



Martin Totti Zeus Lohman

# QGIS-lisäosan laatiminen Maanmittauslaitoksen historiallisten ortokuvien latausta varten

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Maanmittaustekniikka

Insinöörityö

5.5.2024

# Tiivistelmä

Tekijä:	Martin Totti Zeus Lohman
Otsikko:	QGIS-lisäosan laatiminen Maanmittauslaitoksen historiallisten ortokuvien latausta varten
Sivumäärä:	57 sivua + 1 liite
Aika:	5.5.2024
Tutkinto:	Insinööri (AMK)
Tutkinto-ohjelma:	Maanmittaustekniikka
Ohjaajat:	Lehtori Matias Ingman

---

Tässä insinööriyössä laadittiin lisäosa QGIS-paikkatieto-ohjelmistolle, jonka avulla Maanmittauslaitoksen historiallisia ortokuvia voi ladata tiedostomuodossa. Aineisto on avointa dataa, mutta sille ei ole toteutettu latauspalvelua. Sisällöllisesti lisäosan lataamat ortokuvat ovat samoja, joita voi katsoa Paikkatietokannan Historialliset ilmakuvat -osiossa. Vanhimmat ortokuvat ovat 1930-luvulta.

Lisäosa tehtiin Python-ohjelmointikieltä ja PyQGIS-kirjastoa hyödyntäen. Ortokuvat ladataan Maanmittauslaitoksen WCS-rajapinnasta. Ennen lataamista saatavilla oleva aineisto tarkistetaan Maanmittauslaitoksen erillisestä OGC API: Features -rajapinnasta.

Insinööriyöraportissa käsiteltiin lyhyesti saatavilla olevia aineistoja, mutta siinä keskityttiin pääosin lisäosan taustalla oleviin teknologioihin, lisäosan käytännön toteutukseen ja käyttöön. Insinööriyössä laadittu lisäosa on raportin liitteenä.

Laadittu lisäosa laajentaa Maanmittauslaitoksen historiallisten ortokuvien käyttämällisyyksiä nykyisestä. Ladatut tiedostot ovat alkuperäislaatuisia ja georeferoituja, mikä mahdollistaa niiden analysoinnin paikkatieto-ohjelmistoissa. Paikkatieto-ohjelmistoissa tehtävä analysointi on usein edellytys tietosisällön tehokkaalle hyödyntämiselle.

Aineiston parempi hyödynnettävyys voi olla avuksi esimerkiksi luonnonympäristöä, rakennettua ympäristöä ja maa- tai metsätaloutta tutkiville.

Avainsanat: QGIS, PyQGIS, Maanmittauslaitos, historialliset ilmakuvat, WCS, OGC API: Features

---

Tämän opinnäytetyön alkuperä on tarkastettu Turnitin Originality Check -ohjelmalla.

## Abstract

Author: Martin Totti Zeus Lohman  
Title: Creating a QGIS Plugin for Downloading Historical Ortho-photos from National Land Survey of Finland  
Number of Pages: 57 pages + 1 appendix  
Date: 5 May 2024

Degree: Bachelor of Engineering  
Degree Programme: Land Surveying  
Supervisors: Matias Ingman, Senior Lecturer

---

The aim of the final year project was to create a plugin for the QGIS geographic information system whose use would allow the download of historical orthophotos of the National Land Survey of Finland. The orthophotos are open data, but no download service had been provided.

The plugin was developed using the Python programming language and PyQGIS framework. The plugin allowed orthophotos to be saved as georeferenced rasters of original quality, enabling their direct use in geographic information systems. With the plugin, the orthophotos are downloaded from a WCS server, but a separate OGC API: Features server is first queried to determine the years from which orthophotos are available.

The final year project made it possible to easily access, analyse and compare nearly a century's worth of aerial imagery using GIS software. This is valuable to multiple fields of study. Especially fields related to natural environments, built environments, agriculture, or forestry may find the created plugin helpful.

Keywords: QGIS, PyQGIS, historical aerial photos, National Land Survey of Finland, WCS, OGC API: Features

# Sisällys

## Lyhenteet

1	Johdanto	1
2	Ilmakuvaus	3
2.1	Yleistä	3
2.2	Ilmakuvaus Suomessa	5
2.3	Maanmittauslaitoksen ilmakuva-arkisto ja saatavuus	6
2.4	Historiallisen ortokuva-aineiston käyttökohteet	7
3	Paikkatieto ja rajapinnat	11
3.1	Paikkatieto	11
3.2	Paikkatietorajapinnat	13
3.3	WCS-rajapintastandardi	14
3.4	OGC API: Features -rajapintastandardi	18
4	Ohjelmistot	23
4.1	QGIS-paikkatieto-ohjelmisto	23
4.2	Python-ohjelmointikieli	23
4.3	PyQGIS-kirjasto	24
5	QGIS-lisäosan toteutus	25
5.1	Tuontilausekkeet	26
5.2	Pääluokan alustus	26
5.3	Lähtötiedot	28
5.4	Aluevalinta	35
5.5	Lataus	39
5.6	Siivous	45
6	QGIS-lisäosan tiedostot, jakelu, asennus ja käyttö	47
6.1	Tiedostot	47
6.2	Jakelu	48
6.3	Asennus	49
6.4	Käyttö	49

7	Yhteenveto	52
	Lähteet	54
	Liite 1: Toteutettu QGIS-lisäosa	

## Lyhenteet

- API *Application Programming Interface*. Joukko määriteltyjä sääntöjä, jotka mahdollistavat eri ohjelmistojen keskinäisen kommunikoinnin.
- GeoJSON *Geographic JavaScript Object Notation*. Vektorimuotoisen paikkatiedon tallennukseen ja välitykseen kehitetty avoin tiedostostandardi. Käytetään ennen kaikkea paikkatiedon jakamiseen internetin kautta.
- GeoTIFF *Geographic Tag Image File Format*. Rasterimuotoisen paikkatiedon avoin tiedostostandardi, joka mahdollistaa sijaintitiedon sisällyttämisen TIFF-tiedostoihin. Käytetään ennen kaikkea alkuperäislaatuisen rasteriaineiston tallennukseen ja jakeluun.
- HTTP *Hypertext Transfer Protocol*. Protokolla, joka määrittelee, miten tietoa lähetetään ja vastaanotetaan internetissä.
- OGC *Open Geospatial Consortium*. Kansainvälinen järjestö, joka kehittää avoimia standardeja paikkatiedon hallintaan ja jakeluun.
- WCS *Web Coverage Service*. Rajapintastandardi, joka mahdollistaa rasterimuotoisen paikkatiedon jakamisen alkuperäismuodossaan internetin kautta.
- WMS *Web Map Service*. Rajapintastandardi, joka mahdollistaa paikkatiedon jakamisen pakatussa rasterimuodossa internetin kautta, ensisijaisesti visualisointitarkoituksiin.

# 1 Johdanto

Maanmittauslaitoksen ilmakeku-arkisto sisältää runsaasti aineistoa 1930-luvulta lähtien. Arkistoa digitoidaan ja digitoiduista ilmakekuvista laaditaan mittakaavakorrekkeja ja vääristymistä korjattuja ortokuvia. Historiallista ortokuva-aineistoa voidaan hyödyntää muiden muassa ympäristötutkimuksessa, kaupunkisuunnittelussa sekä maanmittaustoimituksissa.

Maanmittauslaitos on toteuttanut verkkoselainpohjaisen palvelun, jonka kautta aineistoa voi katsella ilmaiseksi. Monet käyttötarkoituksista edellyttävät kuitenkin, että aineisto on hyödynnettävissä paikkatieto-ohjelmistoista käsin.

Paikkatieto-ohjelmistoihin aineisto on mahdollista saada, mutta tämä edellyttää lisenssiä. Lisenssin vuosittainen hinta alkaa noin 300 eurosta ja voi käyttömäärästä riippuen nousta tuhansiin euroihin. Yksityishenkilöiden ja pienempien toimijoiden mahdollisuudet hyödyntää historiallista ortokuva-aineistoa ovat siksi rajoitettuja.

Maanmittauslaitos julkaisi vuonna 2023 rajapinnan, jonka kautta aineistoa on mahdollista ladata veloitusetta tiedostomuodossa. Rajapintastandardi on kuitenkin melko tuntematon, eikä esimerkiksi QGIS-paikkatieto-ohjelmisto tue sitä. Myös faktisesti saatavilla olevan aineiston selvittäminen muodostaa lisähaasteen.

Tässä insinööriyössä laaditaan Python-ohjelmointikieltä hyödyntäen QGIS:lle lisäosa, jonka avulla Maanmittauslaitoksen historiallisia ortokuvia voidaan ladata tiedostomuodossa. Tallennettuja tiedostoja on mahdollista käyttää paikkatieto-ohjelmistoista käsin. Lisäosa laajentaa näin ollen mahdollisuuksia hyödyntää kulttuurihistoriallisesti arvokasta ortokuva-aineistoa.

Laadittu QGIS-lisäosa on insinööriyön liitteenä 1, ja luvuissa 6.3 ja 6.4 ohjeistetaan sen asennukseen ja käyttöön.

Insinööriyöraportissa keskitytään lisäosan toimintaan sekä taustalla oleviin rajapintastandardeihin. Ilmakuvausta sekä käytettyjä ohjelmistoja käsitellään pintapuolisemmin.

Työ on tehty insinööriyön tekijän omasta kiinnostuksesta, eikä taustalla ole toimeksiantoa.



## 2 Ilmakuvaus

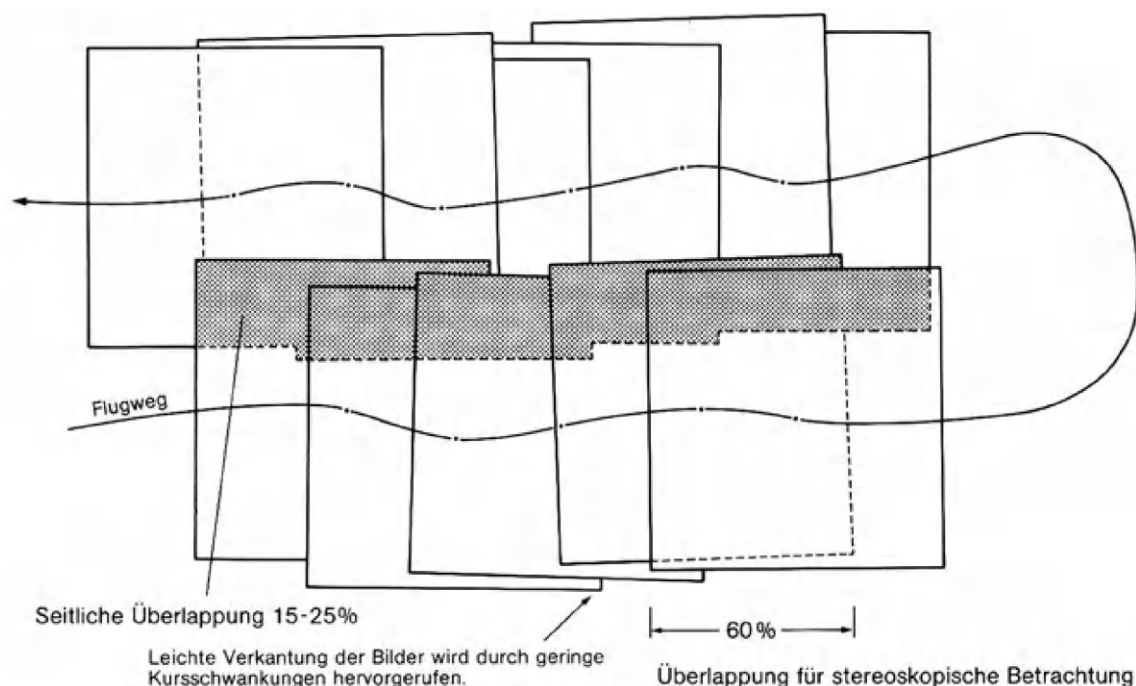
### 2.1 Yleistä

Laajasti määriteltynä ilmakuvia ovat kaikki ilmasta käsin otetut valokuvat. Maanmittausyhteyksissä ymmärretään termi kuitenkin suppeammin, ja esimerkiksi Maanmittauslaitoksen käyttämä määritelmä ilmakuvausta on ”kartoituskäyttöön soveltuva pystykuva” [1]. Kyseistä määritelmää käytetään myös tässä insinööri-työraportissa.

Ilmakuvat otetaan ensisijaisesti lentokoneista ilmakuvaukseen suunnitelluilla erityislaitteistolla. Laajempien alueiden ilmakuvausta suoritetaan lähinnä julkisten toimijoiden, kuten valtion ja kaupunkien, toimeksiannosta. Viime vuosikymmeninä dronet ovat kuitenkin mahdollistaneet pienten alueiden, kuten työmaiden, ilmakuvauksen myös ilman erityislaitteistoa.

Systemaattinen ilmakuvakartoitus sai alkunsa ensimmäisen maailmansodan myötä, ja sodan jälkeen ilmakuvausta alettiin hyödyntää siviilitarkoituksiin. Ilmakuvaus mahdollisti laajojen alueiden verrattain nopean kartoituksen, maasta käsin suoritettuihin mittauksiin verrattuna. Ilmakuvaus on kauan ollut keskeisin laajojen alueiden kartoitusmenetelmä. [2, s. 428.]

Kuvassa 1 on esimerkki ilmakuvauksen käytännön toteutuksesta.

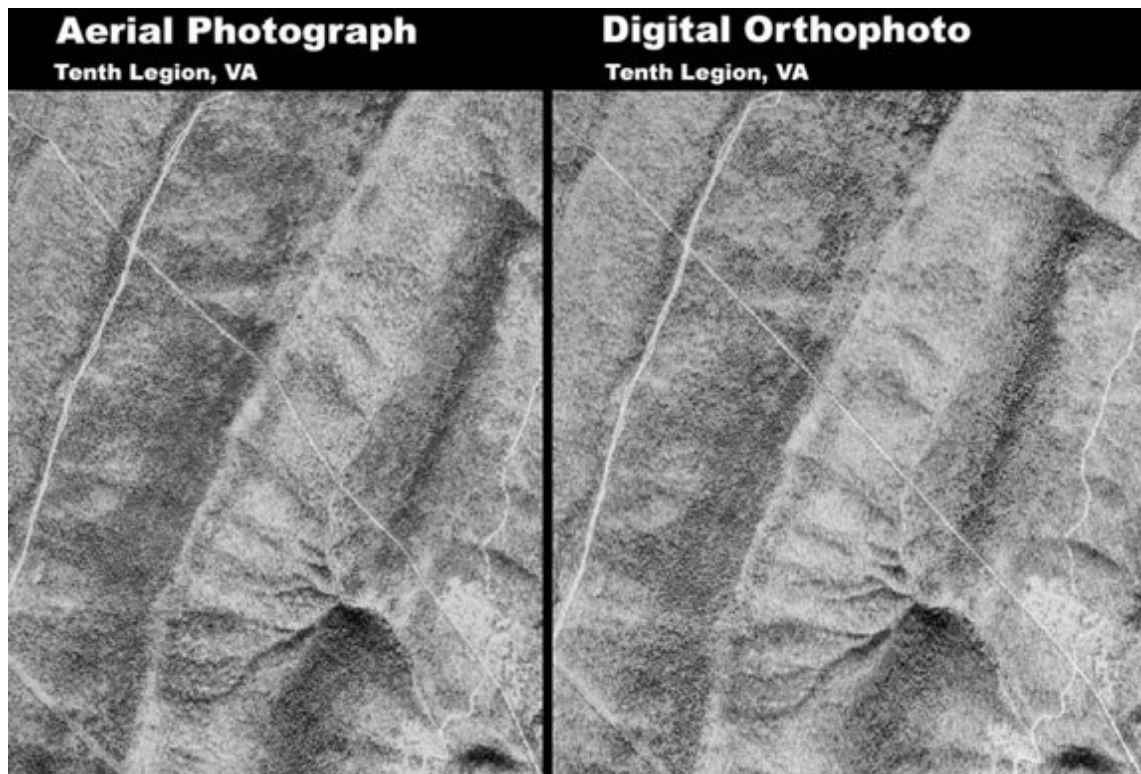


Kuva 1. Viiva osoittaa lentokoneen lentoreitin ilmakuvatussa. Suorakulmiot ovat yksittäisiä ilmakuvia. Kuvasta ilmenee, kuinka tavoitteena on melko suuri päällekkäisyys kuvien välillä. Jokaisesta ilmakuvasta pyritään hyödyntämään ensisijaisesti keskiosa, jossa optiset vääristymät ovat pienimmät. [3, s. 83.]

Ilmakuvauksessa pyritään hallitsemaan ja tuntemaan kuviin vaikuttavat tekijät mahdollisimman hyvin. Tällaisia ovat muiden muassa lentokorkeus, tarkka kuvakulma ja linssin aiheuttamat vääristymät. Näiden tunteminen mahdollistaa kuvien jälkikäsitteilyn ja tietosisällön maksimaalisen hyödyntämisen. Tavoitteena on lähes aina yhdistää monta toisiaan osittain peittävää ilmakuvaa yhtenäiseksi ja kattavaksi peitteeksi kuvatuista alueista.

Raaoissa ilmakuvissa on aina maaston korkeuseroista johtuvaa mittakaavavirhettä. Korkealla ja näin ollen kameraa lähempänä sijaitsevat kohteet vaikuttavat kuvissa suuremmilta kuin matalalla ja kauempana kamerasta sijaitsevat.

Hyödyntämällä tietoa maanpinnan muodosta on mahdollista korjata kyseiset vääristymät. Tuloksena saadaan kauttaaltaan mittakaavakorrekti, ulkoiseen koordinaatistoon sidottu, karttaa vastaava kuvatuote. Tähän viitataan ortokuvana [4]. Kuvasta 2 ilmenee raa'an ilmakuvan ja ortoilmakuvan ero.



Kuva 2. Kuvissa näkyvä vasemmasta yläkulmasta oikeaan alakulmaan johtava viiva on suuri putki. Todellisuudessa se kulkee suoraan. Maaston korkeuserojen takia vasemmalla olevassa raa'assa ilmakuvassa ilmenee kuitenkin vääristymää: putki vaikuttaa tekevän mutkia. Ortokuvassa vääristymä on korjattu. [5.]

## 2.2 Ilmakuvaus Suomessa

Ensimmäiset ilmakuvakartoitukset Suomessa tehtiin silloisen Maanmittaushallituksen toimeksiannosta vuonna 1924. Alkuaikojen tulokset olivat vaihtelevia, ja maankattavan kiintopisteverkoston puutteen takia kuvien sitominen ulkoiseen koordinaatistoon hankalaa. Maanmittaushallituksen säännöllinen ilmakuvakartoitus alkoikin siksi vasta vuonna 1936. [2, s. 428–429.]

Ilmakuvausta on tehty myös muiden toimijoiden toimeksiannosta. Puolustusvoimain ilmakuvaustoiminta alkoi vuonna 1927, jonka lisäksi esimerkiksi kunnat ja kaupungit sekä yksityiset yritykset ovat suorittaneet ilmakuvauksia. Aineistojen välillä on eroavaisuuksia muun muassa katetun alueen sekä mittakaavan suhteen, eri toimijoiden eroavien tarpeiden takia. [2.]

Maanmittauslaitoksen (sittemmin Maanmittauslaitoksen) ilmakuvausohjelma on maan kattavin, ja se on jatkunut vuodesta 1936. Digitaaliseen ilmakuvaukseen siirryttiin vuonna 2009, jota ennen kuvattiin filmille. Pohjoista Lappia lukuun ottamatta ilmakuvaukset uusitaan nykyään joka kolmas vuosi. Ilmakuvista laaditaan oikaistua ortokuvapeitettä, jota käytetään maastotietokannan ylläpitoon. [1.]

### 2.3 Maanmittauslaitoksen ilmakuva-arkisto ja saatavuus

Maanmittauslaitoksen ilmakuva-arkisto sisältää yli miljoona yksittäistä ilmakuvaa. Suurin osa on Maanmittauslaitoksen omaa tuotantoa, mutta arkistoon on luovutettu myös muiden toimijoiden ilmakuvia. [6.]

Filmille kuvattuja ilmakuvia digitoidaan paraikaa. Osittain huolena on filmipohjan hauraus, mutta tavoitteena on lisäksi saattaa aineisto laajemmin yhteiskunnan saataville [7]. Digitoiduista ilmakuvista laaditaan jälkikäteen ortokuvapeite [8].

Maanmittauslaitoksen ortokuva-aineiston jakelukanavat

Kaikki Maanmittauslaitoksen jo digitaalisessa muodossa löytyvä ortokuva-aineisto, joko digilaitteistolla kuvattu tai filmiltä digitoitu, on katsottavissa ilmaiseksi Paikkatietoikkuna-palvelun kautta. Aineistoa ei kuitenkaan ole mahdollista tallentaa omalle tietokoneelle tai saada näkyville paikkatieto-ohjelmistoon, vaan se on ainoastaan katsottavissa verkkoselaimessa. [9.]

Kaikki digitaalisessa muodossa löytyvä ortokuva-aineisto on lisäksi saatavilla Maanmittauslaitoksen maksullisen WMS (Web Map Service) -rajapinnan kautta. Tätä hyödyntäen aineisto on mahdollista saada näkyviin myös paikkatieto-ohjelmistoihin. Rajapinnan vuosittainen hinta alkaa kuitenkin 291,90 eurosta, eikä alkuperäisen laatuista pakkaamatonta aineistoa ole tätäkään kautta mahdollista tallentaa omalle tietokoneelle. [8.]

Ortokuva-aineisto vuodesta 2008 lähtien on ladattavissa ilmaiseksi tiedostomuodossa, Maanmittauslaitoksen Karttapaikka-palvelun kautta. Karttapaikasta ei kuitenkaan löydy vuotta 2008 vanhempaa aineistoa. [10.]

Kaikken vuosien yksittäisiä ilmakuvia on mahdollista tilata Maanmittauslaitokselta tiedostomuodossa, mutta jo ensimmäisestä toimitetusta ilmakuvasta veloitetaan vähintään 125,50 euroa. Kyseessä on tällöin myös raakailmakuva, ei mitataavakorrekti ortokuva. [1.]

### Uusi WCS-rajapinta

Vuonna 2023 Maanmittauslaitos julkaisi WCS (Web Coverage Service) -rasteri-rajapinnan, jonka kautta on mahdollista veloitusetta ladata Maanmittauslaitoksen digitaalisessa muodossa löytyviä ortokuva-aineistoja, kaikilta vuosilta. WCS-rajapintastandardi on kuitenkin verrattain harvinainen, eikä esimerkiksi avoimen lähdekoodin QGIS-paikkatieto-ohjelma tue Maanmittauslaitoksen toteutusta. Tavallisen käyttäjän on siksi hyvin vaikeaa saada historiallista aineistoa tiedostomuodossa. [11.]

Koska aineiston lataaminen veloitusetta on kuitenkin nykyään teknisesti mahdollista, laaditaan tässä insinööriyössä työkalu, jonka avulla myös tavallinen käyttäjä voi hyödyntää uutta WCS-rajapintaa. Työkalun avulla ladatun aineiston saa myös näkyviin paikkatieto-ohjelmiin, ja työkalu mahdollistaa täten näiden tarjoamat laajemmat analyysimahdollisuudet. Ladatun ortokuva-aineiston sijaintitarkkuus on 0,5–2 m [12].

## 2.4 Historiallisen ortokuva-aineiston käyttökohteet

Maanmittauslaitoksen ilmakuva-arkiston vanhimmat digitoidut kuvat ovat vuodelta 1931, ja samasta maantieteellisestä kohdasta löytyy usein aineistoa monilta eri vuosikymmeniltä. Tämä tekee arkistosta kulttuurihistoriallisesti merkittävän ja arvokkaan monelle alalle. Ortokuvista voi ilmetä tietoa ja näkyä muutoksia, joita ei muista lähteistä voi saada.

Maanmittauslaitos selvitti vuonna 2013 historiallisen ilmakuva-arkiston mahdollisia käyttötarkoituksia. Selvitys koostui sekä kirjallisuuskatsauksesta, että Maanmittauslaitoksen sidosryhmilleen tekemästä kyselystä. Selvityksessä tunnistettiin kymmeniä käyttötarkoituksia, joista seuraa lyhyt yhteenveto. [13].

Huomionarvoista on, että suurin osa selvityksessä tunnistetuista käyttötarkoituksista edellyttävät paikkatieto-ohjelmiston käyttöä analyysiin ja seurantaan [14, s. 270–272]. Vaikka aineisto on nykyisellään veloituksetta katseltavissa Paikkatietoikkuna-palvelussa, voi sen käytännön hyödynnettävyys siksi kuitenkin olla heikko. Toteutettu lisäosa, mahdollistaen analyysin paikkatieto-ohjelmistoissa, laajentaa aineiston käyttömahdollisuuksia.

#### Ympäristö, metsätalous ja viljely

Luonnonympäristön muutokset ilmenevät eri vuosien ilmakuvia vertaamalla selvästi. Esimerkkejä tästä ovat esimerkiksi vesistöjen umpeen kasvamiset sekä suoalueiden kuivattamiset. Muutoksia voidaan ortokuvien ja paikkatieto-ohjelmistojen avulla tutkia systemaattisesti, ja niiden tietoja hyödyntää luonnonympäristön entisöimiseen. Myös ilmastotutkimuksessa ortokuvia käytetään ajoittain tutkimusaineistona [15].

Metsätaloudessa vanhoja ortokuvia hyödynnetään esimerkiksi metsäinventointeihin sekä hakkuiden ajoittamiseen [16]. Eri vuosia vertailemalla voidaan seurata metsän kasvua ja tietoa hyödyntämällä laatia tietyille alueille ja kasvuympäristölle optimoituja metsänhoitosuunnitelmia.

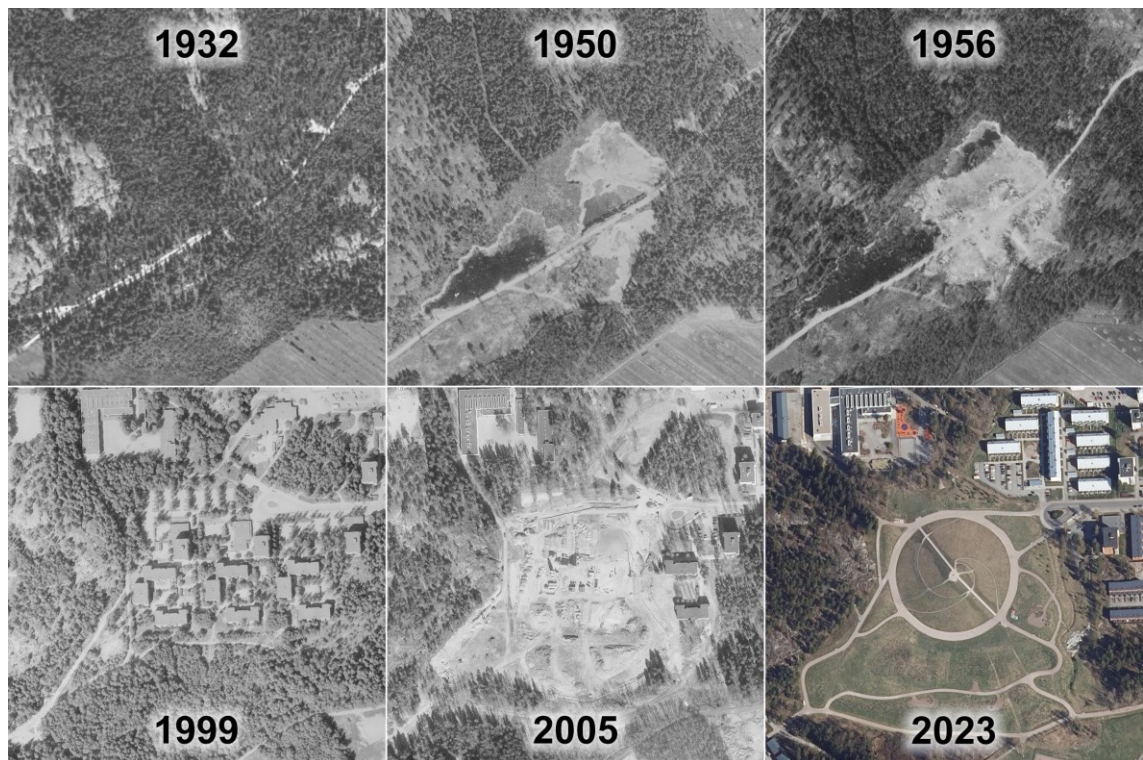
Vastaavasti ilmakuvista ilmenee runsaasti tietoa maatalouden kehityksestä [17]. Ortokuvien avulla on mahdollista tutkia esimerkiksi peltolohkojen kokoa, viljeltyjä kasveja, viljelyolosuhteita sekä arvioida tuotantomääriä.

#### Rakennettu ympäristö ja kaupunkisuunnittelu

Ilmakuvista ilmenee myös rakennetun ympäristön muutokset. Aineiston sisältämä tieto on tärkeä esimerkiksi kaavoituksessa ja muussa suunnittelutyössä,

jossa on selvitettävä alueen aiempi käyttö ja mahdollisesti ajoitettava rakenteita [18].

Varsinkin saastuneiden maa-alueiden ja muiden ympäristövaarojen tunnistamiseen ortokuvat ovat tärkeitä. Suomessa on runsaasti vanhoja ja dokumentoimattomia kaatopaikkoja, jotka on sittemmin peitetty. Kuvassa 3 on tästä konkreettinen esimerkki.



Kuva 3. Ortokuvissa näkyy Helsingin Myllypurossa sijaitsevan alueen käyttö kaatopaikkana 1950-luvulla. Kaatopaikka peitettiin, ja alueelle rakennettiin myöhemmin asuinkerrostalokortteli ja päiväkotii. Nämä jouduttiin vuosituhaten vaihteessa purkamaan, runsaiden ympäristömyrkköjen paljastuttua. Paikalla sijaitsee nykyään täyttömäki. [19.] Maanmittauslaitoksen ortokuva, haettu 03/2024.

### Historiantutkimus

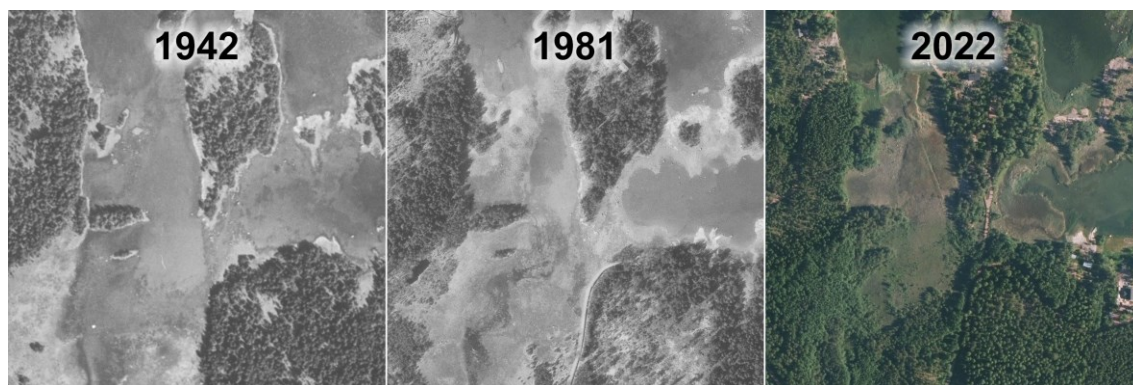
Historiantutkimuksessa vanhoja ilmakuvia käytetään ennen kaikkea alueiden ja yhdyskuntarakenteiden kehityksen tutkimiseen. Esimerkiksi Suomen voimakas

kaupungistuminen 1900-luvulla näkyy selvästi [20]. Aineisto on tärkeä, sillä ortokuvista ilmenee myös sellaisia yksityiskohtia, joita ei karttoihin ole sisällytetty.

Aineistosta on lisäksi ajoittain mahdollista tunnistaa arkeologisia kohteita, jotka ovat sittemmin jääneet kasvillisuuden tai rakenteiden peittoon [15].

#### Maanmittaustoimitukset

Maanmittaustoimituksissa vanhoja ortokuvia hyödynnetään ahkerasti. Niistä on esimerkiksi usein mahdollista erottaa kiinteistörajoja, jotka ovat myöhemmin kasvaneet umpeen, sekä vanhoja tielinjauksia. Ortokuvia voidaan myös hyödyntää vesijätön arviointiin, kuten kuvasta 4 käy ilmi.



Kuva 4. Ortokuvista ilmenee, kuinka Saltvikin kunnassa Ahvenanmaalla sijaitseva Mellanviken-lahti on ajan mittaan muuttunut vesijätöksi. Vertailemalla vanhoja toimituskarttoja ja ortokuvia voidaan arvioida vesijätön laatu ja määrä. Maanmittauslaitoksen ortokuva, haettu 03/2024.

Vanhoilla ilmakuvilla voi myös olla tunnearvoa tai esteettistä arvoa. Tiedostomuodossa olevasta ortokuvasta voi teettää ja kehystää esimerkiksi valokuvan seinälle.



### 3 Paikkatieto ja rajapinnat

#### 3.1 Paikkatieto

Paikkatietoa on tieto, jonka yksi ominaisuus on sijainti. Digitaalisesti tallennettu paikkatieto kategorisoidaan lähtökohtaisesti joko vektori- tai rasterimuotoiseksi. [21].

##### Vektorimuotoinen paikkatieto

Kaksiulotteinen vektorimuotoinen paikkatieto rakentuu geometrioista. Nämä voivat olla pisteitä, viivoja tai suljettuja alueita (polygoneja). Geometrioiden sijainnit ilmaistaan käyttäen yksiselitteisiä x- ja y-koordinaatteja. Yksittäisiin geometrioihin kytkeytyy tavallisesti tietosisältöä eli niin kutsuttuja attribuutteja. [22, s. 49.]

Kiinteistörekisteri on esimerkki vektorimuotoisesta paikkatietoaineistosta. Se sisältää sekä pisteitä (rajamerkit), viivoja (kiinteistöjen rajat) että polygoneja (kiinteistöjen alueet). Näillä kaikilla on yksiselitteiset sijainnit, ja geometrioihin kytkeytyy myös niitä kuvaavia attribuutteja. [23.]

Vektorimuotoinen paikkatieto kuvaa siis ensisijaisesti yksittäisiä kohteita. Se vastaa kysymyksiin missä kohteet ovat, ja millaisia kohteet ovat. Kuvassa 5 (vasemmalla) on esimerkki vektorimuotoisesta paikkatiedosta.



Mustavalkoisissa ortokuvissa pikseliin tallennetaan ainoastaan yksi arvo, joka ilmaisee väri-intensiteetin asteikolla mustasta valkoiseen. Väriortokuvissa pikseliin tallennetaan lähtökohtaisesti kolme arvoa, jotka ilmaisevat väri-intensiteetin erikseen punaisille, vihreille ja sinisille aallonpituuksille. Yhdistämällä nämä kolme arvoa saadaan luotua ihmissilmälle ymmärrettävä väri. [14, s. 272].

### 3.2 Paikkatietorajapinnat

Rajapinnat ovat standardeja, jotka mahdollistavat eri ohjelmistojen välisen vuorovaikutuksen. Ne määrittävät, miten ohjelmistot kommunikoivat keskenään, mahdollistaen niiden yhteentoimivuuden. Paikkatietoyhteyksissä termiä rajapinta käytetään kuitenkin usein viittaamaan ensisijaisesti tiedonsiirtoon internetin välityksellä, näin tehdään myös tässä insinööriyössä. [24.]

Monet paikkatietoaineistot ovat joko kooltaan suuria, tai sisällöltään usein päivittyviä. Maanmittauslaitoksen tuorein ortoilmakuva kattaa koko Suomen, eikä sitä suuren tiedostokoon takia ole mahdollista kokonaisuudessaan tallentaa käyttäjän omalle tietokoneelle. Kiinteistörekisterin sisältö puolestaan muuttuu lähes päivittäin, joten tietokoneelle tallennettu versio vanhentuisi nopeasti.

Rajapinnat mahdollistavat sen, että itse datasta on yksi kattava ja päivitetty versio, jota säilytetään sitä ylläpitävän organisaation palvelimella. Käyttäjät voivat rajapintojen kautta päästä käsiksi juuri sen alueen dataan, jota milloinkin tarvitsevat [25]. Kääntöpuolena on kuitenkin, että suorituskyky on riippuvainen internet-yhteyden nopeudesta ja laadusta. Ajoittain paikkatietoalgoritmit eivät myöskään hyväksy rajapinnasta tulevia lähtötietoja, vaan vaativat aineiston tallentamisen paikallisesti toimiakseen.

Paikkatietorajapintastandardeja on useita, ja soveltuvin riippuu mm. jaettavasta aineistosta (vektori vs. rasteri), käyttötarkoituksesta (analyysi vs. visualisointi) ja ohjelmistosta (paikkatieto-ohjelmisto vs. verkkoselain). Perinteisesti rajapintoja on käytetty lähinnä paikkatieto-ohjelmistojen kautta, mutta uudet rajapintastandardit ovat yhä enenevässä määrin hyödynnettävissä myös verkkoselaimissa.

## Rajapintojen avoimuus

Rajapinnat voivat olla täysin avoimia, jolloin tietosisältö on kaikille saatavilla ilmaiseksi, ilman rekisteröintiä tai kirjautumista. Ne voivat myös olla suljettuja maksullisia, jolloin käyttö vaatii käyttösopimuksen sekä kirjautumisen, ja aineistoista veloitetaan esimerkiksi tietopyyntöjen määrään perustuen.

Tavallinen on välimuoto, jossa aineisto on saatavilla ilmaiseksi, mutta käyttö vaatii rekisteröinnin. Käyttäjä saa rekisteröityään niin kutsutun API-avaimen (Application Programming Interface), jolla tunnistautuu rajapintaan yhdistäessä. API-avaimen avulla organisaatio voi seurata tietopyyntöjä, ja tarvittaessa rajoittaa aineiston saatavuutta, esimerkiksi mikäli yksittäisen käyttäjän tietopyynnöt ovat niin suuria, ettei palvelimen suorituskyky riitä. [26.]

Tässä insinööriyössä toteutettu työkalu hyödyntää Maanmittauslaitoksen tarjoamia WCS- ja OGC API: Features (Open Geospatial Consortium Application Programming Interface: Features) -rajapintoja. Kyseisten rajapintojen käyttö on maksutonta, mutta vaatii rekisteröinnin ja API-avaimen.

### 3.3 WCS-rajapintastandardi

WCS-rajapintastandardi mahdollistaa rasterimuotoisten aineistojen, kuten satelliittikuvien, ortokuvien tai digitaalisten maastomallien, jakelun internetin välityksellä. WCS tukee aineistojen jakamista niiden alkuperäisessä muodossa ja resoluutiolla, mahdollistaen mm. paikkatietoanalyysit ladattua aineistoa hyödyntäen. [27.]

Tämä erottaa WCS:n laajasti käytetystä WMS-rajapintastandardista. WCS:n käyttötarkoitus on aineiston jakelu, kun taas WMS on tarkoitettu aineiston visualisointiin. WMS tarjoaa valmiiksi renderöityjä ja häviöllisesti pakattuja kuvia, alkuperäisen aineiston sijaan.

WCS-rajapinnan kautta tehdyt kyselyt hyödyntävät HTTP- (Hypertext Transfer Protocol) -verkko-osoitteita, ja kyselyjä voi periaatteessa tehdä myös verkkoselaimen kautta. Kyselytyyppejä on kolme:

- GetCapabilities-kysely palauttaa yleistä tietoa WCS-palvelimesta. Tämän kyselytyypin kautta selviää esimerkiksi, mitä koordinaattijärjestelmiä palvelin tukee.
- DescribeCoverage-kysely palauttaa metatietoa tietyistä saatavilla olevasta rasteriaineistosta, esimerkiksi mustavalkoisista ortokuvista. Tämän kyselytyypin kautta selviää muun muassa, miltä ajankohdilta kyseistä aineistoa löytyy.
- GetCoverage-kysely palauttaa itse rasteriaineiston palvelimelta. [28.]

Esimerkkikoodi 1 sisältää esimerkin WCS GetCoverage-kyselystä.

```
https://avoin-karttakuva.maanmittauslaitos.fi/ortokuvat-ja-  
korkeusmallit/wcs/v2?service=WCS&version=2.0.1&request=GetCoverage  
&CoverageID=ortokuva_mustavalko&SUBSET=E(392570.0,394570.0)  
&SUBSET=N(6676879.0,6678879.0)&SubsettingCRS=http://www.opengis.net  
/def/crs/EPSG/0/3067&OutputCRS=http://www.opengis.net/def/crs/EPSG/0  
/3067&api-key=xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxxxxxx&format=image/tiff  
&geotiff:compression=LZW&SUBSET=time("1932-12-31T02:00:00Z")
```

Esimerkkikoodi 1. Esimerkki WCS GetCoverage-kyselystä.

Taulukossa 1 on sama kysely jaoteltu osiin ja selitetty.

Taulukko 1. WCS:n GetCoverage-kysely osiin jaoteltuna ja selitettynä.

Kyselyn kohta	Selitys
https://avoin-karttakuva.maanmittauslaitos.fi/ortokuvat-ja-korkeusmallit/wcs/v2?service=WCS	WCS-palvelimen verkko-osoite
&version=2.0.1	Palvelimen versio
&request=GetCoverage	Kyselyn tyyppi
&CoverageID=ortokuva_mustavalko	Pyydetty aineisto
&SUBSET=E(392570.0,394570.0)	Pyydetyn alueen vasen ja oikea koordinaattirajaus
&SUBSET=N(6676879.0,6678879.0)	Pyydetyn alueen ala- ja ylä- koordinaattirajaus
&SubsettingCRS=http://www.opengis.net/def/crs/EPSG/0/3067	Pyynnössä käytetty koordinaattijärjestelmä
&OutputCRS=http://www.opengis.net/def/crs/EPSG/0/3067	Aineiston haluttu koordinaattijärjestelmä
&api-key=xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx	Käyttäjän API-avain
&format=image/tiff	Aineiston haluttu tiedostomuoto
&geotiff:compression=LZW	Aineiston haluttu pakkausmenetelmä
&SUBSET=time("1932-12-31T02:00:00Z")	Aineiston haluttu ajankohta

Tässä tapauksessa WCS-palvelin palauttaa GetCoverage-kyselyyn GeoTIFF (Geographic Tag Image File Format) -muotoisen rasteritiedoston, joka näkyy kuvassa 6.



Kuva 6. WCS-palvelimen esimerkkikyselyyn palauttama rasteritiedosto. Maanmittauslaitoksen ortokuva, haettu 03/2024.

Maanmittauslaitoksen WCS-palvelimen palauttaman ortokuvarasterin pikselikoko on 0,5 m. Tämä tarkoittaa, että esimerkiksi 1×1 km:n kokoisen alueen ortokuva on 2 000×2 000 pikselin kokoinen.

Maanmittauslaitoksen ortokuva-aineiston alkuperäinen erottelukyky vastaa noin 0,5–1 metrin pikselikokoa [8]. Palvelimelta ladatut ortokuvat ovat näin ollen resoluutioltaan alkuperäisiä vastaavia.

WCS GetCoverage -kyselyä tarkentavat parametrit (&-alkuiset) voi antaa palvelimelle missä järjestyksessä tahansa.

Ennen varsinaisia aineistopyyntöjä GetCoverage-kyselyillä on tutustuttava WCS-palvelimen kykyihin mainituilla GetCapabilities- ja DescribeCoverage-kyselyillä. Tietoa palvelimen tukemista koordinaattijärjestelmistä (GetCapabilities) tarvitaan halutun maantieteellisen alueen määrittelyyn, metatietoa itse aineistosta (DescribeCoverage) oikean aineiston pyytämiseen.

WCS-palvelimen DescribeCoverage-kyselyn vastaus sisältää tietoa myös siitä, miltä ajankohdilta aineistoa on saatavilla. Vastaus voi esimerkiksi kertoa, että mustavalkoisia ortokuvia löytyy vuodelta 1932. Tietoa siitä, miltä maantieteellisiltä alueilta kyseisen vuoden aineistoja on saatavilla, ei kuitenkaan vastaukseen sisälly.

Vuonna 1932 ilmakuvaattiin ainoastaan pääkaupunkiseudulla ja Hämeenlinnassa. Jos laaditaan GetCoverage-kysely ja pyydetään vuoden 1932 ortokuva-aineistoa muualta, esimerkiksi Turun alueelta, on palvelimen palauttama rasteritiedosto tyhjä, sisältäen pelkkää mustaa.

Koska vain pieni osa Suomesta on ilmakuvaattu vuosittain, koskee tyhjiä rasteritiedostojen ongelma kaikkia WCS-palvelimen listaamia vuosia. Jos samasta alueesta tekisi GetCoverage-kyselyn jokaiselle vuodelle, olisi suuri enemmistö palvelimen palauttamista rasteritiedostoista täysin tyhjiä. Tämän takia GetCoverage-aineistokyselyjä on rajattava etukäteen vain sellaisille vuosille, joilta aineistoja tosiasiallisesti on saatavilla halutulta alueelta. Tähän tarvitaan toista rajapintaa.

### 3.4 OGC API: Features -rajapintastandardi

OGC API: Features -rajapintastandardi mahdollistaa vektorimuotoisten paikkatietoaineistojen jakelun internetin välityksellä. Standardi on verrattain uusi:



ydintoiminnallisuudet on määritelty, mutta lisätoiminnallisuuksien kehitystyö jatkuu yhä. [29.]

OGC API: Features -rajapinnat eivät siksi vielä ole kovin laajassa käytössä. Ne tulevat todennäköisesti kuitenkin tulevaisuudessa korvaamaan monet nyt käytössä olevia vektorimuotoisten paikkatietoaineistojen rajapinnat. Uuden standardin vahvuus on ennen kaikkea toimivuus verkkoselaimissa, perinteisten paikkatieto-ohjelmistojen lisäksi. [30.]

Features on osa laajempaa OGC API -hanketta, jossa pyritään laatimaan sarjan yhteneviä ja modulaarisia moderneja rajapintastandardeja erilaisille paikkatietoaineistoille. Monet muutkin paikkatietoalalla nykyään laajasti käytetyt rajapintastandardit korvautunevat tevaisuudessa uusilla OGC API -standardeilla. [31.]

OGC API: Features -kyselyt hyödyntävät HTTP-osoitteita, joten niitä voi tehdä myös verkkoselaimen kautta. Toistaiseksi standardiin kuuluu ainoastaan vektorimuotoisen aineiston hakeminen palvelimelta (ns. GET-metodi), mutta tulevaisuudessa aineistoja voidaan myös lisätä, korvata, muuttaa tai poistaa rajapintastandardia käyttäen.

Esimerkkikoodi 2 sisältää esimerkin OGC API: Features -kyselystä.

```
https://avoin-karttakuva.maanmittauslaitos.fi/ortokuvat-ja-  
korkeusmallit/features/v2/collections/ortokuva_mustavalko/items  
?bbox=392570.0,6676879.0,394570.0,6678879.0&bbox-crs=http://www.  
opengis.net/def/crs/EPSG/0/3067&api-key=xxxxxxxx-xxxx-xxxx-xxxx-  
xxxxxxxxxxxx&f=json
```

Esimerkkikoodi 2. Esimerkki OGC API: Features -kyselystä.

Taulukossa 2 sama kysely on jaoteltu osiin ja selitetty.

Taulukko 2. OGC API: Features -kysely osiin jaoteltuna ja selitettynä.

Kyselyn kohta	Selitys
https://avoin-karttakuva.maanmittauslaitos.fi/ortokuvat-ja-korkeusmallit/features/v2	OGC API: Features -palvelimen verkko-osoite
/collections	Palvelimen tarjoamat aineistot
/ortokuva_mustavalko	Pyydetty aineisto
/items	Pyydetyn aineiston kohteet
?bbox=392570.0,6676879.0,394570.0,6678879.0	Pyydetyn alueen koordinaattirajaus (vasen, ala, oikea, ylä)
&bbox-crs= http://www.opengis.net/def/crs/EPSG/0/3067	Pyynnössä käytetty koordinaattijärjestelmä
&api-key=xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx	Käyttäjän API-avain
&f=json	Aineiston haluttu tiedostomuoto

OGC API: Features -palvelin palauttaa kyselyyn GeoJSON (Geographic JavaScript Object Notation) -muotoisen vastauksen. Esimerkkikoodissa 3 on ote tästä vastauksesta.

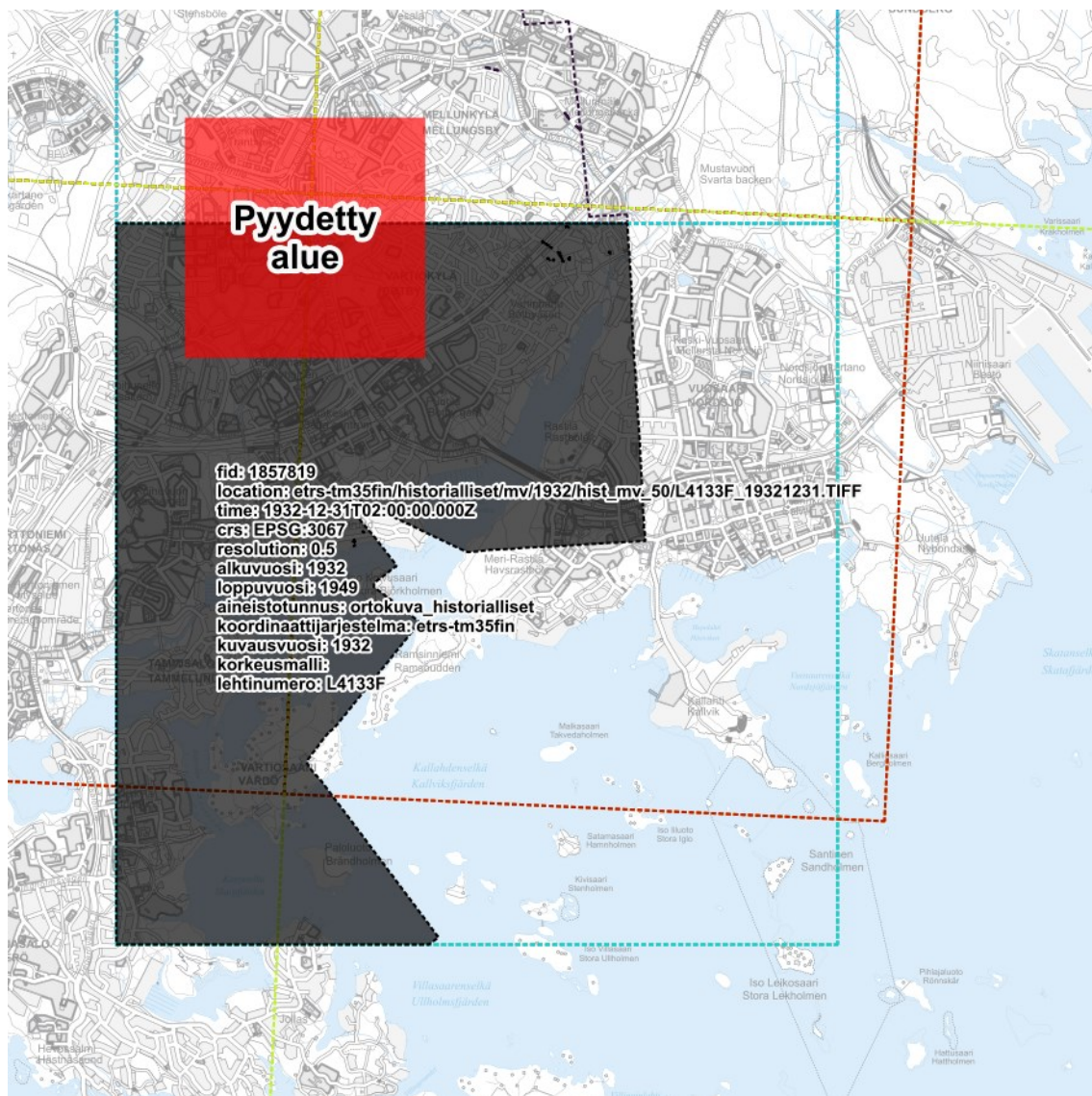
```

{
  "type": "FeatureCollection",
  "features": [
    {
      "type": "Feature",
      "id": 1857819,
      "geometry": {
        "type": "MultiPolygon",
        "coordinates": [
          [
            [
              [
                25.0503325,
                60.2244208
              ],
              [
                25.1270157,
                60.2255256
              ],
              #Kohteen seuraavat koordinaatit jätetty pois
            ]
          ]
        ]
      },
      "properties": {
        "fid": 1857819,
        "location": "etrs-
tm35fin/historialliset/mv/1932/hist_mv_50/L4133F_19321231.TIFF",
        "time": "1932-12-31T02:00:00Z",
        "crs": "EPSG:3067",
        "resolution": "0.5",
        "alkuvuosi": 1932,
        "loppuvuosi": 1949,
        "aineistotunnus": "ortokuva_historialliset",
        "koordinaattijarjestelma": "etrs-tm35fin",
        "kuvausvuosi": "1932",
        "korkeusmalli": "",
        "lehtinumero": "L4133F"
      }
    },
    #Seuraavat kohteet jätetty pois
  ],
  "timeStamp": "2024-02-29T08:04:23.218541Z",
  "links": [
    #Linkit jätetty pois
  ],
  "numberReturned": 31
}

```

**Esimerkkikoodi 3.** Ote OGC API: Features -palvelimen kyselyyn palauttamasta vastauksesta.

Tässä tapauksessa palvelin palauttaa tietoa kyselyssä määriteltyä maantieteellistä aluetta leikkaavista ortokuvista. Määriteltyä aluetta leikkaa yhteensä 31 yksittäistä kuvaa ("numberReturned":31). Palvelin palauttaa näiden aluegeometriat, sekä joukon attribuutteja. Vastaus on visualisoitu kuvassa 7.



Kuva 7. Punaisella on korostettu kyselyssä määritelty alue. Katkoviivat ovat aluetta leikkaavien yksittäisten kuvien geometrioita. Tummennettuna ensimmäinen palautettu kuvageometria. Tämä on sama, joka näkyy esimerkikoodissa 3. Taustakartta Helsingin kaupunki, opaskartta harvanimi harmaa, haettu 03/2024.

Hyödyntämällä OGC API: Features -rajapinnan tietoja ortokuvista, tarkemmin ottaen time-attribuuttia, on mahdollista etukäteen rajata WCS-rajapinnalle tehtyjä aineistopyyntöjä. Jos OGC API: Features -vastauksesta löytyy ajankohta "1932-12-31T02:00:00.000Z", pyydetään tätä myös WCS-rajapinnalta. Jos taas seuraavaa vuotta 1933 ei OGC API: Features -vastauksesta löydy, ei sitä myöskään pyydetä WCS-palvelimelta. Näin voidaan välttää tyhjen rasteritiedostojen lataamista.

## 4 Ohjelmistot

Lähtökohtana oli, että insinööriyössä toteutettavalla työkalulla tulee olla mahdollista graafisen käyttöliittymän kautta ja karttapohjalle piirtäen määritellä alue, jolta aineistoa halutaan ladata. Tämän takia työkalu päätettiin toteuttaa lisäosana QGIS-paikkatieto-ohjelmistoon. QGIS:sta karttapohja sekä tarvittavat koordinaatteihin ja koordinaatistoihin liittyvät toiminnot löytyisivät valmiina.

### 4.1 QGIS-paikkatieto-ohjelmisto

QGIS on laajasti käytössä oleva ilmainen avoimen lähdekoodin paikkatieto-ohjelmisto, joka tarjoaa monipuoliset työkalut paikkatiedon käsittelyyn ja analysointiin. QGIS tukee laajaa valikoimaa eri paikkatietoformaatteja; perinteisten kaksiulotteisten vektori- ja rasteriformaattien lisäksi yhä enenevässä määrin myös kolmiulotteisia aineistoja kuten pistepilviä. QGIS on saatavilla sekä Windows-, Mac- että Linux-käyttöjärjestelmille. [32.]

Avoin lähdekoodi mahdollistaa QGIS:n räätälöinnin ja laajentamisen käyttäjän tarpeiden mukaan. QGIS:lle on saatavilla satoja julkisia lisäosia, jotka laajentavat sen toiminnallisuuksia.

QGIS on ohjelmoitu pääosin C++-ohjelmointikielellä, ja graafinen käyttöliittymä on toteutettu Qt-käyttöliittymäkirjastolla. QGIS on suunniteltu olemaan laajennettavissa Python-ohjelmointikieltä hyödyntäen.

### 4.2 Python-ohjelmointikieli

Ohjelmointikielet ovat työkaluja, joita käytetään tietokoneohjelmien ja skriptien luomiseen. Python-ohjelmointikieli on suunniteltu olemaan mahdollisimman selkeä ja ihmiselle helposti ymmärrettävä. Tämän lisäksi Pythonin mukana tulee laaja standardikirjasto valmiita toiminnallisuuksia. Kyseiset ominaisuudet mahdollistavat sen, että myös aloittelija voi verrattain helposti luoda tarvitsemansa ohjelman Pythonia ja sen valmiita kirjastoja hyödyntäen. Standardikirjastojen

lisäksi Pythoniin on saatavilla runsaasti käyttäjäyhteisön laatimia valmiita kirjastoja spesifisempiin tarkoituksiin. [33.]

Python on laajasti käytössä paikkatietoalalla, ja PyQGIS-kirjaston avulla sitä voidaan hyödyntää QGIS:n toiminnallisuuksien laajentamiseen.

### 4.3 PyQGIS-kirjasto

PyQGIS on Python-kirjasto, joka mahdollistaa QGIS:n laajentamisen, käytön ja automatisoinnin hyödyntäen Pythonia. PyQGIS tarjoaa pääsyn samoihin QGIS:n keskeisiin toimintoihin, joita voi käyttää QGIS:n graafisen käyttöliittymänkin kautta. [34.]

PyQGIS:n avulla on siten mahdollista rakentaa ja automatisoida työkaluja, jotka hyödyntävät sekä QGIS:n toiminnallisuuksia, että Python-ohjelmointikieltä ja sen kirjastoja laajemmin. Tässä insinööriyössä PyQGIS:ia käytetään tällä tavalla. Laadittu lisäosa hyödyntää QGIS:n toiminnallisuuksia ja käyttöliittymää maantieteellisen alueen valintaan, kun taas itse ortokuvien lataus toteutetaan käyttäen Python-ohjelmointikielen omia kirjastoja.

PyQGIS-kirjasto koostuu useista eri osista, jotka tarjoavat eri osa-alueita QGIS:n toiminnallisuuksista. Tässä insinööriyössä käytetään PyQGIS:n core- ja gui-osia. core sisältää QGIS:n ydintoiminnallisuudet, kuten paikkatietoaineistojen hallinnan ja käsittelyn. gui puolestaan mahdollistaa QGIS:n graafiseen käyttöliittymään hyödyntämisen Pythonin kautta. [35.]

PyQGIS-ohjelman tai skriptin laatiminen edellyttää Python-ohjelmointikielen alkeiden tuntemista. Internetistä on löydettävissä vain rajallinen määrä PyQGIS-kursseja ja -resursseja, joten usein halutun toiminnallisuuden toteuttamiseen ei löydy valmista ratkaisua. Tällöin vaaditaan PyQGIS:n tekniseen dokumentaation paneutumista.

## 5 QGIS-lisäosan toteutus

Lisäosa päätettiin jakaa kolmeen selkeästi eroteltuun toiminnalliseen vaiheeseen, joita ovat lähtötiedot, aluevalinta ja lataus. Lisäosa etenee vaiheittain siirtymisen seuraavaan vasta edellisen vaiheen onnistuttua.

Kolmen toiminnallisen päävaiheen lisäksi lisäosaan voidaan katsoa kuuluvan kolme apuvaihetta. Näistä kaksi, tuontilausekkeet ja pääluokan alustus, tapahtuvat jo ennen lisäosan toiminnallisuuden käynnistämistä, kolmas (siivous) puolestaan lisäosan lopettaessa toimintansa.

Kuvasta 8 ilmenee lisäosan rakenne sekä vaiheiden tehtävät.



Kuva 8. Toteutetun lisäosan rakenne sekä vaiheiden tehtävät.

Vaiheita käsitellään seuraavaksi yksityiskohtaisemmin. Lisäosan koodi (liite 1, tiedosto ortokuvalataaja.py) on jäsennelty ja otsikoitu vastaavalla tavalla.

## 5.1 Tuontilausekkeet

Tuontilausekevaiheessa tuodaan ne Python-kirjastot, joita lisäosa tarvitsee toimiakseen. Kirjastot ja niiden käyttötarkoitukset käydään läpi taulukossa 3.

Taulukko 3. Lisäosan hyödyntämät kirjastot käyttötarkoituksineen.

Kirjasto	Käyttötarkoitus
os	os tarjoaa tiedosto- ja kansiohallintaan liittyviä toimintoja. Sitä tarvitaan ortokuvien tallennuskansion valitsemiseen, ladatun tiedoston koon tarkistamiseen sekä mahdollisten tyhjen tiedostojen poistamiseen.
json	json tarjoaa tavan käsitellä json-muotoisia tietorakenteita. Sitä tarvitaan OGC API: Features -vastauksen jäsentämiseen merkkijonosta tietorakenteeksi. Tämä mahdollistaa vastauksen tietosisällön hyödyntämisen.
urllib.request	urllib.request tarjoaa verkkosisältöjen hakemiseen liittyviä toimintoja. urlopen-toimintoa tarvitaan OGC API: Features -kyselyn tekemiseen, urlretrieve-toimintoa puolestaan itse ortokuvatiedostojen lataamiseen.
qgis.core	qgis.core tarjoaa QGIS:n ydintoiminnallisuudet. Sitä tarvitaan koordinaattijärjestelmien sekä karttatasojen hallintaan.
qgis.gui	qgis.gui tarjoaa pääsyn QGIS:n graafiseen käyttöliittymään. Sitä tarvitaan aluevalinnassa. Kirjastolla voidaan muuntaa hiiren sijainti näytöllä maantieteellisiksi koordinaateiksi, ja valittu alue visualisoida.
PyQt5.QtGui	PyQt5.QtGui tarjoaa graafisten Qt-käyttöliittymien perustoiminnallisuuksia. Sitä tarvitaan mm. määrittelemään värejä.
PyQt5.QtWidgets	PyQt5.QtWidgets tarjoaa graafisten Qt-käyttöliittymien perustoiminnallisuuksia. Sitä tarvitaan ikkunoiden, painikkeiden sekä tekstilaatikkojen luomiseen.

## 5.2 Pääluokan alustus

Pääluokan alustus -vaiheessa ortokuvalataaja-niminen pääluokka otetaan käyttöön QGIS:ssa. Käytännössä tämä tarkoittaa, että lisäosan toiminnallisuus



integroidaan osaksi QGIS:ia, ja lisäosa valmistellaan käyttöä varten. Alustus ei kuitenkaan vielä käynnistä lisäosan toimintaa, tämä tapahtuu vasta käyttäjän aloitteesta.

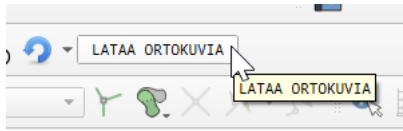
Alustusvaihe koostuu kolmesta eri funktiosta, jotka ovat `__init__`, `initGui` ja `unload`.

### `__init__`

`__init__`-funktio asettaa pohjan lisäosan toiminnallisuudelle QGIS:ssa. Funktio yhdistää lisäosan QGIS:iin, ja määrittää sellaisia muuttujia, joita lisäosa tarvitsee toimiakseen.

### `initGui`

`initGui`-funktio lisää QGIS:n käyttöliittymään painikkeen, jossa lukee "LATAA ORTOKUVIA" (kuva 9). Painikkeeseen liitetään tapahtumakäsittelijä, joka reagoi, jos painiketta klikataan. Kun niin tapahtuu, käynnistetään `lahtotiedot`-niminen funktio, joka aloittaa lisäosan varsinaisen toiminnan.



Kuva 9. QGIS:n käyttöliittymään lisätty "LATAA ORTOKUVIA" -painike.

### `unload`

`unload`-funktio määrittää toimet, jotka suoritetaan, kun lisäosa deaktivoidaan tai poistetaan. Tässä tapauksessa "LATAA ORTOKUVIA" -painike sekä siihen liitetty tapahtumakäsittelijä poistetaan.

### 5.3 Lähtötiedot

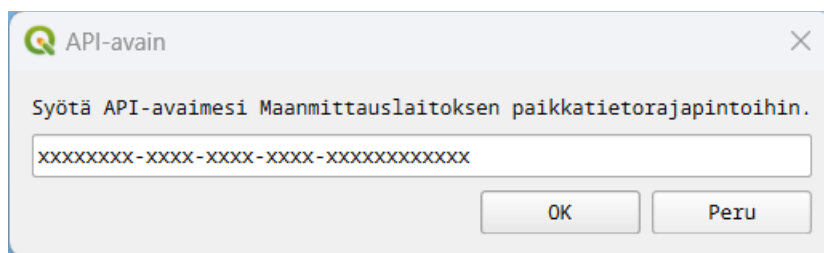
Lisäosan varsinainen toiminta alkaa lähtötietovaiheesta. Lähtötietovaiheessa pyydetään käyttäjältä ne lähtötiedot, joita lisäosa tarvitsee. Nämä ovat:

- API-avain Maanmittauslaitoksen paikkatietorajapintoihin
- nimi tallennetuille tiedostoille
- kansio, johon tiedostot tallennetaan
- koordinaattijärjestelmä, jota tiedostojen halutaan käyttävän.

Kun käyttäjä klikkaa ”LATAA ORTOKUVIA” -painiketta, käynnistyy lähtötiedot-niminen funktio. Funktio pyytää käyttäjältä yksitellen mainitut lähtötiedot, dialogi-ikkunoita hyödyntäen. Mikäli lähtötietoa ei käyttäjältä saada, lisäosan toiminta keskeytyy välittömästi ja se siirtyy suoraan viimeiseen siivousvaiheeseen.

#### API-avain

Hyödyntäen Qt:n `QInputDialog.getText`-funktiota näytetään käyttäjälle dialogi-ikkuna, johon pyydetään syöttämään API-avain Maanmittauslaitoksen paikkatietorajapintoihin. API-avainta pyytävä ikkuna näkyy kuvassa 10.



Kuva 10. Käyttäjältä API-avainta pyytävä ikkuna.

Funktio palauttaa kaksi arvoa. Ensimmäinen on käyttäjän syöttämä teksti merkkijonona ja toinen on boolean (True/False). Jos dialogi-ikkuna suljetaan OK:ta klikkaamalla, on boolean True. Jos dialogi-ikkuna on peruutettu tai suljettu x:stä, on boolean False.

Ensimmäinen palautettu arvo säilötään `api_avain`-nimiseen muuttujaan, boolean säilötään `api_avain_b`-nimiseen muuttujaan. Tämän jälkeen tarkistetaan, että `api_avain_b`-muuttuja on `True` (käyttäjä on klikannut OK:ta), sekä että `api_avain`-muuttuja ei ole tyhjä (käyttäjä on syöttänyt jotain).

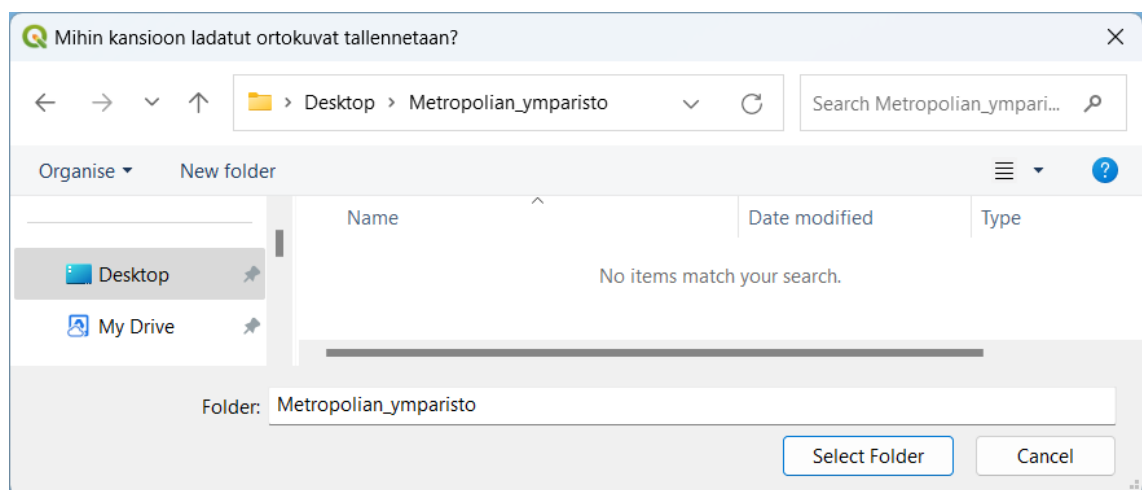
Jos ehdot täyttyvät, jatkaa lisäosa pyytämään tallennuskansiota. Jos ehdot eivät täyty, toiminta keskeytyy ja lisäosa siirtyy suoraan viimeiseen siivousvaiheeseen.

### Tiedostonimi

Kauttaaltaan samaa menetelmää hyödyntäen, kuin API-avaimen kohdalla, pyydetään käyttäjää syöttämään tiedostonimi tallennetuille ortokuville. Tämä säilötään nimi-nimiseen muuttujaan.

### Tallennuskansio

Hyödyntäen Qt:n `QFileDialog.getExistingDirectory`-funktioita näytetään käyttäjälle dialogi-ikkuna. Dialogi-ikkunassa (kuva 11) pyydetään valitsemaan kansio, johon ladatut ortokuvat tallennetaan.



Kuva 11. Käyttäjältä tallennuskansiota pyytävä ikkuna.

Funktio palauttaa tallennuskansion polun merkkijonona. Tämä säilötään tallennuskansio-nimiseen muuttujaan. Tämän jälkeen tarkistetaan, että tallennuskansio-muuttuja ei ole tyhjä (käyttäjä on valinnut kansion).

Jos ehto täyttyy, jatkaa lisäosa koordinaattijärjestelmiin. Jos ehto ei täyty, toiminta keskeytyy ja lisäosa siirtyy suoraan viimeiseen siivousvaiheeseen.

#### Tuettujen koordinaattijärjestelmien määrittäminen

Maanmittauslaitoksen OGC API: Features- ja WCS-palvelimet tukevat Suomessa käytössä olevia TM35FIN- ja GKnnFIN-tasokoordinaattijärjestelmiä. OGC API: Features- ja WCS-kyselyissä on määriteltävä, missä koordinaattijärjestelmässä kyselyssä annetut koordinaatit ovat.

TM35FIN-koordinaattijärjestelmä kattaa koko Suomen, kuitenkin tarkkuuden kustannuksella. Varsinkin maan länsi- ja itäosissa mittakaavavirhe kasvaa verrattain suureksi.

Paikallisissa GKnnFIN-koordinaattijärjestelmissä Suomi on jaettu yhden asteen levysiin siivuihin länsi-itäsuunnassa. Läntisin siivu (koordinaattijärjestelmä GK19FIN) on 19. meridiaanin molemmin puolin, itäisin siivu (koordinaattijärjestelmä GK31FIN) vastaavasti 31. meridiaanin molemmin puolin. Tarkoituksenmukainen koordinaattijärjestelmä riippuu sijainnista. Esimerkiksi Maarianhaminassa hyödynnetään GK20FIN-koordinaattijärjestelmää, Helsingissä puolestaan GK25FIN-koordinaattijärjestelmää.

Lisäosassa koordinaattijärjestelmiä varten luodaan sanakirja, joka tallennetaan muuttujaan nimeltä `tuetut_koordinaattijarjestelmat`. Sanakirja sisältää rajapintojen tukemat koordinaattijärjestelmät, ja sen perusrakenne ilmenee esimerkkikoodista 4.

```
tuetut_koordinaattijarjestelmat = {
    "Nimi":["EPSG:wxyz","http://www.opengis.net/def/crs/EPSG/0/wxyz"],
}
```

Esimerkkikoodi 4. tuetut\_koordinaattijarjestelmat-sanakirjan rakenne.

Sanakirjan avaimina toimivat käyttäjälle ymmärrettävät koordinaattijärjestelmien nimet, esimerkiksi "TM35FIN (EPSG:3067)". Sanakirjan arvoina ovat puolestaan listarakenteet. Listan ensimmäinen alkio on koordinaattijärjestelmän tunnistetunnus (esimerkiksi "EPSG:3067"), toinen alkio puolestaan linkki koordinaattijärjestelmän määrittelysivulle (esimerkiksi "http://www.opengis.net/def/crs/EPSG/0/3067").

Sanakirjan luonnin jälkeen tehdään käyttäjälle ymmärrettävistä avaimista erillinen listarakenne, mutta ilman sanakirjan arvoja. Tämä tallennetaan muuttujaan nimeltä lista\_koordinaattijarjestelmista. Kyseinen lista näkyy esimerkkikoodissa 5.

```
lista_koordinaattijarjestelmista = ["TM35FIN (EPSG:3067)", "GK19FIN (EPSG:3873)", "GK20FIN (EPSG:3874)", "GK21FIN (EPSG:3875)", "GK22FIN (EPSG:3876)", "GK23FIN (EPSG:3877)", "GK24FIN (EPSG:3878)", "GK25FIN (EPSG:3879)", "GK26FIN (EPSG:3880)", "GK27FIN (EPSG:3881)", "GK28FIN (EPSG:3882)", "GK29FIN (EPSG:3883)", "GK30FIN (EPSG:3884)", "GK31FIN (EPSG:3885)"]
```

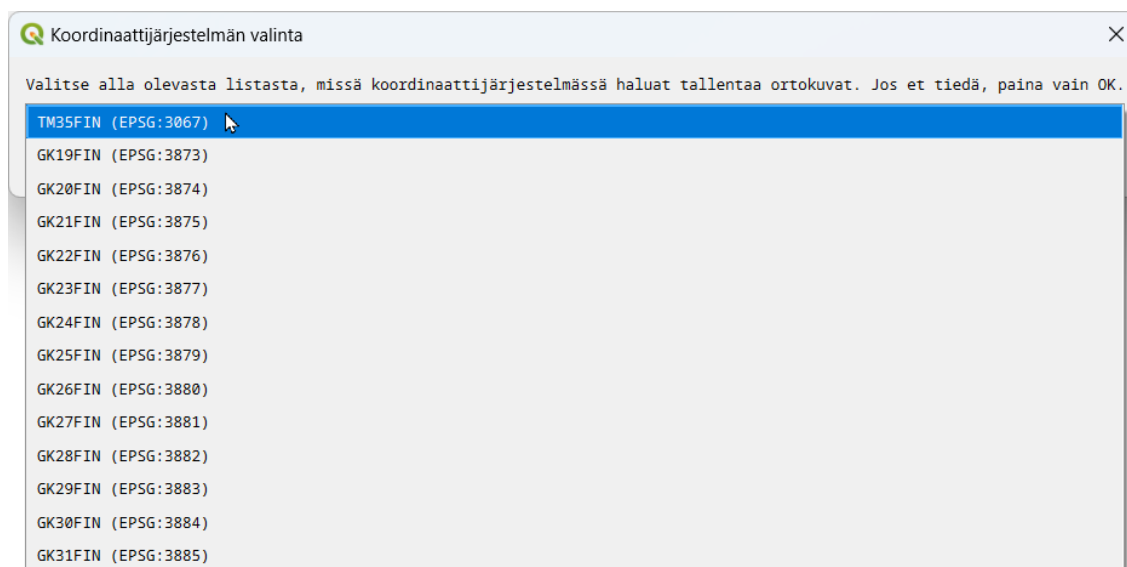
Esimerkkikoodi 5. lista\_koordinaattijarjestelmista-muuttujan sisältö.

lista\_koordinaattijarjestelmista-muuttuja on tarpeen, sillä seuraavaksi toteutettava dialogi-ikkuna odottaa listarakennetta.

Koordinaattijärjestelmän valinta

Hyödyntäen Qt:n QDialog.getItem() -funktiota näytetään käyttäjälle dialogi-ikkuna, jossa pyydetään valitsemaan haluttu koordinaattijärjestelmä ennalta määritellystä listasta. Dialogi-ikkuna hyödyntää juuri luotua lista\_koordinaattijarjestelmista-muuttujaa. Kuvasta 12 ilmenee, miltä kyseinen vaihe näyttää käyttäjälle.

Oletusarvoksi on asetettu maankattava "TM35FIN (EPSG:3067)", joka on turvallinen valinta, ellei käyttäjä tiedä tarvitsevansa toista koordinaattijärjestelmää.



Kuva 12. Käyttäjältä koordinaattijärjestelmää pyytävä ikkuna.

`QinputDialog.getItem`-funktio palauttaa kaksi arvoa. Ensimmäinen on käyttäjän valitsema vaihtoehto merkkijonona, toinen on boolean (`True/False`). Jos dialogi-ikkuna suljetaan OK:ta klikkaamalla, on boolean `True`. Jos dialogi-ikkuna on peruutettu tai suljettu x:stä, on boolean `False`.

Ensimmäinen palautettu arvo säilötään `valitun_nimi`-nimiseen muuttujaan, boolean säilötään `valitun_nimi_b`-nimiseen muuttujaan. Tämän jälkeen tarkistetaan, että `valitun_nimi_b`-muuttuja on `True` (käyttäjä on klikannut OK:ta).

Jos ehto ei täyty, toiminta keskeytyy ja lisäosa siirtyy suoraan viimeiseen siivousvaiheeseen. Jos ehto täyttyy, lisäosa jatkaa toimintaansa.

`valitun_nimi`-muuttuja sisältää siis ihmiselle ymmärrettävän koordinaattijärjestelmän nimen, esimerkiksi "TM35FIN (EPSG:3067)". Tämä on myös sama, kuin aiemmin luodun `tuetut_koordinaattijarjestelmat`-sanakirjan avain kyseiselle koordinaattijärjestelmälle.

Avaimen avulla haetaan sanakirjasta kyseisen koordinaattijärjestelmän arvo, ja tallennetaan se erilliseen `valittu_koordinaattijarjestelma`-nimiseen muuttujaan. `valittu_koordinaattijarjestelma`-muuttujasta löytyy nyt valitun koordinaattijärjestelmän tunniste (ensimmäinen alkio), sekä linkki sen määrittelysivulle (toinen alkio). `valittu_koordinaattijarjestelma`-muuttuja sisältää esimerkiksi esimerkikoodissa 6 näkyvät arvot, jos koordinaattijärjestelmäksi on valittu TM35FIN:

```
valittu_koordinaattijarjestelma = ["EPSG:3067",
"http://www.opengis.net/def/crs/EPSSG/0/3067"]
```

Esimerkkikoodi 6. Esimerkki `valittu_koordinaattijarjestelma`-muuttujan sisällöstä.

Ensimmäistä alkioita hyödynnetään QGIS:n koordinaattijärjestelmähallintaan, toista puolestaan Maanmittauslaitoksen rajapintojen kyselyissä.

### Aluevalintaan siirtyminen

Lähtötietovaiheen viimeinen tehtävä on valmistella ja käynnistää sitä seuraava aluevalintavaihe.

QGIS-karttaikkuna käyttää entuudestaan jotakin koordinaattijärjestelmää. Tämä tallennetaan väliaikaisesti muuttujaan nimeltä `alkuperainen_koordinaattijarjestelma`. Näin voidaan lisäosan toiminnan loppuksi palauttaa karttaikkunan alkuperäinen koordinaattijärjestelmä käyttöön.

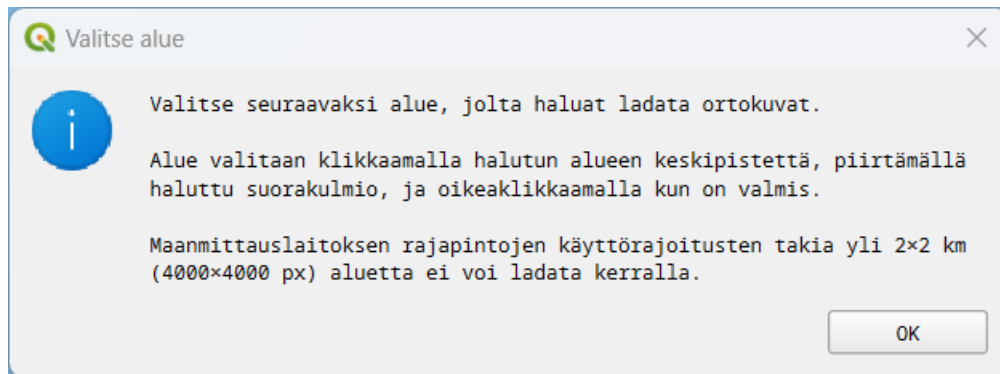
Seuraavaksi karttaikkuna asetetaan käyttämään juuri valittua koordinaattijärjestelmää, hyödyntäen koordinaattijärjestelmän tunnistetta (muuttujan `valittu_koordinaattijarjestelma` ensimmäinen alkio). Alkuperäisen koordinaattijärjestelmän tallentaminen ja uuden asettaminen näkyvät esimerkikoodissa 7.

```
self.alkuperainen_koordinaattijarjestelma=QgsProject.instance().crs()
QgsProject.instance().setCrs(QgsCoordinateReferenceSystem(self.valittu_koordinaattijarjestelma[0]))
```

Esimerkkikoodi 7. QGIS-karttaikkunan alkuperäinen koordinaattijärjestelmä tallennetaan, jonka jälkeen karttaikkuna asetetaan käyttämään valittua koordinaattijärjestelmää.

Tämän myötä hiiren sijainti karttaikkunassa saadaan helposti muunnettua asetetun koordinaattijärjestelmän koordinaateiksi. Tätä hyödynnetään aluevalintavaiheessa.

Käyttäjälle näytetään nyt infoikkuna (kuva 13) Qt:n QMessageBox.information-funktion avulla. Ikkunassa opastetaan, miten aluevalinta käytännössä tehdään.



Kuva 13. Aluevalintaan opastava infoikkuna.

Aluevalintavaihe on toteutettu QGIS:n karttatyökaluna. Sillä on oma luokkansa (aluevalintaluokka), joka perii QgsMapTool-luokan.

aluevalintaluokka-luokasta luodaan nyt ilmentymä. Se yhdistetään QGIS:n karttaikkunaan, ja sijoitetaan aluevalintatyökalu-nimiseen muuttujaan. Tämän jälkeen kyseinen aluevalintatyökalu asetetaan QGIS:n aktiiviseksi karttatyökaluksi. Nämä vaiheet näkyvät esimerkkikoodissa 8.

```
self.aluevalintatyokalu = self.aluevalintaluokka(
self iface.mapCanvas(), self)
self iface.mapCanvas().setMapTool(self.aluevalintatyokalu)
```

Esimerkkikoodi 8. aluevalintaluokan ilmentymän luominen, tallentaminen muuttujaan ja asettaminen aktiiviseksi karttatyökaluksi.

Aluevalintatyökalu on nyt aktivoitu, ja lähtötietovaihe on tullut päätökseen.

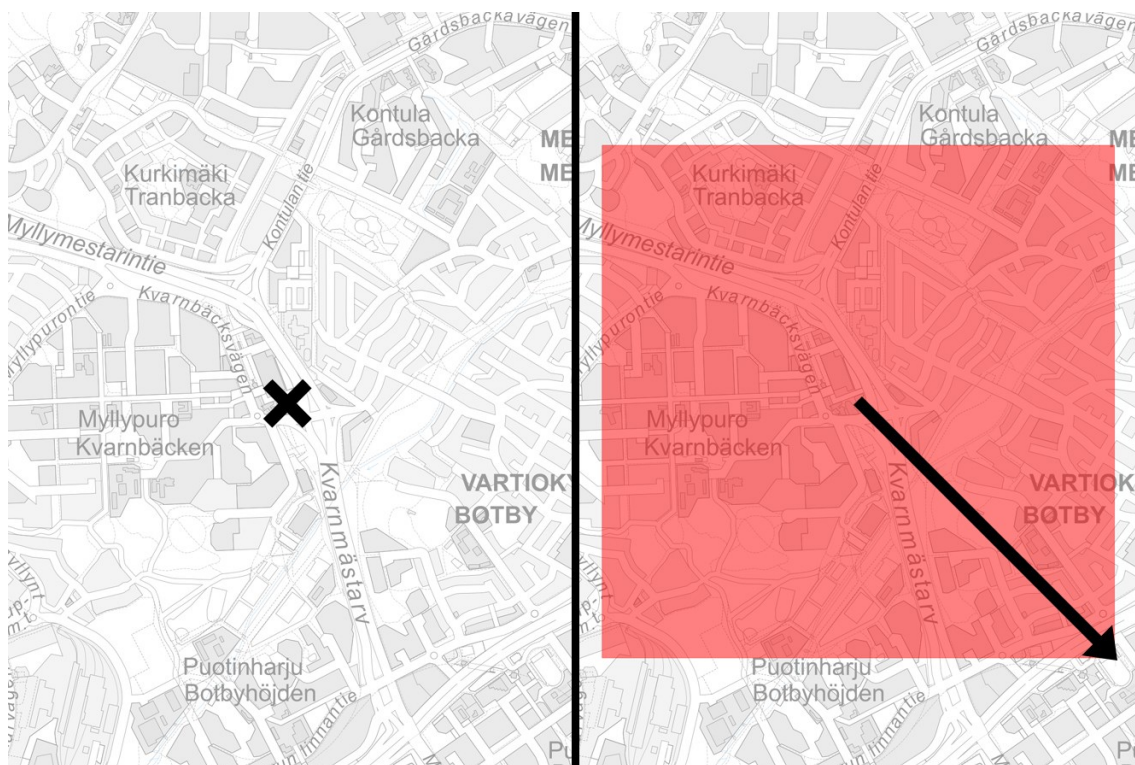


## 5.4 Aluevalinta

Aluevalintavaiheessa käyttäjä valitsee alueen, jolta ortokuvia ladataan. Tämä on toteutettu siten, että:

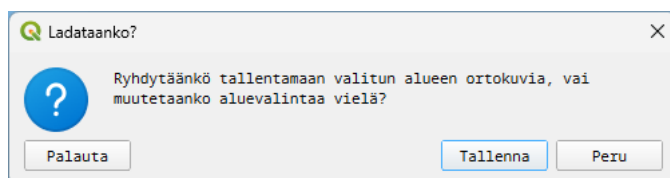
- Käyttäjä klikkaa halutun alueen keskipistettä hiiren vasemmalla painikkeella.
- Käyttäjä liikuttaa hiirtä, jolloin luodaan aluevalinta keskipisteen ympärille ja visualisoidaan se.
- Käyttäjä klikkaa hiiren oikealla painikkeella, kun on tyytyväinen aluevalintaan, mikä avaa dialogi-ikkunan.

Aluevalinnan toimintaperiaate on visualisoitu kuvassa 14.



Kuva 14. Käyttäjä klikkaa haluttua keskipistettä hiiren vasemmalla painikkeella (vasen kuva). Kun hiiri tämän jälkeen liikkuu, päivittyä aluevalinta automaattisesti keskipisteen ympärille, ja valittu alue visualisoidaan punaisena (oikea kuva). Taustakartta Helsingin kaupunki, opaskartta harvanimi harmaa, haettu 03/2024.

Aluevalintavaiheessa hiiren oikeata painiketta klikkaamalla avautuva dialogi-ikkuna näkyy puolestaan kuvassa 15.



Kuva 15. Aluevalintavaiheen dialogi-ikkuna.

Mikäli käyttäjä on tyytyväinen aluevalintaan, etenee lisäosa seuraavaan eli latausvaiheeseen. Mikäli käyttäjä peruuttaa, lisäosan toiminta keskeytyy ja siirrytään suoraan viimeiseen siivousvaiheeseen. Mikäli käyttäjä vielä haluaa muokata aluetta, ei tapahdu mitään, vaan palataan aluevalintaan.

#### Toteutus

Aluevalintavaihe on toteutettu omana luokkana (aluevalintaluokka). Kyseessä on QGIS:n karttatyökalu, joka perii QgsMapTool-luokan ominaisuudet.

Aluevalintaluokka sisältää neljä funktiota:

- `__init__` valmistelee aluevalintatyökalun käyttöä varten yhdistämällä se karttaikkunaan. Se luo samalla tarvittavia muuttujia.
- `canvasPressEvent` on tapahtumakäsittelijä, joka määrittelee toiminnallisuuden hiiren painalluksille.
- `canvasMoveEvent` on tapahtumakäsittelijä, joka määrittelee mitä tapahtuu, kun hiirtä liikutetaan karttanäkymässä.
- `lopeta_aluevalinta` toteuttaa kuvan 13 dialogi-ikkunan ja sen toiminnot.

Varsinainen toiminta alkaa klikkaamalla hiiren vasenta painiketta. Tällöin `canvasPressEvent`-funktio lukee hiiren sen hetkisen sijainnin karttaikkunassa, muuntaa sijainnin pikseleistä maantieteellisiin koordinaatteihin, ja tallentaa maantieteellisen sijainnin muuttujaan nimeltä `keskipiste`. Tämä toistuu aina, kun hiiren vasenta painiketta klikataan, joten keskipistettä voi muuttaa helposti.

Klikkauksen sijainnin lukeminen, muuntaminen ja tallentaminen tapahtuu samalla tavalla kuin esimerkikoodissa 9, liikkuvan hiiren sijaintia seurattaessa. Sitä ei siksi esitetä tässä erikseen.

Kun keskipiste kerran on määritelty, alkaa aluevalintatyökalu canvasMoveEvent-funktion avulla seurata hiiren liikkeitä karttaikkunassa. Aina, kun liikettä havaitaan, ajetaan esimerkikoodi 9.

```

pikselisijainti_x = event.pos().x()
pikselisijainti_y = event.pos().y()
hiiren_sijainti = self.canvas.getCoordinateTransform()
.toMapCoordinates(pikselisijainti_x, pikselisijainti_y)
alueen_leveys = 2*abs(self.keskipiste.x()-hiiren_sijainti.x())
alueen_korkeus = 2*abs(self.keskipiste.y()-hiiren_sijainti.y())
if alueen_leveys > 2000:
    alueen_leveys = 2000
if alueen_korkeus > 2000:
    alueen_korkeus = 2000
suorakulmio = QgsRectangle().fromCenterAndSize(
self.keskipiste, alueen_leveys, alueen_korkeus)
self.polygoni = QgsGeometry.fromRect(suorakulmio)
self.aluevisualisointi.reset()
self.aluevisualisointi = QgsRubberBand(self.plugin.iface.mapCanvas(),
Qgis.GeometryType.Polygon)
self.aluevisualisointi.setColor(QColor(255, 0, 0, 127))
self.aluevisualisointi.addGeometry(self.polygoni)
self.plugin.iface.mapCanvas().refresh()

```

**Esimerkkikoodi 9. canvasMoveEvent-funktion toiminnallisuus.**

Funktio lukee hiiren sijainnin karttaikkunassa, muuntaa sen pikseleistä maantieteellisiin koordinaatteihin ja tallentaa maantieteellisen sijainnin muuttujaan hiiren\_sijainti.

Tämän jälkeen funktio laskee etäisyyden keskipisteen x-koordinaatista hiiren sijainnin x-koordinaattiin, ja vastaavasti keskipisteen y-koordinaatista hiiren sijainnin y-koordinaattiin. Ottamalla näiden itseisarvot ja kertomalla ne kahdella saadaan valitun alueen leveys ja korkeus metreissä. Molemmat kuitenkin rajoitetaan 2 000 metriin, sillä Maanmittauslaitoksen WCS-rajapinta ei palauta 2×2 km suurempia alueita.

Hyödyntämällä keskipisteen sijaintia sekä alueen leveyttä ja korkeutta laaditaan alueesta QgsRectangle-objekti, joka tallennetaan muuttujaan suorakulmio.

Suorakulmiosta puolestaan laaditaan QgsGeometry-objekti, joka tallennetaan muuttujaan polygoni.

Seuraavaksi alue visualisoidaan. Tämä tapahtuu QgsRubberBand-luokan avulla, joka on tarkoitettu väliaikaisiin karttapohjavisualisointeihin. Ensin aiempi visualisointi resetoidaan, jonka jälkeen tilalle laaditaan uusi. Tämä määritetään punaiseksi ja läpikuultavaksi, minkä jälkeen sille annetaan tuore polygoni-muuttuja visualisoitavaksi, ja päivitetään karttapohja.

canvasMoveEvent-funktio ajetaan joka kerta kun hiiressä havaitaan liikettä, kymmeniä kertoja sekunnissa. Käyttäjää näkee ainoastaan, kuinka punainen alue muuttuu hiirtä liikuttaessa.

Kun hiiren oikeata painiketta klikataan, käynnistyy lopeta\_aluevalinta-funktio. Tämän myötä aluevalinta ei enää päivity. Käyttäjälle näytetään kuvan 15 dialogi-ikkuna, hyödyntäen Qt:n QMessageBox.question-funktiota.

Jos käyttäjä dialogi-ikkunassa ilmaisee olevansa tyytyväinen aluevalintaan ja haluavansa siirtyä lataamaan ortokuvia, ajetaan esimerkkikoodi 10:

```
global bbox_pyoristetty
bbox_pyoristetty = self.polygoni.boundingBox().snappedToGrid(1)
self.aluevisualisointi.reset()
self.plugin.iface.actionPan().trigger()
self.plugin.lataus()
return
```

Esimerkkikoodi 10. Aluevalintavaiheesta latausvaiheeseen siirtyminen.

Esimerkkikoodissa 10 laaditaan ensin globaali muuttuja bbox\_pyoristetty. Valittu alueen koordinaatit pyöristetään lähimpään täyteen metriin, jonka jälkeen valittu alue tallennetaan laadittuun bbox\_pyoristetty-muuttujaan. Sillä laadittu muuttuja on globaali, on valittu alue nyt hyödynnettävissä myös aluevalintatyökalun ulkopuolella, latausvaiheessa.

Punainen aluevisualisointi resetoidaan. Aktiiviseksi karttatyökaluksi asetetaan QGIS:n oma "Vieritä karttaa" (Pan Map) -työkalu, joka samalla deaktivoi aluevalintatyökalun. Lopuksi käynnistetään latausvaihe.

## 5.5 Lataus

Latausvaiheessa:

- Kootaan OGC API: Features -kyselyjen osoitteet.
- Tarkistetaan OGC API: Features -rajapinnasta, miltä ajankohdilta ortokuvia on saatavilla. Ajankohdat tallennetaan listoihin.
- Kootaan WCS-kyselyjen osoitteet.
- Haetaan WCS-rajapinnasta itse ortokuvat, hyödyntämällä listoissa olevia ajankohtia. Tallennetut ortokuvat myös lisätään QGIS:iin.

OGC API: Features -kyselyjen kokoaminen

Taulukosta 2 (luku 3) ilmenee OGC API: Features -kyselyn rakenne. Vastaavat kyselyt kootaan nyt lisäosassa.

Osittain kysely on muuttumaton ja kovakoodattu lisäosaan merkkijonoina. Muuttuvat osat kyselystä lisäosa on puolestaan selvittänyt. Lähtötietovaiheessa käyttäjältä kysyttiin API-avainta ja koordinaattijärjestelmää, jotka tallennettiin muuttujiin `api_avain` sekä `valittu_koordinaattijarjestelma`. Aluevalintavaiheessa määritettiin koordinaattirajaus halutulle alueelle, joka tallennettiin muuttujaan `bbox_pyoristetty`.

Ketjuttamalla kiinteät ja selvitettyt osat saadaan merkkijonojen yhdistämisellä laadittua lopulliset OGC API: Features -kyselyt. Tämä vaihe näkyy esimerkkikoodissa 11.

```

f_url_mv = 'https://avoin-karttakuva.maanmittauslaitos.fi/ortokuvat-
ja-korkeusmallit/features/v2/collections/ortokuva_mustavalko/items?'
f_url_vari = 'https://avoin-karttakuva.maanmittauslaitos.fi/ortokuvat-
ja-korkeusmallit/features/v2/collections/ortokuva_vari/items?'
f_url_bbox = 'bbox='+str(bbox_pyoristetty.xMinimum())+', '
+str(bbox_pyoristetty.yMinimum())+', '+str(bbox_pyoristetty.xMaximum())
+', '+str(bbox_pyoristetty.yMaximum())
f_url_crs = '&bbox-crs='+self.valittu_koordinaattijarjestelma[1]
f_url_api = '&api-key='+self.api_avain
f_url_format = '&f=json'
ajankohdat_url_mv = f_url_mv+f_url_bbox+f_url_crs+f_url_api
+f_url_format
ajankohdat_url_vari = f_url_vari+f_url_bbox+f_url_crs+f_url_api
+f_url_format

```

**Esimerkkikoodi 11. OGC API: Features -kyselyt laaditaan merkkijonojen yhdistämisellä.**

Muuttuja `api_avain` on merkkijono, ja sellaisenaan yhdistettävissä kyselyyn.

Muuttuja `valittu_koordinaattijarjestelma` on listarakenne. Siitä tarvitaan toinen alio ( `valittu_koordinaattijarjestelma[1]` ), joka sisältää linkin koordinaattijärjestelmän määrittelysivulle.

Muuttuja `bbox_pyoristetty` on QGIS:n `QgsRectangle`-objekti. Siitä saadaan poimittua kyselyssä tarvittu koordinaatit (vasen, ala, oikea, ylä) hyödyntämällä metodeja `xMinimum`, `yMinimum`, `xMaximum` sekä `yMaximum`.

Kyselyjä laaditaan kaksi (muuttujat `ajankohdat_url_mv` sekä `ajankohdat_url_vari`), sillä Maanmittauslaitos säilyttää mustavalkoisia ortokuva-aineistoja ja väriortokuva-aineistoja erikseen.

**Ajankohtien tarkistus**

Ajankohtien tarkistus tapahtuu `tarkista_ajankohdat`-funktiolla, joka on kokonaisuudessaan esimerkkikoodissa 12.

```
def tarkista_ajankohdat(ajankohtahaku_url):
    itse_sivu = urlopen(ajankohtahaku_url)
    sivun_sisalto = itse_sivu.read()
    sisalto_json = json.loads(sivun_sisalto)
    ajankohtalista = []
    for i in sisalto_json['features']:
        ajankohtalista.append(i['properties']['time'])
    ajankohtalista = list(set(ajankohtalista))
    ajankohtalista.sort()
    return ajankohtalista
```

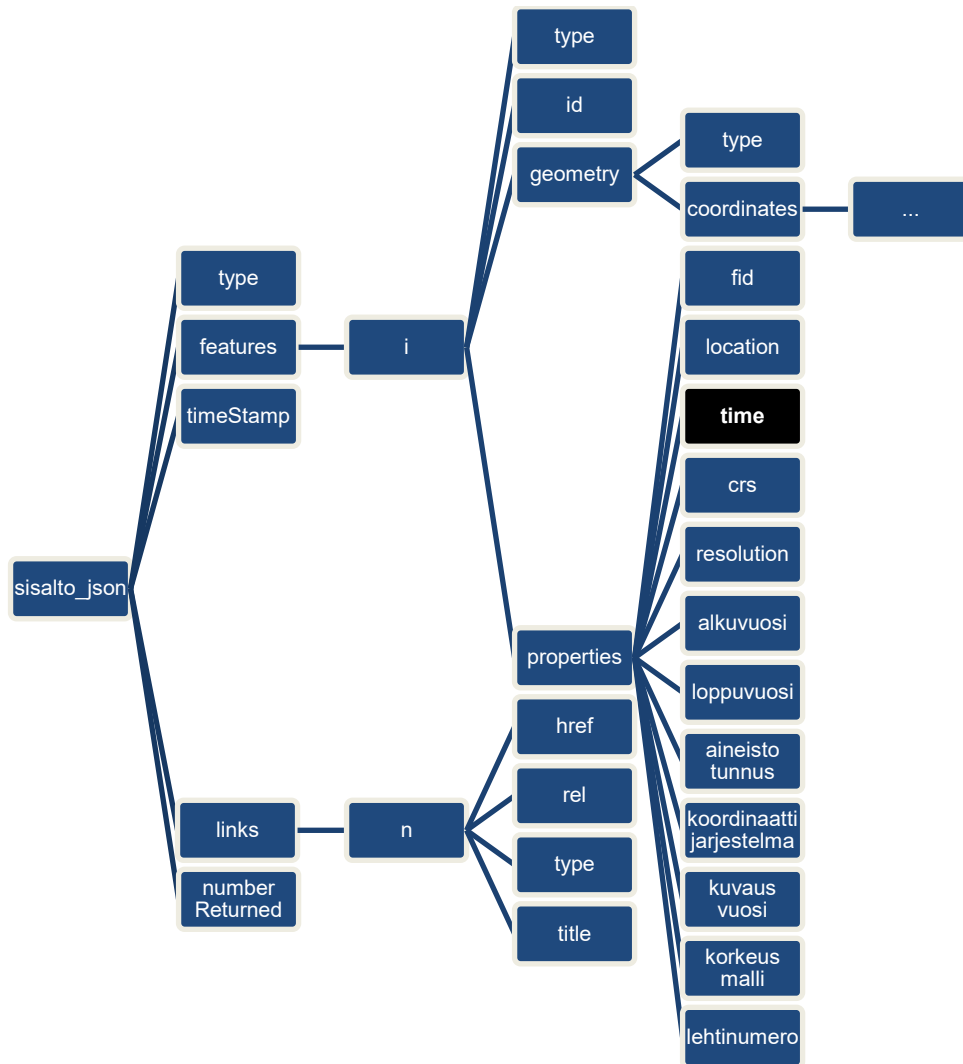
Esimerkkikoodi 12. Ajankohtien tarkistus OGC API: Features -palvelimelta. tarkista\_ajankohdat-funktio ottaa argumentiksi juuri kootun OGC API: Features -kyselyn, ja se on ajettava erikseen mustavalkoisille ja värillisille ortokuville.

Funktio tekee urllib.request-kirjaston urlopen-toiminnolla kyselyn OGC API: Features -palvelimelle. Vastaus tallennetaan itse\_sivu-muuttujaan. Muuttuja itse\_sivu luetaan, ja tallennetaan merkkijonomuodossa sivun\_sisalto-muuttujaan.

Esimerkkikoodissa 3 (luku 3) on ote OGC API: Features -palvelimen vastauksesta. Saman tyyppinen, mutta huomattavasti pidempi, vastaus saadaan nyt.

Seuraavaksi merkkijonona oleva vastaus (sivun\_sisalto) jäsennetään, json-kirjastoa hyödyntäen, ja tallennetaan sisalto\_json-muuttujaan. Kun sisältö saadaan jäsennettyä tietorakenteeseen, voidaan sitä hyödyntää ohjelmallisesti.

Jäsennetyn OGC API: Features -vastauksen rakenne ilmenee kuvasta 16.



Kuva 16. Jäsennetyn OGC API: Features -vastauksen rakenne. Jäsennettyä vastausta on mahdollista navigoida. Esimerkiksi time-attribuuttiin pääsee käsiksi hierarkiaa etenemällä: `sisalto_json['features']['i']['properties']['time']`.

Kun vastaus on jäsennetty tietorakenteeseen, sitä voi iteroida for-logiikalla. Jokaisen yksittäisen palautetun ortokuvan (i) time-attribuutti (korostettu) lisätään näin ajankohtalista-muuttujaan.

Lopuksi ajankohtalista-muuttujasta poistetaan mahdolliset duplikaatit, ja se järjestetään kronologisesti. Tulos on esimerkkikoodin 13 tapainen.



```
ajankohtalista = ['1932-12-31T02:00:00Z', '1950-12-31T02:00:00Z',  
'1956-12-31T02:00:00Z', '1964-12-31T02:00:00Z', '1965-12-  
31T02:00:00Z', '1996-12-31T02:00:00Z', '1999-12-31T02:00:00Z', '2004-  
12-31T02:00:00Z', '2005-12-31T02:00:00Z', '2006-12-31T02:00:00Z']
```

**Esimerkkikoodi 13. OGC API: Features -palvelimen palauttamia ortokuvien ajankohtia järjestettyinä.**

Lista ajankohdista palautetaan, ja tallennetaan ajankohtalista\_mv-muuttujaan. Sama toistetaan väriortokuville, joiden ajankohdat puolestaan tallennetaan ajankohtalista\_vari-muuttujaan.

Nämä kaksi listaa sisältävät siis ne ajankohdat, joilta on digitoituja ortokuvia halutulta alueelta. Ajankohtia hyödynnetään rasteriaineiston lataamiseen WCS-palvelimelta.

WCS-kyselyjen kokoaminen

Taulukosta 1 (luku 3) ilmenee WCS-kyselyn rakenne. Vastaavat kyselyt koostaan nyt lisäosassa. Tämä tehdään samalla tavalla, kuin OGC API: Features -kyselyt koottiin. Kovakoodatut osat kyselystä yhdistetään lisäosan selvittämiin muuttuviin osiin.

Myös WCS-kyselyjä on laadittava kaksi: yksi mustavalkoisille ortokuville, toinen väriortokuville. WCS-kyselyihin ei kuitenkaan vielä tässä vaiheessa lisätä ajankohtaparametria.

Kuvien tallennus ja lisäys QGIS:iin

Ennen ortokuvien lataamista laaditaan QGIS:n tasovalintäkymään uusi ryhmä, johon ladatut ortokuvat sijoitetaan. Ryhmän nimeksi tulee se tiedostonimi, jonka käyttäjä antoi lähtötietovaiheessa. Tämä vaihe löytyy esimerkkikoodista 14.

```
QgsProject.instance().layerTreeRoot().insertGroup(0, self.nimi)
self.tasoryhma = QgsProject.instance().layerTreeRoot()
.findGroup(self.nimi)
```

**Esimerkkikoodi 14.** Uuden ryhmän luominen QGIS:n tasovalintänäkömään. Viittaus laadittuun ryhmään tallennetaan tasoryhma-nimiseen muuttujaan.

**Varsinainen kuvien lataus ja lisääminen QGIS:iin tapahtuu esimerkkikoodilla 15:**

```
def tallenna_kuvat(ajankohtalista, kuvahaku_url):
    for i in ajankohtalista:
        wcs_url_aika = '&SUBSET=time(%22'+i+'%22)'
        tiedostonimi = self.nimi+'_'+i[:4]+'.tiff'
        tiedoston_polku = self.tallennuskansio+tiedostonimi
        urlretrieve(kuvahaku_url+wcs_url_aika, tiedoston_polku)
        if os.path.getsize(tiedoston_polku) > 20000:
            rasteritaso = QgsRasterLayer(tiedoston_polku,
str(tiedostonimi), "gdal")
            QgsProject.instance().addMapLayer(rasteritaso, False)
            self.tasoryhma.insertChildNode(0,
QgsLayerTreeLayer(rasteritaso))
            rasteritaso_node = QgsProject.instance().layerTreeRoot()
            .findLayer(rasteritaso)
            rasteritaso_node.setExpanded(False)
            rasteritaso_node.setItemVisibilityChecked(False)
        else:
            os.remove(tiedoston_polku)
```

**Esimerkkikoodi 15.** Ortokuvat lataava ja QGIS:iin lisäävä funktio.

Funktio ottaa argumenteikseen aiemmin laaditun listan saatavilla olevista ajankohdista (ks. esimerkkikoodi 13), sekä WCS-kyselyn osoitteen (ilman ajankoh-  
taparametria). Funktio ajetaan erikseen mustavalkoisille ja värillisille ortokuville,  
kumpikin omilla ajankohdillaan ja WCS-kyselyosoitteillaan.

Jokaiselle annetussa ajankohtalistassa olevalle ajankohdalle funktio tekee seuraavan:

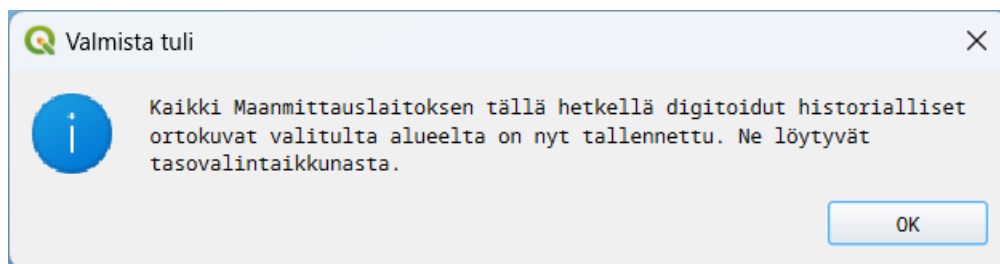
- Laatii WCS-kyselyn loppuun ajankohtaparametrin, jossa on kyseinen ajankohta.
- Määrittelee nimen tallennettavalle ortokuvulle. Nimi koostuu käyttäjän lähtötietovaiheesta antamasta tiedostonimestä, johon lisätään nyt ladattavan ortokuvan vuosiluku sekä tiedostopääte (esimerkiksi "Metropolian\_ymparisto\_1932.tiff").
- Lataa urllib.request-kirjaston urlretrieve-toimintoa hyödyntäen kyseisen ajankohdan ortokuvan WCS-rajapinnasta, ja tallentaa sen

määritellyllä nimellä käyttäjän lähtötietovaiheessa määrittelemään tallennuskansioon.

- Tarkistaa, mikäli juuri ladattu tiedosto on yli 20 kilotavua suuri.
- Lisää ladatun tiedoston rasteritasona avoimeen QGIS-projektiin sekä tasovalintanäkymään laadittuun ryhmään, jos se on yli 20 kilotavua suuri.
- Poistaa ladatun tiedoston, jos se puolestaan on alle 20 kilotavua suuri. Hyvin pienikokoisissa rasteritiedostoissa on ainoastaan yksittäisiä pikseleitä aineistoa. Näitä saattaa ilmestyä aivan ilmakuvattujen alueiden rajoilla.

Käytännössä tallenna\_kuvat-funktio ikään kuin liimaa OGC API: Features -rajapinnasta selvitetyn ajankohdan muuten valmiiseen WCS-kyselyyn, ja lataa kyseisen ortokuvan. Tämä toistetaan, kunnes kaikki listassa olevat ajankohdat on ladattu.

Kun sekä mustavalkoiset että värilliset ortokuvat on ladattu onnistuneesti, tiedotetaan käyttäjää tästä infoikkunalla (kuva 17). Infoikkuna käyttää samaa QMessageBox.information-funktiota, jota hyödynnettiin aluevalintavaiheen käytön opastuksessa.



Kuva 17. Lisäosan käyttäjälle näyttämä infoikkuna, kun lataus on päättynyt.

Lopuksi siirrytään viimeiseen siivousvaiheeseen.

## 5.6 Siivous

Siivousvaiheessa QGIS:n karttaikkuna asetetaan käyttämään samaa koordinaattijärjestelmää, joka sillä oli ennen lisäosan käynnistämistä. Tähän

hyödynnetään alkuperäinen\_koordinaattijärjestelmä-muuttujaa, joka laadittiin ennen koordinaattijärjestelmän vaihtamista.

Lisäksi poistetaan globaali muuttuja bbox\_pyoristetty, joka aluevalintavaiheessa luotiin. Tämän myötä lisäosan toiminta on päättynyt.

## 6 QGIS-lisäosan tiedostot, jakelu, asennus ja käyttö

### 6.1 Tiedostot

QGIS-lisäosan hakemistosta on löydyttävä vähintään kolme tiedostoa. Päätoiminnallisuuden sisältävän tiedoston lisäksi vaaditaan kaksi oheistiedostoa (metadata.txt sekä \_\_init.py\_\_). [36.]

Insinööriyössä toteutettu lisäosa on toiminnallisuuksiltaan alkeellinen, ja tarkoitettu rajoitettuun käyttöön. Tämän takia se käyttää vain näitä kolmea tiedostoa. Toiminnallisuuksiltaan laajemmat lisäosat koostuvat kymmenistä tai sadoista tiedostoista. QGIS:n julkiseen lisäosarepositorioon julkaistaville lisäosille on myös huomattava määrä lisävaatimuksia, joita ei käsitellä tässä.

metadata.txt

metadata.txt on tekstipohjainen tiedosto, joka sisältää lisäosaa kuvailevaa metatietoa. QGIS:n käyttöliittymä hyödyntää tiedostoa esimerkiksi lisäosia hallinnoimassa, näyttäen käyttäjälle asennetun lisäosan nimen, version ja kuvauksen. [36.]

Vähimmäisvaatimus metadata.txt-tiedoston sisällölle näkyy esimerkkikoodissa 16. Insinööriyössä toteutetun QGIS-lisäosan metadata.txt-tiedosto sisältyy liitteeseen 1.

```
[general]
name=Lisäosan nimi (esimerkiksi Ortokuvalataaja)
version=Lisäosan versio (esimerkiksi 1.0)
description=Lyhyt, noin lauseen pituinen kuvaus lisäosasta.
about=Pidempi kuvaus lisäosasta ja sen toiminnallisuuksista.
author=Lisäosan laatija
email=Laatijan sähköpostiosoite
qgisMinimumVersion=Vanhin QGIS, jolla lisäosa toimii (esim. 3.0)
repository=verkko-osoite, josta lisäosan lähdekoodi löytyy
```

Esimerkkikoodi 16. metadata.txt-tiedoston vähimmäisvaatimus.

`__init.py__`

`__init.py__` on Python-alustustiedosto. Tämän niminen tiedosto viestii, että kyseinen hakemisto sisältää Python-koodia. Tiedosto sisältää useimmiten vain lyhyehkön koodipätkän. Koodin avulla haetaan itse päätoiminnallisuus, joka sijaitsee toisessa tiedostossa samassa hakemistossa. [36.]

QGIS edellyttää, että lisäosan `__init.py__`-tiedostosta löytyy ainakin `classFactory`-niminen funktio. Esimerkki `classFactory`-funktioista näkyy esimerkkikoodissa 17. Insinööriyössä toteutetun QGIS-lisäosan `__init.py__`-tiedosto sisältyy liitteeseen 1.

```
def classFactory(iface):
    from .paatiedosto import paatoiminnallisuus
    return paatoiminnallisuus(iface)
```

Esimerkkikoodi 17. QGIS-lisäosan `__init.py__`-tiedoston vähimmäisisältö.

Yksinkertaistetusti tämä koodi kertoo QGIS:lle, että se löytää lisäosan päätoiminnallisuuden `paatiedosto`-nimisestä tiedostosta, joka sijaitsee samassa hakemistossa.

`ortokuvaalataaja.py`

Kolmas tiedosto sisältää lisäosan varsinaisen toiminnallisuuden. Kaikki luvussa 5 (QGIS-lisäosan toteutus) selostettu tapahtuu tässä päätiedostossa. Päätiedoston voi nimetä vapaasti, kunhan vastaavaa nimeä käyttää alustustiedostossa. Insinööriyössä laadittu päätiedosto (`ortokuvaalataaja.py`) sisältyy liitteeseen 1.

## 6.2 Jakelu

Toteutettu QGIS-lisäosa jaetaan ja asennetaan zip-tiedostona. Lisäosan zip-tiedosto laaditaan pakkaamalla se hakemisto, joka sisältää lisäosan tarvitsemat kolme tiedostoa (`metadata.txt`, `__init.py__`, `ortokuvaalataaja.py`). Pakkaus on suoritettava juuri tiedostot sisältävälle hakemistolle, ei itse tiedostoille.

### 6.3 Asennus

Lisäosan asennus QGIS:iin tapahtuu graafisen käyttöliittymän kautta, kohdasta Lisäosat – Hallitse ja asenna lisäosia – Asenna ZIP-tiedostosta (Plugins – Manage and install plugins – Install from ZIP). Oikea zip-tiedosto valitaan, jonka jälkeen klikataan Asenna lisäosa (Install Plugin). Myös lisäosan poistaminen tapahtuu QGIS:n graafisen käyttöliittymän kautta, kohdasta Lisäosat – Hallitse ja asenna lisäosia – Asennetut (Plugins – Manage and install plugins – Installed).

Lisäosa asentuu tyypillisesti johonkin seuraavista sijainneista, käyttöjärjestelmän mukaan.

- C:\Users\kayttajanimi\AppData\Roaming\QGIS\QGIS3\profiles\default\python\plugins\ (Windows)
- /home/kayttajanimi/.local/share/QGIS/QGIS3/profiles/default/python/plugins/ (Linux)
- /Users/kayttajanimi/Library/Application Support/QGIS/QGIS3/profiles/default/python/plugins/ (Mac)

### 6.4 Käyttö

Lisäosan asennuksen myötä QGIS:n käyttöliittymään ilmestyy ”LATAA ORTO-KUVIA” -painike. Tämä näkyy kuvassa 9 (luku 5). Lisäosa käynnistyy painiketta klikkaamalla. Ennen lisäosan käynnistämistä on QGIS:iin lisättävä tarkoituksenmukainen taustakartta, jotta halutun alueen valinta olisi mahdollista.

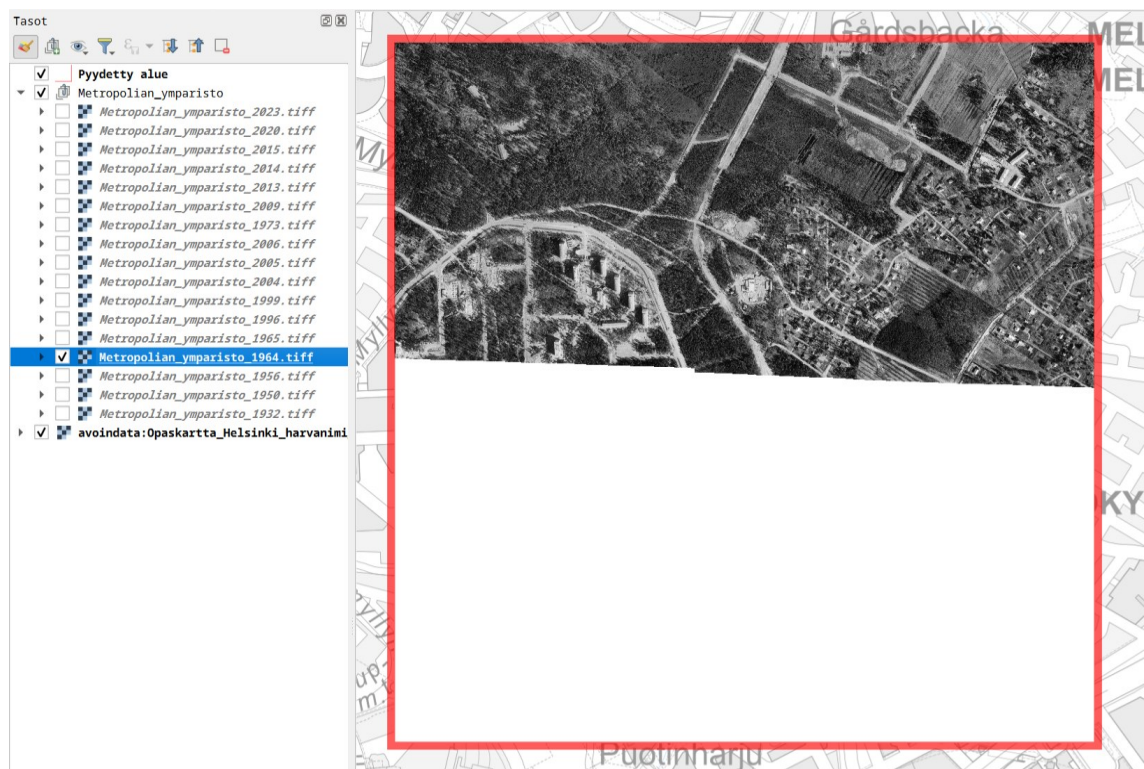
Lisäosan käyttö vaatii API-avaimen Maanmittauslaitoksen paikkatietorajapintoihin. API-avain on ilmainen. Osoitteesta <<https://www.maanmittauslaitos.fi/rajapinnat/api-avaimen-ohje>> löytyy ohjeet API-avaimen luomiseen.

Lisäosan käynnistyttyä se pyytää ensin syöttämään tai valitsemaan tarvitsemansa lähtötiedot. Nämä ovat API-avain, tiedostonimi, tallennuskansio, sekä haluttu koordinaattijärjestelmä. Koordinaattijärjestelmäksi kannattaa valita oleksena oleva TM35FIN, ellei tiedä tarvitsevasa muuta.

Seuraavaksi määritellään alue, jolta ortokuvia haluaa ladata. Se tapahtuu klikkaamalla halutun alueen keskipistettä hiiren vasemmalla painikkeella, piirtämällä suorakulmio, ja lopuksi klikkaamalla hiiren oikeata painiketta. Yli 2×2 km suuruisia alueita ei ole mahdollista ladata kerralla, Maanmittauslaitoksen asettamien rajoitusten takia.

Tämän jälkeen lisäosa tarkistaa saatavilla olevat historialliset ortokuvat, ja lataa ne. Kyseinen vaihe on näkymätön käyttäjälle. QGIS lakkaa hetkeksi reagoimasta, kunnes ilmoittaa latauksen onnistuneen. Tyypillisesti lataus kestää joitain kymmeniä sekunteja.

Ladatut tiedostot löytyvät aiemmin valitusta tallennuskansiosta, jonka lisäksi ne ilmestyvät QGIS:n tasovalintänäkymään kuvan 18 esittämällä tavalla.



Kuva 18. Ladatut ortokuvat ilmestyvät QGIS:n tasovalintänäkymään latauksen valmistuttua. Osa ladatusta ortokuvasta voi ajoittain olla tyhjä, kuten tässä esimerkissä. Kyseisenä vuonna ilmakuvatun alueen raja on sattunut ladatulle alueelle, ja aineistoa on saatavilla vain pohjoispuolelta. Maanmittauslaitoksen ortokuva, haettu 03/2024. Taustakartta Helsingin kaupunki, opaskartta harvanimi harmaa, haettu 03/2024.



## Ongelmatilanteet

Virheen sattuessa lisäosa pyrkii näyttämään dialogi-ikkunan, jossa on lisätietoa kohdatusta ongelmasta. Mikäli ongelman havaitaan johtuvan suoraan käyttäjältä, esimerkiksi puuttuvan lähtötiedon tapauksessa, tämä ilmaistaan selvästi. Teknisille virheille ei kuitenkaan ole toteutettu kattavaa virnehallintaa, ja mahdollinen vianmääritys on suoritettava manuaalisesti.

Vianmäärityksessä on apua QGIS:n Lokiviestit-paneelistä (Log Messages Panel), jonka saa avattua klikkaamalla esimerkiksi "LATAA ORTOKUVIA" -painiketta hiiren oikealla painikkeella. Lokiviestit-paneeli on avattava ennen lisäosan käynnistämistä ja virheen ilmestymistä.

## 7 Yhteenveto

Insinööriyössä pyrittiin laatimaan QGIS:lle lisäosa, jonka avulla Maanmittauslaitoksen historiallisia ortokuvia halutulta alueelta voisi tallentaa tiedostomuodossa. Näin aineisto saataisiin veloituksetta hyödynnettäväksi paikkatieto-ohjelmistoihin. Insinööriyön tavoitteessa onnistuttiin.

Laadittu ortokuva-lataaja-lisäosa toteutettiin Python-ohjelmointikielellä. Ortokuva-aineisto ladataan Maanmittauslaitoksen WCS-rajapinnasta. Ennen latausta tarkistetaan saatavilla oleva aineisto Maanmittauslaitoksen erillisestä OGC API: Features -rajapinnasta.

Vaikka insinööriyössä onnistuttiin luomaan toimiva lisäosa, on sillä kuitenkin rajoitteita, jotka tulee ottaa huomioon.

Ensinnäkin aineistoa löytyy myös muilta toimijoilta kuin Maanmittauslaitokselta. Varsinkin suuremmat kaupungit ovat suorittaneet ilmakuvauksia itsenäisesti. Digitoituja ortokuvia voi siksi olla saatavilla myös muilta vuosilta kuin mitä lisäosa hakee Maanmittauslaitokselta; näitä on etsittävä erikseen. [37.]

Lisäksi Maanmittauslaitoksen ilmakuva-arkiston digitointi on vielä kesken, eikä lisäosa ilmoita digitoimattomasta aineistosta. Mikäli aineistoa tarvitaan vielä digitoimattomilta vuosilta, voi yksittäisten ilmakuvien digitointeja tilata Maanmittauslaitokselta maksua vastaan. Maanmittauslaitoksella on erillinen karttapalvelu, josta voi tarkistaa minä kaikkina vuosina tietty alue on ilmakuvattu [38].

Kolmannen rajoituksen muodostaa metatiedon puute. Lisäosan tallentamista ortokuvista tiedetään ainoastaan kuvausvuosi. Maanmittauslaitoksen vuonna 2013 tekemästä käyttötarkoitusselvityksestä ilmenee, että osalle toimijoista olisi tärkeää tietää lisäksi esimerkiksi kuvan lentokorkeus, käytetty laitteisto ja tarkka ottopäivämäärä [13]. Maanmittauslaitos ei näitä tietoja julkaise. Tarvittaessa kuvien lisätietoja on mahdollista tiedustella, mutta tämäkin on maksullista.

Jotta lisäosan saisi laajemmin käyttöön, pitäisi se julkaista QGIS:n julkiseen lisäosarepositorioon, josta muut käyttäjät voisivat sen helposti löytää ja asentaa. Tämä asettaisi kuitenkin teknisiä lisävaatimuksia lisäosalle, joihin ei tämän insinööriyön puitteissa ole mahdollista paneutua. Jakelu tapahtuu siksi ainoastaan insinööriyön liitteenä 1 olevan zip-tiedoston kautta, mikä rajoittanee sen laajempaa hyödyntämistä.

Historiallisen ortokuva-aineiston laajemman hyödyntämisen kannalta olisi ihanteellista, jos Maanmittauslaitos toteuttaisi aineistolle selainpohjaisen latauspalvelun. Esimerkiksi Ruotsin vastaava viranomainen, Lantmäteriet, tarjoaa jo mahdollisuuden ladata historiallisia ortokuvia veloitusetta tiedostomuodossa metatietoineen [39].

## Lähteet

- 1 Maanmittauslaitoksen ilmakekuva. Verkkoaineisto. Maanmittauslaitos. <<https://www.maanmittauslaitos.fi/kartat-ja-paikkatieto/aineistot-ja-rajapinnat/tuotekuvaukset/ilmakekuva>>. Luettu 25.3.2024.
- 2 Jaakkola, Matti. 1983. Valtakunnallinen ilmakekuvatoiminta. Teoksessa Maanmittaus Suomessa 1633–1983. Helsinki: Maanmittaushallitus, 428–447.
- 3 Löffler, Ernst. 1985. Geographie und Fernerkundung. Wiesbaden: Springer Fachmedien.
- 4 Haggrén, Henrik. 2002. Fotogrammetrinen kartoitus, luento 7. Verkkoaineisto. Aalto-yliopisto. <<https://foto.aalto.fi/opetus/220/luennot/7/7.html>>. 11.10.2002. Luettu 25.3.2024.
- 5 Aerial Photograph vs. Orthoimage. Verkkoaineisto. United States Geological Survey. <<https://www.usgs.gov/media/images/aerial-photograph-vs-orthoimage>>. Luettu 25.3.2024.
- 6 Valkama, Heikki & Huttunen, Samuli. 2021. Katso upeat kuvat: Maanmittauslaitoksen historialliset ilmakekuvat avautuivat suomalaisten käyttöön – kiinnostus ollut heti valtavaa. Verkkoaineisto. Yle. <<https://yle.fi/a/3-11959338>>. 1.6.2021. Luettu 25.3.2024.
- 7 Maanmittauslaitoksen historiallisten ilmakekuvien arkisto avautuu verkossa. Verkkoaineisto. Maanmittauslaitos. <<https://www.sttinfo.fi/tiedote/69910901/maanmittauslaitoksen-historiallisten-ilmakekuvien-arkisto-avautuu-verkossa>>. 1.6.2021. Luettu 25.3.2024.
- 8 Tekninen kuvaus (WMS). Verkkoaineisto. Maanmittauslaitos. <<https://www.maanmittauslaitos.fi/karttakuvapalvelu/tekninen-kuvaus-wms>>. Luettu 25.3.2024.
- 9 Paikkatietoikkuna: Historialliset ilmakekuvat. Verkkoaineisto. Maanmittauslaitos. <<http://www.paikkatietoikkuna.fi/historiallisetilmakuva>>. Luettu 25.3.2024.
- 10 Karttapaikka: Lataa paikkatietoaineistoja. Verkkoaineisto. Maanmittauslaitos. <<https://asiointi.maanmittauslaitos.fi/karttapaikka/tiedostopalvelu/ortoilmakuva>>. Luettu 25.3.2024.

- 11 Ortokuvien ja korkeusmallien kyselypalvelu (WCS). Verkkoaineisto. Maanmittauslaitos. <<https://www.maanmittauslaitos.fi/ortokuvien-ja-korkeusmallien-kyselypalvelu>>. Luettu 25.3.2024.
- 12 Maanmittauslaitoksen ortokuva. Verkkoaineisto. Maanmittauslaitos. <<https://www.maanmittauslaitos.fi/kartat-ja-paikkatieto/aineistot-ja-rajapinnat/tuotekuvaukset/ortokuva>>. Luettu 25.3.2024.
- 13 Ruotsala, Anni-Helena. 2013. Selvitys vanhojen ilmakuvien käyttötarkoituksista. Raportti. Helsinki: Maanmittauslaitos.
- 14 Bolstad, Paul. 2019. GIS Fundamentals: A First Text on Geographic Information Systems. 6th edition. Ann Arbor: XanEdu.
- 15 Laine, Pirita. 2015. Vanhan ilmakuvan käyttö tutkimuksessa modernein menetelmin. Kandidaatintyö. Aalto-yliopisto, Insinöörیتieteiden korkeakoulu. Aaltodoc-tietokanta.
- 16 Lundmark, H.; Östlund, L. & Josefsson, T. 2021. Continuity forest or second-generation forest? Historic aerial photos provide evidence of early clear-cutting in northern Sweden. *Silva Fennica*. Vol. 55, article id 10460.
- 17 Ihse, Margareta. 1995. Swedish agricultural landscapes - patterns and changes during the last 50 years, studied by aerial photos. *Landscape and Urban Planning*. Vol. 31, s. 21–37.
- 18 Hartikainen, Lauri. 2011. Historialliset ilmapuva-aineistot osana kaupunkisuunnittelua. *Maankäyttö* 2/2011, s. 23–26.
- 19 Suolahti, Ida. 2022. Kaatopaikkojen Helsinki. Verkkoaineisto. Helsingin kaupunki. <<https://historia.hel.fi/fi/ilmiot/muuttuva-kaupunkiluonto/kaatopaikkojen-helsinki>>. 10.3.2022. Luettu 25.3.2024.
- 20 Taleva, Katariina. 2021. Näin Suomi on muuttunut ilmasta katsottuna – katso historialliset kuvaparit 10 kaupungista. Verkkoaineisto. Tekniikka ja talous. <<https://www.tekniikkatalous.fi/uutiset/nain-suomi-on-muuttunut-ilmasta-katsottuna-katso-historialliset-kuvaparit-10-kaupungista/72fd6562-8c87-48e6-9238-fc34d9ebb5d7>>. 29.6.2021. Luettu 4.4.2024.
- 21 Learn more about raster and vector map data. 2018. Verkkoaineisto. Ordnance Survey. <<https://www.youtube.com/watch?v=hbOGp51nYGI>>. Katsottu 4.4.2024.
- 22 Kaiser, Peter. 2021. ArcGIS Pro: Das deutschsprachige Handbuch inklusive Einstieg in ArcGIS Online. 2., neu bearbeitete und erweiterte Auflage. Berlin: Wichmann.

- 23 Kiinteistötietojen kyselypalvelu, avoimet tuotteet. Verkkoaineisto. Maanmittauslaitos. <<https://avoin-paikkatieto.maanmittauslaitos.fi/kiinteisto-avoin/simple-features/v3/collections>>. Luettu 3.4.2024.
- 24 Rajapinta. Verkkoaineisto. TEPA-termipankki. Sanastokeskus. <<https://termipankki.fi/tepa/fi/haku/rajapinta>>. Luettu 25.3.2024.
- 25 Yleistä rajapintapalveluista. Verkkoaineisto. Maanmittauslaitos. <<https://www.maanmittauslaitos.fi/kartat-ja-paikkatieto/aineistot-ja-rajapinnat/yleista-rajapintapalveluista>>. Luettu 25.3.2024.
- 26 Ohje API-avaimen käyttöön. Verkkoaineisto. Maanmittauslaitos. <<https://www.maanmittauslaitos.fi/rajapinnat/api-avaimen-ohje>>. Luettu 25.3.2024.
- 27 Rahkonen, Jukka. 2019. Paikkatietopeitteen kyselypalvelut (WCS) -webinaari 7.11.2019. Verkkoaineisto. Maanmittauslaitos. <<https://www.youtube.com/watch?v=P3h8mByXWjY>>. 13.11.2019. Katsottu 25.3.2024.
- 28 Tekninen kuvaus Ortokuvien ja korkeusmallien kyselypalvelu (WCS). Verkkoaineisto. Maanmittauslaitos. <<https://www.maanmittauslaitos.fi/ortokuvien-ja-korkeusmallien-kyselypalvelu/tekninen-kuvaus>>. Luettu 25.3.2024.
- 29 OGC API – Features. Verkkoaineisto. Open Geospatial Consortium. <<https://ogcapi.ogc.org/features/>>. Luettu 25.3.2024.
- 30 OGC-API: Rajapintapalvelujen tulevaisuus paikkatietoalalla. 2019. Verkkoaineisto. Gispo Oy. <<https://www.gispo.fi/blogi/ogc-api-rajapintapalvelujen-tulevaisuus-paikkatietoalalla/>>. 30.9.2019. Luettu 3.4.2024.
- 31 Reini, Jari. 2021. OGC API Features Webinaari. Verkkoaineisto. Maanmittauslaitos. <<https://www.youtube.com/watch?v=QDsTnenFIds>>. 7.4.2021. Katsottu 25.3.2024.
- 32 QGIS User Guide. Verkkoaineisto. QGIS project. <[https://docs.qgis.org/3.28/en/docs/user\\_manual/index.html](https://docs.qgis.org/3.28/en/docs/user_manual/index.html)>. Luettu 25.3.2024.
- 33 General Python FAQ. Verkkoaineisto. Python Software Foundation. <<https://docs.python.org/3/faq/general.html>>. Luettu 25.3.2024.
- 34 PyQGIS Developer Cookbook: Introduction. Verkkoaineisto. QGIS project. <[https://docs.qgis.org/3.28/en/docs/pyqgis\\_developer\\_cookbook/intro.html](https://docs.qgis.org/3.28/en/docs/pyqgis_developer_cookbook/intro.html)>. Luettu 25.3.2024.

- 35 QGIS Python API Documentation. Verkkoaineisto. QGIS project. <<https://qgis.org/pyqgis/3.28/index.html>>. Luettu 25.3.2024.
- 36 PyQGIS Developer Cookbook: 16.1. Structuring Python Plugins. Verkkoaineisto. QGIS project. <[https://docs.qgis.org/3.28/en/docs/pyqgis\\_developer\\_cookbook/plugins/plugins.html](https://docs.qgis.org/3.28/en/docs/pyqgis_developer_cookbook/plugins/plugins.html)>. Luettu 25.3.2024.
- 37 Helsingin ortoilmakuvat. Verkkoaineisto. Helsinki Region Infoshare. <<https://hri.fi/data/fi/dataset/helsingin-ortoilmakuvat>>. Luettu 3.4.2024.
- 38 Ilmakuvat ja ortot 1931 - 2025 sekä laserkeilaukset 2008 – 2025. Verkkoaineisto. Maanmittauslaitos. <<https://hkp.maanmittauslaitos.fi/hkp/published/fi/4343c1b4-7d8f-4473-896a-70f930f36be1>>. Luettu 25.3.2024.
- 39 Informationsinnehåll – ortofoto historiska Nedladdning. Verkkoaineisto. Lantmäteriet. <<https://geotorget.lantmateriet.se/dokumentation/GEO-DOK/31/latest/informationsinnehall.html>>. Luettu 25.3.2024.

## **Toteutettu QGIS-lisäosa**

*Liite on erillinen tiedosto nimeltä ortokuvaalataaja.zip*