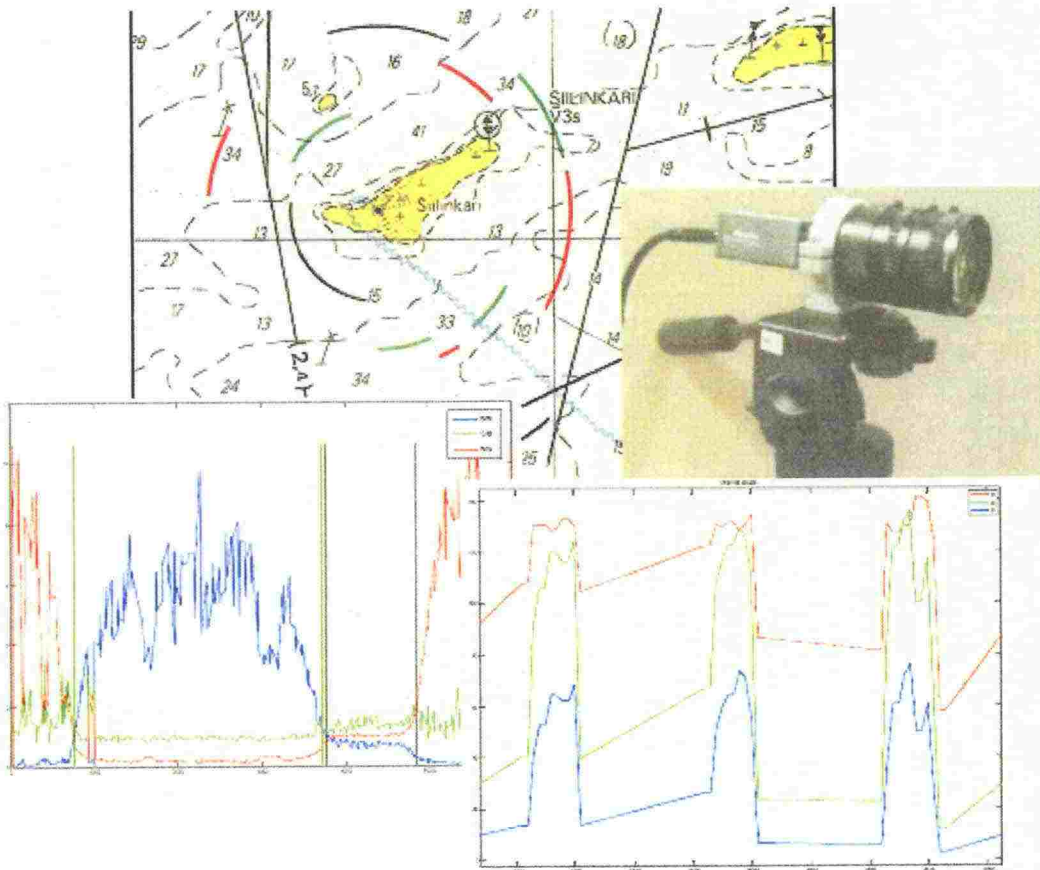


SEKTORILOISTOJEN VÄRIRAJOJEN MITTAUS KUVAAN PERUSTUVAN MITTAUKSEN KEINAIN - JATKOHANKE





| | | | |
|---|-----------------------|---|--------------------|
| Tekijät (toimielimestä: toimielimen nimi, puheenjohtaja, sihteeri) | | Julkaisun laji Merenkululaitoksen sisäisiä julkaisuja | |
| TTY/Mittaus- ja informaatiotekniikan laitos/Kalle Ryyänen, Antti Vehkaoja | | Toimeksiantaja MKL/Väylänpito/DI Risto Joro | |
| | | Toimielimen asettamispäivämäärä 15.02.2006 | |
| Julkaisun nimi SEKTORILOISTOJEN VÄRIRAJOJEN MITTAUS KUVAAN PERUSTUVAN MITTAUKSEN KEINAIN - JATKOHANKE | | | |
| <p>Tiivistelmä</p> <p>SEKLO- eli sektoriloistoprojekti on Suomen Merenkululaitoksen ja Tampereen teknillisen yliopiston Mittaus- ja Informaatiotekniikan laitoksen yhteistyöprojekti, jonka tavoitteena on sektoriloiston värirajojenmääritys digitaalikuvaan perustuvan mittauksen avulla. Käytännön tutkimus toteutettiin Mittaus- ja Informaatiotekniikan laitoksella Merenkululaitoksen tarpeiden mukaisesti.</p> <p>Sektoriloisto on tärkeä merenkulun turvalaite, jonka avulla alukset voivat paikantaa itsensä ajettavan väylän suhteen. Tällä hetkellä sektorirajojen tarkastus perustuu havainnointiin paljaalla silmällä. Tarkastajasta riippuen sektorirajan sijainti on saattanut vaihdella jopa yli 100 metriä. Tarkan rajan määrittäminen pelkän näköhavainnon perusteella on myös vaatinut useita tarkistuskertoja ts. tarkastusalue ajaa tiettyä sektorirajaa edestakaisin ja rajan sijainti määritetään tehtyjen havaintojen keskiarvona.</p> <p>Tarkennusta sektorirajojen havainnointiin haettiin konenäön avulla. Projektin ensimmäinen vaihe perustui Tampereen teknillisen yliopiston Mittaus- ja informaatiotekniikan syksyllä 2005 tekemään esiselvitykseen mahdollisuudesta käyttää kameraa apuna sektorien rajojen määrittämisessä. Esiselvityksen positiivisten tulosten perusteella aloitettiin varsinaisen projektin ensimmäinen vaihe. Projektin ensimmäisessä vaiheessa värin määrittäminen tapahtui lähes ideaaliolosuhteissa.</p> | | | |
| Avainsanat (asiasanat) Sektoriloisto, avauskulma, konenäkö, valoimpulssi | | | |
| Muut tiedot | | | |
| Sarjan nimi ja numero Merenkululaitoksen sis. julkaisuja 2/07 | | ISSN 1456-9442 | ISBN |
| Kokonaissivumäärä 42 | Kieli Suomi | Hinta | Luottamuksellisuus |
| Jakaja | | Kustantaja | |

Sisällysluettelo

| | |
|---|-----------|
| 1. Johdanto | 3 |
| 2. Laitteisto | 4 |
| 2.1. Kamera | 4 |
| 2.2. Optiikka | 6 |
| 2.3. Tietokone ja ohjelmisto | 7 |
| 2.4. Lisälaitteistot | 7 |
| GPS | 7 |
| Lisäjännitelähde | 8 |
| 3. Kuvankäsittelyalgoritmi | 9 |
| 3.1. Kohteen paikannus | 9 |
| 3.2. Värin määrittäminen | 9 |
| 4. Tulokset | 13 |
| 5. Yhteenveto ja jatkotoimenpiteet | 16 |
| Lähteet | 18 |
| Liitteet | 19 |
| Liite A – Laskenta-algoritmi | 19 |
| Liite B | 25 |



1. Johdanto

SEKLO- eli sektoriloistoprojekti on Suomen merenkululaitoksen ja Tampereen teknillisen yliopiston Mittaus- ja Informaatiotekniikan laitoksen yhteistyöprojekti, jonka tavoitteena on sektoriloiston väri rajojen määrittäminen digitaalikuvaan perustuvan mittauksen avulla. Käytännön tutkimus toteutettiin Mittaus- ja Informaatiotekniikan laitoksella Merenkululaitoksen tarpeiden mukaisesti.

Sektoriloisto on tärkeä merenkulun turvalaite, jonka avulla alukset voivat paikantaa itsensä ajettavan väylän suhteen. Tällä hetkellä sektorirajojen tarkastus perustuu havainnointiin paljaalla silmällä. Tarkastajasta riippuen sektorirajan sijainti on saattanut vaihdella jopa yli 100 metriä. Tarkan rajan määrittäminen pelkän näköhavainnon perusteella on myös vaatinut useita tarkistuskertoja ts. tarkastusalus ajaa tiettyä sektorirajaa edestakaisin ja rajan sijainti määritetään tehtyjen havaintojen keskiarvona. Tarkennusta sektorirajojen havainnointiin haettiin konenäön avulla.

Projektin ensimmäinen vaihe perustui Tampereen teknillisen yliopiston Mittaus- ja informaatiotekniikan tutkijoiden Antti Vehkaojan ja Petri Österbergin syksyllä 2005 tekemään esiselvitykseen mahdollisuudesta käyttää kameraa apuna sektorien rajojen määrittämisessä [1]. Esiselvityksen positiivisten tulosten perusteella aloitettiin varsinaisen projektin ensimmäinen vaihe. Projektin ensimmäisessä vaiheessa värin määrittäminen tapahtui lähes ideaaliolosuhteissa.

2. Laitteisto

Tässä kappaleessa kuvataan lyhyesti kuvaamiseen käytetty laitteisto ja perusteet välineiden valinnoille. Pääasialliset laitteet ovat kuvauksessa käytettävä kamera, kameran optiikka sekä kuvadatan tallettava ja signaalinkäsittelyn suorittava tietokone.

2.1. Kamera

Kuvausjärjestelmä ei aseta tarkkoja rajoja kameran valinnalle. Kameran tulee olla video- tai niin sanottu konenäkökamera, jotta jatkuvan datan tuottaminen on mahdollista. Videokameroiden heikkous tieteellisesti tarkasteltuna on niiden automatiikka. Automaattiset valkotasapainon korjaukset ja valaistuksen korjaukset aiheuttavat ongelmia vertailtaessa eri voimakkuuden omaavia valonlähteitä tai taustavalon voimakkuuden tai värin vaihdellessa. Teollisuuskäyttöön tarkoitetuissa konenäkökameroissa kameran parametrit saa asetettua halutuksi manuaalisesti. Konenäkökameran yksi tärkeä ominaisuus on raakadatan tuottaminen eli vältytään kuvan pakkaamisessa aiheutuvalta häiriöiltä. Värejä vertailtaessa on tietenkin tärkeää, että kamera on värikamera. Sovelluksessa ei tarvita suurta resoluutiota, vaan on tärkeämpää saada mitattava kohde jaettua laajalle alueelle ilman kirkkaan pistemäisen valonlähteen helposti aiheuttamaa kuvapisteyden ylivuotoa. Valon jakaminen laajalle alueelle asettaa vaatimuksen kennon herkkyydelle eli kennon tulee olla mahdollisimman herkkä, jotta saadaan rekisteröityä mahdollisimman paljon valoa. Matalan resoluutiovaatimuksen avulla voidaan käyttää halvempia bayer-matriisikameroita kalliiden foveon- ja 3CCD-kameroiden sijasta.

Konenäkökameroiden liittäminen tietokoneeseen voi tapahtua useiden eri väylävaihtoehtojen kautta. Yleisimpiä on Cameralink, PCI, Ethernet, USB1.1 ja 2.0 ja IEEE1394 a ja b (eli FireWire™ 400Mbs ja 800Mbs). Väylävaihtoehtoja on useita, mutta tässä nyt muutama mainittuna.

IEEE1394 eli kansankielellä FireWire on standardoitu väyläratkaisu. FireWire kehitettiin 1990 luvun alussa Apple™ tietokoneisiin. Standardoitu väyläratkaisu

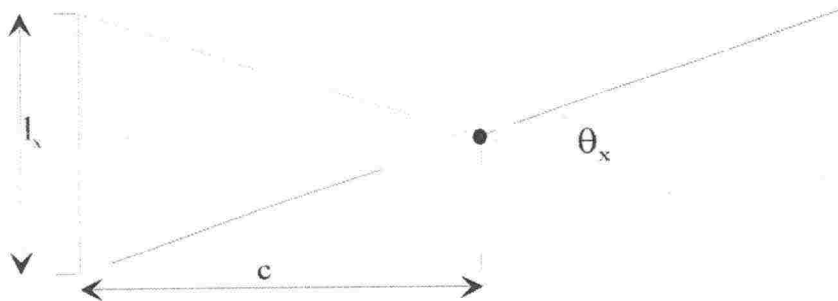
helpottaa yhteensopivuutta eri tietokonevalmistajien kanssa. Lisäksi FireWire:n etuna on sen kyky antaa kameralle käyttöjännite. PCMCIA-porttiin liitettävää FireWire-korttia käytettäessä tulee käyttää ulkoista jännitelähdettä, koska PCMCIA-portti pystyy maksimissaan antamaan ainoastaan kolme voltia minimissään vaadittavan kahdeksan voltin sijasta.

Testilaitteistoon valittiin kameraksi NET FOculus 432SC, sen kennon herkkyyden takia. Testikamerassa on 1392 x 1040 pikselin 1/2" kenno. Suomeen FOculus kameroita maahan tuo konenäön erikoisliike Metric.

2.2. Optiikka

Kuvatessa kilometrin päästä vaaditaan optiikalta hieman pidempää polttoväliä kuin peruslähikuvauksessa, jotta kohde saadaan kasvatettua kuvalle riittävän suureksi analysointia varten. Epäfokuksen avulla saadaan valopistettä myös kasvatettua, mutta samalla mahdolliset virheelliset valonlähteet suurenevät.

Pitkällä polttovälillä saadaan samalla rajattua kuva-aluetta suppeammaksi. Suppeamman kuva-alueen häittana on avauskulman pieneneminen polttovälin kasvaessa. Veneeltä kuvattaessa pieni avauskulma saattaa koitua ongelmaksi aluksen keinuessa voimakkaasti. Kuva 2.1. esittää avauskulman määräytymistä polttovälin suhteen. Kuvan 2.1 perusteella voidaan johtaa kaavat 2.1–2.3 avauskulman laskemiseksi, trigonometrian peruskaavojen avulla.



Kuva 2.1: Avauskulma

$$\tan\left(\frac{\theta_x}{2}\right) = \frac{l_x/2}{c} \quad (2.1)$$

$$\arctan\left(\frac{l_x/2}{c}\right) = \frac{\theta_x}{2} \quad (2.2)$$

$$\theta_x = \arctan\left(\frac{l_x/2}{c}\right) \quad (2.3)$$

Missä θ_x on avautumiskulma, l_x on kennon fyysinen koko ja c polttoväli.

Testijärjestelmän optiikaksi valittiin 8-48mm Pentax zoom-objektiivi. Pentax zoom soveltuu testaukseen hyvin manuaalisesti säädettävien aukon, tarkennuksen ja zoomin ansiosta. Jokaiselle säädölle objektiivissa on lukitusruuvi säätöjen ryöminnan estämiseksi. Säätöjen ryömintä on ollut joissain aikaisemmissa kuvauksissa ongelma käytettäessä objektiivia, jossa ei ole ollut kyseisiä lukitusruuveja.

2.3. Tietokone ja ohjelmisto

Vaatimukset tietokoneelle on FireWire yhteensopivuus eli FireWire liitin, sekä tarvittavat ajurit. Tietokoneeseen tulee sen lisäksi asentaa kameran ohjausohjelmisto. Kameran mukana tulevat ohjelmistot saattavat vaikuttaa ohjauksen toteutukseen.

WindowsTM-käyttöjärjestelmää käytettäessä tulee ladata korjauspaketti FireWire:n vapauttamiseksi sen todelliseen nopeuteen. Windows:ssa service pack 2 rajoittaa FireWire 100Mbs, normaalin 400 (a) tai 800 (b) Mbs sijasta.

Laskenta toteutetaan SEKLO-projektin alkuvaiheessa MATLAB® ohjelman avulla. MATLAB ohjelmiston etuna ovat valmiit funktiopaketit kuvan käsittelyä varten.

2.4. Lisälaitteistot

Kappaleessa on esitetty kuvausta tukevia ja helpottavia lisälaitteistoja

GPS

Tutkimuksen apuvälineenä käytettiin Magellan 315TM GPS-paikanninta (Global Position System). GPS:n avulla voidaan määrittää sijainti muutaman metrin tarkkuudella. Sijainnin tarkka tietäminen auttaa sektorin vaihtumiskohtaa tarkasteltaessa. Ongelmana normaalin käsikäyttöisen GPS-paikantimen käyttämisessä kuitenkin on että sillä saatavia koordinaatteja on vaikea sitoa kuvadataan siten että voitaisiin koko ajan tarkasti tietää, missä kohdassa ollaan kullakin kuvaushetkellä oltu. Järjestelmään liitettävän GPS:n avulla voidaan määrittää tarkasti kuljettu reitti ja erityisesti vaihtumishetkien koordinaatit.

Lisäjännitelähde

Kannettavan tietokoneen PCMCIA-portin jännitteensyöttökyky rajoittuu 3V ja kameran käyttämiseen tarvitaan minimissään 8V. Jännitevajeen takia tulee käyttää ulkoista jännitelähdettä. Sovelluksen tutkimusvaiheessa päädyttiin käyttämään pienestä laboratorioakusta saatavaa 12V jännitettä. Akun käyttämisen etuna on myös sen tasainen virransyöttö.

3. Kuvankäsittelyalgoritmi

Tässä kappaleessa käydään läpi kuvankäsittelyyn kehitetyn algoritmin perusosat, joiden avulla voidaan määrittellä sektoriloiston valon väri jokaisella ajanhetkellä erikseen.

3.1. Kohteen paikannus

Sektorivalon paikantaminen kuvasta, perustuu kuvan intensiteettitasoihin. Kuvasta etsitään tietyn intensiteettitason ylittäviä alueita, jotka ovat kooltaan yli 50 pikseliä. Kokoa rajaamalla voidaan välttää roskien ja muiden pienten poikkeaminen tallentumisen muodostettavaan maskiin. Valonalueen löydyttyä alue täytetään eli poistetaan roskat löydetyn valon alueelta ja korvataan ne ympäristön arvoilla, jotta maski on yhtenäinen.

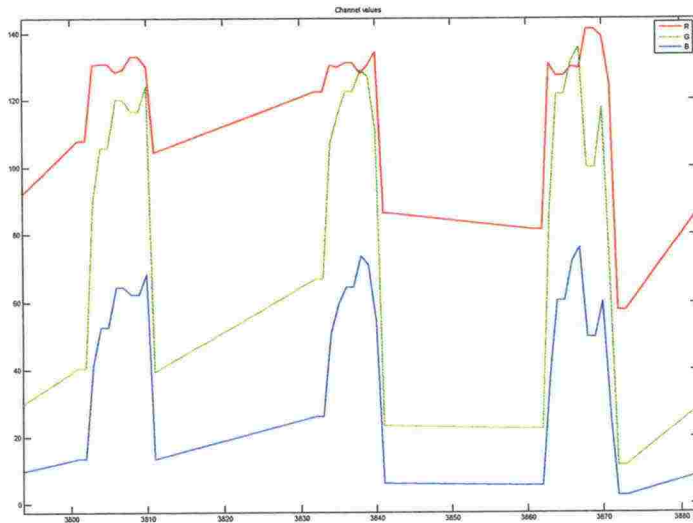
Projektin alkuvaiheessa perusoletuksena on, että kuva-alueella ei esiinny muita intensiteetiltään korkeita valolähteitä, jolloin havaittu korkean intensiteetin alue voidaan tulkita tutkittavaksi valolähteeksi.

Löydetyn valonalueen avulla muodostetaan maski, jossa täytetty valonalue asetetaan ykköseksi ja kaikki muu nolllaksi. Tällä maskilla kerrottaessa saadaan tarkasteltua ainoastaan valittua kohdetta ja ympäristön häiritsevät valolähteet yms. roskat jää laskujen ulkopuolelle.

3.2. Värien määrittäminen

Maskin alueelta lasketaan kunkin värikanavan summa. Värikanavia RGB-järjestelmässä on kolme, punainen (Red), vihreä (Green) ja sininen (Blue). Summa kertoo kunkin värin kokonaismäärän maskin alueella. Algoritmin toteutusta vaikeutti valon vilkkuminen, koska syttyessään ja sammuessaan sektoriloiston valo on täydellä teholla palamista punasävyisempi.

Tämän takia algoritmiin on kehitetty osa, joka poistaa impulssin alusta ja lopusta 25%. Jäljelle jäävästä 50% lasketaan keskiarvo ja sijoitetaan se koko valoimpulssin leveydelle. Kuvassa 3.1. on esitetty väri vaihtelut valoimpulssin aikana.



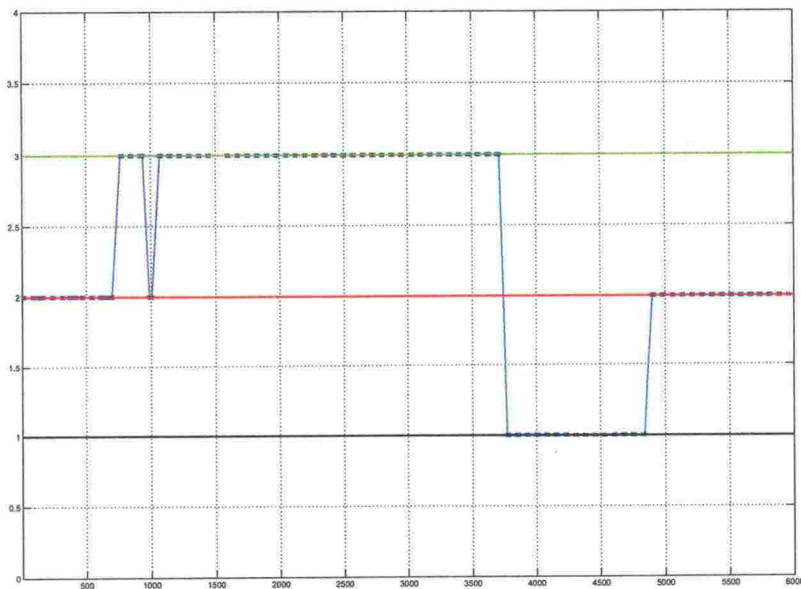
Kuva 3.1: Valoimpulssin muodon vaihtelut

Mitattujen datojen värikanavien suhteita empiirisesti tutkimalla on voitu arvioida värin vaihtumiselle ominaisia tunnuslukuja. Tutkittaessa värikanavien suhteilla poistuu valon vaihtelevasta intensiteetistä aiheutuvat ongelmat, koska suhteet ovat lähes riippumattomia intensiteetistä. Erittäin matala tai korkea intensiteetti saattaa aiheuttaa havainnointi virheitä, tämän takia ei voida sanoa absoluuttisesti suhteiden olevan riippumattomia intensiteetti-arvoista. Taustavaloista, valojen väri virheistä, kameran automatiikasta ja olosuhteista johtuen yksittäisien tunnuslukujen avulla ei voida päättää väriä vaan värinmääritys tapahtuu yhdistämällä useita värien ominaisuuksia. Tutkitun datan avulla on päästy taulukossa taulukon 3.1. esitettyihin ehtoihin.

Taulukko 3.1: Eräitä mahdollisia värinmäärittäyssääntöjä

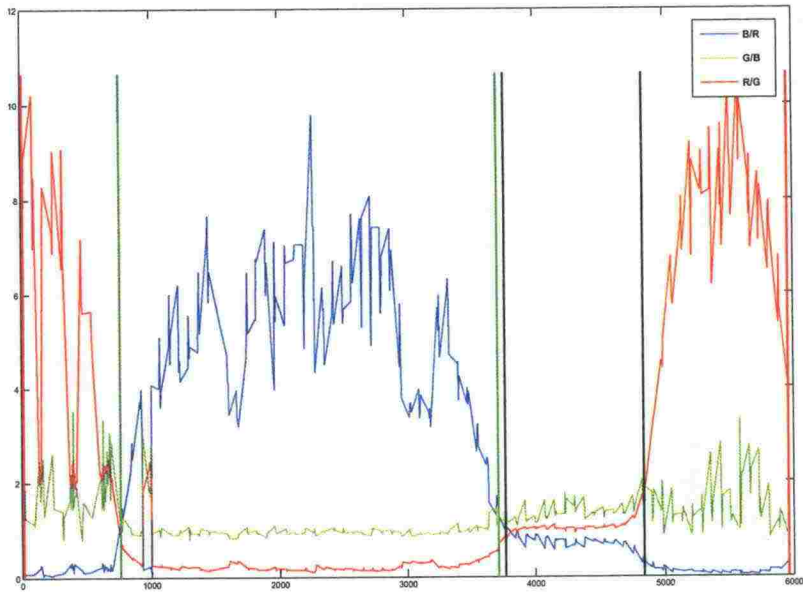
| Väri | Ehto |
|-----------|-------------|
| Valkoinen | $G/B > 1.7$ |
| Punainen | $R/G > 10$ |
| Vihreä | $B/R > R/G$ |

Jokaisen kuvakehyksen (freimin) havaittu väri tallennetaan yhteen vektoriin. Kun kaikki kehykset on tarkasteltu, poistetaan normaalitasosta poikkeavat piikit vertaamalla jokaista kehystä erikseen ympäröiviin kehyksiin eli mediaanisuodatetaan tämä värivektori. Tällä toimenpiteellä saadaan poistettua virheelliset kehykset. Oletetaan siis että väri ei oikeasti vaihdu edestakaisin kolmen kehyksen aikana. Tulosten havainnollistamiseksi piirretään käyrä, jossa jokainen väri on kuvattu omalle tasolleen. Väritasokäyrä on esitetty kuvassa 3.2.



Kuva 3.2: Värien havainnollistus tasoina. Vaihtokohdissa saattaa ajoittain olla pientä vaihtelua, ennen värin tasoittumista.

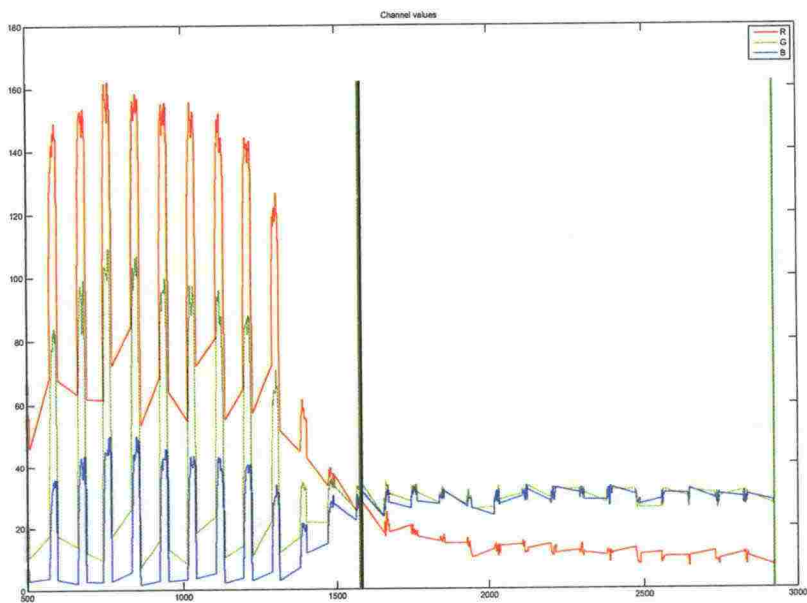
Algoritmi merkitsee automaattisesti myös värisuhdekäyrien kanssa samaan kuvaan havaittujen värisektoreiden vaihtumiskohtat sektorin väristen pystyviivojen avulla. Kuvassa 3.3 on esimerkki vaihtumiskohtien merkitsemisistä.



Kuva 3.3: Värien suhteet.

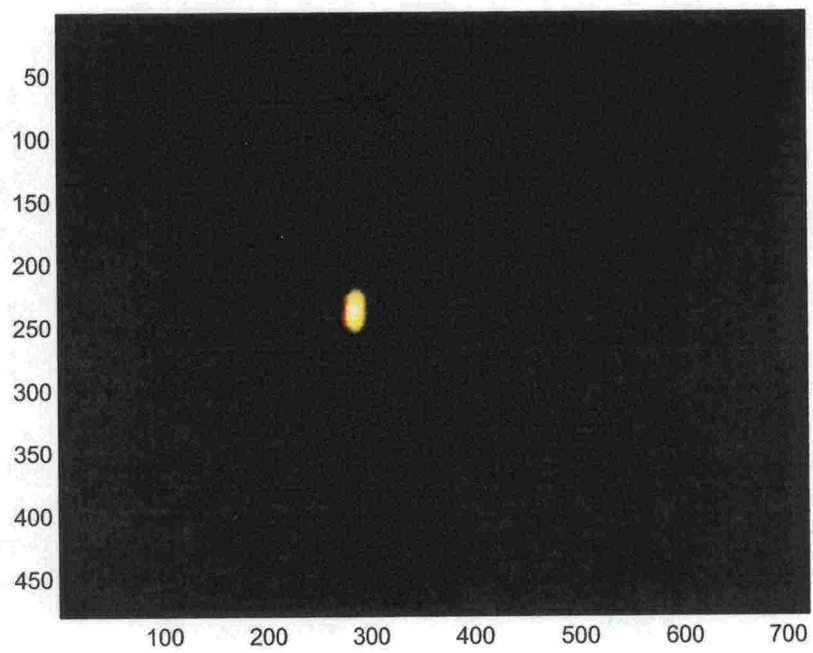
4. Tulokset

Useista mittausdatoista saatiin havaittua sektorin värin vaihtuminen tarkasti samaan kohtaan. Kuvassa 4.1 on esitetty algoritmin havaitsema värin vaihtumiskohta värikanavien intensiteettien avulla. Algoritmi havaitsee sektorin valon aina samaan kohtaan, mutta tällä hetkellä käytettävissä olevilla laitteistoilla ja tehdyillä mittauksilla ei voida vielä sanoa, miten värin vaihtumiskohta on suhteessa merenkulun ammattilaisten havainnointiin tai DGPS:n ja merikortin antamiin teoreettisiin sektorin värin vaihtumiskohtaan. Kuvissa 4.2 – 4.6 on esitetty kutakin väriä esittävät yksittäiset kuvakehykset. Kuvausta vaikeuttivat taustalla näkyvät valot, kuten vastarannan katuvalot.

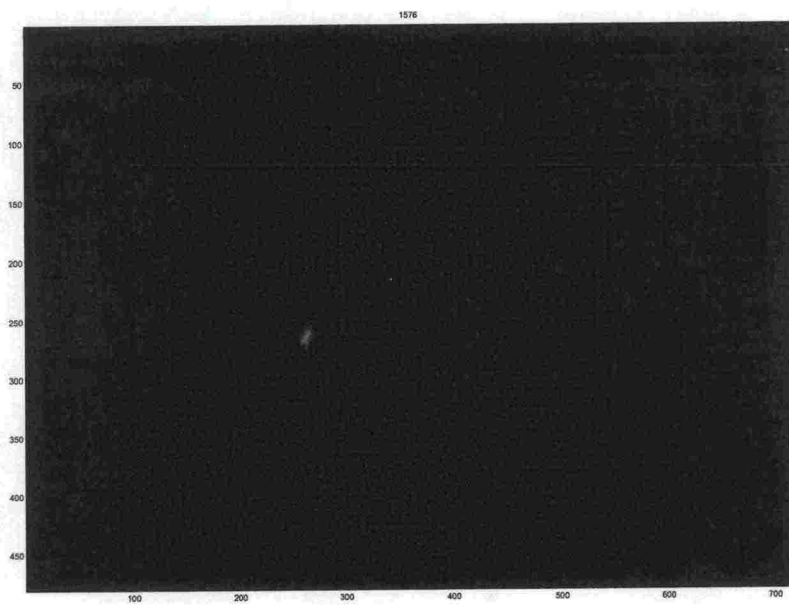


Kuva 4.1: Värien havainnointi värikanavien avulla, jossa väri vaihtuu valkoisesta vihreään.

Kuvasta 4.2 voidaan havaita, että puhdas valkoinen sisältää huomattavasti enemmän punaista kuin muita värikanavien arvoja. Tämä voidaan tulkita ns. lämpöiseksi valkoiseksi. Toinen kuvasta 4.1 huomioitava kohta on värin havainnointi. Algoritmi tulkitsee valon vihreäksi vasta, kun väri on aidosti vihreää. Tällä hetkellä algoritmi tulkitsee koko siirtymäalueen valkoisesta vihreään valkoiseksi. Lisää mittausdataa esitetty liitteessä B.

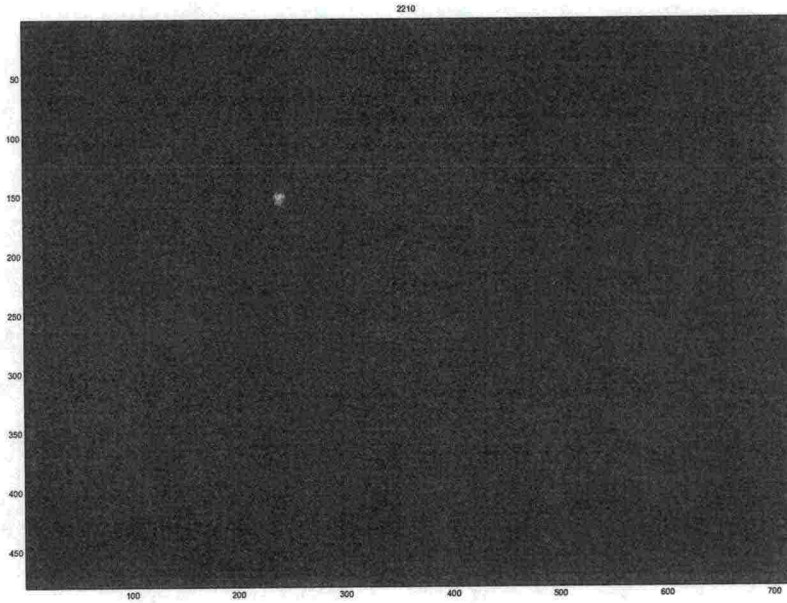


Kuva 4.2: Yksittäinen kehys valkoisesta valosta.

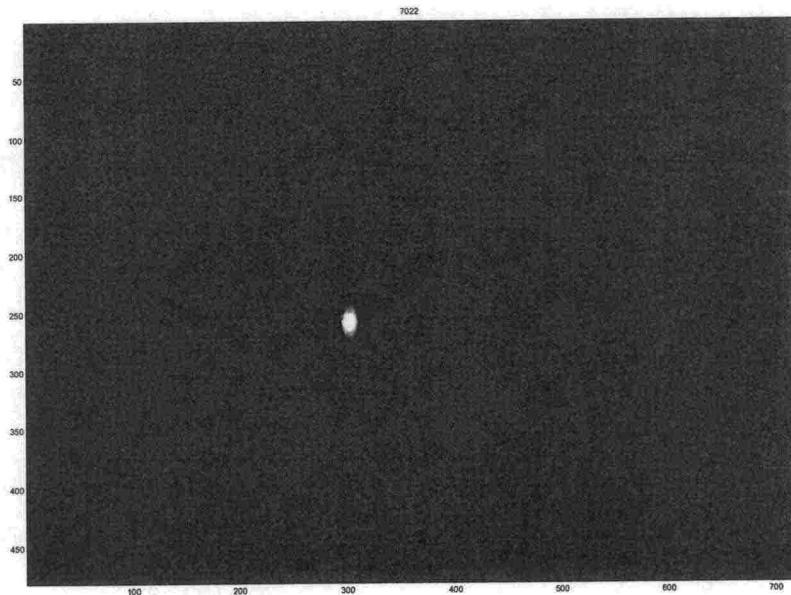


Kuva 4.3: Yksittäinen kehys vihreän ja valkoisen vaihtumiskohdasta.

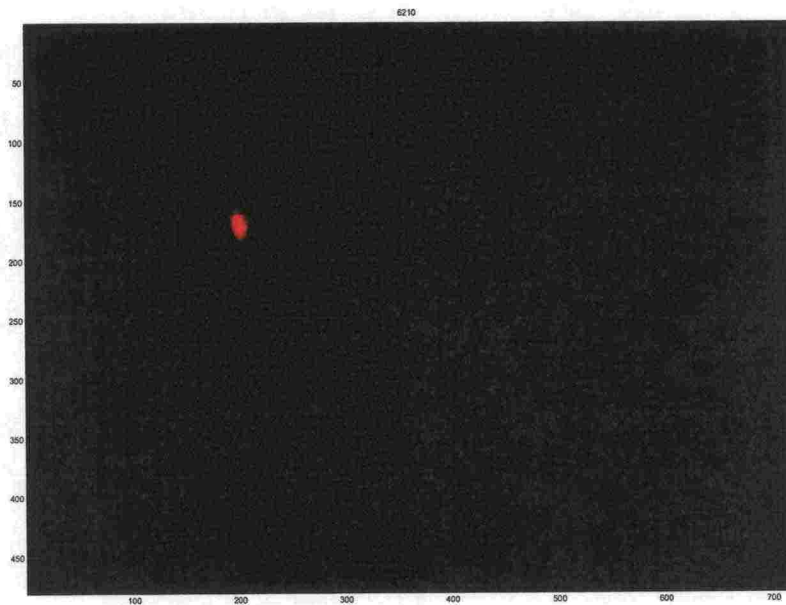
Vihreän valon intensiteetti jäi huomattavasti valkoista ja punaista valoa alemmaksi. Kysymyksessä saattaa osittain olla myös käytettävän kameran kennon suurempi herkkyys punaisille aallonpituuksille kuin sinisille tai vihreille.



Kuva 4.4: Yksittäinen kehys vihreästä valosta.



Kuva 4.5: Yksittäinen kehys punaisen ja valkoisen valon vaihtumiskohdasta.



Kuva 4.6: Yksittäinen kehys punaisesta valosta.

5. Yhteenveto ja jatkotoimenpiteet

Projektin tässä vaiheessa oli tavoitteena värin luotettava havainnointi, kun kuvan alueella on vain yksi ainoa valonlähde. Tässä tavoitteessa onnistuttiin ja sektorin värin vaihtuminen saatiin tarkistettua ja havainnoitua aina samaan kohtaan. Ongelmaksi saattaa koitua eri valaistusolosuhteissa kuvattaessa tai eri sävyisiä loistoja kuvattaessa. Tätä varten tarvitaan vielä jatkotutkimusta. Tämänhetkiset mittaukset kuitenkin osoittavat siihen suuntaan, että myös iltahämärällä kuvaus onnistuu ja voidaan erottaa sektorin väri ennen pimeän tuloa. Tällä hetkellä on kuvattu järjestelmällä ainoastaan Tampereen Näsijärvellä sijaitsevaa Siilinkarin sektoriloistoa, joten muiden loistojen mahdolliset sävyerojen aiheuttavat virheet ovat tässä vaiheessa vielä huomioimatta. Myös mahdollisen epävarmuusalueen (alue väri vaihtuessa) määrittäminen on vielä kesken. Epävarmuusalueen avulla voidaan määrittää, koska väri alkaa vaihtua ja koska väri on täysin vaihtunut.

Jatkossa on myös tarkoitus tutkia mitattavan valonlähteen tunnistaminen useiden valonlähteiden joukosta. Tähän keinona voidaan käyttää apriori tietoa eli aiempaa tietoa valon paikasta kuvakehyksessä. Käytettäessä edellisen kehyksen paikkatietoa vilkkuva mittauskohde aiheuttaa ongelmia, joten on suotavaa, että mitattaessa valo on asetettu palamaan kiinteästi. Vilkkuva valon aiheuttaa ongelmia, kun ylimääräinen valolähde on sävyltään ja kooltaan samanlainen tutkittavan valonlähteen kanssa. Erytisesti ongelmia ilmenee silloin, kun mitattava valolähde on pois päältä ja kuvassa näkyy vain ns. ylimääräisiä valonlähteitä. Avovesikaudella loistojen asettaminen kiinteästi palavaksi aiheuttaa mittavia toimenpiteitä, joten nämä mittaukset on jätetty tehtäväksi talvella, jolloin toivottavasti jää kantaa mittaushenkilön kalustoineen. Jäältä mittaus auttaa myös paikkatiedon saamista. Kiinteällä alustalla voidaan määrittää GPS:n avulla tarkasti kuljettu reitti. Jolloin myös merikortin mukaisen väylän paikan määrittäminen helpottuu ja voidaan verrata kuvasta tunnistetun sektorirajan paikkaa merikorttiin merkattuun. Kun paikkatieto voidaan määrittää tarkasti, voidaan myös tehdä helposti ja tarkasti toistuvuus ja uusittavuus kokeita. Tämän vaatimuksena on DGPS:n kytkettävyys käytettävään mittausjärjestelmään, jolloin paikkatieto voidaan yhdistää suoraan kameran tuottamaan dataan.

Ensimmäisessä vaiheessa käytetty MATLAB-ohjelmisto saattaa väistyä muiden ohjelmointiympäristöjen tieltä projektin edetessä. Tutkimuskäytössä MATLAB on oivallinen työkalu tutkimukseen ja tuotekehitykseen, sillä sen avulla pystyy helposti ja nopeasti kokeilemaan erilaisten laskentamenetelmien toimivuutta. MATLABilla tehtävä kuvankäsittely vaatii kuitenkin huomattavasti enemmän laskentatehoa yleisiin ohjelmistokieliin, kuten esimerkiksi C ja C++ verrattuna eikä sillä sen vuoksi ole mahdollista tehdä niin monimutkaisia, reaaliaikaisesti toimivia, laskenta-algoritmeja.

Tutkimuksen aikana on myös esiin noussut tarve LabView-ohjelmointikielen käyttösovelluksen tekemisestä. LabView:n etuna ovat sen monipuoliset funktiopaketit tietokoneen ja kameran rajapintaan. Kameran rajapinta on MATLAB:lle vaikeasti käsiteltävä kohta, joten kameranohjaus tarvitsee todennäköisesti kuitenkin jonkin muun apuohjelman tai -ohjelmiston mittausta suoritettaessa. Lisäksi myös kuvankäsittely voitaisiin toteuttaa LabView:n valmiilla ohjelmistokokonaisuuksilla.

Lähteet

1. A. Vehkaoja ja P. Österberg SELO-projektin esiselvitys, 2005
2. IALA Recommendation E-108, On Surface Colours used as Visual Signals on Aids to Navigation, Ed. 2, May 2004
3. Janesick, James R. *Scientific Charge-Coupled Devices*, 2001
4. Gonzales, R. C. and Woods, R. E., *Digital Image Processing*, 1994
5. IALA Guideline No. 1041, On Sector Lights, Dec 2004

Liitteet

Liite A – Laskenta-algoritmi

```
function [M] = seklo();
% Indicates each frames colour and gives time limits of each colour
% sector
% [M] = seklo();
% M = seccol2, variR, variG, variB
%
% seccol = Indicates frame colour (G = 3, R = 2, W = 1)
% variR, variG, variB = vector of channel values in frames
%
% Kalle Ryynänen (c) 12.7.2006

rootdir = 'E:\seklo\data\nassy20061002\';
[filename, pathname] = uigetfile(rootdir, 'Select video file');

ai = aviinfo([pathname filename]); % Luetaan videoclipin tiedot
NumberOfFrames = ai.NumFrames % Luetaan ja tulostetaan näytölle
% kehysten lukumäärä
% Kysytään käyttäjältä lukuväli ja lukutaajuus
beg = input('Type a number of a first frame: ');
fs = input('Type a sample frequency: ');
fin = input('Type a number of a last frame: ');
% Määritetään kuvan keskipiste myöhempää käyttöä varten
center{beg-fs} = [ai.Width/2 ai.Height/2];

% Kello käyntiin
tic;

% Aloitetaan kehysten käsittely yksitellen
for ii = beg:fs:fin;

% Luetaan ensimmäinen kehys käsittelyä varten
avi = aviread([pathname filename],ii);

I = avi.cdata;
% figure; imagesc(I); title(ii);

% Etsitään kynnystämällä valolähteitä kuvasta
% Täyttää reiät siinä valoalueella ja poistaa kaiken pikku roskan kok
% alle 50 pikseliä
BW = double(sum(I,3) > 50);
mask = imfill(bwareaopen(BW,30),'holes');
% figure; imagesc(mask); colormap summer;

% Numeroi löydetyt alueet 0,1,2,3...
BWL = bwlabel(mask);
MI{ii} = regionprops(BWL); % Antaa labeloitujen alueiden koordinaatit

% distcomp vertailee edellisten freimien alueiden liikkuvuutta ja
% valitsee vähiten liikkuneen valopisteen
center{ii} = distcomp(MI{ii},BW, center{ii-fs});

% Luodaan uusi maski, jossa käytetään ainoastaan valittua aluetta
maskC = crmask(BWL, center{ii}, mask);
```

```

% figure; imagesc(maskC); colormap summer;

R = double(I(:,:,1)); % jaetaan kuva kanaviinsa
G = double(I(:,:,2));
B = double(I(:,:,3));

% Tyhjä frame saa suoraan arvon 0
if length(find(maskC)) > 0;
    variR(ii) = sum(sum(R.*maskC))/length(find(maskC)); % Laskee
    punaisen kanavan värin keskiarvon valon alueella
    variG(ii) = sum(sum(G.*maskC))/length(find(maskC)); % Vihreälle
    sama kuin edellä
    variB(ii) = sum(sum(B.*maskC))/length(find(maskC)); % Siniselle
    sama kuin edellä
    varikokR(ii) = sum(sum(R.*maskC)); % Laskee punaisesta kanavasta
    erotetulla valonalueella olevan valon kokonaismäärän
    varikokG(ii) = sum(sum(G.*maskC)); % Vihreälle sama kuin edellä
    varikokB(ii) = sum(sum(B.*maskC)); % Siniselle sama kuin edellä
else
end

end

%figure; plot(1:length(seccol), seccol);

for ii = 1:(length(in)-1);
    if in(ii+1)-in(ii)>4;
        sl = (round((in(ii+1)-(in(ii)+1))*0.25));
        mxR((in(ii)+1):(in(ii+1)-1)) =
max(variR((in(ii)+1+sl):(in(ii+1)-1-sl)));
        mxG((in(ii)+1):(in(ii+1)-1)) =
max(variG((in(ii)+1+sl):(in(ii+1)-1-sl)));
        mxB((in(ii)+1):(in(ii+1)-1)) =
max(variB((in(ii)+1+sl):(in(ii+1)-1-sl)));
        elseif (in(ii+1)-in(ii)>1)&&(in(ii+1)-in(ii)<=4);
            mxR((in(ii)+1):(in(ii+1)-1)) =
max(variR((in(ii)+1):(in(ii+1)-1)));
            mxG((in(ii)+1):(in(ii+1)-1)) =
max(variG((in(ii)+1):(in(ii+1)-1)));
            mxB((in(ii)+1):(in(ii+1)-1)) =
max(variB((in(ii)+1):(in(ii+1)-1)));
        else
            end
    end
end

```



```
% Määritetään jokaisen framin väri ja indeksoidaan havaitut värit
```

```
for ii=1:(length(mxR));  
if mxR(ii) == 0;  
    seccol(ii) = 0; %'no colour';  
elseif mxR(ii)/mxG(ii)>10; % R/G ratio > 2 for normal video camera  
    seccol(ii) = 2; %'red';  
elseif (mxB(ii)/mxR(ii))>(mxR(ii)/mxG(ii));%||variR(ii)<variG(ii);  
    seccol(ii) = 3; %'green'; % toimii myös foculuksen kanssa  
elseif mxG(ii)/mxB(ii)>1.7;  
    seccol(ii) = 1; % 'white'  
else  
    seccol(ii) = 0; %changing color  
end  
end
```

```
% Poistetaan äkilliset poikkeama piikit datasta vertailemalla  
% edellisiin kehyksiin
```

```
seccol2 = [];  
for ii = (beg+2*fs):fs:(fin-(2*fs));  
    if (seccol(ii)~=seccol(ii-  
2*fs))&&(seccol(ii)~=seccol(ii+2*fs))&&(seccol(ii)>0);  
        if seccol(ii-2*fs)>0;  
            seccol2(ii)=seccol(ii-2*fs);  
        else  
            if ii>100;  
                for fst = 100:(-2*fs):(2*fs);  
                    if seccol(ii-fst)>0;  
                        seccol2(ii)=seccol(ii-fst);  
                    else  
                        end  
                end  
            else  
                for fst = ii-2*fs:-2*fs:2*fs;  
                    if seccol(ii-fst)>0;  
                        seccol2(ii)=seccol2(ii-fst);  
                    else  
                        end  
                end  
            end  
        end  
        seccol2(ii)=seccol(ii);  
    else  
        end  
end
```

```
%% Tulostus  
% Havaittujen värien tulostus
```

```
in2 = find(seccol2>0);  
figure; plot(in2,seccol2(in2),'-x');axis([beg fin 0 4]);  
line([beg fin], [3 3],'color',[0 0.75 0],'LineWidth',2);  
line([beg fin], [2 2],'color','r','LineWidth',2);  
line([beg fin], [1 1],'color','k','LineWidth',2);
```

```

in = find(seccol2>0);
inG = find(seccol2==3);
inR = find(seccol2==2);
inW = find(seccol2==1);

```

```

% Etsitään suhteiden maksimiarvojen maksimi

```

```

ind = find(variR>0);
maksr=max([max(variB(ind)./variR(ind)) max(variG(ind)./variB(ind))
max(variR(ind)./variG(ind))]);

```

```

% Etsitään värikanavien maksimit

```

```

maksv = max([max(variR) max(variG) max(B)]);

```

```

%
figure;plot(ind,variB(ind)./variR(ind),'b',ind,variG(ind)./variB(ind)
,'g',ind,variR(ind)./variG(ind),'r'); legend('B/R', 'G/B', 'R/G');
title('Colour ratios');
for ii=2:(length(in)-1);
    if (seccol2(in(ii))~=seccol2(in(ii+1)))&&(seccol2(in(ii))==3);
line([in(ii) in(ii)], [0 maksr],'color',[0 0.75 0],'LineWidth',2);
        elseif
(seccol2(in(ii))~=seccol2(in(ii+1)))&&(seccol2(in(ii))==2);
line([in(ii) in(ii)], [0 maksr],'color','red','LineWidth',2);
        elseif
(seccol2(in(ii))~=seccol2(in(ii+1)))&&(seccol2(in(ii))==1);
line([in(ii) in(ii)], [0 maksr],'color','black','LineWidth',2);
        elseif (seccol2(in(ii))~=seccol2(in(ii-1)))&&
(seccol2(in(ii))==2); line([in(ii) in(ii)], [0 maksr],'color',[0 0.75
0],'LineWidth',2);
        elseif (seccol2(in(ii))~=seccol2(in(ii-1)))&&
(seccol2(in(ii))==2); line([in(ii) in(ii)], [0
maksr],'color','red','LineWidth',2);
        elseif (seccol2(in(ii))~=seccol2(in(ii-1)))&&
(seccol2(in(ii))==1); line([in(ii) in(ii)], [0
maksr],'color','black','LineWidth',2);
        else
end
end
end

```

```

figure;plot(ind,variR(ind),'r',ind,variG(ind),'g',ind,variB(ind),'b')
; legend('R','G','B'); title('Channel values');
for ii=2:(length(in)-1);
    if (seccol2(in(ii))~=seccol2(in(ii+1)))&&(seccol2(in(ii))==3);
line([in(ii) in(ii)], [0 maksv],'color',[0 0.75 0],'LineWidth',2);
        elseif
(seccol2(in(ii))~=seccol2(in(ii+1)))&&(seccol2(in(ii))==2);
line([in(ii) in(ii)], [0 maksv],'color','red','LineWidth',2);
        elseif
(seccol2(in(ii))~=seccol2(in(ii+1)))&&(seccol2(in(ii))==1);
line([in(ii) in(ii)], [0 maksv],'color','black','LineWidth',2);
        elseif (seccol2(in(ii))~=seccol2(in(ii-1)))&&
(seccol2(in(ii))==3); line([in(ii) in(ii)], [0 maksv],'color',[0 0.75
0],'LineWidth',2);
        elseif (seccol2(in(ii))~=seccol2(in(ii-1)))&&
(seccol2(in(ii))==2); line([in(ii) in(ii)], [0
maksv],'color','red','LineWidth',2);
        elseif (seccol2(in(ii))~=seccol2(in(ii-1)))&&

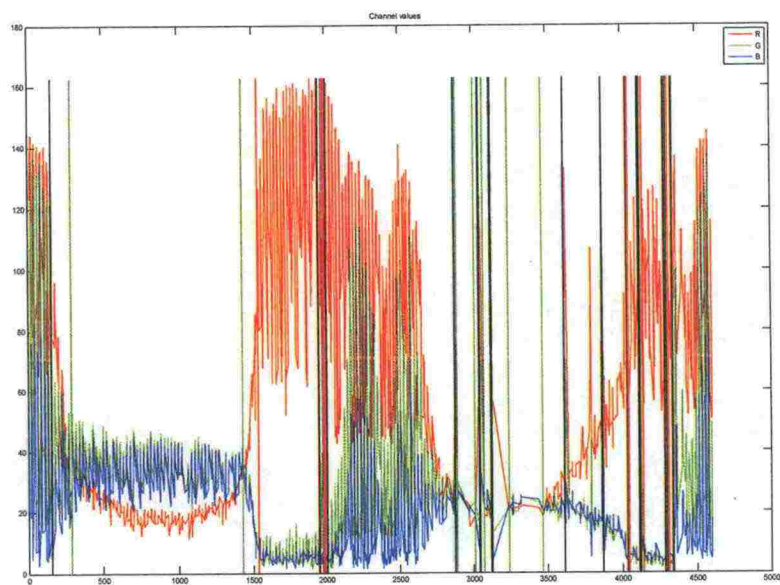
```



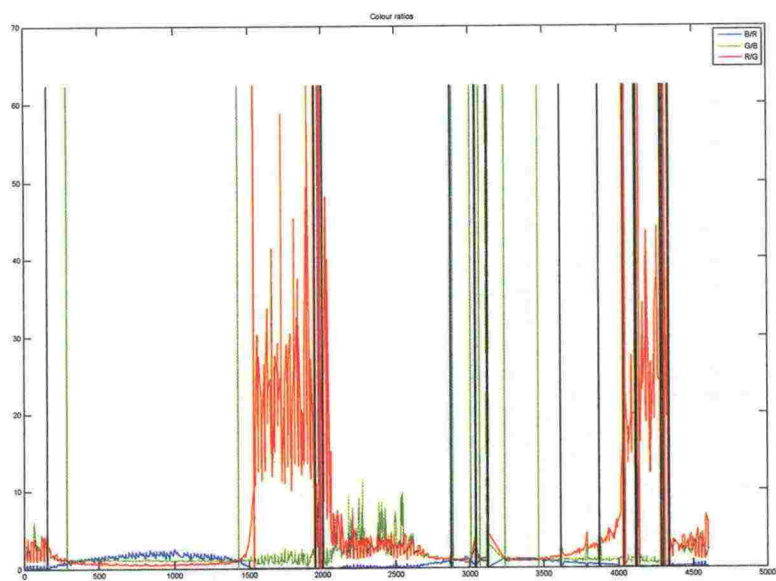
```
        end
    else
    end
end
```

```
M.seccol = seccol2;
M.variR = variR;
M.variG = variG;
M.variB = variB;
toc;
```

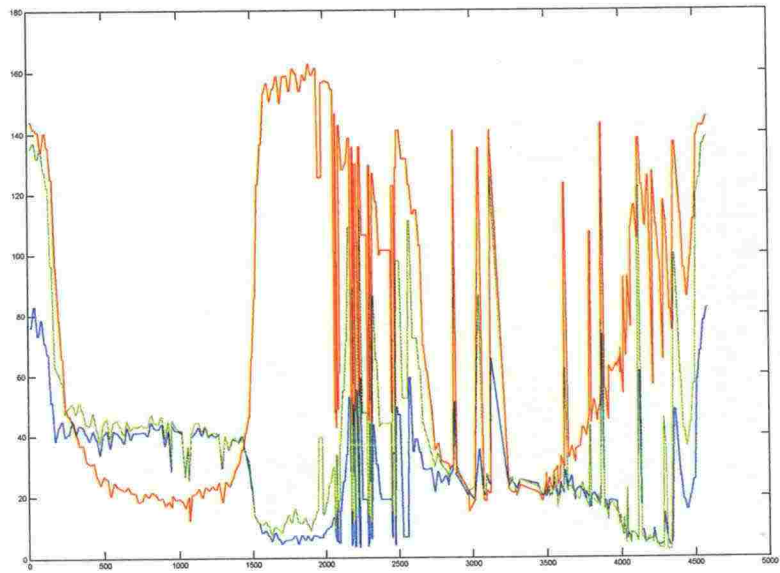
Liite B



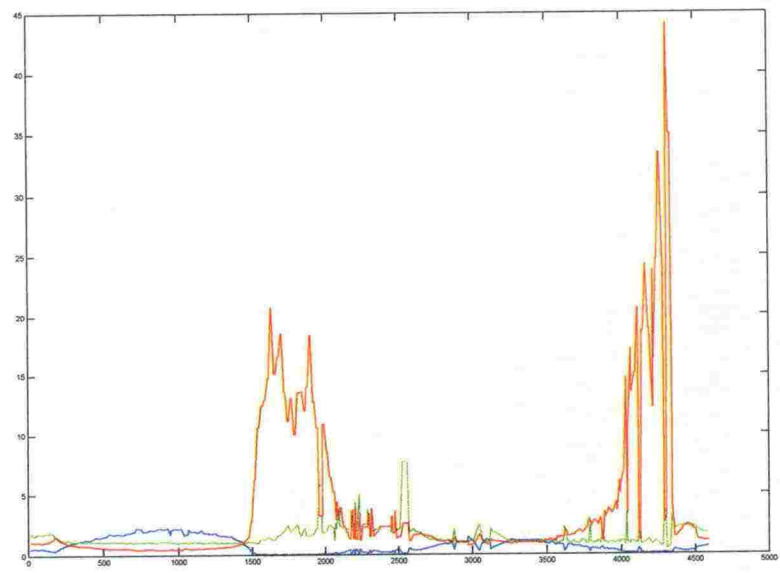
Kuva 1: Mittauslegin yksi mitatut värikanavien arvot



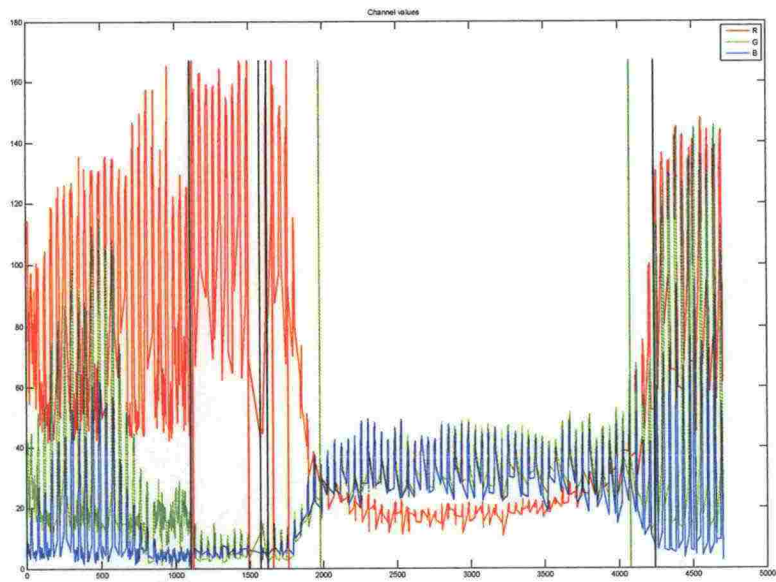
Kuva 2: Mittauslegin yksi mitatut värikanavien suhteiden arvot



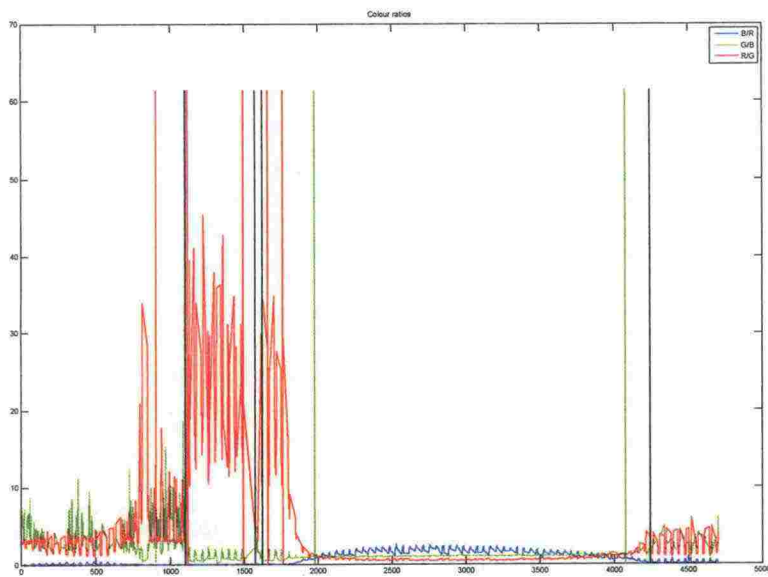
Kuva 3: Mittauslegin yksi keskiarvostetut värikanavien arvot



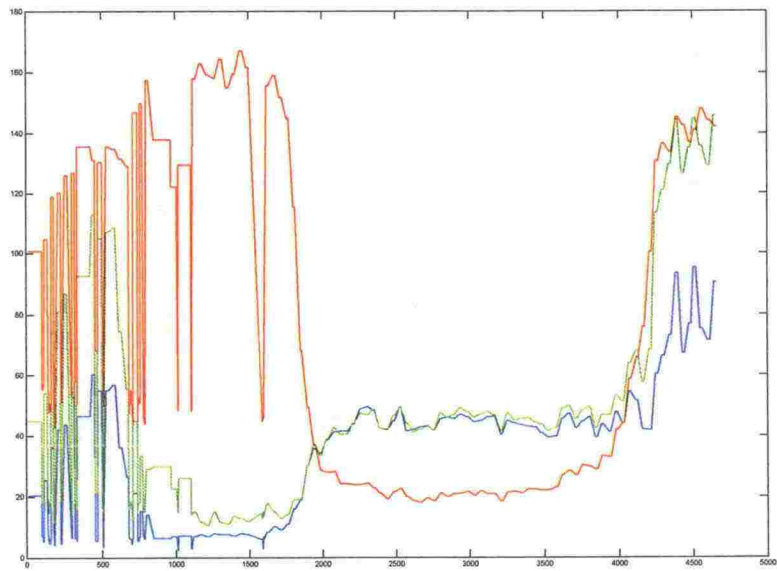
Kuva 4: Mittauslegin yksi keskiarvostetut värikanavien suhteiden arvot



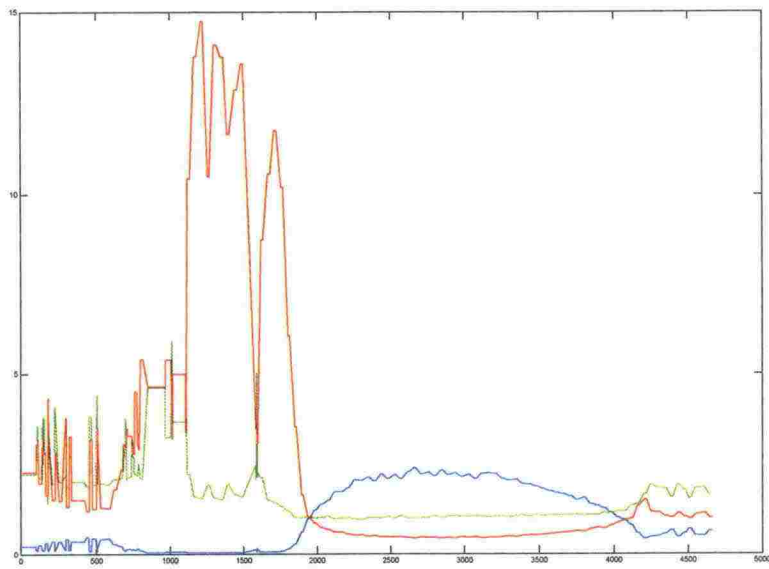
Kuva 5: Mittauslegin kaksi mitatut värikanavien arvot



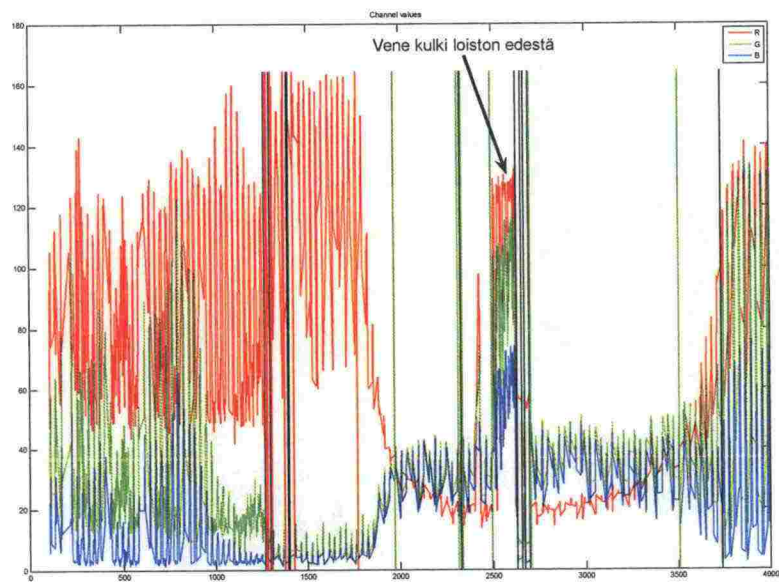
Kuva 6: Mittauslegin kaksi mitatut värikanavien suhteiden arvot



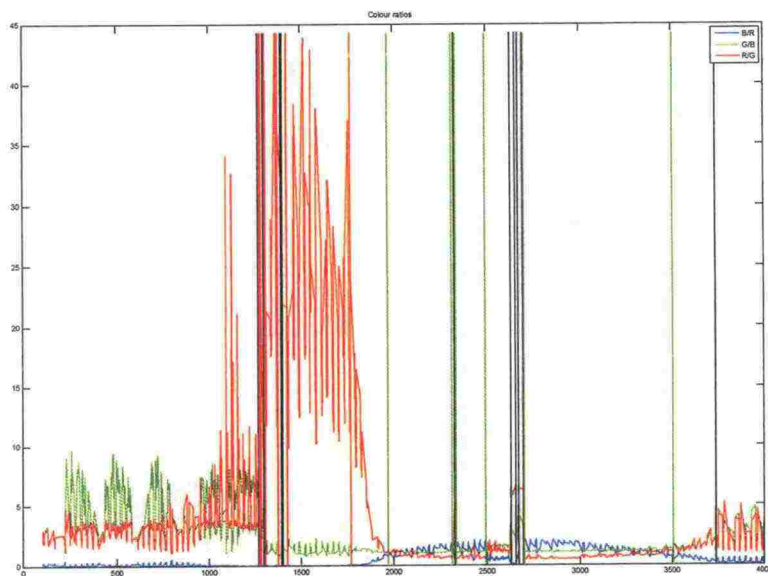
Kuva 7: Mittauslegin kaksi keskiarvostetut värikanavien arvot



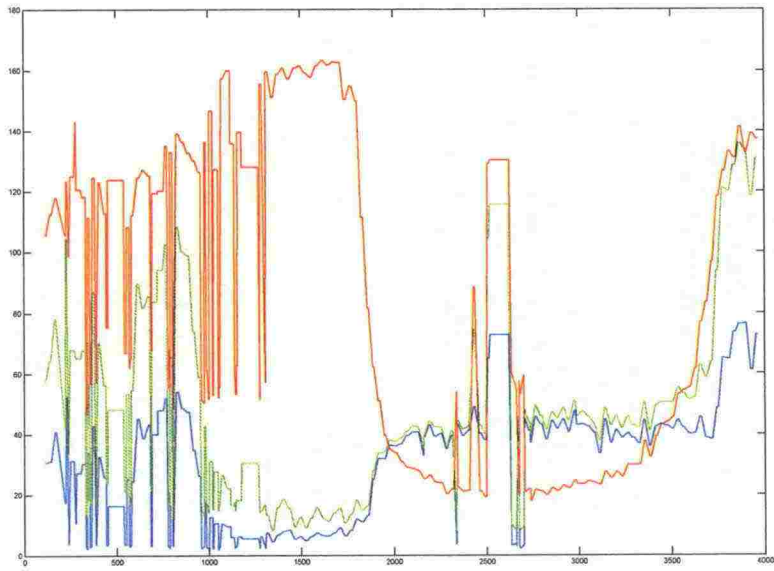
Kuva 8: Mittauslegin kaksi keskiarvostetut värikanavien suhteiden arvot



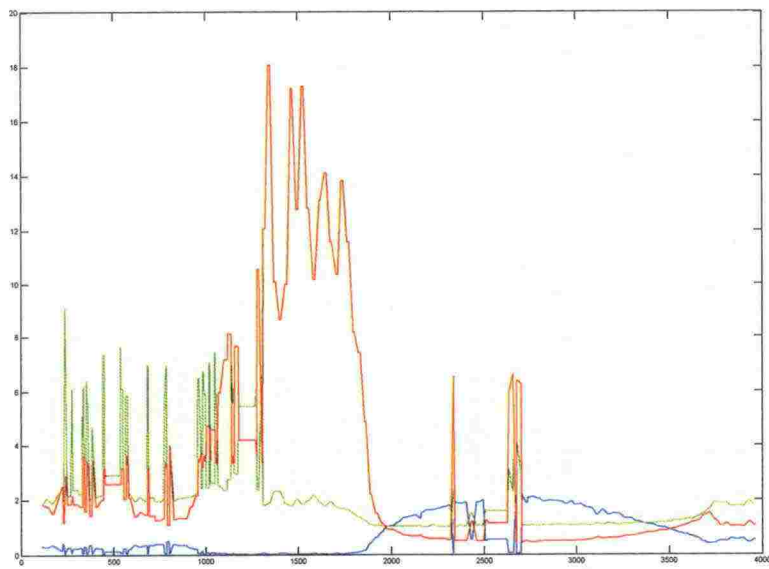
Kuva 9: Mittauslegin kolme mitatut väriskanavien arvot



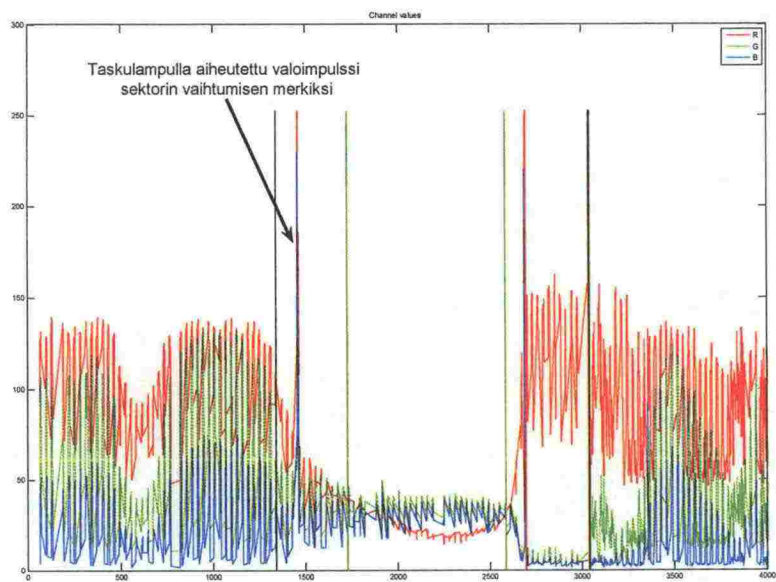
Kuva 10: Mittauslegin kolme mitatut väriskanavien suhteiden arvot



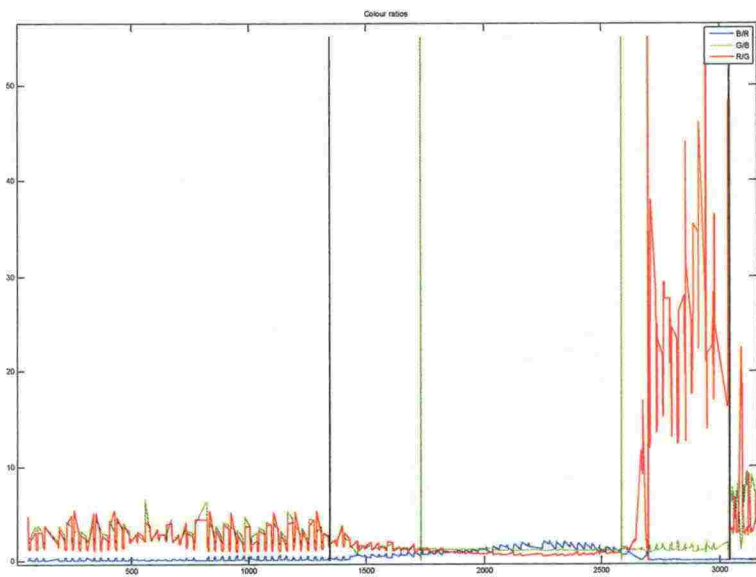
Kuva 11: Mittauslegin kolme keskiarvostetut värikanavien arvot



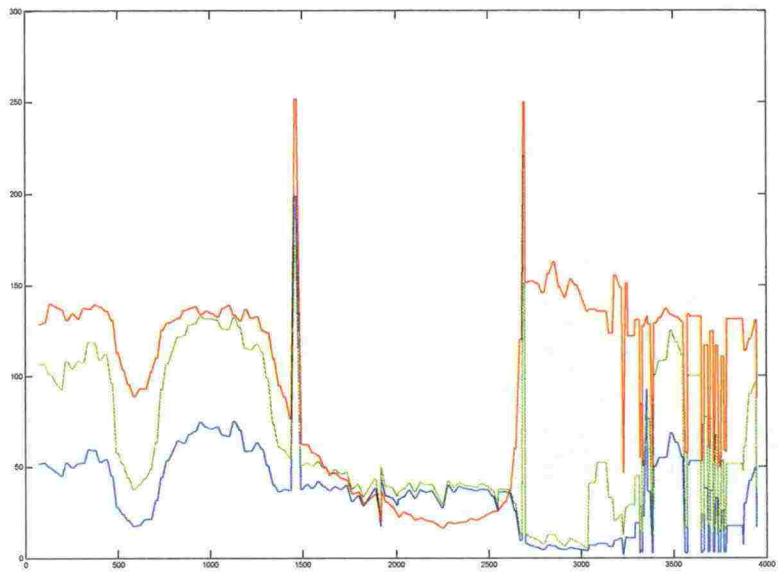
Kuva 12: Mittauslegin kolme keskiarvostetut värikanavien suhteiden arvot



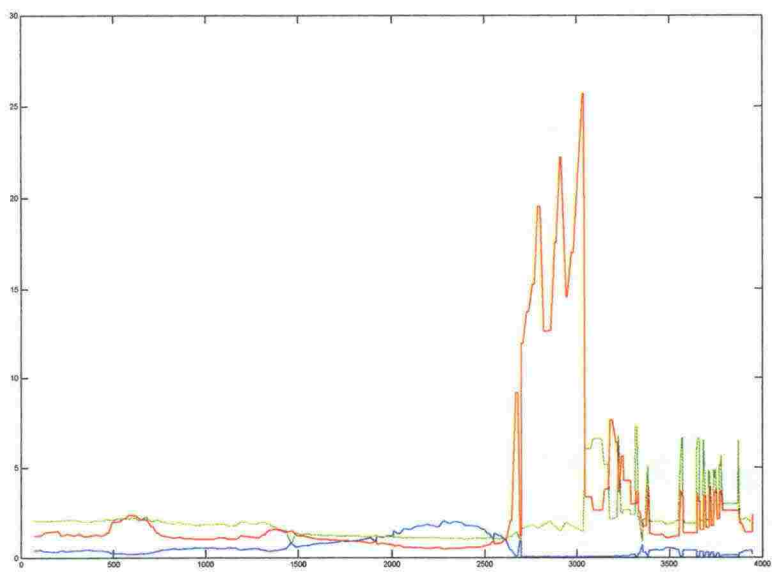
Kuva 13: Mittauslegin neljä mitatut värikanavien arvot



Kuva 14: Mittauslegin neljä mitatut värikanavien suhteiden arvot



Kuva 15: Mittauslegin neljä keskiarvostetut värikanavien arvot



Kuva 16: Mittauslegin neljä keskiarvostetut värikanavien suhteiden arvot