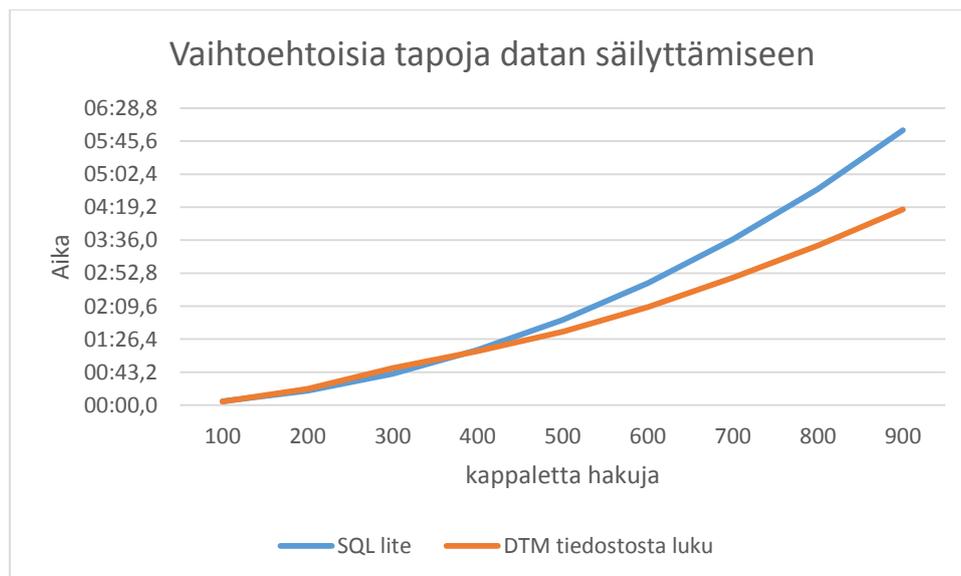
ja laskee tarkan gps-sijainnin näiden kahden tuloksen erotuksesta. Virhe missä tahansa toimenpiteessä vaikuttaa mittauspisteiden tarkkuuteen.

Jos DTM tiedostoa ei ole, vierekkäisten ruutujen maapisteitä voitaisiin vertailla keskenään maapisteen varmistamiseksi. Vastaavasti tällöin maaston mäkyisyys voisi sekoittaa laskelmat.

Koska maasto ruudutetaan viivasuorasti ja vain sen sisältöä käytetään Kasvuluokan ja kiireellisuuden määrittämiseen pelkillä keskiarvoilla ”pelaaminen” aiheuttaa väistämättä virhearvioita. Tätä voitaisiin vähentää vertaamalla vierekkäisiä ruutuja tai vain 1-2 metrin ruudun ulkopuolelle jääviä mittauspisteitä ruudun tuloksiin. Näin paljon suurempaa vertailua käyttäen voitaisiin parantaa ruutujen luokituksen laatua. Tämä tosin vaatisi paljon hienovaraisempaa laskemista ja vertailua pisteiden välillä.

#### 5.4 Muistinhallinnan kokeilut.

Koska 32-bittinen käyttöjärjestelmä ei antanut tarpeeksi tilaa muistinvaraukselle, projektin aikana testattiin datan säilyttämistä tietokannassa laskelmien aikana. Koska ongelmana oli erityisesti DTM-data, testattiin myös tapaa hakea jokaiselle laserpisteelle tarvittava DTM-luku vasta laskuvaiheessa. Ongelmaksi nousi liian pitkät luku- ja kirjoitusajat (ks. kuvio 23.).



Kuvio 26. Datan nouto SQLite tietokannasta ja DTM -tiedostosta.

Muistiin ladattuna saman tuhannen laskutoimituksen käsittely onnistui alle yhdessä millisekunnissa.

## 5.5 LAS-datan lukeminen.

Valitettavasti projektin kehitystyössä loppui aika kesken, joten joitakin ominaisuuksia jäi implementoimatta. Suurimmaksi puutteeksi jäi alkuperäisen LAS-datan suora käyttö. Tätä ongelmaa yritettiin saada ratkaistuksi käyttämällä kolmannen osapuolen kirjastoja, kuten vapaan lähdekoodin LASlib ja LibLas kirjastoja.

Liblas on ASPRS kehittämä C++ pohjainen kirjasto jota käytetään muunmuassa OSGeo4W ohjelmistopakettissa. Kirjaston käytössä törmättiin DLL-tiedostojen toimimattomuusongelmiin, ja kun kirjastoa pyrittiin kääntämään avoimesta koodista C++ käännösvirheiden määrä kasvoi liian suureksi selvitettäväksi (Liblas. 2015.).

Laslib on ravidlasso GmbH:n LAsTools ohjelmistopakettin pohjalla toimiva C++ kirjasto. Kirjaston kääntämisessä törmättiin samanlaisiin ongelmiin kuten liblas kirjastossa, joten tämänkään implementointi ei onnistunut. (Lastools. N.d)

Järkevin ratkaisu olisi ollut kirjoittaa kokonaan oma lukurajapinta kuten Hannu Siren teki omassa opinnäytetyössään (Siren. 2015.). Näin säästytään turhilta riippuvuuksilta kolmannen osapuolen virityksiin.

Jos taas haluttaisiin hyödyntää jompaa kumpaa aiemmin mainituista kirjastoista ja C++-projektit saadaan kääntymään, implementointi C#-ohjelmaan onnistuisi lisäämällä C++-projekti C#-projektin sisältävään ratkaisuarkeeseen ja rakentamalla väliin Cli-Wrapper-projekti jota kautta C# pystyisi hyödyntämään C++ ohjelman luokkia ja metodeja (CLI esimerkki. 2014.)

## 5.6 Työmääräarviot ja tulevaisuus.

Ohjelman alkuperäiseen suunnitelmaan oli kirjattu mahdollisuus laskea työmääräarvioita lasketusta kasvu- ja kiireellisyysluokkadatasta. Tämä ominaisuus jäi ajanpuutteen johdosta suunnitteluasteelle, mutta ominaisuuden toimintaperiaate selvitettiin.

Halutulta rajatulta alueelta poimitaan kaikki kasvu- ja kiireellisyysluokkien arvot. Yksinkertaistamisen vuoksi nämä alueet voidaan laskea luokittain yhteen, jotta saadaan jokaisen luokan määrä hehtaareissa. Aiempaa metsänhoitotyökokemusta hyödyntäen voidaan jokainen luokka pisteyttää, kuinka paljon kukin alue tarvitsee työtunteja per 100 neliometriä/1000 neliometriä/mikä tahansa haluttu mittasuhte. Näillä arvoilla pystytään helposti laskemaan kuinka paljon työtä menee rajatun alueen hoitamiseen. Tällainen tehokas arviointi on pääsyy LAS-datan tiivistämiseen CLM-muotoon.

## Lähteet

- CLI esimerkki. 12.10.2014. Using C++ in C# by Example by Dorodnic. Viitattu 16.5.2016. <http://www.dorodnic.com/blog/2014/12/10/calling-cpp-by-example/>
- Digital Preservation. 29.1.2015. LAS 1.4 esittely. Viitattu 14.5.2016. <http://www.digitalpreservation.gov/formats/fdd/fdd000418.shtml>
- Digital terrain model. N.d. wikipedia artikkeli. viitattu 10.5.2016 [https://en.wikipedia.org/wiki/Digital\\_terrain\\_model](https://en.wikipedia.org/wiki/Digital_terrain_model)
- Lastools. N.d. LAStools esittelysivusto. <http://www.cs.unc.edu/~isenburg/lastools/>
- LASZip documentation. N.d. LASzip esittelydokumentaatio. Viitattu 14.5. <http://www.laszip.org/>
- Liblas. 6.11.2015. Wikipedia – Vapaa tietosanakirja. Viitattu 16.5.2016. <https://en.wikipedia.org/wiki/LibLAS>
- Luke. N.d. Luonnonvarakeskuksen esittelysivu. Viitattu 14.5.2016. <https://www.luke.fi/luke/>
- MVVM-malli. 2012. Wikipedia – Vapaa tietosanakirja. Viitattu 18.5.2016.
- MVVM-artikkeli. N.d. Wikipedia – vapaa tietosanakirja. Viitattu 17.5.2016. <https://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93viewmodel>
- Maanmittauslaitoksen laserkeilausaineisto. N.d. Laserkeilausaineiston esittelysivut. Viitattu 10.5.2016. <http://www.maanmittauslaitos.fi/digituotteet/laserkeilausaineisto>
- Maanmittauslaitoksen vuositilastoja. 2014. dokumentin sivu 15. Viitattu 15.5.2016. [http://www.maanmittauslaitos.fi/sites/default/files/mml\\_vuositilasto\\_2014.pdf](http://www.maanmittauslaitos.fi/sites/default/files/mml_vuositilasto_2014.pdf)
- Metsänomistajat Haapavesi-Kärsämäki ja Siikalakeus-lehti, 2016, sivu 26. Viitattu 18.5.2016. [http://www.e-julkaisu.fi/metsanhoitoyhdistys/haapavesi-karsamaki\\_siikalakeus/1-2016/mobile.html#&gid=1&pid=26](http://www.e-julkaisu.fi/metsanhoitoyhdistys/haapavesi-karsamaki_siikalakeus/1-2016/mobile.html#&gid=1&pid=26)
- Metsäkeskus. N.d. Metsäkeskuksen esittelysivut. viitattu 10.5.2016 <http://www.metsakeskus.fi/organisaatio>
- Metsaan. N.d. Metsaan.fi palvelun esittely. Viitattu 17.5.2016. <http://www.metsaan.fi/metsatiedot>
- Metla. N.d. Tutkimusmetsät. Viitattu 10.5.2016 <http://www.metla.fi/metsat/>
- Microsoft, N.d, .Net Dokumentaatio. BackgroundWorker, viitattu 5.10.2016. <https://msdn.microsoft.com/en-us/library/system.componentmodel.backgroundworker%28v-vs.110%29.aspx>
- Oxyplot. N.d. Oxyplot kotisivut. Viitattu 14.5.2016 <http://www.oxyplot.org/>

Oxyplot license. N.d. Oxyplot git lisenssi. Viitattu 14.5.2016  
<https://github.com/oxyplot/oxyplot/blob/develop/LICENSE>

Puuston kehitysluokat. Julkaistu 08.04.2013. Suomen Metsäurheiluliiton tiedosto. Viitattu 10.5.2016. <http://www.riihimaki.fi/wp-content/uploads/sites/3/2015/01/Mets%C3%A4suunnitelma-Puuston-kehitysluokat.pdf>

Siren, H. 2015. Metsän kehitysluokkien tunnistus lentokonekeilausaineistojen perusteella Python-ohjelmointina. <https://www.theseus.fi/handle/10024/98022>

Suomen Metsäurheiluliitto, N.d. Dokumentti puuston kehitysluokista, viitattu 10.5.2016.  
[http://www.helsinki.fi/hyvtiala/smul/saannot/dokumentit/saannot\\_kehitysluokat\\_w\\_atermark.pdf](http://www.helsinki.fi/hyvtiala/smul/saannot/dokumentit/saannot_kehitysluokat_w_atermark.pdf)

Yrittävä maaseutu. 1/2016. Artikkeliverkkolehdessä. Viitattu 10.5.2016.  
[https://issuu.com/maakaista.fi/docs/yritta\\_va\\_maaseutu\\_2016/1](https://issuu.com/maakaista.fi/docs/yritta_va_maaseutu_2016/1)

Wikipedia DTM ja DSM. 16.10.2015. Wikipedia – vapaa tietosanakirja. Viitattu 18.5.2016. [https://en.wikipedia.org/wiki/File:DTM\\_DSM.svg](https://en.wikipedia.org/wiki/File:DTM_DSM.svg)