



TAMPEREEN TEKNILLINEN YLIOPISTO

LAURI SOINI

DIGITAALISEN KUVAMANIPULOINNIN HAVAITSEMINEN

Diplomityö

Tarkastaja: professori Olli Yli-Harja
Tarkastaja ja aihe hyväksytty
Automaatio-, kone- ja materiaalitek-
niikan tiedekuntaneuvoston
kokouksessa 6. syyskuuta 2010

TIIVISTELMÄ

TAMPEREEN TEKNILLINEN YLIOPISTO

Automaatiotekniikan koulutusohjelma

SOINI, LAURI: Digitaalisen kuvamanipuloinnin havaitseminen

Diplomityö, 60 sivua

Elokuu 2011

Pääaine: Signaalinkäsittely

Tarkastaja: professori Olli Yli-Harja

Avainsanat: Kuvamanipulointi, digitaalikamera, kuvantamisprosessi

Kuvankäsittelyohjelmistojen saatavuuden ja helppokäyttöisyyden kehittyminen on kyseenalaistanut digitaalisen valokuvan käytön todisteena, mikäli kuvan alkuperäisyydestä ei voida olla varmoja. Tästä syystä digitaalisen kuvamanipuloinnin havaitsemisesta on muodostunut yhä tärkeämpi tutkimusaihe. Tässä työssä keskitytään manipuloidun alueen, sekä JPEG-uudelleenpakkauksen havaitsemiseen kuluttajaluokan digitaalikameran kuvantamisprosessia hyväksikäyttäen. Työn tavoitteena on selvittää eri menetelmien toimivuus erilaisilla kuvamanipulaatioilla, sekä luoda helppokäyttöinen toteutus menetelmien testausalustaksi.

Lukija perehdytetään ensin kirjallisuusselvityksen tapaan digitaalisen kameran komponentteihin ja kuvankäsittelyprosesseihin. Tämän jälkeen esitellään kuvamanipulaatiotyypit, sekä matemaattisia menetelmiä kuvantamisprosessin jälkeisen kuvamanipuloinnin havaitsemiseksi. Esitellyn teorian pohjalta esitellään toteutukset kolmea eri kuvantamisprosessin vaihetta hyväksikäyttävälle menetelmälle. Toteutetut menetelmät perustuvat kameran objektiivin epätäydellisyyteen, kuvasensorin aiheuttamaan kohinaan, sekä JPEG-uudelleenpakkauksen aiheuttamiin artefakteihin diskreetissä kosinimuunnoshistogrammissa. Kokeellisessa osuudessa toteutettuja menetelmiä testataan kuvilla, joiden sisältöä on muutettu tietyllä alueella, sekä jälkikäsitelty koko kuvan alalta sisällönmuutoksen jälkeen. Muokatut kuvat pakataan sekä häviöttömästi, että JPEG-algoritmilla. Toteutuksen ja testitilanteiden pohjalta lukijan on helpompi arvioida toteutetun menetelmän käyttötilanteiden rajoituksia ja soveltuvuutta kuvamanipuloinnin havaitsemissovelluksiin.

ABSTRACT

TAMPERE UNIVERSITY OF TECHNOLOGY
Master's Degree Programme in Automation Technology

SOINI, LAURI: Detection of digital image manipulation

Master of Science Thesis, 60 pages

August 2011

Major: Signal processing

Examiner: Professor Olli Yli-Harja

Keywords: Digital image manipulation, digital camera, image processing pipeline

As image manipulation software has become more available and easier to use, digital photograph as evidence has become questionable if the authenticity of the photograph cannot be proven. This is why detecting digital image manipulation has become more important field of study. In this thesis we concentrate on detecting manipulated area and double JPEG-compression using the image processing pipeline found in consumer class digital cameras. The objective for this thesis is to clarify the usability of different methods in different cases of image manipulation and to create easy-to-use implementation platform for testing the implemented methods.

In the literature study part, background for the thesis is being explored. The background includes components and image processing pipeline of a generic digital camera, types of manipulation and mathematical methods for detecting image manipulation. Thereafter, implementations based on some of the previously explored methods are presented. The implemented methods are based on imperfections in the objective, sensor noise and the artifacts in the discrete cosine transform histogram after second JPEG-compression. In the experimental part, the implemented methods are tested with images which have been manipulated on certain area(s) and also enhanced on the area of the entire image after the manipulation. The manipulated images are compressed with lossless and JPEG-algorithms. The implementation and different test-cases provide the reader with understanding of suitability and limitations of an implemented method for future implementations of image manipulation detection software.

ALKUSANAT

Kuvamanipulointi ja sen tunnistaminen ovat varsin mielenkiintoisia aiheita. Siksi tein henkilökohtaisen valinnan jatkaa projektia, vaikka työn aloitusvaihe kärsi monenlaisista käytännön ongelmista.

Rikosinsinööri Antti Lehmussolan panos työn ideoinnissa ja erityisesti työhön motiivoinnissa on ollut merkittävä. Ilman häntä koko työtä tuskin olisi edes aloitettu. Haluan kiittää myös professori Olli Yli-Harjaa työn sisältöön ja rakenteeseen liittyvistä ohjeista. Lopuksi ilmaisen lämpimimmät kiitokseni perheelleni, jotka ovat tukeneet minua koko opiskelun ajan. Kiitos kuuluu myös kihlatulleni arvokkaasta tuesta.

Aurassa, 3. heinäkuuta 2011

Lauri Soini

SISÄLLYS

1	Johdanto	10
2	Digitaalinen kuvantamisprosessi.....	12
2.1	Kameran optiikka	12
2.2	Kuvasensori.....	13
2.3	Värierottelu	13
2.4	Tavallisimmat kuvanmuokkausprosessit	14
2.4.1	A/D-muunnos ja tasokorjaus	15
2.4.2	Valkotasapainotus	15
2.4.3	Väri-interpolointi	15
2.4.4	Väri- ja gammakorjaus.....	16
2.4.5	Muut kuvanmuokkausprosessit	16
2.5	Kuvan tallennus ja pakkausmenetelmät	17
2.6	Syntyprosessin jälkeinen manipulointi	19
2.6.1	Komposiittikuva.....	20
2.6.2	Metamorfoosi.....	21
2.6.3	Retusointi	21
2.6.4	Kuvan parantaminen	21
2.6.5	Kuvan luominen.....	21
3	Kuvamanipuloinnin havaitseminen.....	22
3.1	Valaistuksen yhdenmukaisuus	22
3.2	Värivärästymät	22
3.3	Värisuodinmatriisi ja interpolointi	24
3.4	Sensorikohina.....	26
3.5	JPEG-uudelleenpakkaus.....	30
3.6	Kopioi-siirrä-manipuloinnin tunnistaminen.....	31
3.6.1	Pääkomponenttianalyysi	31
3.6.2	SIFT-algoritmi	33
4	Toteutus.....	35
4.1	Värivärästymiin perustuvan menetelmän toteutus.....	35
4.1.1	Gradienttien estimointi	35
4.1.2	Väärästymämallien estimointi.....	36
4.1.3	Luokitin.....	37
4.2	Sensorikohinamenetelmän toteutus.....	37
4.2.1	Sisällönpoistosuodatin	37
4.2.2	Sensoriestimaatti.....	38
4.2.3	Opetusnäytteet	38
4.2.4	Luokitin.....	39
4.3	DCT-histogrammimenetelmän toteutus	40
5	Tulokset.....	42
5.1	Testikuvat.....	42

5.2	Sensorikohinamenetelmä	44
5.2.1	Esimerkkikuvien testitulokset.....	47
5.2.2	JPEG-uudelleenpakkauksen vaikutukset testituloksiin	48
5.3	Värivääristymämenetelmä.....	49
5.4	DCT-histogrammimenetelmä.....	52
6	Johtopäätökset.....	54
	Lähteet.....	56

TERMIT JA NIIDEN MÄÄRITELMÄT

α	Vääristymäparametri
$\alpha_{u,v}$	Interpolointimatriisin alkio
γ	Gammakorjauskerroin
θ_a	Valon tulokulma linssille
θ_b	Valon lähtökulma linssiltä
λ_j	Ominaisarvo
μ	Ikkunan sisältämä paikallinen keskiarvo
Θ	Satunnaiset kohinakomponentit
Θ_q	Kvantisointikohina
Λ	Sensorin satunnaiset kohinaelementit
ρ	Kahden vektorimuotoisen kuva-alueen välinen korrelaatio
\mathcal{E}	Sisältösuodatuksen aiheuttamat kohinatermit
φ	Normaalijakautunut, nollakeskinen virhe tuntemattomalla keskihajonnalla
σ	Normaalijakauman keskihajonta
σ_η	Ikkunanfunktion η paikallinen keskihajonta
σ_g	Gauss-funktion keskihajonta
σ_n	Pikseliarvon normaalijakautunut keskihajonta
ω	Taajuusparametri
a	Kvantisointiaskelparametri
a_j	Pääkomponenttialkio
a_n	Signaalin arvosta riippumaton kohinakomponentti
b	Kvantisointiaskelparametri toisella JPEG-pakkauskerroilla
b_f	Suodatettu tulosalkio
b_n	Signaalin arvosta riippuva kohinakomponentti
c	Kameran indeksi
c_0	Kosinimuunnoksessa käytetty funktio
d	Gauss-erotuskuva
d_x	Yksiulotteinen, x-akselin suuntainen derivaattasuodin
d_y	Yksiulotteinen, y-akselin suuntainen derivaattasuodin
E	Virhepinta
f	Diskreetin kuvasignaalin yhden kanavan yksi lukuarvo
f_0	Gammakorjattu näkymäkomponentti
\hat{f}_0	Approksimaatio gammakorjatusta näkymäkomponentista
f_g	Vihreä värikaista
f_i	Väri-interpoloidun kuvan yksi värikaista
f_k	Opetuskuva
f_r	Punainen värikaista

f_{ra}	Vääristymäkorjattu punainen värিকাista
f_y	Kohinaton näkymäkomponentti
∇f	Gradientti
∇f_x	Gradientti x-akselin suunnassa
∇f_y	Gradientti y-akselin suunnassa
F	Diskreetin kosinimuunnoksen alkio
F_n	Kohinasuodatin
F_q	Kvantisoitu kosinimuunnosalkio
g	Gauss-funktio
g_a	Vihreän värikanavan arvo
g_f	Värikanavan vahvistuskerroin
G	Normaalijakauman kertymäfunktio
I	Yhteinen informaatio
k	Diskreetti valoherkkyyskerroinfunktio
k_g	Skaalauskerroin
\hat{k}	Estimoitu valoherkkyyskerroinfunktio
l	Gauss-skaalauskuva
M_1	Luokka interpolointimalliin kuuluville pikseleille
M_2	Luokka interpolointimallin ulkopuolisille pikseleille
n_a	Ilman taittokerroin
n_b	Linssin taittokerroin
N_c	Opetusnäyttekuvien määrä
N_p	Kuvasta leikattujen kuvalohkojen määrä
N_r	Luokittimen opetukseen käytettävien kuvalohkojen määrä
N_s	Suuntavektorien määrä
p	Pikseleiden määrä kuvalohkossa
p_x	Yksiulotteinen, x-akselin suuntainen alipäästösuodin
p_y	Yksiulotteinen, y-akselin suuntainen alipäästösuodin
P	Todennäköisyys
P_r	Prioritodennäköisyys
q_a	Kvantisoitu DCT-arvo
q_{ab}	Kahteen kertaan kvantisoitu DCT-arvo
Q	Kvantisointiaskeleen suuruus
Q_c	JPEG-laatuparametri
r_a	Punaisen värikanavan arvo
s	Taajuusriippuvainen kvantisointiaskel
w	Sisältösuodatettu kohinakuva
w_e	Virhepinnan painoparametri
$\underline{\theta}$	Mitattu paikallinen suuntavirhevektori
$\bar{\theta}$	Paikallinen suuntavirhevektoriestimaatti

\underline{a}	Pääkomponenttivektori
\underline{a}_{iq}	Kvantisoitu pääkomponenttivektori
\underline{e}	Ominaisvektori
\underline{s}_i	Pääkomponenttivektorimatriisin sarake
\underline{x}	Vektorimuotoinen kuvalohko
A	Interpolointimatriisi
C	Taajuusriippuvainen kvantisointimatriisi
\mathbf{C}_x	Kovarianssimatriisi
S	Pääkomponenttivektorimatriisi
AFP	Uutistoimisto (Agence France-Presse)
CCD	Eng. Charge-coupled device
CMOS	Eng. Complementary metal-oxide-semiconductor
DCT	Diskreetti kosinimuunnos (Eng. Discrete cosine transformation)
EM	Eng. Expectation-maximization
EXIF	Tiedostomuoto (Exchangeable Image File Format)
Flatfielding	Valovasteen epäyhdenmukaisuuden poistamiseen käytetty tekniikka
FPN	Muuttumaton sensorikohina (Eng. Fixed pattern noise)
HVS	Ihmisen näkökyky (Eng. Human Visual System)
JPEG	Kuvanpakkausstandardi (Joint Photographic Experts Group)
JPEG2000	Kuvanpakkausstandardi (Joint Photographic Experts Group 2000)
Kvantamisprosessi	Valmiin kuvan aikaansaamiseksi suoritettava prosessi
Kvanttitehokkuus	Eng. Quantum efficiency
ML	Suurin uskottavuus (Eng. Maximum likelihood)
MMSE	Pienin keskineliövirhe (Eng. Minimum mean square error)
PCA	Pääkomponenttianalyysi (Eng. Principle component analysis)
PNU	Pikselikohinan epäyhdenmukaisuus (Eng. Pixel noise non-uniformity)
PRNU	Valovasteen epäyhdenmukaisuus (Eng. Photoresponse non-uniformity)
SIFT	Hahmontunnistusmenetelmä (Eng. Scale-invariant feature transform)
TIFF	Häviötön kuvanpakkausstandardi (Eng. Tagged Image File Format)
TIFF/EP	Tiedostomuoto (Eng. Tagged Image File Format / Electronic Photography)
Värisuodinmatriisi	Eng. Color filter array

1 JOHDANTO

Valokuva on perinteisesti kelvannut todistusaineistoksi tai lehtikuvaksi sellaisenaan. Perinteisen valokuvan muokkaus on hankalaa ja vaatii erityistä asiantuntemusta. Nykyiset digitaaliset kamerat ja ohjelmistot tekevät taidokkaasta kuvamanipuloinnista helppoa. Mikäli kuvan lähteeseen ei voida luottaa, digitaalisen valokuvan todistusarvo on joutunut kyseenalaiseksi.

Kuvamanipuloinnista on erityistä haittaa rikostekniselle tutkinnalle, jonka tarkoitus on todistaa jokin tapahtuma joko tapahtuneeksi tai tapahtumattomaksi. Rikosteknisen tutkinnan lisäksi kuvamanipuloinnin yleistymisestä on haittaa lehdistölle, jolle oikeellisten tietojen välittäminen on uskottavuuskysymys. Haittaa tai mielipahaa kuvamanipuloinnista voi myös koitua yksityiselle henkilölle tai organisaatiolle, mikäli manipuloitua kuvaa käytetään valheellisen tiedon levittämiseen tai suoranaiseen huijaukseen.

Kuvamanipuloinnin havaitsemista vaikeuttaa digitaalisten kuvien monimuotoisuus. Digitaalisen kuvan lähdelaitteena voi olla esimerkiksi kamera, kuvanlukija tai muu laite, joiden kuvantamisprosessit ovat täysin erilaisia. Myös kuvantamisprosessin lopputuloksena syntyneen kuvan ominaisuudet vaihtelevat. Kuva voi olla pakattu häviöttömällä tai häviöllisellä menetelmällä, skaalattu, käännetty ja niin edelleen. Kuvalle voidaan myös jälkikäteen tehdä koko joukko operaatioita, jotka eivät varsinaisesti muuta kuvan sisältöä, mutta muuttavat kuvan ominaisuuksia. Tällaisia operaatioita ovat esimerkiksi kontrastin muokkaus tai gammakorjaus, joita jokainen digitaalinen kamera tekee automaattisesti.

Aiheesta tehty tutkimus voidaan jakaa karkeasti ottaen kolmeen eri osa-alueeseen, joita ovat kuvan lähdelaitteen tunnistaminen, keinotekoisesti luotujen kuvien tunnistaminen, sekä kuvamanipuloinnin havaitseminen. Tämä työ keskittyy kuvamanipuloinnin havaitsemiseen, joskin jotkin esitellyistä menetelmien teorioista sopivat myös muihin osa-alueisiin. Ala on alkutekijöissään, sillä aihetta on käsitelty suurimmaksi osaksi julkaisuissa ja aiheesta on vain joitakin diplomitöitä.

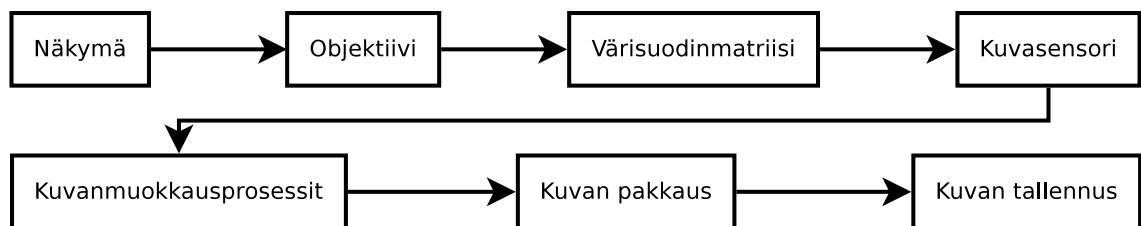
Erilaisten kuvamanipulointityyppien havaitsemiseen on kehitetty massoittain erilaisia menetelmiä, mutta niillä kaikilla on omat käyttötilanteita rajoittavat ongelmansa. Yhdellä menetelmällä ei myöskään voida osoittaa kuvan olevan oikea, osittain muokattu, useasta kuvasta koostuva komposiittikuva tai kokonaan keinotekoisesti tehty. Kannattaa huomioida, että uusien menetelmien kehittäjät pyrkivät usein antamaan mahdollisimman hyvän kuvan menetelmänsä toimivuudesta esittämällä julkaisuissaan menetelmän kannalta suotuisia testitapauksia. Kehitetyt kuvamanipuloinnihavaitsemismenetelmät perustuvat yleensä johonkin tiettyyn vaiheeseen digitaalisessa kuvantamis- tai

tallennusprosessissa. Tästä johtuen osa havaitsemismenetelmistä jää auttamattomasti hyödyttömäksi niiden teoreettisen pohjan poistuessa edellä mainituista prosesseista.

Kappaleessa kaksi lukija perehdytetään kirjallisuusselvityksen tapaan digitaalisen kameran komponentteihin, kuvankäsittelyprosesseihin, sekä kuvamanipulaatiotyyppeihin. Kappaleessa kolme esitellään matemaattisia menetelmiä syntyprosessin jälkeisen kuvamanipuloinnin havaitsemiseksi. Tämän jälkeen kappaleessa neljä esitellään teorian pohjalta toteutettuja menetelmiä. Kappaleen viisi kokeellisessa osuudessa toteutettuja menetelmiä testataan kuvilla, joiden sisältöä on muutettu tietyllä alueella, sekä jälkikäsitelty koko kuvan alalta sisällönmuutoksen jälkeen. Tässä työssä toteutettuja menetelmiä ei ole aikaisemmin testattu jälkikäsitellyillä ja uudelleenpakatuilla testikuvilla. Toteutuksen ja testitilanteiden pohjalta lukijan on helpompi arvioida toteutetun menetelmän käyttötilanteiden rajoituksia ja soveltuvuutta kuvamanipuloinnihavaitsemissovelluksiin.

2 DIGITAALINEN KUVANTAMISPROSESSI

Tässä kappaleessa selvitetään digitaalisen kameran teknisiä taustoja, sekä käydään läpi kuvantamisprosessia siltä osin kuin se on työn kannalta mielekästä. Kuvantamisprosessia on käsitelty laajasti toisaalla [1; 2]. Kuvantamisprosessiin kuuluvat kameran komponentit, sekä kuvanmuokkausprosessit, joita tarvitaan valmiin kuvan aikaansaamiseksi. Ensiksi esitellään kameran kuvantamisprosessia yleisellä tasolla, jonka jälkeen esitellään tarkemmin kameran optiikkaa, sensoritekniikkaa, värisuodatinmatriiseja, yleisempiä kuvanmuokkausprosesseja, sekä talletusmuotoja. Tämän jälkeen esitellään lyhyesti erilaisia kuvantamisprosessin jälkeisiä manipulointityyppejä, joiden jälkeen kuvaa ei voida enää väittää alkuperäiseksi.



Kuva 2.1. Tyypillisen kuluttajaluokan digitaalikameran kuvantamisprosessin eri vaiheet.

Nykyaikaiset digitaalikamerat sisältävät koko prosessiketjun raakadatan keräämisestä aina valmiin digitaalisen kuvan tallentamiseen, kuten kuvassa 2.1 on havainnollistettu. Kuvasensorilta luettava raakadata poikkeaa valmiista kuvasta erittäin paljon ja sitä on käsiteltävä valmiin kuvan aikaansaamiseksi. Useimmissa kameroissa on mahdollisuus myös raakadatan tallentamiseen ja prosessoida raakadata myöhemmin ilman kameran laitteiston asettamia rajoituksia. Valmiiksi prosessoitu pakkaamaton kuva veisi varsin suuren määrän muistia tallennettaessa, joten on kehitetty erilaisia häviöllisiä ja häviöttömiä pakkausmenetelmiä muistinkäyttöongelman ratkaisemiseksi. Valmiiseen kuvatiedostoon on yleensä tallennettu kuvan lisäksi tietoja kamerasta, sekä kuvanottohetkellä vallinneista olosuhteista.

2.1 Kameran optiikka

Digitaalikameran prosessiketjussa ovat ensimmäisenä optiset komponentit. Optisia komponentteja ovat tyypillisesti linssit, aukko, sekä suljin. Yhdessä näitä komponentteja kutsutaan objektiiviksi. Objektiivin pääasiallinen tehtävä on kerätä valoa kuvattavasta kohteesta kuvasensorille ja huolehtia oikean valomäärän keräämisestä. Joidenkin kameroiden objektiivit sisältävät myös kuvanvakaimen, mutta järjestelmä voidaan toteuttaa

myös objektiivin ulkopuolella. Objektiivin komponenteista suljin ja aukko kontrolloivat kerättävän valon määrää. Sulkimen tehtävä on säätää valotusajan pituutta. Aukolla säädetään tarkennusalueen suuruutta, sekä valon keräysnopeutta. Useimmissa tapauksissa näitä kahta komponenttia säädetään automaattisesti, joskin käsikäyttö on mahdollista [2]. Kaikkein halvimmista kameroissa säätö tapahtuu aina automaattisesti ja suljin on toteutettu elektronisesti kuvasensorin avulla. [1]

Useimmissa objektiiveissa on muuttuva polttoväli, sekä automaattinen tarkennusjärjestelmä. Toisaalta on olemassa myös kiinteällä polttovälillä varustettuja objektiiveja. Yleisimmin tällaisia objektiiveja löytyy hyvin halvoista kameroista. Toisaalta myös kalliita objektiiveja on saatavilla kiinteällä polttovälillä, sillä niiden kuvanlaatu on parempi verrattuna samanhintaisiin muuttuvalla polttovälillä varustettuihin objektiiveihin. [1]

2.2 Kuvasensori

Kuvasensori muuttaa fotoneita (valoa) elektroneiksi. Kaikki fotonit eivät kuitenkaan muutu elektroneiksi, vaan suhde on alhaisempi. Tätä suhdetta kutsutaan kvanttitehokkuudeksi (Quantum efficiency). Kvanttitehokkuutta alentavat esimerkiksi valoa absorboivat materiaalit sensorin päällä, sekä piin aiheuttama valon takaisinheijastuminen. Lisäksi kuvasensorilla on ominainen valon taajuudesta riippuva spektrivaste. Useimmat digitaaliset kuvasensorit on suunniteltu siten, että näkyvän valon kvanttitehokkuus on noin 90–100 prosenttia. Kuvasensori on herkkä myös näkyvän valon ulkopuolella oleville aallonpituuksille, joten ei-halutut aallonpituudet suodatetaan optisilla yli- ja alipäästösuotimilla ennen kuvasensoria. [1]

Yleisesti kameroissa käytetään CCD- (Charge-coupled device) tai CMOS-kuvasensoria (Complementary metal-oxide-semiconductor). Ensin mainitussa CCD-kennotyyppissä jokaisen valoherkän elementin sisältämä varaus muunnetaan jännitteeksi käyttämällä ulkoista muunninta. Tämä on varsin hidas prosessi verrattuna CMOS-kuvasensorin tapaan muuntaa varaus jännitteeksi erikseen jokaisella valoherkällä elementillä, mahdollisten myös muiden komponenttien integroinnin kuvasensorille. Näin voidaan vähentää järjestelmän hintaa, mutta integroidun järjestelmän tuottamassa kuvassa on enemmän kohinaa. [2]

2.3 Värierottelu

Väri-informaatio koodataan kolmella eri värillä. Näitä värejä ovat punainen, vihreä ja sininen. Värimalli perustuu ihmissilmän tapaan erottaa värejä kolmella erityyppisellä värireseptorilla. Näin ollen myös kuvasensorin tulee olla herkkä ihmissilmän havaitsemille aallonpituuksille. Kuitenkin kaikki piihin perustuvat kuvasensorit tuottavat vain monokromaattisia värisävyjä. Värien erotteluun on kehitetty neljä eri menetelmää: Väripyörä, kolmen kuvasensorin käyttö, värisuodinmatriisi, sekä Foveon®. [1]

Väripyörää käytettäessä kenno valotetaan peräkkäin eri värisuotimien lävitse. Käytännössä tämä tarkoittaa muita tekniikoita pidempää valotusaikaa ja sopii näin ollen vain täysin liikkumattomien kuvaamiseen tai tarvitsee voimakkaan ulkoisen valonlähteen valotusajan lyhentämiseksi.

Kolmen kuvasensorin menetelmässä valo jaetaan eri värikomponentteihin prisman avulla. Tämän jälkeen eri värikanavat ohjataan omille kuvasensoreille. Tätä tekniikkaa käytettäessä saavutetaan hyvä resoluutio, sekä värien erottelu. Kuitenkin menetelmän toteuttaminen on kallista ja järjestelmästä tulee fyysiseltä kooltaan suuri.

Värisuodinmatriisimenetelmässä (Color filter array) yksittäiselle kuvasensorin valoherkkää solua on käsitelty siten, että se reagoi vain rajoitetulle aallonpituusalueelle. Käytännössä jokaisen pikselin eteen on asetettu värisuodin. Koska yhdellä pikselillä voi olla vain yksi väriarvo, kaksi muuta väriä joudutaan interpoloimaan lähellä olevien eri väristen pikselien intensiteettiä arvosta. Tätä operaatiota kutsutaan väri-interpoloinniksi. Värisuodinmatriisina voidaan käyttää montaa erilaista värimatriisia, joista yleisimmin käytetään Bayer-matriisia [3]. Bayer-matriisissa jokainen neliönmuotoinen neljän pikselin alue koostuu punaisesta, sinisestä, sekä kahdesta vihreästä pikselistä. Muita värimatriiseja ovat muun muassa SMY, GMY+G, RGB+turkoosi ja niin edelleen. Värisuodatinmatriisimenetelmän ongelmana on laskostumisilmiön kasvanut riski. Laskostumista ilmenee, mikäli halutaan kuvata kohdetta, joka vaatii suurempaa resoluutiota kuin samanväristen pikselien spatiaalinen resoluutio muuten sallisi. Ongelma on ratkaistu asettamalla värisuotimen eteen kuvaa sumentava spatiaalinen alipäästösuodin. Näin ollen kuvan terävyyttä joudutaan parantamaan digitaalisesti myöhemmässä prosessointivaiheessa. Värisuodinmatriisin toteuttaminen on halpaa, mutta resoluutio on heikko ja menetelmä vaatii muita menetelmiä monimutkaisempaa kuvankäsittelyä.

Foveon® sensorissa käytetään päällekkäisiä valoherkkiä elementtejä [4]. Tekniikka käyttää hyväkseen valon pidempien aallonpituuksien ominaisuutta absorboitua syvemällä piissä lyhyempiin aallonpituuksiin verrattuna [5]. Näin ollen Foveon®-kuvasensori pystyy hyödyntämään enemmän fotoneja kuin värisuodattimeen perustuva ratkaisu, jossa suurin osa näkyvästä valosta suodatetaan pois ennen kuvasensoria. Tästä huolimatta menetelmän tuottama värierottelu on heikko johtuen lisääntyneestä raakadatan prosessoinnin tarpeesta. Ongelma korostuu erityisesti, mikäli halutaan kuvata alhaisissa valo-olosuhteissa. Koska menetelmä ei vaadi väri-interpolointia, lopullisessa värikuvassa on vain kolmasosa valoherkkien elementtien tuottamista pikseleistä.

Värisuodin Bayer-matriisilla on yleisimmin digitaalikameroissa käytetty värierottelumenetelmä. Muita menetelmiä käytetään lähinnä erikois- ja ammattilaiskameroissa. [1]

2.4 Tavallisimmat kuvanmuokkausprosessit

Tässä kappaleessa käydään läpi joitakin yleisimpiä automaattisia kuvanmuokkausprosesseja, joita tarvitaan käytettäessä värisuodinmatriisia. Tällaisia prosesseja ovat suoritussjärjestyksessä signaalinvahvistus, A/D-muunnos, tasokorjaus, valkotasapainotus,

interpolointi, värikorjaukset ja gammakorjaus. Edellä mainittujen lisäksi tehdään myös muita kuvanmuokkausprosesseja, kuten kuvanparannusta, kontrastikorjauksia, kuvan terävöittämistä ja niin edelleen. Signaalivahvistusta käytetään lyhentämään valotusai-kaa, mutta suuri signaalivahvistus lisää kohinaa. [1]

Kameran toteutus saattaa vaatia muokkausprosessien poistamista, lisäämistä tai prosessin vaiheiden järjestyksen muuttamista. Yleensä jokaisella kameravalmistajalla tai jopa yksittäisellä kameramallilla on oma yksilöllinen kuvanmuokkausprosessi. Näin ollen prosessia on lähes mahdotonta mallintaa yksittäisen, universaalien mallin pohjalta. Prosessoinnin jälkeen raakadatasta on saatu lopullinen kuva, joka talletetaan jollakin pakkausmenetelmällä.

2.4.1 A/D-muunnos ja tasokorjaus

Signaalivahvistuksen ja A/D-muunnoksen jälkeen signaali esitetään yleisemmin 8–14 bittisinä kokonaislukuina. Tämän jälkeen sisäinen prosessointi toimii yleensä suurem- malla bittimäärällä ja lisäksi A/D-muunnettu signaali normalisoidaan uudelle lukualu- eelle pyöristysvirheiden välttämiseksi. [1.] Tasokorjausta tarvitaan sensorissa aiheutu- van pimeän virran sekä objektiivissa aiheutuvien heijastumien poistamiseksi. Lisäksi kuvattavan kohteen vaatima dynaaminen alue saattaa ylittää tai kattaa vain osan senso- rin dynaamisesta alueesta. Tämä johtaa käytännössä siihen, että A/D-muunnetun signaa- lin nollataso ei vastaa kohteen mustaa tasoa. Tasokorjaus tehdään tyypillisimmillään raakadatan histogrammianalyysia hyväksikäyttäen. [6]

2.4.2 Valkotasapainotus

Valkotasapaino tarkoittaa kuvan värikanavien oikeaa suhdetta, jolla kuvan väritoisto näyttää luonnolliselta. Kohteen heijastaman valon spektri on riippuvainen valonlähtei- den spektristä. Ihmisten näköaisti osaa mukautua eri tilanteissa siten, että kohteen värit havainnoidaan tarkasti valonlähteen spektristä riippumatta. Kuvasektorin spektrivaste on vakio, mikä johtaa huonoon väritoistoon. Tästä syystä tarvitaan valkotasapainotusta väritoiston korjaamiseksi. [1]

Eri tilanteissa ympäristöä valaistaan valonlähteillä, joilla on erilaiset spektrit. Sa- massa kuvassa voi olla myös useita erilaisia valonlähteitä. Mahdollisimman lähellä oi- keaa olevan valkotasapainon löytäminen automaattisesti edellyttää sellaisen alueen löy- tämistä kuvasta, joka on saturoimaton ja väreiltään neutraali eli heijastaa kaikkia väri- kanavia yhtä paljon. [2]

2.4.3 Väri-interpolointi

Väri-interpoloinnin tehtävä on rekonstruoida jokaiselle pikselille kaikki värikanavat värisuodatinmatriisiin jälkeisestä raakadatasta. Yleisin värisuodatinmatriisi on Bayer- matriisi. Bayer-matriisin käyttö johtaa siihen, että näytteistystaajuus vihreälle kanavalle on puolet sensorin näytteistystaajuudesta ja neljäsosa muille kanaville. Eri näytteistys- taajuudet asettavat korkean vaatimustason käytettävälle interpolointimenetelmälle. Vä-

rikanavat korreloivat usein toistensa kanssa, joten useat väri-interpolointimenetelmät käyttävät hyväksi tätä ominaisuutta. [2.] Gunturk et al. [7] ovat tehneet artikkelin erilaisista väri-interpolointimenetelmistä. Väri-interpoloinnin jälkeen kaikkien värikanavien jokaisella pikselillä on jokin väriarvo.

2.4.4 Väri- ja gammakorjaus

Sensorilta ulostuleva RGB-raakadata riippuu suuresti käytetyn sensorin spektrivasteesta ja muista komponenteista. Puhutaan laitteen sisäisestä väriavaruudesta. Ilman yleisesti tunnettua väriavaruutta, värien toisto laitteen ulkopuolella olisi automaattisesti väärä. Tästä syystä RGB-raakadata korjataan johonkin yleisesti tunnettuun, muissa laitteissa käytettyyn väriavaruuteen. Nyt värit toistuvat periaatteessa tarkasti samaa väriavaruutta tukevissa laitteissa, kuten esimerkiksi monitoreissa ja tulostimissa. Yleisesti käytettyjä väriavaruuksia ovat sRGB [8] ja Adobe RGB [9].

Yksittäisen valoherkän elementin ulostulo on suoraan verrannollinen siihen törmäävien fotonien määrään [1]. Näin pikselin intensiteettivaste on lineaarinen. Kuitenkin monitoreiden, kuten LCD-paneelien siirtofunktio sähköstä valoksi on epälineaarinen, tarkemmin ottaen eksponentiaalinen. Tämän lisäksi ihmisen näkökyky (HVS, Human Visual System) reagoi herkemmin intensiteetin välisiin eroihin matalilla kuin korkeilla intensiteeteillä. Tästä syystä gammakorjattu kuvasignaali johtaa lineaariseen havaitsemiskokemukseen. Lisäksi gammakorjaus parantaa HVS:n ominaisuuksia hyödyntävän perustuvien pakkausmenetelmien tehokkuutta, mutta toisaalta aiheuttaa ongelmia väri-toiston kanssa erilaisten laitteiden välillä. Wandell [10] on käsitellyt HVS:n ominaisuuksia kattavasti. [1; 2]

2.4.5 Muut kuvanmuokkausprosessit

Gammakorjauksen jälkeiset kuvanmuokkauksen koostuvat kohinanpoistosta, terävöittämisestä, kuolleiden pikselien poistosta, objektiivin geometriavääristymien korjaamisesta, sekä kirkkauden ja kontrastin säädöstä.

Kameran kuvasensori ja siihen liittyvät järjestelmät tuottavat lopulliseen pakkaamattomaan kuvaan monenlaista kohinakomponentteja eri lähteistä [11]. Tällaisia kohinatyyppisiä ovat esimerkiksi signaaliriippuvainen raekohina, ei-satunnaiset kohinat ja satunnaisiin muuttujiin (elektronisten piirien tuottama kohina, kvantisointikohina, lämpökohina ja niin edelleen) rinnastettavat kohinalähteet. Kohinakomponentteja mallinnetaan yleensä nollakeskeisenä normaalijakaumaa noudattavana funktiona, jonka varianssi on

$$\sigma_n(x, y) = a_n + b_n f(x, y). \quad (2.1)$$

Muuttuja a_n mallintaa signaalista riippumattomia komponentteja ja muuttujalla b_n mallinnetaan yhden kanavan kuvasignaalista $f(x, y)$ riippuvia komponentteja. Ei-satunnaisia kohinatyyppisiä käsitellään tarkemmin kappaleessa 3.4.

Kuvanottohetkellä tapahtuu useita kuvaa sumentavia tekijöitä. Kuvaa sumentavat esimerkiksi kameran ja objektiivin liikkeet valotusaikana, objektiivin tarkennusvirheet, objektiivin linssien epätäydellisyys, kuvan sumennus ennen värisuodatinmatriisia, sekä väri-interpolointi. Tästä syystä kuvaa terävöitetään kohinanpoiston jälkeen. Sumentumista mallinnetaan pistejakaumafunktion avulla ja näin ollen voidaan yrittää poistaa jonkin dekonvoluutiomenetelmän avulla. Dekonvoluution jälkeen kuvaa voidaan vielä lisäksi terävöittää vahvistamalla korkeita taajuuksia ylipäästösuotimen avulla. [1]

Kuolleet pikselit eivät vastaa valoon ollenkaan tai niiden varaus ei vastaa niihin osuneiden fotonien määrää. Kuolleita pikseleitä voidaan mallintaa niin sanotun suola- ja pippurikohinan avulla. Havaitut kuolleet pikselit poistetaan kuvasta ja niiden sisältämää arvoa mallinnetaan ympäröivien pikseleiden avulla. Muokkausprosessin jälkeen jäljellä on valmis kuva, joka tyypillisesti tallennetaan käyttämällä jotakin pakkausmenetelmää tilan säästämiseksi. [1]

2.5 Kuvan tallennus ja pakkausmenetelmät

Prosessoinnin jälkeen kuva yleensä pakataan häviöllisellä tallennusmuodolla tilan säästämiseksi. Häviöllinen pakkausmuoto tarkoittaa, että pakattua kuvaa ei saada purettua täysin alkuperäiseksi, vaan informaatiota hävitetään pakkausprosessissa tarkoituksellisesti. Nykyään yleisin häviöllinen pakkausmuoto on JPEG (Joint Photographic Experts Group) [12], jonka yleisimpään profiiliin tutustutaan tarkemmin tässä kappaleessa.

Kuvallisen informaation lisäksi JPEG-tiedosto sisältää EXIF-standardin (Exchangeable Image File Format) [13] mukaisia tietoja muun muassa kamerasta, väriprofiilista, sekä kameran asetuksista kuvanottohetkellä. Kuva voidaan pakata myös häviöttömästi käyttäen TIFF/EP-tiedostomuotoa (Tagged Image File Format / Electronic Photography) [14], joka perustuu TIFF-pakkausstandardiin (Tagged Image File Format) [15]. Kuvan lisäksi TIFF/EP-tiedostoon talletetaan samanlaisia tietoja kuvaolosuhteista kuin EXIF-standardissa. Kahden edellisen lisäksi kuva pakata myös uudemmassa JPEG2000-tiedostomuodossa (Joint Photographic Experts Group, 2000) [16], joka ei ole kuitenkaan yleistynyt suurempien resurssivaatimusten takia.

JPEG-standardiin [12] kuuluu kolme eri osiota: Baseline, parempaan pakkaussuhteen tarjoava laajennettu pakkausmenetelmä ja häviötön pakkausmenetelmä. Kaikkien JPEG-yhteensopivien laitteiden pitää osata vähintään baseline-menetelmä, joka esitellään tässä kappaleessa. Purku tapahtuu käänteisessä järjestyksessä pakkaukseen nähden. Menetelmä toimii kolmessa vaiheessa, joita ovat diskreetti kosinimuunnos, kvantisointi ja entropiapakkaus:

1. Diskreetti kosinimuunnos (DCT, Discrete cosine transformation) [17]: Kuva muunnetaan YCbCr-väriavaruuteen ja jaetaan 8 kertaa 8 -pikselin kokoisiin alueisiin. Tämän jälkeen pikseliarvojen lukualetta siirretään etumerkittömistä etumerkillisiksi. Tyypillisesti pikselissä on 8-bittiä (lukualue [0, 255]), jolloin siirto etumerkilliseksi muuttaa

arvon lukualueelle $[-128, 127]$. Lukualueen siirron jälkeen suoritetaan diskreetti kosinimuunnos

$$F(\omega_x, \omega_y) = \frac{1}{4} c_0(\omega_x) c_0(\omega_y) \sum_{x=0}^7 \sum_{y=0}^7 f(x, y) \cos\left(\frac{(2x+1)\omega_x\pi}{16}\right) \cos\left(\frac{(2y+1)\omega_y\pi}{16}\right), \quad (2.2)$$

jossa $f(x, y)$ tarkoittaa alkuperäisen lohkon kuvasignaalin yhtä arvoa. DCT-lohkon pikselin indeksiä vastaavat taajuusparametrit $\omega_x, \omega_y = 0, \dots, 7$ ja funktio $c_0(\omega) = 1/\sqrt{2}$, jos $\omega = 0$, muuten $c_0(\omega) = 1$. Muunnoksen jälkeen lohkot voidaan periaatteessa muuttaa vielä takaisin alkuperäiseksi. Kosinimuunnettu arvo $\omega_x, \omega_y = 0$ on lohkon DC-komponentti ja muut arvot ovat AC-komponentteja [18].

2. Kvantisointi: Tässä vaiheessa DCT-arvot pyöristetään, niin että saavutetaan haluttu pakkaustaso. Baseline-profiilissa kvantisointi tapahtuu tasaisella kvantisointiaskeleen pituudella. Parempi pakkaussuhde saavutetaan, mikäli askeleet ovat tarkempia nollan läheisyydessä. Tämä ominaisuus ei kuitenkaan kuulu baseline-profiilin ominaisuuksiin. Baseline-profiilissa kvantisointi suoritetaan siis alkioittain kaikille kuvan DCT-arvoille:

$$F_q(\omega_x, \omega_y) = \left\lfloor \frac{F(\omega_x, \omega_y)}{s(\omega_x, \omega_y)} + \frac{1}{2} \right\rfloor \quad (2.3)$$

jossa funktio $s(\omega_x, \omega_y)$ on taajuusriippuvainen kvantisointiaskel, joka määritellään kaavalla

$$s(\omega_x, \omega_y) = \begin{cases} \left\lfloor \frac{200 - 2Q_c}{100} \mathbf{C}(\omega_x, \omega_y) + \frac{1}{2} \right\rfloor, & 50 \leq Q_c \leq 100 \\ \left\lfloor \frac{50}{Q_c} \mathbf{C}(\omega_x, \omega_y) + \frac{1}{2} \right\rfloor, & 0 \leq Q_c \leq 50 \end{cases} \quad (2.4)$$

jossa Q_c on JPEG-laatu parametri ja matriisi \mathbf{C} on kokeellisesti määritetty taajuusriippuvainen kvantisointimatriisi [19].

3. Entropiapakkaus: Viimeinen askel sisältää kvantisoitujen DCT-arvojen häviöttämän pakkauksen. Yleisimmin käytetään Huffman-koodausta [20], joskin standardi tukee myös aritmeettista pakkausta.

2.6 Syntyprosessin jälkeinen manipulointi

Farid [21] listaa raportissaan yleisimpiä tietokoneella suoritettavia kuvankäsittelymenetelmiä, joihin tutustutaan tässä kappaleessa. Esiteltyjä kuvankäsittelymenetelmiä ovat komposiittikuva, metamorfoosi, retusointi, kuvan parantaminen, sekä kuvan luominen.

Kuvamanipuloinnin ei kaikissa tapauksissa tarvitse olla kovinkaan taidokkaasti tehty ollakseen uskottava vastaanottajalle. Tästä esimerkkinä kuva 2.2, joka julkaistiin useiden amerikkalaisten sanomalehtien etusivuilla uutistoimisto Agence France-Pressen (AFP) [22] levittämänä. Kuva oli alun perin saatu Iranilaisen Sepah News-uutistoimiston verkkosivuilta [23]. Kuvassa esitetään neljää maasta nousevaa ohjusta Iranin ohjusteihin liittyen. Kuitenkin seuraavana päivänä uutistoimisto AFP levitti uutta, lähes identtistä kuvaa ohjuslaukaisusta, jossa oli vain kolme ohjusta (kuva 2.3). Uusi kuva osoittaa aikaisemmin julkaistun kuvan manipuloiduksi. [24]



Kuva 2.2. Uutistoimisto AFP:n julkaisema manipuloitu kuva. Kuvan julkaisivat muun muassa *The Los Angeles Times*, *The New York Times* ja *The Chicago Tribune* etusivuillaan [24].



Kuva 2.3. Kuvan 2.2 jälkeen julkaistu kuva ohjuslaukaisusta. Nyt havaitaan, että myös laukaisemattoman ohjuksen laukaisualusta on peitetty retusoimalla ja kopioimalla materiaalia samasta kuvasta [24].

Kuvasta 2.2 selviää tarkemmin katsomalla, että yksi ohjuksista koostuu kahdesta vierekkäisestä ohjuksesta. Toisesta ohjuksesta on kopioitu pölypilvi ja toisesta itse ohjus. Lisäksi ohjuksen savuvanaa on jatkettu keinotekoisesti. Kuvaa 2.3 tarkastelemalla selviää lisäksi, että kuvasta 2.2 on myös poistettu laukaisemattoman ohjuksen laukaisualusta.

2.6.1 Komposiittikuva

Kuvan muodostaminen useasta eri kuvasta on tavallisin kuvamanipulaation tyyppi. Tämänkaltaista kuvaa kutsutaan komposiittikuvaksi. Komposiittia käytetään usein kohteen lisäykseen tai poistamiseen kuvasta. Tämä tapahtuu kopioimalla alue yhdestä kuvasta ja liittämällä alue muokattavaan kuvaan. Manipuloinnin kohteena on usein jokin kuvassa oleva henkilö. Esimerkiksi kuvassa olevan henkilön pään tilalle sijoitetaan toisen henkilön pää. Kuvan uskottavuus ihmisilmälle riippuu pitkälle manipuloijan kyvyistä ja siitä, miten hyvin lähdemateriaali sopii yhteen. Mikäli jokin asia halutaan piilottaa, voidaan alue myös kopioida ja siirtää kopioitu alue samaan kuvaan piilotettavan alueen päälle. Tämänkaltaisen kopioi-siirrä-kuvanmanipulointi jättää vähemmän manipuloinnin jälkiä kuvaan, sillä valaisuolosuhteet, kokosuhteet ja tarkennusalueen ominaisuudet ovat yhdenmukaisia muun kuvan kanssa [25].

2.6.2 Metamorfoosi

Kuvan metamorfoosilla tarkoitetaan tietokoneella suoritettavaa kuvanmuokkausta, jossa yksi kuva muunnetaan vähitellen toiseksi. Näin ollen välimuotoisissa kuvissa on piirteitä molemmista kuvista.

2.6.3 Retusointi

Kuvan retusoinnilla tarkoitetaan operaatioita, joilla parannellaan tai muutetaan kuvan sisältöä ilman sisällön lisäystä toisesta kuvasta. Tavallisimpia retusointeja ovat esimerkiksi ryppyjen silottaminen henkilön kasvoilta tai muiden tällaisten piirteiden silottaminen käyttäen hyväksi piirteen välittömässä läheisyydessä olevaa tietoa.

2.6.4 Kuvan parantaminen

Koko kuvaan kohdistuvat operaatiot eivät välttämättä muuta kuvan sisältöä, mutta ne lasketaan silti manipuloinniksi. Kuvan paranteluksi voidaan laskea esimerkiksi kontrastin ja kirkkauden muokkaus esimerkiksi niin, että sää tai vuorokaudenaika näyttää erilaiselta kuin alun perin. Kuvassa olevaa yksityiskohtaa voidaan korostaa tai piilottaa, riippuen tarkoituksesta.

2.6.5 Kuvan luominen

Kuva voidaan myös luoda alusta alkaen itse. Tällöin kuitenkin tarvitaan erityisiä taitoja, eikä prosessi ole yhtä nopea kuin edellä mainituissa menetelmissä. Kuva voidaan luoda digitaalisesti joko kuvanmuokkausohjelmalla piirtämällä tai käyttäen 3D-mallia. Tällöin saadaan periaatteessa aina haluttu lopputulos, mutta esimerkiksi realistinen ihmisen 3D-mallintaminen on hyvin vaikea tehtävä. Kuitenkin on osoitettu [26], että mallinnustekniikan kehittymisen myötä ihmisten on yhä vaikeampi erottaa 3D-mallinnettuja kuvia aidoista valokuvista.

3 KUVAMANIPULOINNIN HAVAITSEMINEN

Tässä kappaleessa esitellään teoriapohjia eri kuvantamisprosessin vaiheita hyväksikäytettäville kuvamanipuloinnin havaitsemismenetelmille. Kuvantamisprosessi on esitelty kameran ja kuvamanipuloinnin osalta kappaleessa 2. Kuitenkin tässä kappaleessa esitellään myös menetelmä, joka käyttää hyväkseen itse kuvan sisältöä. Esitellyt menetelmät perustuvat valaistuksen yhdenmukaisuuteen, objektiivin tuottamiin väriääristymiin, värisuodinmatriisiin ja interpoloinnin tuottamiin jaksollisiin signaaleihin, kuvasensorin aiheuttamaan kohinaan, JPEG-pakkaukseen, sekä kopioidun alueen havaitsemiseen pikselitasolla.

Vaikka kappaleessa ei varmastikaan esitellä kaikkia mahdollisia tai nykyisellään luotuja menetelmiä, tämän kappaleen on tarkoitus antaa lukijalle käsitys miten kuvantamisprosessin vaiheita voidaan hyväksikäyttää kuvamanipuloinnin havaitsemisessa. Tässä kappaleessa esitellyt menetelmät on pyritty valitsemaan niin, että ne olisivat uskottavia ja toteuttamiskelpoisia vaihtoehtoja tietyntyyppisen kuvamanipuloinnin havaitsemiseen. Kuvamanipuloinnin havaitsemiseen on myös ehdotettu koko kameran komponentteja yhdistelevää menetelmää [27] ja koko kameran käsittävää kameran vastefunktiota [28].

3.1 Valaistuksen yhdenmukaisuus

Johnson & Farid [29] ovat ehdottaneet manipuloitujen alueiden havaitsemista perustuen valaistuksen yhdenmukaisuuden puuttumiseen. Komposiittikuvia luotaessa lähdemateriaalin valaistusolosuhteet ovat usein erilaisia ja periaatteessa havaittavissa. Menetelmä olettaa valojen sijaitsevan etäällä kohteesta, sekä heijastavan pinnan olevan pallomainen mattapinta, joka heijastaa valoa tasaisesti joka suuntaan. Valojen määrää ei kuitenkaan ole rajoitettu. Myös kameran vastefunktion oletetaan olevan lineaarinen. Menetelmä pyrkii näitä oletuksia hyväksikäyttäen mallintamaan vallitsevat valaistusolosuhteet. Kuvaa voidaan epäillä manipuloiduksi, mikäli mallinnettujen valonlähteiden suunnat poikkeavat toisistaan suuresti kuvan eri osissa.

3.2 Väriääristymät

Ideaalinen objektiivinen kerää kaikki valonsäteet haluttuun kohtaan kuvasensorilla. Käytännössä valon tiellä on hiukkasia, jotka vääristävät valonsäteiden kulkua. Myös kameran optiikka ja siihen kuuluvat linssit taittavat eri tavalla eri aallonpituuksia, joka aiheuttaa sensorille ja lopulliseen kuvaan väriääristymiä. Tässä kappaleessa tutustutaan Johnsonin & Faridin [30] ehdottamaan menetelmään sivusuuntaisen väriääristymän käytös-

tä kuvan aitouden määrittämiseksi. Menetelmässä käytettävän oletuksen mukaan, kuvamanipulointi aiheuttaa yleensä rikkeen värivääristymän yhdenmukaisuuteen ja voidaan näin ollen havaita.

Mikäli sivusuuntaista värivääristymää approksimoidaan ensimmäisen asteen kaavalla, voidaan käyttää Snellin lakia [31]:

$$n_a \sin(\theta_a) = n_b \sin(\theta_b), \quad (3.1)$$

jossa θ_a on valonsäteen tulokulma linssille, θ_b on säteen lähtökulma linssiltä, taittokertoimet n_a ja n_b määräytyvät rajapinnan materiaalien taittokertoimista ja valonsäteen taajuuden mukaan. Nyrkkisääntönä korkeampitaajuiset valonsäteet taittuvat enemmän kuin matalammat. Myös valonsäteiden lähtökulma suurenee suoraan verrannollisesti tulokulmaan nähden. Sivusuuntainen värivääristymä aiheuttaa siis eri taajuuksien valonsäteiden taittumisen eri kohtaan kuvasensorilla.

Tarkemmassa mallissa värivääristymä mallinnetaan punaisen ja vihreän, sekä vihreän ja sinisen kaistan välillä. Mallit johdetaan vain punaisen ja vihreän kanavan väliselle vääritysmälle, sillä parametrien estimointiin käytetyt mallit ovat samanlaisia kaikkien värikanavien välisille väärityksille. Mallissa on kolme parametria. Parametrit x_1 , y_1 ja α_1 mallintavat siirtymää punaisen ja vihreän kaistan välillä. Parametrit x_1 ja y_1 merkitsevät väärityksen keskipistettä (pikseliä) sensorilla, jossa vääritystä ei ole. Parametri α_1 on vääritysmävakio.

Oletetaan kuvassa olevan punainen ja vihreä värikanava: $f_r(x, y)$, ja $f_g(x, y)$. Punaisen kanavan vääritysmäkorjattua versiota kuvataan funktiolla $f_{ra}(x_r, y_r)$, jossa

$$x_r = \alpha_1(x - x_1) + x_1, \quad x \in \mathbb{N}, x_r, x_1 \in \mathbb{R}, \quad (3.2)$$

$$y_r = \alpha_1(y - y_1) + y_1, \quad y \in \mathbb{N}, y_r, y_1 \in \mathbb{R}, \quad (3.3)$$

jossa koordinaatti (x_r, y_r) on punaisen valonsäteiden vääritysmäkorjattu osumakohta kuvasensorilla. Parametrit x_1 , y_1 ja α_1 estimoidaan maksimoimalla värikanavien välinen informaatio kanavien $f_{ra}(x_r, y_r)$ ja $f_g(x, y)$ välillä. Etsitään siis

$$\operatorname{argmax}_{x_1, y_1, \alpha_1} I(f_{ra}, f_g), \quad (3.4)$$

jossa yhteistä informaatiota merkitään

$$I(f_{ra}, f_g) = \sum_{r \in f_r} \sum_{g \in f_g} P(r_a | g_a) \log \left(\frac{P(r_a | g_a)}{P(r_a)P(g_a)} \right), \quad (3.5)$$

jossa kokonaisluvut r_a ja g_a kuvaavat tiettyä arvoa punaiselle ja vihreälle pikselille. Kaavassa on ehdollinen todennäköisyys $P(r_a | g_a)$, sekä tilastolliset todennäköisyydet $P(r_a)$ ja $P(g_a)$.

Johnsonin & Faridin toteutuksessa [30] yhteisen informaation globaalin maksimin toteuttavat parametrit maksimikohta etsittiin brute-force-menetelmällä, joka tekee algoritmista laskennallisessa mielessä hyvin raskaan. Mikään ei kuitenkaan periaatteessa estä gradienttiin perustuvien algoritmien käyttöä. Mallin maksimiparametrit etsitään koko kuvan kattavalla globaalilla tasolla, sekä vain osan kuvasta käsittävällä paikallisella tasolla.

Laskettujen paikallisten $\underline{v}_1(x, y)$ vääristymien ja globaalin vääristymän $\underline{v}_0(x, y)$ välinen suuntaerovektori lasketaan kaavalla

$$\underline{\theta}(x, y) = \cos^{-1} \left(\frac{\underline{v}_0(x, y) \cdot \underline{v}_1(x, y)}{\|\underline{v}_0(x, y)\| \|\underline{v}_1(x, y)\|} \right). \quad (3.6)$$

Luokitukseen käytettävä suuntavirhe lasketaan luotettavuuden parantamiseksi käytämällä paikallisten suuntavirheiden keskiarvoja. Keskiarvoon lasketaan vain ne suuntavirhevektorit $\underline{\theta}(x, y)$, joiden normin suuruus on suurempi kuin 0,01 pikseliä:

$$\hat{\underline{\theta}} = \frac{1}{N_s} \sum_{x,y} \underline{\theta}(x, y), \quad (3.7)$$

jossa vakio N_s tarkoittaa keskiarvon laskennassa käytettyjen suuntavirhevektorien määrää. Väri-vääristymiin perustuvan menetelmän toteutus esitellään kappaleessa 4.1.

3.3 Värisuodinmatriisi ja interpolointi

Yksi ensimmäisistä kuvamanipuloinnin havaitsemismenetelmistä perustui väri-interpoloinnin jättämien taajuuskomponenttien havaitsemiseen. Kappaleessa 2.3 on mainittu erilaisia värisuodinmatriisien tyyppisiä. Kuvaan jää viitteitä sekä käytetystä värisuodinmatriisista, että käytetystä interpolointimenetelmästä. Värisuodinmatriisin käyttö jättää lopulliseen kuvaan jaksollisia signaaleja. Tämä johtuu siitä, että yleensä värisuodinmatriisi värisuotimet on sijoitettu tasaisin välein ja yhden tietyn värisen pikselin kohdalla muut värit joudutaan laskemaan viereisten pikselien perusteella. Kuva on todennäköisesti manipuloitu, mikäli jaksollisuus puuttuu joltain kuvan alueelta tai puuttuu kuvasta kokonaan. Kameran käyttämä väri-interpolointimenetelmä ja värisuodinmatriisin tyyppi ovat tuntemattomia. Tästä syystä Popescu & Farid [32] ovat käyttäneet EM-algoritmia (Expectation-maximization) [33] interpoloinnin uskottavuusfunktion maksimoimiseksi, sekä lineaarisesta mallista interpolointiprosessin approksimoimiseksi. Tässä kappaleessa tutustutaan edellä mainittuun menetelmään.

Kuvatkoon f_i väri-interpoloidun kuvan yhtä värikanavaa ja $f_i(x, y)$ yhden värikanavan yhtä pikselin arvoa. Oletuksena on, että kukin näyte $f_i(x, y)$ kuuluu toiseen kahdesta mallista, joista käytetään notaatiota M_1 ja M_2 . Malli M_1 kuvaa lineaarisesti muista näytteistä riippuvia näytteitä, eli niitä näytteitä joiden havaitaan kuuluvan väri-

interpoloituun kuvaan. Malliin M_2 kuuluvat kaikki ne näytteet, jotka eivät korreloi lähellä olevien näytteiden kanssa eli siis ovat mahdollisesti kuvamanipuloinnin tulosta. Malliin M_1 kuuluvat pikselit $f_i(x, y)$ toteuttavat siis ehdon

$$f(x, y) = \sum_{u,v=-N}^N \alpha_{u,v} f_i(x+u, y+v) + \varphi(x, y), \quad (3.8)$$

jossa alkio $\alpha_{u,v}$ kuuluvat matriisiin $\mathbf{A} = \{\alpha_{u,v} | -N \leq u, v \leq N\}$, $N \in \mathbb{Z}$, $\alpha_{0,0} = 0$ ja $\varphi(x, y)$ kuvaa normaalijakautunutta nollakeskistä virhettä tuntemattomalla keskihajonnalla. Algoritmin tarkoitus on löytää parametrit $\alpha_{u,v}$, jotka kuvaavat tuntematonta lineaarista interpolointiprosessia.

Algoritmina EM on kaksivaiheinen ja koostuu nimensä mukaisesti E- ja M-askeleista. Ensin laskettavalla E-askeleella lasketaan kunkin näytteen kuulumuus kuhunkin malliin. Jälkimmäisellä M-askeleella päivitetään mallia M_1 kuvaavan matriisin \mathbf{A} parametrit virhepinnan $E(\mathbf{A})$ minimoimiseksi. Tarkemmin ilmaistuna, E-askeleella lasketaan todennäköisyys pikselin $f_i(x, y)$ kuulumisesta luokkaan M_1 :

$$\begin{aligned} & P\{f_i(x, y) \in M_1 | f_i(x, y)\} \\ &= \frac{P\{f_i(x, y) | f_i(x, y) \in M_1\} P_r\{f_i(x, y) \in M_1\}}{\sum_{i=1}^2 P\{f_i(x, y) | f_i(x, y) \in M_i\} P_r\{f_i(x, y) \in M_i\}}, \end{aligned} \quad (3.9)$$

jossa prioritodennäköisyydet $P_r\{f_i(x, y) \in M_1\}$ ja $P_r\{f_i(x, y) \in M_2\}$ ovat molemmat oletettavasti 0,5. Ehdollinen todennäköisyys näytteen $f_i(x, y)$ kuulumisesta malliin M_1 merkitään

$$\begin{aligned} & P\{f_i(x, y) | f_i(x, y) \in M_1\} \\ &= \frac{1}{\sigma\sqrt{2\pi}} \exp \left[\frac{1}{2\sigma^2} \left(f_i(x, y) - \sum_{u,v=-N}^N \alpha_{u,v} f_i(x+u, y+v) \right)^2 \right], \end{aligned} \quad (3.10)$$

jossa normaalijakauman varianssi σ^2 on estimoitu edellisellä M-askeleella. Mallin M_2 oletetaan olevan tasaisesti jakautunut. Uudella M-askeleella päivitetään matriisin \mathbf{A} alkioita minimoimalla painoparametreilla w_e painotettu virhepinta

$$E(\mathbf{A}) = \sum_{x,y} w_e(x, y) \left(f_i(x, y) - \sum_{u,v=-N}^N \alpha_{u,v} f_i(x+u, y+v) \right)^2, \quad (3.11)$$

jossa painoparametri $w_e(x, y) = P_r\{f_i(x, y) | f_i(x, y) \in M_1\}$. Virhepinta minimoidaan asettamalla virhepinnan $E(\mathbf{A})$ gradientti alkion $\alpha_{s,t}$:n suhteen nolaksi. Ratkaistaan siis

$$\begin{aligned}
& \frac{\partial E}{\partial \alpha_{s,t}} = 0 \\
\Leftrightarrow & -2 \sum_{x,y} w_e(x,y) f_i(x+s,y+t) \left(f(x,y) - \sum_{u,v=-N}^N \alpha_{u,v} f_i(x+u,y+v) \right) = 0 \\
& \Leftrightarrow \sum_{x,y} w_e(x,y) f_i(x+s,y+t) \sum_{u,v=-N}^N \alpha_{u,v} f_i(x+u,y+v) \\
& = \sum_{x,y} w_e(x,y) f_i(x+s,y+t) f_i(x,y) \\
& \Leftrightarrow \sum_{u,v=-N}^N \alpha_{u,v} \left(\sum_{x,y} w_e(x,y) f_i(x+s,y+t) f_i(x+u,y+v) \right) \\
& = \sum_{x,y} w_e(x,y) f_i(x+s,y+t) f_i(x,y). \tag{3.12}
\end{aligned}$$

Sama prosessi toistetaan matriisin \mathbf{A} jokaiselle alkioille $\alpha_{s,t}$. Askelia E ja M suoritetaan, kunnes saavutetaan (tarpeeksi) vakaa estimaatti matriisin \mathbf{A} alkiolle. Tuloksena matriisi \mathbf{A} saa uskottavimmat parametrit interpoloitujen näytteiden estimoimiseksi. Menetelmän ulostulona saadaan todennäköisyyskartta malliin M_1 kuuluvista näytteistä. Oletuksen mukaisilla jaksollisilla signaaleilla on suurin teho pysty-, vaaka- ja vinoitaisakselilla, joten todennäköisyyskartan Fourier-muunnoksessa kuuluu olla taajuuspiikkejä. Mikäli taajuuspiikkejä ei löydy, kuvaa on todennäköisesti manipuloitu.

Menetelmä toimii testien mukaan [32] luotettavasti vain häviöttömästi pakatuille ja hyvin korkealaatuisille häviöllisesti pakatuille kuville. Menetelmää on myös helppo huijata simuloimalla värisuodatinmatriisia ja interpoloimalla manipuloitu kuva uudestaan lineaarisella menetelmällä, jolloin taajuuspiikit palautuvat menetelmän ulostuloon.

3.4 Sensorikohina

Lukáš et al. [34] ehdottivat kameran yksilöimiseen menetelmää, joka hyödyntää CCD-kvasensorin pysyviä kohinakomponentteja. Edellä mainittua periaatetta on kuitenkin sovellettu kuvamanipuloinnin tunnistamiseen [35; 36]. Tässä kappaleessa selvitetään menetelmän teoriaa ja taustoja niin kuin niitä on käsitelty artikkeleissa [34; 35]. Menetelmän jatkokehittämistä on myös käsitelty eri julkaisuissa [36; 37].

Suurin osa kappaleessa 2.4.5 mainitusta kuvasensorin kohinatyypeistä ovat satunnaismuuttujiin verrattavia suureita, mutta on olemassa kaksi ei-satunnaista kohinatyyppiä: Muuttumaton sensorikohina (FPN, fixed pattern noise) ja valovasteen epäyhdenmukaisuus (PRNU, Photoresponse non-uniformity). Muuttumaton sensorikohina johtuu sensorin vuotovirrasta, joka on aina vakio. Näin ollen se yleensä poistetaan tehokkaasti erottamalla kameran automaattisesti ottama ”pimeä kuva” sensorin myöhemmin tuotta-

mista kuvista. Kohinatyyppiltään PRNU-kohina on mielenkiintoinen, sillä se sisältyy sensorin jokaisella pikselille erillisenä kerroinkomponenttina. Näin ollen se ei poistu jos samalla kameralla otettuja kuvia summattaisiin alkioittain. Sensorin PRNU-kohina on suhteellisen pysyvä kohinakomponentti koko CCD-kennon eliniän ajan. Myös toisella yleisesti käytetyllä, CMOS-kennotyypillä esiintyy PRNU-kohinaa, sillä kaikki nykyiset kennot valmistetaan samanlaisella puolijohdetekniikalla [35, katso 38].

Kuitenkin PRNU-kohina on mahdollista poistaa tekniikalla, jota kutsutaan nimellä flatfielding. Flatfielding suoritetaan muuttumattoman sensorikohinan poiston jälkeen. Tekniikassa otetaan kuva tasaisesti valotetusta näkymästä, jonka sensorivaste tunnetaan. Tämän jälkeen sensorilta saatu prosessoimaton kuva muutetaan vastaamaan tunnettua näkymää määräämällä jokaiselle pikselille oma korjauskerroin. Kuitenkin tekniikkaa sovelletaan harvoin kuluttajaluokan kameroissa. Tämä johtuu siitä, että tasaisen valotuksen saaminen kennolle on vaikeaa, vaikka näkymä olisikin tasaisesti valaistu.

Koska PRNU-kohina sisältyy jokaiseen värikanavan pikseliarvoon kertoimena, sitä ei havaita mustissa tai täysin valkoisessa, saturoituneessa pikseleissä. Kohinalla on kaksi komponenttia: pikselikohinan epäyhdenmukaisuus (PNU, Pixel noise non-uniformity) ja optisilla komponenteilla olevien epäpuhtauksien aiheuttama kohina. Pikselikohinan epäyhdenmukaisuus aiheuttaa useimmiten korkeille taajuuksille painottuneita kohinakomponentteja. Sensorilla ja optiikassa olevien hiukkaset aiheuttavat yleensä hyvin matalataajuisia kohinakomponentteja ja voivat ajan mittaan muuttua. Tästä syystä halutaan aina mitata PNU-kohinaa. Pikselikohinan epäyhdenmukaisuus johtuu sensorin valmistusprosessista, joten se ei ole kovinkaan herkkä ympäristön olosuhteille, kuten lämpötilalle. Parhaiten PNU-kohinan mittaaminen onnistuu kuvan osissa, jotka sisältävät hyvin matalataajuisia komponentteja kohinan lisäksi. Käytännössä mittausta varten halutaan löytää kuva, jossa väripinta on kaikille kanaville mahdollisimman yhdenmukainen, lähellä saturoitumisrajaa, mutta ei-saturoitunut ja sisältää mahdollisimman vähän värien vaihtelua kuten teräviä reunoja. Hyvä kuva PNU-kohinan mittausta varten voisi olla esimerkiksi tarkentamaton kuva harmaasta taivaasta.

Lopullista kuvaa $f(x, y)$ mallinnetaan sensorimallin [39] avulla:

$$f(x, y) = g_f^\gamma \left[(1 + k(x, y)) f_y(x, y) + \Lambda(x, y) \right]^\gamma + \Theta_q(x, y), \quad (3.13)$$

jossa $f(x, y)$ kuvaa värikanavan pikselin arvoa. Parametri g_f on värikanavan vahvistusparametri. Parametri γ on gammakorjauskerroin. Funktiolla k mallinnetaan pikselien valoherkkyysero. Funktiolla f_y mallinnetaan täydellistä näkymäkomponenttia ilman virheitä. Funktio Λ mallintaa muita sensorin kohinaelementtejä ja funktio Θ_q on mallintaa kvantisointikohinaa. Taylorin laajennoksen jälkeen saadaan

$$f(x, y) = \left(g_f f_y(x, y) \right)^\gamma \left[1 + k(x, y) + \frac{\Lambda(x, y)}{f_y(x, y)} \right]^\gamma + \Theta_q(x, y)$$

$$= (g_f f_y(x, y))^{\gamma} \left[1 + \gamma k(x, y) + \frac{\gamma \Lambda(x, y)}{f_y(x, y)} \right] + \Theta_q(x, y). \quad (3.14)$$

Kaavaa yksinkertaistetaan sisällyttämällä gammakorjauskerroin γ , valoherkkyyskerroin-funktioon k . Merkitään myös

$$f_0(x, y) = (g_f f_0(x, y))^{\gamma} \quad (3.15)$$

ja yhdistetään satunnaista kohinaa kuvaavat termit

$$\Theta(x, y) = \gamma f_0(x, y) \frac{\Lambda(x, y)}{f_y(x, y)} + \Theta_q(x, y). \quad (3.16)$$

Kaavat (3.14), (3.15), ja (3.16) yhdistämällä saadaan lopullinen yksinkertaistettu malli

$$f(x, y) = f_0(x, y) + f_0(x, y)k(x, y) + \Theta(x, y), \quad (3.17)$$

jossa manipuloidut alueet voidaan teorian mukaan tunnistaa manipuloiduiksi, mikäli tutkitun alueen PNU-kohina ei vastaa referenssialueen PNU-kohinaa. Toisin sanoen kameralle etsitään estimaatti diskreetistä valoherkkyysfunktioista k käyttämällä kameral-la otettuja opetusnäytekuvia ja verrataan luokiteltavaa kuvaa kameras valoherkkyys-funktioon. Tarvittavien opetuskuvien määrää voidaan vähentää poistamalla kuvista näkymäkomponentti f_0 ennen valoherkkyysfunktion approksimointia. Käytännössä näky-mää voidaan approksimoida suodattamalla kohinaa kuvasta f suotimella F_n . Kohinasuodatettua kuvaa merkitään

$$F_n(f) = \hat{f}_0. \quad (3.18)$$

Nyt ei olla kuitenkaan kiinnostuneita kuvan sisältämästä näkymästä, vaan PRNU- komponentista $f_0(x, y)k(x, y)$. Näin ollen näkymä poistetaan erottamalla kohinasuodatettu kuva alkuperäisestä kuvasta. Tätä prosessia kutsutaan sisältösuodatukseksi. Sisäl-tösuodatuksen jälkeistä kuvaa merkitään

$$\begin{aligned} w(x, y) &= f(x, y) - \hat{f}_0(x, y) \\ &= f(x, y)k(x, y) + f_0(x, y) - \hat{f}_0(x, y) + (f_0(x, y) - f(x, y))k(x, y) \\ &\quad + \Theta(x, y) = f(x, y)k(x, y) + \Xi(x, y), \end{aligned} \quad (3.19)$$

jossa kohinakomponentti Ξ kuvaa suodatuksen F_n kuvaan lisäämiä termejä.

Suodatuksen jälkeen halutun PRNU-komponentin ja muun datan signaali-kohinasuhde paranee, vaikka kohinakuva w sisältää uusia kohinakomponentteja. Par-haaksi sisällönpoistosuotimeksi F_n on todettu [40] eräs pienimmän keskineliövirheen

(MMSE, minimum mean square error) suodin [41]. Koska sisällönpoistosuodin F_n on aina ylipäästösuodin, hiukkasten aiheuttama PRNU-komponentti vaimenee automaattisesti. Näin ollen sisältösuodatetussa kuvassa on aina suhteessa enemmän PNU-komponentin aiheuttamaa signaalia kuin alkuperäisessä suodattamattomassa kuvassa. Koska sisällön täydellinen poistaminen kuvasta on hyvin vaikeaa, kameran c opetusnäyttekuvien f_k määräksi N_c suositellaan [34] enemmän kuin 50 kuvaa.

Seuraavaksi voidaan estimoida valoherkkyyskerroimet \hat{k}_c käyttäen hyväksi opetuskuvia $f_k, = 1, 2, \dots, N_c$. Valoherkkyyskerroinestimaatti \hat{k}_c saadaan laskemalla alkioden keskiarvo opetusnäytteiden kohinakuvista w_k tai vaihtoehtoisesti voidaan laskea suurimman uskottavuuden (ML, Maximum likelihood) estimaatti:

$$\hat{k}(x, y) = \frac{\sum_{k=1}^{N_c} w_k(x, y) f_k(x, y)}{\sum_{k=1}^{N_c} (f_k(x, y))^2}. \quad (3.20)$$

Saatu valoherkkyyskerroinestimaatti käsitellään tiettyjen artefaktien poistamiseksi. Tällaisia artefaktit johtuvat esimerkiksi väri-interpoloinnista, sekä muista kamerassa tehtävistä kuvasensorin pysty- ja vaakavektoreille kohdistuvista operaatiosta. Nämä virheet poistetaan vähentämällä jokaiselta estimaatin rivivektorilta tämän oma keskiarvo. Sama toistetaan sarakevektoreille. Mikäli estimaattiin on päässyt näkyvää sisältöä alkuperäisistä kuvista, suodatetaan estimaatti vielä kertaalleen sisällönpoistosuotimella.

Luokiteltavassa kuvassa olevan alueen \mathcal{R} luokitteluksi, suodatetaan kuvasta ensin kohinakuva w_i . Tämän jälkeen lasketaan korrelaatio alueen \mathcal{R} vektorimuotoisten kohinakuvan ja valoherkkyysestimaatin välillä. Merkitään siis

$$\rho(\underline{w}_i(\mathcal{R}), \underline{\hat{k}}_c(\mathcal{R})) = \frac{(\underline{w}_i(\mathcal{R}) - \bar{\underline{w}}_i(\mathcal{R})) \cdot (\underline{\hat{k}}_c(\mathcal{R}) - \bar{\underline{\hat{k}}}_c(\mathcal{R}))}{\|\underline{w}_i(\mathcal{R}) - \bar{\underline{w}}_i(\mathcal{R})\| \|\underline{\hat{k}}_c(\mathcal{R}) - \bar{\underline{\hat{k}}}_c(\mathcal{R})\|}, \quad (3.21)$$

jossa yläviivalla merkityt vektorit merkitsevät vastaavan vektorin keskiarvoa. Kohinasuodatin F_n muuttuu tehottomaksi alueella, jossa on paljon reunoja, joten tämänkaltaiset alueet tulisi tunnistaa ja jättää luokittelematta. Myös dynaamisen alueen ääripäissä oleville alueille tulisi jättää huomiotta, sillä näille alueille korrelaatio ρ on hyvin pieni.

Alueen \mathcal{R} luokitusta varten lasketaan korrelaatiot $\rho(\underline{w}_i(Q_j), \underline{\hat{k}}_c(\mathcal{R}))$, $j = 1, 2, \dots, N_r$. Alueet Q_j ovat samankokoisia ja muotoisia alueen \mathcal{R} kanssa mutta tulevat kuvan eri alueilta tai kokonaan eri kameran ottamista kuvista. Tällaisia korrelaatioita kutsutaan myöhemmin sijainniltaan vastaamattomaksi ja niiden jakaumaa mallinnetaan normaalijakauman mukaan. Normaalijakauman kertymäfunktiota sijainniltaan vastaamattomalle luokalle merkitään $G(x)$. Lasketaan siis todennäköisyys tapahtumalle ”alue \mathcal{R} kuuluu sijainniltaan vastaamattomaan luokkaan”:

$$P_i = 1 - G\left(\rho\left(\underline{w}_i(\mathcal{R}), \hat{\underline{k}}_c(\mathcal{R})\right)\right). \quad (3.22)$$

Alue \mathcal{R} luokitellaan manipuloiduksi mikäli todennäköisyys $P_i > 10^{-3}$. Mikäli alue \mathcal{R} luokiteltiin manipuloiduksi, tulos pitää vielä tarkistaa jollakin luokitusmenetelmällä. Kappaleessa 4.2 toteutetaan kuvamanipuloinnihavaitsemismenetelmä tässä kappaleessa esiteltyyn teoriaan perustuen.

3.5 JPEG-uudelleenpakkaus

Tässä työssä JPEG-pakkausta käytetään hyväksi kahteen kertaan pakatun kuvan tunnistamisessa. Kahteen kertaan pakatun kuvan havaitseminen ei vielä kerro mitään kuvan sisällön muokkauksesta. Kuva saattaa olla pakattu toiseen kertaan esimerkiksi tilan säästämiseksi. Epäilyksiä kuvan aitoudesta on kuitenkin syytä herättää, mikäli alkuperäiseksi väitetystä kuvasta on merkkejä JPEG-uudelleenpakkauksesta.

Popescu [19] on todennut, että JPEG-uudelleenpakkaus voidaan havaita tarkastelemalla kuvan DCT-histogrammia. Kappaleessa 2.5 esitetyn JPEG-pakkausprosessin mukaan diskreetin kosinimuunnoksen tuloksena syntyneet arvot kvantisoidaan riippuen halutusta pakkauslaadusta. Alkuperäisten DCT-arvojen kvantisoidut arvot voidaan ilmaista yksinkertaistetusti kaavalla

$$q_a(u) = \left\lfloor \frac{u}{a} \right\rfloor, \quad (3.23)$$

jossa parametri a on kvantisointiaskel ja pakattava DCT-arvo u on kappaleessa 2.5 mainitulla DCT-muunnoksen arvoalueella. Purkamisvaiheessa näytteet puretaan kaavalla

$$q_a^{-1} = au. \quad (3.24)$$

Kannattaa huomata, että funktiolla $q_a(u)$ ei ole olemassa käänteisfunktiota, joka muuntaisi kvantisoidut arvot alkuperäisiksi. Tästä syystä uudelleenpakkauksen vaikutusta havainnollistetaan katsomalla, mitä arvoille tapahtuu seuraavalla kvantisointikerralla:

$$q_{ab}(u) = \left\lfloor \left\lfloor \frac{u}{b} \right\rfloor \frac{b}{a} \right\rfloor, \quad (3.25)$$

jossa muuttujat a ja b ovat kvantisointiaskelia.

Ensimmäisessä vaiheessa pakatut näytteet noudattavat yleensä normaalijakaumaa. Mikäli kvantisointiaskeleet a ja b ovat erisuuruisia, toiseen kertaan pakattaessa normaalijakauma ei enää toteudu ja jakaumaan syntyy erilaisia artefakteja. Käytännössä tämä ehto toteutuu, mikäli kappaleessa 2.5 mainittu taajuusriippuvainen kvantisointiaskel on

jälkimmäisellä pakkauskeralla erisuuruinen tai käytetään toista JPEG-standardia. Tämä ehto toteutuu hyvin usein, sillä kuvankäsittelyohjelmat ja kamerat käyttävät erilaisia kvantisointimatriiseja. Myöskään kameran käyttämä JPEG-laatuparametri ei yleensä ole tunnettu. Artefaktit voivat olla esimerkiksi joidenkin DCT-arvojen puuttumista tai kvantisoitujen arvojen kerääntymistä esimerkiksi joka toiseen arvoon. Tällaisia helposti havaittavia artefakteja esiintyy vain harvoin kertaalleen pakatussa luonnollisessa signaalissa. Kappaleessa 4.3 on esitelty tämän kappaleen teoriaa hyödyntävä DCT-histogrammimenetelmän toteutus.

3.6 Kopioi-siirrä-manipuloinnin tunnistaminen

Kopioi-siirrä-kuvamanipuloinnilla tarkoitetaan kuvan sisällön kopioimista samaan kuvaan.. Tämäntyyppinen kuvamanipulointi on yksi yleisimmistä kuvamanipuloinnin tyypeistä [25] ja tästä syystä se käsitellään omana osa-alueenaan, vaikka työ keskittyy digitaaliseen kuvantamisprosessiin ja sen hyväksikäyttämiseen kuvamanipuloinnin tunnistamisessa.

Periaatteessa kopioi-siirrä-kuvamanipuloinnin havaitseminen on yksinkertaista: Jokaista kuvan osa-aluetta verrataan muuhun kuvaan ja saadun korrelaation perusteella päätetään onko alue kopioitu samasta kuvasta. Käytännössä tämä on kuitenkin laskentatehokkaasti äärimmäisen raskas operaatio (kompleksisuus $O(N^2)$). Lisäksi kopioitua aluetta on saatettu skaalata, kääntää tai muuten jälkikäsitellä, joten on kehitetty älykkäämpiä menetelmiä tämänkaltaisen kuvamanipuloinnin tunnistamiseen. Tässä kappaleessa esitellään kaksi eri menetelmää kopioi-siirrä-kuvamanipuloinnin tunnistamiseen. Toinen menetelmä perustuu pääkomponenttianalyysiin (PCA, Principle component analysis) ja toinen SIFT-algoritmiin (Scale-invariant feature transform) [42].

3.6.1 Pääkomponenttianalyysi

Popescu & Farid [43] ovat ehdottaneet pääkomponenttianalyysin käyttöä kompleksisuusongelman ratkaisemiseksi, sekä pakkausvirheiden kohinasta johtuvien virheluokituksen vähentämiseksi. Kuten useassa muussakin menetelmässä, kuva jaetaan lohkoihin ja niille lasketaan pääkomponentit. Piirteenirrotuksen jälkeen vektorimuotoiset piirrektorit asetetaan tietorakenteeseen siten, että jokaiselle lohkolle tunnetaan toinen samankaltaisin lohko. Suurin laskentateho kuluu nyt pääkomponenttien suuruusjärjestyksen määrittämiseen, joten algoritmin kompleksisuus on nyt $O(N \log N)$.

Mustavalkoisen kuvan analyysi toteutetaan seuraavalla tavalla. Kuvassa on N pikseliä, joka jaetaan neliönmuotoisiin lohkoihin. Yhdessä lohossa on nyt p pikseliä. Yksittäisen lohkon tulee olla huomattavasti pienempi kuin kopioidun alueen koko. Kaksiulotteinen lohko muutetaan vektorimuotoon \underline{x}_i , joiden määrä on

$$N_p = (\sqrt{N} - \sqrt{p} + 1)^2. \quad (3.26)$$

Nyt vektorimuotoiset lohkot $\underline{x}_i, i = 1, \dots, N_p$ oletetaan odotuskeskiarvoltaan nolllaksi. Mikäli näin ei ole, tulee jokaisesta vektorista \underline{x}_i vähentää keskiarvovektori $\underline{\mu}_x = \frac{1}{N_p} \sum_{i=1}^{N_p} \underline{x}_i$. Nyt lohkoille voidaan laskea kovarianssimatriisi

$$\mathbf{C}_x = \sum_{i=1}^{N_p} \underline{x}_i \underline{x}_i^T. \quad (3.27)$$

Matriisin \mathbf{C}_x ominaisvektoreita merkitään \underline{e}_j ja vektorin ominaisarvot λ_j , täyttävät ehdon

$$\mathbf{C}_x \underline{e}_j = \lambda_j \underline{e}_j, \quad (3.28)$$

jossa $j = 1, \dots, p$ ja $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_p$. Vektorimuotoiset lohkot \underline{x}_i voidaan nyt muodostaa lineaarisesti käyttäen hyväksi ominaisvektoreita \underline{e}_j :

$$\underline{x}_i = \sum_{j=1}^b a_j \underline{e}_j, \quad (3.29)$$

jossa alkio $a_j = \underline{x}_i^T \underline{e}_j$ ja alkioista a_j muodostettu vektori $\underline{a}_i = [a_1, \dots, a_p]^T$ muodostavat pääkomponentin jokaiselle vektorille \underline{x}_i . Nyt laskennan monimutkaisuutta voidaan säätää yksinkertaisesti katkaisemalla vektori \underline{a}_i siten, että jäljelle jää N_i ensimmäistä alkioita.

Mikäli vektoreiden \underline{x}_i jakauma on moniulotteisen normaalijakauman mukainen, jäljelle jää pienimmän neliösumman N_i -ulotteinen approksimaatio alkuperäisestä vektorista \underline{x}_i . Katkaistua vektoria \underline{x}_i kutsutaan nyt pääkomponenttivektoriksi. Näin ollen samanaisten lohkojen havaitseminen helpottuu sillä pääkomponenttivektorin katkaiseminen poistaa kohinaa ja pieniä intensiteettieroja lohkoista. Katkaisun lisäksi pääkomponenttien alkioit kvantisoidaan kaavalla

$$\underline{a}_{iq} = \lfloor \underline{a}_i / Q \rfloor, \quad (3.30)$$

jossa Q :lla merkitään kvantisointiaskeleen suuruutta. Nyt pääkomponenttivektoreista muodostetaan matriisi \mathbf{S} , siten että sarakejärjestyksessä suurimmat arvo omaava pääkomponentti on matriisissa ensimmäisenä. Merkitään matriisin \mathbf{S} saraketta i vektorilla \underline{s}_i ja kyseisen vektorin kuvaaman kuvlohkon vasenta yläkulmaa koordinaatilla (x_i, y_i) . Algoritmi vertailee kaikkia sellaisia vektoreita \underline{s}_i , joiden sarake-etäisyys $|i - j|$ on vähemmän kuin ennalta määrätty enimmäisetäisyys. Seuraavaksi lasketaan lohkojen koordinaattien siirtymä x - ja y akselilla. Lohkojen kuvaama alue merkitään kopioiduksi,

mikäli monella lähekkäisellä lohkollla on identtinen siirtymä. Väärin kopioiduksi luokiteltujen alueiden vähentämiseksi asetetaan jokin raja-arvo lohkon minimisiirtymälle.

Menetelmää voidaan soveltaa värikuville ainakin kahdella eri tapaa. Ensimmäisessä tavassa menetelmää sovelletaan jokaiselle värikanavalle erikseen. Toisessa lähestymistavassa jokainen värikanava otetaan mukaan vektorimuotoiseen lohkoon, jolloin vektorista tulee kolme kertaa pidempi.

3.6.2 SIFT-algoritmi

Huang et al. [25] ehdottivat SIFT-piirrevektoreihin pohjautuvaa kopioi-siirräkuvamanipulaation tunnistamiseen. SIFT-piirrevektoria kutsutaan avainpisteeksi. Avainpisteet ovat invariantteja kuvan skaalaukselle, kiertämiselle, sekä osittain myös kirkkaustason ja kamerakuvakulman muutokselle. Piirteenirrotus toimii neljässä vaiheessa [44]. Vaiheet ovat skaalausvaruuden ääriarvojen määrittäminen, avainpisteiden sijainnin- ja suunnan määrittäminen, sekä avainpistealueen piirrevektorin määrittäminen.

Ensimmäisenä etsitään kuvan ääriarvot koko kuvasta eri kuvan skaaloilla. Kuvaa skaalataan käyttäen kaksiulotteista Gauss-funktiota. Skaalauskuva määritellään

$$l(x, y, \sigma_g) = g(x, y, \sigma_g) * f(x, y), \quad (3.31)$$

jossa operaattori $*$ merkitsee konvoluutiota ja $f(x, y)$ tarkoittaa alkuperäisen kuvan yhtä alkioita. Parametri σ_g on Gauss-funktion keskihajonta. Kaksiulotteinen Gauss-funktio määritellään seuraavasti:

$$g(x, y, \sigma_g) = \frac{1}{2\pi\sigma_g^2} e^{-\frac{x^2+y^2}{2\sigma_g^2}}. \quad (3.32)$$

Skaalattuja kuvia käytetään avainpisteiden määrittämiseen. Avainpisteet etsitään kuvasta, jolle on suoritettu konvoluutio kahden Gauss-funktion erotuksen kanssa. Tällaista kuvaa kutsutaan Gauss-erotuskuvaksi. Vähennettävä Gauss-funktion keskihajonta skaalataan kertoimella k_g :

$$\begin{aligned} d(x, y, \sigma_g) &= [g(x, y, k_g\sigma_g) - g(x, y, \sigma_g)] * f(x, y) \\ &= l(x, y, k_g\sigma_g) - l(x, y, \sigma_g), \end{aligned} \quad (3.33)$$

jossa skaalauskerroin $k_g \neq 1$. Gauss-erotuskuvat d ryhmitellään oktaaveittain siten että toiselle oktaaville siirtymistä kuvaa skaalausparametrin σ_g kaksinkertaistaminen. Kerroimen k_g arvot valitaan niin, että jokaiselle oktaaville saadaan sama määrä näytekuvia. Ennen seuraavalle oktaaville siirtymistä, oktaavissa olevat kuvat desimoidaan poistamalla joka toinen näyte riveittäin ja sarakkeittain.

Tämän jälkeen Gauss-erotuskuvista etsitään paikallisia minimi- ja maksimiarvoja vertaamalla jokaista pikseliä kahdeksaan ympärillä olevaan pikseliin. Tämän jälkeen pikseliä verrataan alemmassa ja ylemmässä skaalassa olevaan Gauss-erotuskuvien yhdeksään vastaavaan pikseliin. Keskellä oleva pikseli valitaan kandidaatiksi avainpisteelle, mikäli se osoittautuu suuremmaksi tai pienemmäksi kuin muut verrattavat pikselit. Avainpisteiden valintaa ja sijainnin määrittäystä on käsitelty tarkemmin Lowen artikkelissa [44].

Seuraavaksi jokaiselle jäljellä olevalle avainpisteelle määritetään suuntahistogrammi. Suuntahistogrammi on jaettu 10 asteen välein, joten yhdessä histogrammissa on 36 lokeroa. Suuntahistogrammiin talletetaan avainpisteen lähellä olevien avainpisteiden suunta painotettuna gradientin suuruudella, sekä Gauss-ikkunafunktiolla, jonka keskihajonta on 1,5 kertaa kyseisen avainpisteen skaalauskerroimesta σ_g . Suuntahistogrammin maksimiarvo vastaa nyt avainpisteen suuntaa ja se talletetaan avainpisteen gradientin suunnaksi. Mikäli suuntahistogrammista löytyy paikallisia huippuja, jotka ovat vähintään 80 % maksimihuipun suuruudesta, luodaan uusi avainpiste ja talletetaan tämän pisteen suunnaksi löydetty paikallishuippu. [44]

Seuraavaksi luodaan avainpisteen kuvaaja. Nykyisellään avainpiste on invariantti sijainnille, suunnalle, sekä skaalaukselle. Halutaan luoda sellainen kuvaaja, joka on invariantti myös kirkkauden vaihteluille. Avainpisteen kuvaaja luodaan sivultaan kuudentoista pikselin neliönmuotoisen alueen sisältämistä avainpisteistä. Kuvaajan sisältämä alue jaetaan sivultaan neljän pikselin kokoisiin alueisiin ja jokaisen jaetun alueen avainpisteiden suuntavektoreista luodaan kuvaajan piirrevektorina toimiva suuntahistogrammi. Jokainen avainpisteen kuvaaja sisältää nyt 16 suuntahistogrammia, joissa kussakin on 8 suuntalokeroa. Näin ollen SIFT-piirrevektori sisältää yhteensä 128 muuttujaa. [44]

Piirrevektorien luonnin jälkeen kaikkia luotuja piirrevektoreita verrataan toisiinsa. Vastaavaksi piirrevektoriksi saadaan se piirrevektori, jonka euklidinen etäisyys on kaikkein pienin. Tarkkuuden parantamiseksi löydetty piirrevektori luokitellaan kopioksi vain jos etäisyyden suhde toiseksi lähimpään piirrevektoriin on vähemmän kuin käyttäjän määrittämä vertailuarvo. [25]

Koska piirrevektorit ovat hyvin korkealotteisista ja raskaita laskennalliselta kannalta, on järkevää käyttää kd-puualgoritmista jatkokehitettyä ”Paras lokero ensin” -menetelmää (BBF, Best bin first) [45] lähimmän piirrevektorin etsintään. Menetelmän on osoitettu [25] löytävän kopioitu alue JPEG laatukertoimella 40, kuten myös kohinanlisäyksen ja kopioidun alueen käännöksen jälkeen.

4 TOTEUTUS

Tässä kappaleessa toteutetaan kolme kappaleessa 3 esitellystä menetelmästä. Toteutetut menetelmät ovat valittu teoriaosuudesta siten, että ne hyödyntävät kuvantamisprosessia ja ovat järkevästi toteutettavissa tämän työn puitteissa. Toteutus koostuu yksinkertaisesta käyttöliittymästä, sekä yhteisen rajapinnan omaavista menetelmien toteutuksista. Menetelmien toteutus kuvataan niiltä osin, kuin se sopii tämän työn sisältöön. Toisin sanoen työssä ei esitellä rajapintoja tai tarkkoja ohjelmistoteknisiä ratkaisuja.

Toteutuksessa käytetään laajasti Qt- [46] ja OpenCV-kirjastoja [47]. Kirjastoista Qt:a käytettiin pääosin käyttöliittymässä, mutta joitakin komponentteja kuten rinnakaistusta, sekä tietorakenteita käytettiin myös menetelmien toteutuksissa. OpenCV-kirjastoa käytettiin menetelmätasolla kaikissa kuvankäsittelyyn liittyvissä operaatioissa. Laskentaan käytetyssä laitteistossa oli kaksisydinprossessori ja neljä gigatavua keskusmuistia. Toteutuksen lähdekoodi on saatavissa verkko-osoitteesta [48].

4.1 Värivääristymiin perustuvan menetelmän toteutus

Tässä kappaleessa esitellään kappaleessa 3.2 esitellyn värivääristymiin perustuvan menetelmän toteutus. Tämä toteutus noudattaa mahdollisimman tarkasti Johnsonin & Faridin [30] esittämää toteutusta mainittuja raja-arvoja myöten. Tosin menetelmän käyttämät raja-arvot ja muut parametrit ovat vapaasti käyttäjän muokattavissa. Menetelmä ei tarvitse opetusnäytteitä toimiakseen. Mikäli opetusnäytteitä haluttaisiin käyttää, polttovälin ja aukon koon tulee olla samat kaikissa opetusnäytteissä, sekä luokiteltavassa kuvassa. Tämän lisäksi opetuskuvien ja luokiteltavan kuvan tulee sisältää reunoja yms. jotta menetelmä voi laskea mallin värivääristymistä.

4.1.1 Gradienttien estimointi

Luokiteltava kuva jaetaan 300:n pikselin kokoisiin neliönmallisiin lohkoihin. Suuren koon takia lohkot ovat päällekkäisiä. Lohkojen luokituskelpoisuuden määrittämiseksi jokaiselle lohkolle lasketaan gradienttikuva. Gradienttikuva lasketaan lohkon kuvaa vastaavasta harmaasävykuvasta, jossa eri harmaasävytasoja on yhteensä 256. Paikallisten estimaattien laskennassa huomiotta jätetään lohkot, joiden keskiarvgradientti on vähemmän kuin 2,5 harmaasävytasoa pikseliä kohden. Keskiarvgradientti lasketaan seuraavasti:

$$\nabla f(x, y) = \sqrt{\nabla f_x^2(x, y) + \nabla f_y^2(x, y)}, \quad (4.1)$$

jossa $\nabla f_x^2(x, y)$ ja $\nabla f_y^2(x, y)$ ovat gradienttien neliöitä x- ja y-akselin suuntaisesti. Gradientit saadaan suodattamalla alkuperäinen kuva yksiulotteisilla derivaattasuotimella d ja alipäästösuotimella p . Suotimet ja on kuvattu tarkemmin Faridin & Simoncellin artikkelissa [49]. Suodatus tapahtuu seuraavasti:

$$\nabla f_x(x, y) = (f(x, y) * d_x(x)) * p_y(y), \quad (4.2)$$

$$\nabla f_y(x, y) = (f(x, y) * d_y(y)) * p_x(x), \quad (4.3)$$

jossa operaattori $*$ merkitsee konvoluutioita. Suotimet p_x ja d_x ovat x-akselin suuntaisia ja vastaavasti suotimet p_y ja d_y ovat y-akselin suuntaisia. Tässä toteutuksessa käytettyjen suotimien aste on kolme kaikille suotimille.

4.1.2 Vääristymämallien estimointi

Johnsonin & Faridin [30] raportissa on mainittu kaksi eri tapaa globaalien vääristymämallien estimoimiseksi. Ensimmäisessä tavassa globaali estimaatti lasketaan hyväksikäyttämällä 50 suurimman gradientin omaavaa lohkoa. Globaali estimaatti lasketaan nyt lokaalin estimaatin käyttämällä yhden lohkon sijaan maksimoimalla kanavien välinen informaatio kaikkien 50 lohkon sisällä samanaikaisesti. Toinen lähestymistapa on laskea ensin paikalliset estimaatit jokaiselle lohkolle. Tämän jälkeen globaali estimaatti lasketaan minimoimalla lokaalien estimaattien ja globaalien estimaatin välinen virheen suuruus. Toteutuksessa käytetään tätä lähestymistapaa, sillä se on huomattavasti ensin mainittua menetelmää nopeampi.

Toteutuksessa jokaiselle lohkolle lasketaan paikallinen estimaatti joko punainen-vihreä- tai sininen-vihreä-värikanavien väri-vääristymistä. Paikallisten ja globaalien estimaattien laskentaan käytetään samalla periaatteella toimivaa algoritmia. Algoritmit ovat iteratiivisia ja toimivat brute-force-periaatteella, joka käyttää hyväkseen tiettyä määrää testipisteitä hakualueen sisällä. Jokaisella iteraatiolla lasketaan uudet tarkemmat arvot vääristymän keskipisteelle, hakualueelle ja väri-vääristymän suuruutta kuvaavalle parametrille α .

Käytännössä algoritmit laskevat ensin mallin summittaisen väri-vääristymän keskipisteen jollakin vääristymäparametrin α :n alkuarvolla tasaisesti hakualueelle jaetuilla testipisteillä. Hakualueena toimii ensimmäisellä iteraatiolla kuvan koko alue. Summittaisen keskipisteen laskemisen jälkeen lasketaan tarkempi arvo vääristymäparametri α :lle. Parametrien hakualueita tarkennetaan jokaiseen iteraation jälkeen. Käytettävien hakualueiden testipisteiden määrä on käyttäjän säädettävissä erikseen molemmille iteraation vaiheille.

4.1.3 Luokitin

Jokaisen lohkon piirrevektori lasketaan hyväksikäyttäen luokiteltavan lohkon kanssa päällekkäin olevia lohkoja. Piirrevektoriin kuuluu vääristymän suunta x -, sekä y -akselilla. Lohko luokitellaan manipuloiduksi, jos lohkon paikallisen väriääristymän suunta poikkeaa globaalista estimaatista enemmän kuin 60 astetta, sekä lohkon paikallisen väriääristymän suuruus on enemmän kuin 0,15 pikseliä.

Menetelmän suurin ongelma on luotettavien paikallisten estimaattien löytäminen. Estimaatti löydetään yleensä vain, mikäli lohkon alueella on paljon jyrkkiä ja monensuuntaisia reunoja. Mikäli lohkon sisällä on vain yhdensuuntainen reuna, estimaatti on yleensä epäluotettava ja näin ollen lohko luokitellaan helposti väärin. Mielenkiintoisilla alueilla, kuten kasvoilla on harvemmin tarpeeksi suuri gradientin keskiarvo ja näin ollen alue jätetään kokonaan luokittelematta.

4.2 Sensorikohinamenetelmän toteutus

Tässä kappaleessa esitellään sensoriestimaattimenetelmän toteutus. Kuvattu toteutus poikkeaa kappaleessa 3.4 kuvatusta menetelmästä varsin merkittävästi toteutusteknisistä syistä. Menetelmän toimintaperiaate pysyy kuitenkin samana. Kuten teoriaosuudessa on mainittu, opetuskuviin on oltava manipuloimattomia oikeellisten tulosten saamiseksi. Lisäksi toteutuksessa opetuskuviin ja luokiteltavien kuviin on oltava samankokoisia.

4.2.1 Sisällönpoistosuodatin

Toisin kuin kappaleessa 3.4 on mainittu, toteutuksessa käytetty sisällönpoistosuodin on adaptiivinen Wiener-suodin [50]. Suodinta voidaan käyttää, sillä se on teoriaosuudessa mainittu MMSE-suodin ja sen havaittiin käyttäytyvän hyvin reunojen läheisyydessä. Sama kohinasuodin löytyy esimerkiksi Matlab-ohjelmistosta funktionimellä *wiener2* [51]. Kuitenkaan nyt ei haluta poistaa kohinaa, vaan signaalin sisältökomponentti. Kohinakomponentti saadaan alkuperäisen ja suodatetun kuvan erotuksena.

Kuvataan seuraavaksi suodimen toimintaa. Ensimmäiseksi estimoidaan jokaisen pikselin $f(x, y)$ ympärillä olevan ikkunan η paikallinen keskiarvo. Diskreetit muuttujat n ja m kuuluvat ikkunaan η :

$$\mu(x, y) = \frac{1}{NM} \sum_{n, m \in \eta} f(n, m), \quad (4.4)$$

jossa $f(n, m)$ kuvaa alkuperäisen kuvan pikseliarvoja. Seuraavaksi estimoidaan paikalliset varianssit

$$\sigma_{\eta}^2(x, y) = \sum_{n, m \in \eta} [f(n, m) - \mu^2(x, y)]. \quad (4.5)$$

Kaavoissa (4.4) ja (4.5) käytetty ikkuna η on kooltaan N kertaa M -pikseliä. Käytännössä hyväksi havaittiin ikkunakoko 3 kertaa 3 -pikseliä. Tämän jälkeen estimaatteja käytetään hyväksi Wiener-suotimessa, joka on määritelty seuraavasti:

$$b_f(x, y) = \mu(x, y) + \frac{\sigma_\eta^2(x, y) - v^2}{\sigma_\eta^2(x, y)} [f(x, y) - \mu(x, y)], \quad (4.6)$$

jossa muuttuja v^2 kuvaa kohinan varianssia ja $b_f(x, y)$ yksittäistä tulosalkiota. Kohinan varianssin v^2 estimaattina käytetään kaikkien paikallisten varianssien σ_η^2 keskiarvoa.

4.2.2 Sensoriestimaatti

Toteutusta testattaessa havaittiin, että ML-estimaatin antamat luokitustulokset eivät poikkea merkittävästi kohinakuvien alkioiden keskiarvoista muodostetusta estimaatista. Keskiarvoestimaatti on myös laskentatehollisesti vähemmän vaativa. Näin ollen toteutus pohjautuu keskiarvoestimaattiin. Menetelmää testattaessa korrelaatioarvojen havaittiin käyttäytyvän epäjohdonmukaisesti käyttäytymistä molemmilla estimaattityypeillä. Syyksi paljastui 32-bittisen liukuluvun tarkkuuden loppuminen nollan läheisyydessä. Ongelma kierretään yksinkertaisesti summaamalla opetusnäytteistä suodatettujen kohinakuvien alkioita ilman jakolaskua. Keskiarvoestimaatti jättää lopulliseen estimaattiin paljon opetusnäytteissä esiintyvää kuvan sisällöstä. Tämän takia lopullinen estimaatti suodatettiin vielä kerran sisällönpoistosuotimella, jonka jälkeen sisältöä ei ollut enää silminnähtävissä.

4.2.3 Opetusnäytteet

Sensoriestimaatin lisäksi tarvitaan manipuloimattomia opetusnäytteitä luokittimen oikean toiminnan varmistamiseksi. Mikäli sensoriestimaattiin ja opetusnäytteisiin käytettäisiin samoja kuvia, piirrevektorien korrelaatioista tulisi liian suuria ja luokittimen luokat erottuisivat liian optimistisesti. Näin ollen luokittimen opetusnäytteet eivät saa olla samoja sensoriestimaatin opetukseen käytettyjen kuvien kanssa. Käytännössä tämä näkyy toteutuksen opetusvaiheessa. Ensimmäisellä opetuskerralla avattuja kuvia käytetään sensoriestimaatin laskemiseen. Tämän jälkeen menetelmä opetetaan toiseen kertaan, jolloin avattuja kuvia käytetään luokittimen opetusnäytteinä.

Saturoituneiden alueiden poistamiseen liittyvien raja-arvojen opetusnäytteiden että luokitusnäytteiden joukosta on käyttäjän hallittavissa. Näyte luokitellaan saturoituneeksi, mikäli kaikki sen värikanavista ovat alarajan alapuolella tai vastaavasti ylärajan yläpuolella. Käyttäjä voi muokata molempia raja-arvoja. Näytettä siis käytetään, mikäli yksi värikanavista ei ole saturoituneella alueella.

4.2.4 Luokitin

Luokitin toteutettiin yksiulotteisena Bayes-luokittimena. Luokitin käyttää kunkin lohkon eri värikanaville lasketun korrelaationkertoimien arvoista vain suurinta. Laskuajan säästämiseksi lohkoista \mathcal{R} luokitellaan vain ne, joiden luokiteltavan kuvan ja sensories-timaatin välinen sijainniltaan vastaava korrelaatio ρ on kaikkein alhaisin. Käyttäjä päättää miten suuri osa alueista luokitellaan tarkasti. Muut alueet merkitään automaattisesti manipuloimattomiksi. Tätä vaihetta kutsutaan myöhemmin esiluokittimeksi.

Jokainen luokitettava alue \mathcal{R} käyttää omaa luokitinta, joilla on omat opetusnäytteet. Luokitukseen valittuja opetusnäytteiden lohkon sijaintia merkitään symbolilla Q_i ja opetusnäytteeseen käytettyä kuvaa merkitään vektorimuodossa \underline{w}_j . Opetusmenetelmä toimii seuraavasti:

1. Valitaan opetusnäytteistä käyttäjän määrittelemä määrä RGB-histogrammiltaan samanlaisia alueita.
2. Opetetaan luokittimelle manipuloimaton luokka sijainniltaan vastaavista korrelaatioista $\rho(\underline{w}_j(Q_i), \hat{\underline{k}}_c(Q_i))$.
3. Opetetaan luokittimelle manipuloitu luokka sijainniltaan vastaamattomista korrelaatioista $\rho(\underline{w}_j(Q_i), \hat{\underline{k}}_c(\mathcal{R}))$.
4. Alueen luokka on korrelaation $\rho(\underline{w}_p(\mathcal{R}), \hat{\underline{k}}_c(\mathcal{R}))$ luokitustulos.
5. Mikäli alue luokitetaan manipuloituksi, talletetaan alueen sijainti.

Käyttäjä voi ohjata luokittimen toimintaa muuttamalla manipuloimattoman luokan prioritodennäköisyysparametria. Yleensä prioritodennäköidydeksi halutaan arvo, joka on enemmän tai yhtä kuin 50 prosenttia.

Kun kaikki alueet on luokiteltu, suodatetaan manipuloituksi luokitellut alueet virheluokitusten vähentämiseksi. Tulossuodin on eräänlainen mediaanisuodin, joka vaatii luokiteltavilta lohkoilta päällekkäisyyttä. Suodin toimii seuraavasti:

1. Lasketaan kuinka monta kertaa jokainen kuvan pikseleistä kuuluu manipuloituksi luokiteltuihin lohkoihin ja talletetaan saatu arvo.
2. Lasketaan vähintään kerran manipuloituun alueeseen kuuluvien pikselien mediaaniarvo.
3. Mikäli yksittäisen pikselin arvo on vähemmän kuin mediaaniarvo, merkitään pikseli manipuloimattomaksi.
4. Katsotaan mitkä alun perin manipuloituksi luokitelluista lohkoista sisältävät vain manipuloituja pikseleitä ja merkitään nämä lohkot lopullisesti manipuloituksi.

Hyväksi lohkojen päällekkäisyydeksi havaittiin 50–70 prosenttia. Alle 50 prosentin arvoilla tulossuodin joudutaan kytkemään pois päältä ja yli 70 prosentin arvolla luoki-

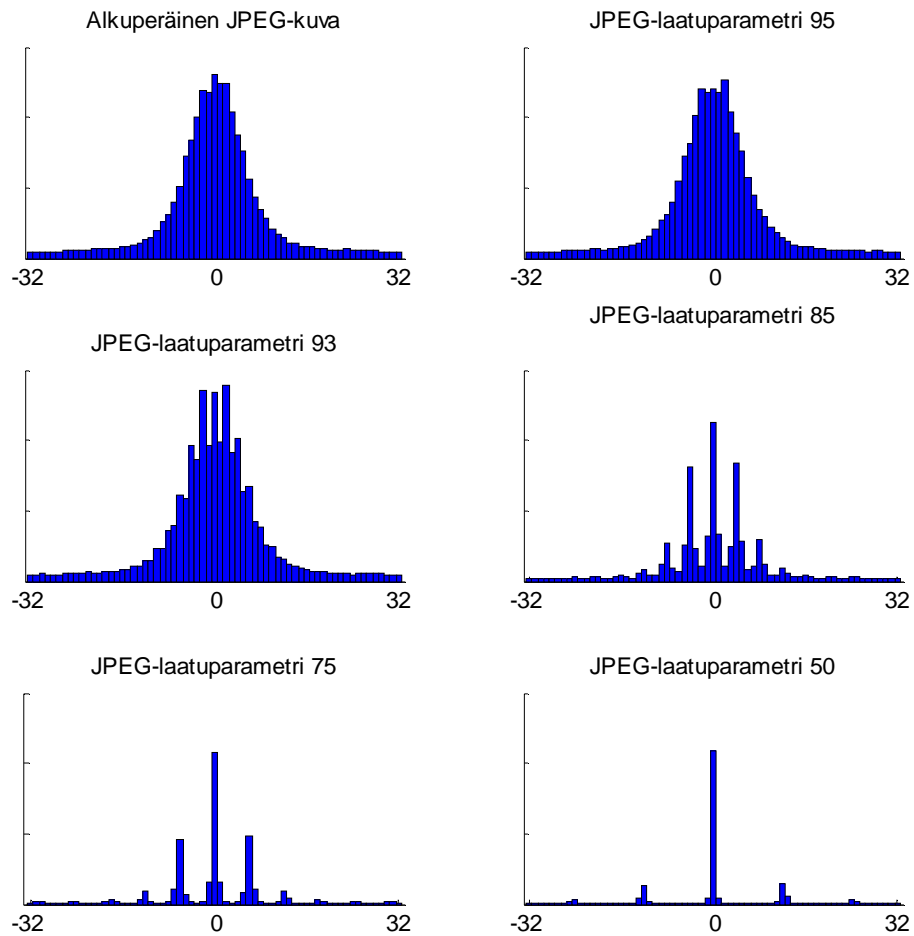
tukseen kului käytetyllä laitteistolla useita minutteja. Käytännön testeissä havaittiin myös, että suuri ikkuna antaa luotettavampia tuloksia kun lohkon sisällä ei ole reunaa tai vastaavaa alueen tilastollisia ominaisuuksia muuttavaa tekijää. Havaittiin myös, että erikokoisten lohkojen käyttö samaan aikaan toimii parhaiten. Prioritodennäköisyydelle sopivaksi arvoksi havaittiin noin 50 prosenttia.

4.3 DCT-histogrammimenetelmän toteutus

Toteutus JPEG-uudelleenpakkauksen havaitsemiseksi tehtiin kappaleessa 3.5 kuvatun DCT-arvojen jakauman artefakteihin perustuvan teorian mukaisesti. Muista toteutetuista menetelmistä poiketen DCT-histogrammimenetelmälle ei tehty automaattista luokitinta, sillä tulokset ovat helposti ihmisen tulkittavissa.

Menetelmä käyttää hyväkseen kappaleessa 2.5 esitellyn JPEG-pakkausprosessin ensimmäistä vaihetta. Varmin tapa saada oikeat DCT-arvot olisi lukea JPEG-tiedosto ja prosessoida tiedoston sisältämää dataa kunnes jäljellä olisi DCT-lohkojen sisältämät arvot. Kuitenkin toteutusteknisistä syistä JPEG-pakkauksen ensimmäistä vaihetta joudutaan simuloimaan suorittamalla DCT-muunnos jo valmiiksi purettuun kuvalle. Näin olleen käyttäjän tulee varmistua kuvan olevan alun perin JPEG-pakattu kuva. Seuraavassa kappaleessa selitetään tarkemmin menetelmässä käytettävät operaatiot.

Menetelmä muuttaa purettuun kuvan RGB-väriavaruuden YCbCr-väriavaruuteen. Tämän jälkeen jätetään jäljelle vain Y-kanavan sisältämä informaatio, sillä muut kanavat desimoidaan, sekä kvantisoidaan yleensä suuremmalla askeleella [52]. Kanavan jokaisesta arvoista vähennetään kokonaisluku 128. Seuraavaksi kuva jaetaan 8 pikselin kokosiin neliönmuotoisiin lohkoihin. Mikäli kuvan oikeassa reunassa tai alareunassa ei ole tarpeeksi pikseleitä täyttämään viimeistä lohkoa kokonaan, jätetään reuna-alue huomiotta. Luoduille lohkoille suoritetaan DCT-muunnos ja histogrammi luodaan vain DCT-alkion $\omega_x, \omega_y = 1$ arvoista. Histogrammin arvoalueeksi on valittu $[-32, 32]$, jonka ulkopuolisia arvoja ei oteta huomioon histogrammin ääripäissä. Mainitun arvoalueen havaittiin olevan riittävän suuri tuloksen todentamiseen, mutta ei sisällä liikaa informaatiota ihmisen tulkittavaksi. Toteutuksesta johtuen luvut esitetään alusta lähtien 32 bittisinä liukulukuina. Kuitenkin DCT-muunnoksen jälkeen jäljellä tulisi periaatteessa olla jäljellä vain kokonaislukuja. Tästä syystä histogrammin arvot on valittu niin, että yksi palkki edustaa yhden kokonaisluvun arvoaluetta. Tämän jälkeen histogrammi on valmis esitettäväksi käyttäjälle.



Kuva 4.1. Esimerkkihistogrammit JPEG-uudelleenpakkauksen vaikutuksesta manipuloimattoman kuvan Y -kanavan DCT-alkion $\omega_x, \omega_y = 1$ histogrammiin arvoalueella $[-32,32]$. Histogrammit ylhäältä alas ja vasemmalta oikealle: Tuntemattomalla laadulla otettu alkuperäinen JPEG-kuva, sekä tästä kuvasta uudelleenpakatut versiot laatuparametreilla 95, 93, 85, 75 ja 50.

Kuvassa 4.1 on esitettyä esitettyä esimerkkejä säännönmukaisista artefakteista eri JPEG-pakkauslaaduilla. Mikäli histogrammissa on selkeitä artefakteja, kuva on todennäköisesti pakattu kahteen kertaan JPEG-algoritmilla. Kuvasta 4.1 selviää, että artefaktien huiput ovat toisistaan sitä kauempana, mitä pienempää laatuparametria käytetään.

5 TULOKSET

Tässä kappaleessa käydään läpi eri menetelmien toimivuutta erilaisilla esimerkkikuva-manipulaatioilla, jotka esitellään tarkemmin kappaleessa 5.1. Kuvamanipulaatiot luotiin Adobe Photoshop -kuvankäsittelyohjelmalla [53]. Sisällön muuttamisen jälkeen kuvista luotiin myös jälkikäsitellyt versiot. Nyt kaikki luodut kuvat pakattiin sekä häviöttömästi, että JPEG-algoritmillä uudelleenpakkauksen vaikutuksen havainnoimiseksi. Mainittuja jälkikäsitelyoperaatioita ovat keskiarvo- ja mediaanisuodatus, kontrastin (+30 ja -30) tai kirkkauden (+30 ja -30) muuttaminen, gammakorjaus (parametreilla 0,7 ja 1,3), kuvan kääntäminen ja leikkaus, sekä valkoisen kohinan lisäys (parametri 12,5 %). Molempien suotimien koko oli 5 pikseliä, sekä pysty- että vaakasuunassa.

Sensorikohinamenetelmää testataan komposiittikuvilla, sekä retusoidulla kuvalla. Sensorikohinamenetelmän tuloksia käsitellään kappaleessa 5.2. Värivärästymämenetelmälle suoritettiin testejä vain yhdellä kopioi-siirrä-menetelmällä manipuloidulla kuvalla, sillä luotettavien estimaattien laskeminen onnistui vain erikoistapauksissa. Värivärästymämenetelmän tulokset löytyvät kappaleesta 5.3. Molempien menetelmien testikuvien uudelleenpakkaus suoritettiin Photoshopin JPEG-kuvanlaatuasetuksella yhdeksän, sekä Baseline optimized -parametrilla.

DCT-histogrammimenetelmän tapauksessa JPEG-uudelleenpakkaukseen käytettiin GIMP-ohjelmaa [54], sillä se tarjoaa tarkemman asteikon JPEG-kuvanlaatuparametrille. Käytetty asteikko on myös yhteensopiva kappaleen 2.5 teorian kanssa, sekä kuvassa 4.1 esitetylle esimerkille laatuparametrin vaikutuksesta menetelmän tuloksiin. Uudelleenpakkaukseen käytettiin laatuparametria 90. DCT-histogrammimenetelmän tuloksia käsitellään kappaleessa 5.4.

5.1 Testikuvat

Tässä kappaleessa esitellään menetelmien testaukseen käytetyt alkuperäiset kuvat 5.1 a), b), c) ja d), sekä näistä kuvista manipuloituja kuvia merkitään vastaavasti kuvilla 5.2 a), b), c) ja d). Alkuperäiset esimerkkikuvat on pakattu kamerassa korkeimmalla JPEG-kuvanlaatuasetuksella. Kuva 5.1 d) on otettu Canon D50 -järjestelmäkameralla ja muut kolme kuvaa on otettu Fujifilm FinePix F10 -taskukameralla.



Kuva 5.1. Alkuperäiset testikuvat alkaen oikeasta yläkulmasta a), b), c) ja d).

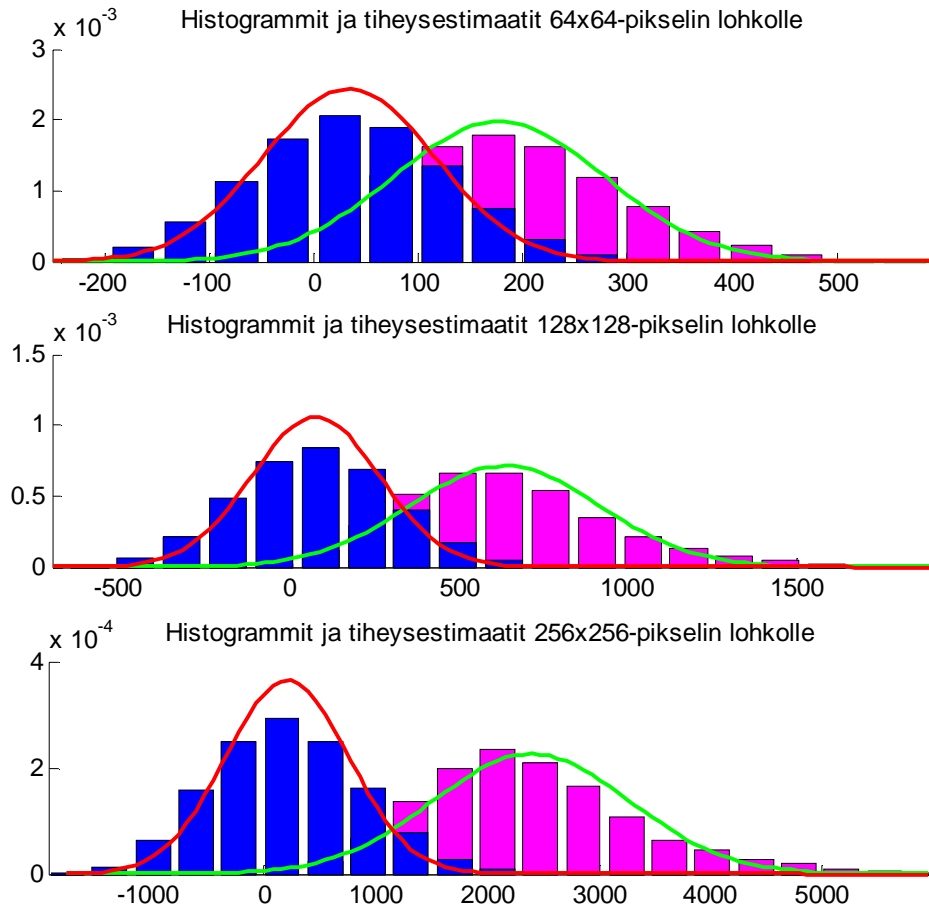
Kuva 5.2 a) on komposiitti kahdesta samalla kameralla otetusta kuvasta, jossa kopioitua aluetta on siirretty jonkin verran. Alkuperäisestä kuvasta on häivytetty alueita kopioimalla osia samasta kuvasta, mikäli alue ei peittynyt muualta kopioidun alueen alle. Ihmisen on lähes mahdotonta erottaa manipuloitua aluetta, sillä valaistusolosuhteet ja kameran sijainti pysyvät lähes muuttumattomina kuvien oton välillä. Kuva 5.2 b) on myös komposiittikuva, mutta kopioituun alueeseen on tehty enemmän kuvankäsittelyoperaatioita, kuten kääntäminen ja koon muuttaminen. Kopioitu alue on myös sensorikohinamenetelmän kannalta ongelmallinen, sillä kopioidut alueet ovat hyvin kapeita, vaikeuttaen menetelmän toimintaa. Kuvaa 5.2 c) on retusoitu ilman suoraa kopiointia. Kuvassa esiintyvän henkilön kasvoja on muokattu kasvokuville tyypillisellä tavalla poistamalla hikipisaroita, luomia ja muita vastaavia kohteita. Kuvan 5.2 d) keskellä oleva kala on liitetty kolmeen kohtaan erilaisten tulosten havainnollistamiseksi. Tämän lisäksi kuva sisältää paljon kohinan kaltaisia korkeita taajuuksia, mikä vaikeuttaa sensorikohinamenetelmän toimintaa. Toisaalta taas värivärähtelymenetelmän tapauksessa kuva oli ainoa testijoukosta, jolle saatiin laskettua teorian mukainen värähtelyäestimaatti.



Kuva 5.2. Muokatut testikuvat a), b), c) ja d) esiteltynä vasemmasta yläkulmasta alkaen. Kuvassa a) folionpala on liitetty toisesta, samalla kameralla otetusta kuvasta. Kuvassa b) on lisätty tekstiä toisesta kuvasta. Kuvassa c) olevan henkilön kasvoja on tasoitettu, poistettu hikipisaroita ja niin edelleen. Kuvan d) keskellä oleva kala on kopioitu muualle kuvaan useaan kertaan.

5.2 Sensorikohinamenetelmä

Kappaleessa 4.2 toteutettu sensorikohinamenetelmä osoittautui toimivaksi menetelmäksi manipuloitujen alueiden tunnistamiseen. Kuvassa 5.3 on kuvattuna sensoriestimaatin ja kohinakuvan korrelaatiot erikokoisille lohkoille. Sijainniltaan vastaamattomien lohkojen korrelaation keskiarvo on nollan läheisyydessä ja korreloivien lohkojen huomattavasti suurempi, riippuen lohkon koosta. Kuvasta voidaan havaita, että jakauma vastaa hyvin Gaussin jakaumaa pienellä ikkunakoolla ja kohtalaisesti suuremmalla. Syyksi tähän epäillään sitä, että suuremman lohkon tilastolliset ominaisuudet muuttuvat suuremmalla todennäköisyydellä lohkon sisällä. Tilastollisilla ominaisuuksien muutoksella tarkoitetaan esimerkiksi lohkon sisältä löytyvää teräviä reunoja ja värimuutoksia. Kuitenkin päätösrajapinnan läheisyydessä Gaussin jakauma vastaa kohtalaisen tarkasti esitimoitua jakaumaa kaikilla testatuilla lohkotyypeillä.



Kuva 5.3. Sijainniltaan vastaamattomien (sininen) ja vastaavien (magenta) lohkojen tiheysestimaatit lohkon kohinakuvas välisistä korrelaatioista. Tiheyttä mallinnetaan Gauss-funktioilla (punainen ja vihreä).

Taulukossa 5.1 kuvataan teoreettiset luokitusvirheet kuvassa 5.3 esiintyville lohko-tyypeille. Suurempaa lohkoa käytettäessä virheluokitusten määrä pienenee, mutta toisaalta pienellä lohkon koolla voidaan havaita pienempiä manipuloituja alueita. Tästä syystä menetelmän testausvaiheessa päädyttiin käyttämään erikokoisia lohkoja. Näin voidaan käyttää hyväksi suurten lohkojen varmempi luokittelutulos, sekä pienempien lohkojen tarkempi erottelukyky. Mitä pienempiä alueita halutaan luokitella, sitä pienempää ikkunakokoa ja suurempaa lohkojen päällekkäisyyskerrointa halutaan käyttää.

Luokituksen jälkeinen tulossuodatin poistaa tehokkaasti yksittäiset virheellisesti positiiviseksi luokitellut lohkot. Manipuloiduiksi luokiteltujen päällekkäisten lohkojen määrän havaittiin olevan riittävä tarpeeksi suurilla manipuloiduilla alueilla, joten nämä alueet jäivät jäljelle suodatuksen jälkeenkin. Tulossuodatusta käytettäessä kannattaa myös huomata, että jos kaikki manipuloidut alueet merkitään manipuloimattomaksi, jos suurimmalla osalla kuvan pikseleistä on sama määrä manipulaatioluokituksia. Siis jos kuvan kaikki lohkot on luokiteltu manipuloiduiksi, suodatusvaiheessa kaikki luokitukset muutetaan manipuloimattomaksi. Tästä syystä on kannattavaa käyttää esiluokitinta, joka

jättää eniten sensoriestimaatin kanssa korreloivat lohkot luokittelematta. Käytännön testeissä kustakin lohkotyyppistä luokiteltiin vain vähiten korreloiva neljäsosa, joka osoittautui hyväksi kompromissiksi laskentanopeuden ja luokitustulosten välillä. Vaarana näin yksinkertaisessa esiluokittimessa on laajojen manipuloitujen alueiden havaitseminen. Mikäli suurta osaa kuvasta on manipuloitu, esiluokitin voi jättää luokittelematta myös manipuloitulle alueelle kuuluvia lohkoja. Toinen vaihtoehto on luokitella kaikki lohkot ilman esiluokitinta ja tarkastaa manipuloituiksi luokitellut kuvat silmämääräisesti.

Neliölohkon sivu (pikseliä)	Oikea negatiivinen	Väärä negatiivinen	Oikea Positiivinen	Väärä positiivinen
64	0,37	0,09	0,41	0,13
128	0,43	0,04	0,46	0,07
256	0,48	0,03	0,47	0,02

Taulukko 5.1. Sijainniltaan vastaavan (negatiivinen) ja sijainniltaan vastaamattomien (positiivinen) luokkien tiheyttä Gauss-funktiolla mallintavan Bayes-luokittimen teoreettiset luokitusvirheet erikokoisille neliölohkoille.

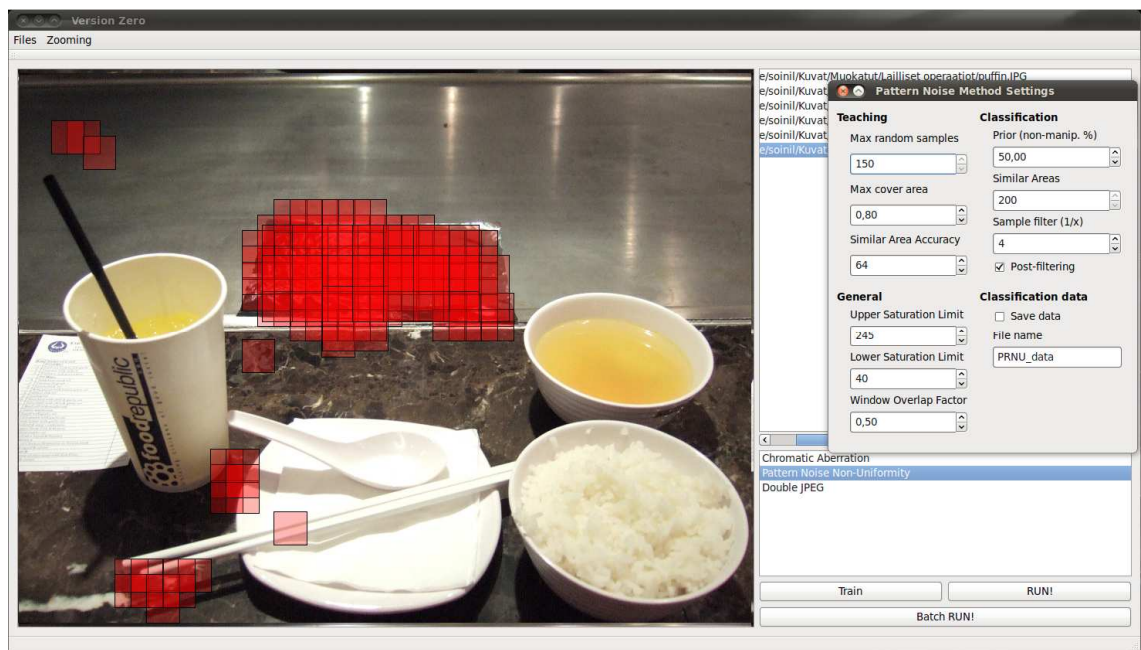
Kaikissa neljässä testitapauksessa testit suoritettiin samoilla asetuksilla. Sensoriestimaattiin käytettiin 25 kuvaa ja luokittimen opetusnäytteisiin 10 kuvaa. Opetuskuvat olivat peräisin laajemmasta valikoimasta, eikä niitä ole erikseen otettu tarkoitusta varten. Sensoriestimaattiin valitut kuvat sisältävät mahdollisimman vähän tummia alueita ja opetusnäytekuvat ovat väreiltään ja sisällöltään mahdollisimman erilaisia. Lohkon kokona käytettiin samanaikaisesti sivultaan 128:n ja 256:n pikselin kokoisia neliömallisia lohkoja. Jokaisen luokiteltavan lohkon sijainniltaan vastaavaa luokkaa varten etsittiin 200 värilliseltä histogrammiltaan mahdollisimman samanlaista opetusnäytelohkoa. Sijainniltaan vastaamaton luokka koottiin edellä mainittujen lohkojen kohinakuva ja luokiteltavan lohkon kohdalla olevan sensoriestimaatin välisestä korrelaatiosta. Luokiteltavien lohkojen päällekkäisyysarvona on 50 prosenttia ja prioritodennäköisyytenä käytettiin 50 prosentin todennäköisyyttä. Esiluokittimen asetuksena käytettiin yhtä neljäsosaa ja tulossuodatin oli käytössä kaikissa testeissä.

Kaiken kaikkiaan luokittustulosten laadun havaittiin riippuvan todella paljon käytetyistä opetusnäytteistä. Tästä syystä, mikäli mahdollista, opetuksen laatua tulisi testata tunnetuilla näytekuvilla ennen varsinaisen luokituksen aloittamista. Testituloksia kerran JPEG-pakatuilla kuvilla käsitellään tarkemmin kappaleessa 5.2.1 ja uudelleen JPEG-pakattujen kuvien testituloksia käsitellään kappaleessa 5.2.2.

5.2.1 Esimerkkikuvien testitulokset

Tässä kappaleessa käydään läpi kertaalleen JPEG-pakkauksen jälkeen manipuloitujen kuvien testitulokset ja havainnot. Tämän jälkeen käydään pääkohdat manipuloinnin jälkeen käsiteltyjen kuvien tuloksista.

Kuvan 5.2 a) tapauksessa muokattu alue (foliopaperi) havaittiin onnistuneesti. Kuvassa on paljon tummia alueita ja siirtymisiä vaaleasta alueesta tummaan, joka johti muutamaan virheluokitukseen. Toteutuksen käyttäjälle esittämä luokitustulos on nähtävissä kuvassa 5.4. Kuvassa 5.2 b) osa siirretystä alueesta havaittiin onnistuneesti. Kannattaa huomata että manipuloiduksi luokitellaan alueita vain pienemmällä luokitukseen käytettävällä 128:n pikselin kokoisella loholla, joka mahtui kokonaisuudessaan manipuloidun alueen sisään.



Kuva 5.4. Kuvakaappaus toteutuksen työnäkymästä, sekä kuvan 5.2 a) luokitustulos. Manipuloiduiksi luokiteltuja alueita merkitään punaisella neliöllä. Oikealla on kuvasensorimenetelmän asetussikkuna.

Kuvan 5.2 c) retusointityyppinen manipulointi osoittautui hankalaksi sensoriestimaattimenetelmälle. Kuvassa olevat retusointioperaatiot muokkaavat varsin pientä aluetta kerrallaan. Manipuloituja pikseleitä sisältävän lohkon sisältä löytyy myös paljon manipuloimattomia pikseleitä. Tämä johtaa siihen, että myös manipuloituja alueita sisältävä lohko luokitellaan manipuloimattomaksi. Tästä syystä menetelmä ei ole kovin herkkä havaitsemaan retusointia. Kuvamanipulointi havaittiin vain otsan alueella, jossa retusointioperaatioita on eniten pinta-alan kannalta katsoen.

Kuvassa 5.2 d) on sisältösuodatuksen jälkeen testijoukosta eniten sisältöä jäljellä. Vain oikeanpuoleisin manipuloitu alue havaittiin onnistuneesti. Tulokseen vaikuttaa myös se, että kuva on otettu eri kameralla kuin muut kuvat ja sillä on myös omat opeusnäytteet. Näin ollen kuvan tulokset eivät ole täysin yhteensopivia muiden testikuvien

kanssa. Kuitenkin tuloksesta nähdään, että toteutettu menetelmä ei välttämättä havaitse manipuloitua aluetta mikäli, kuvassa on paljon korkeita taajuuksia. Tämä johtuu sisäl-
tösuodattimen epäonnistumisesta erottaa sisältö ja kohina toisistaan.

Kaikkien testikuvien tapauksessa kontrastin, kirkkauden muutokset, sekä gamma-
korjaus eivät muuta merkittävästi luokitustulosta. Edellä mainitut operaatiot muuttavat
lohkojen kirkkauden keskiarvoa, jolloin luokitellaan sellaisia lohkoja jotka muuten jätet-
täisiin luokittelematta ja toisinpäin. Kaiken kaikkiaan edellä mainitut operaatiot lisäävät
yksittäisiä virheluokituksia. Manipuloidut alueet tunnistetaan kuitenkin edelleen edellä
kuvatulla tavalla ja yksittäiset virheluokitukset poistuvat tehokkaasti tulossuodatuksen
avulla. Kuvan kääntäminen, zoomaus, kohinan lisäys, keskiarvosuodatus, sekä mediaa-
nisuodatus johtavat siihen, että suurin osa kuvan pinta-alasta luokitellaan manipuloiduk-
si riippumatta tulossuodattimen käytöstä. Mikäli kuvaan on tehty jokin edellä luetelluis-
ta jälkikäsitteilyoperaatiosta, manipuloidun alueen piirrevektorina käytetty korrelaatio ei
enää eroa merkittävästi manipuloimattomista alueista. Tällöin manipuloitua aluetta on
mahdotonta erottaa luokitustulosten perusteella. Kohinan lisäys kuvan 5.2 d) tapaukses-
sa johti oikealla ja vasemmalla olevien manipuloitujen alueiden havaitsemiseen, mutta
suurin osa positiivisista luokituksista oli virheellisiä.

5.2.2 JPEG-uudelleenpakkauksen vaikutukset testituloksiin

Tässä kappaleessa käydään läpi JPEG-uudelleenpakkauksen vaikutukset testituloksiin.
Taulukosta 5.2 voidaan havaita, että luokiteltavan kuvan pakkaaminen toiseen kertaan
heikentää luokittimen toimintaa kertaalleen pakattuihin kuviin verrattuna. Ilmiö havait-
tiin myös käytännössä, sillä yleisesti ottaen JPEG-uudelleenpakkaus lisäsi virheluoki-
tusten määrää kaikissa testitapauksissa. Virheellisesti manipuloiduiksi luokiteltuja loh-
kojen määrä lisääntyi huomattavasti enemmän kuin virheellisesti manipuloimattomaksi
luokiteltuihin lohkoihin verrattuna. Oikeasti manipuloitua aluetta oli nyt lähes mahdo-
tonta erottaa virheluokitusten joukosta, sillä yksittäisten lohkojen sijaan virheluokitukset
käsittivät kokonaisia alueita.

Neliölohkon sivu (pikseliä)	Oikea nega- tiivinen	Väärä nega- tiivinen	Oikea Posi- tiivinen	Väärä posi- tiivinen
64	0,30	0,17	0,33	0,20
128	0,34	0,10	0,40	0,16
256	0,41	0,05	0,45	0,08

Taulukko 5.2. Sijainniltaan vastaavan (negatiivinen) ja sijainniltaan vastaamattomien (positiivinen) luokkien tiheyttä Gauss-funktiolla mallintavan Bayes-luokittimen teoreettiset luokitusvirheet erikokoisille JPEG-uudelleenpakatuille neliölohkoille.

JPEG-uudelleenpakkaus lisäsi kuvien 5.2 a), b) ja c) tapauksessa yksittäisiä virhe-
luokituksia, mutta toisaalta jälkikäsittelemättömien kuvien manipuloitu alue oli luokitel-
tu tarkemmin kuvien 5.2 b) ja c) tapauksessa. Yllättäen JPEG-uudelleenpakatuista ku-

vasta 5.2 d) havaittiin kaikki kolme manipuloitua aluetta, mutta toisaalta tuloksessa oli suuri määrä väärin manipuloituksi luokiteltuja lohkoja. Myös JPEG-uudelleenpakattujen testikuvien tapauksessa kontrastin, kirkkauden muutokset, sekä gammakorjaus lisäsivät edelleen virheluokitusten määrää. Virheluokituksia oli nyt siinä määrin, ettei tuloksista pystynyt tulkitsemaan oikein manipuloitua ja väärin manipuloituksi luokiteltuja alueita. Mediaani- ja keskiarvosuodatus johtivat kuvan 5.2 d) kuvan tapauksessa yhteen suureen manipuloituksi luokiteltuun alueeseen. Luokitustulos oli kuitenkin kokonaan virheellinen. Muiden kuvien tapauksessa luokitustulokset olivat sattumanvaraisia. Kohinan lisäys aiheutti nyt kaikkien kuvien tapauksessa sattumanvaraisia luokitustuloksia.

5.3 Väriääritysmenetelmä

Kappaleessa 4.1 toteutettu väriääritysmiin perustuva menetelmä osoittautui melko epävarmaksi menetelmäksi manipuloitun alueen tunnistamiseen. Kuitenkin menetelmä toimi parhaiten kuvalla 5.2 d), jolla sensoriestimaattimenetelmä toimi heikoiten.

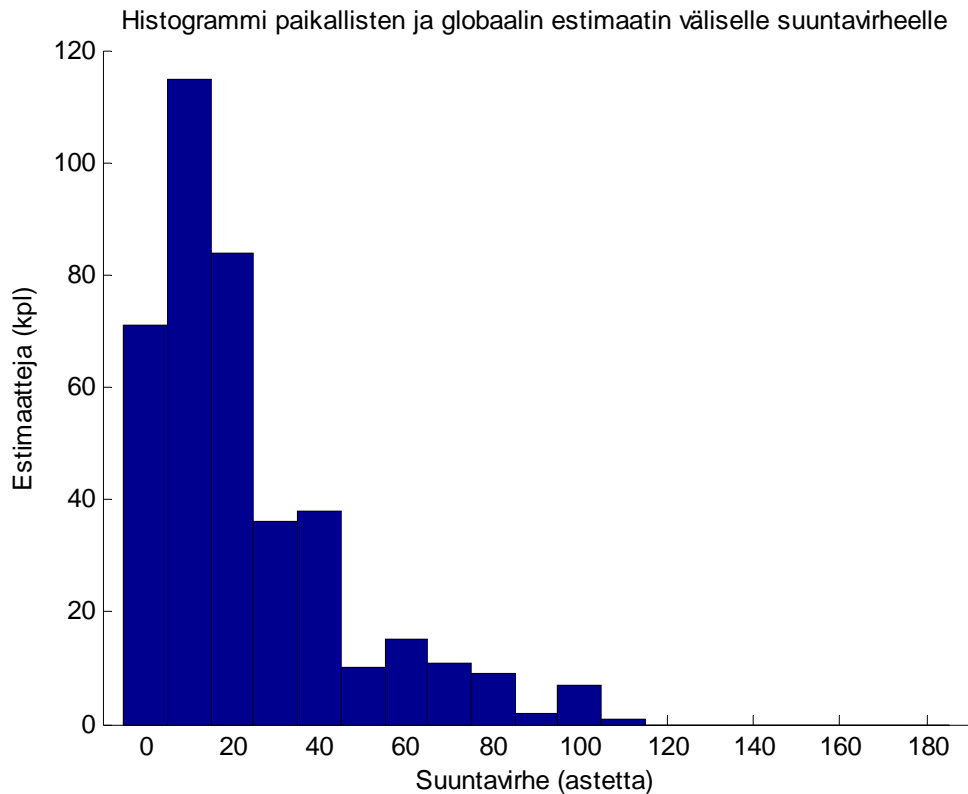
Taulukossa 5.3 on kuvattuna paikallisten estimaattien ja globaalien estimaattien väliset suuntaerot asteina kappaleen 5.2 sensoriestimaatin opetuskuville, sekä kuvalla 5.1 d). Taulukosta selviää, että opetuskuviin estimaattien väliset suuntaerot ovat huomattavia. Tästä syystä menetelmän käyttäminen kuvamanipuloinnin tunnistamiseen tulee kysymykseen vain, jos kuvan globaali ja paikalliset estimaatit poikkeavat toisistaan vain vähän. Jotta luotettava estimaatti saataisiin laskettua, kahden estimaattiin käytetyn värikanavan intensiteeteissä tulee olla paljon jyrkkiä vaihteluita. Tilanne on mahdollinen jos kuvassa on esimerkiksi puun oksien läpi näkyvää valoa. Kannattaa huomata, että myös manipuloitun alueen luokitus vaatii luotettavan estimaatin laskemista alueelta.

Lohkojen alkuperä	Gradienttien keskiarvo	Gradienttien mediaani	Väriäärityksen keskipituus (pikseliä)	Estimaattien keskim. suuntaero (astetta)
Opetuskuvat	4,373	3,824	0,0519	77,62
Kuva 5.1 d)	4,372	4,119	0,816	17,35

Taulukko 5.3. Eri kuvista saatujen lohkojen tilastollisten ominaisuuksien vertailua.

Taulukosta 5.3 selviää myös, että mustavalkokuvan gradientin keskiarvo ei ole kovinkaan hyvä keino lohkon sopivuuden määrittämiseksi väriääritysmenetelmälle. Edellä mainittujen opetuskuviin ja kuvan 5.1 d) luokitteluun käytettyjen lohkojen gradienttien keskiarvo ja mediaani ovat hyvin lähellä toisiaan ja reilusti raja-arvon yläpuolella, mutta paikallisten ja globaalien mallien välinen keskimääräinen suuntavirhe on opetuskuvilla huomattavasti suurempi. Myös mitattujen väriääritysmien keskipituus poikkeaa huomattavasti toisistaan, vaikka kuvien ottamiseen on käytetty samaa kameraa.

Kuvassa 5.5 on histogrammi paikallisten estimaattien ja globaalin estimaatin välisille suuntaeroille kuvan 5.1 d) estimaatteja käyttäen. Globaalin ja paikallisten estimaatin erot ovat tässä tapauksessa vähäisiä. Muiden testikuvien (kuvat 5.1 a), b) ja c)) tapauksessa vääristymäestimaatit eivät muodostaneet teorian mukaista globaalia estimaattia ja estimaattien väliset virheet olivat liian suuria luotettavaan luokitukseen tai lohkoja ei luokiteltu mitatun värivääristymän pienuuden takia.



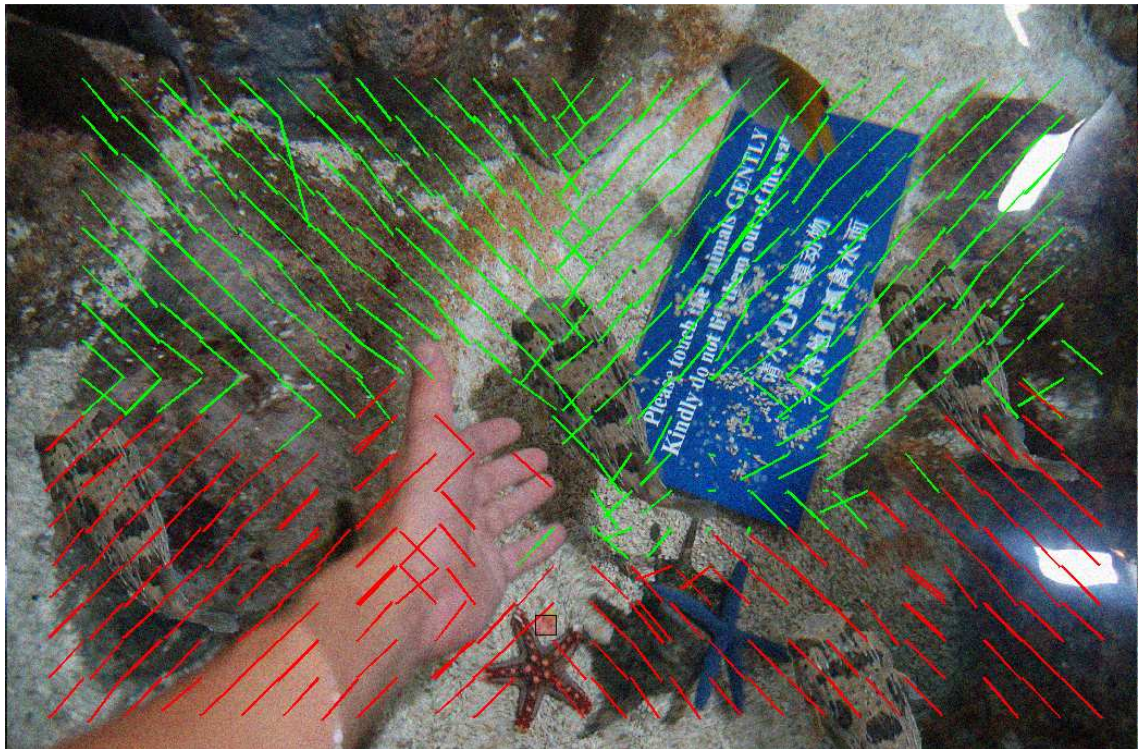
Kuva 5.5. Histogrammi paikallisten ja globaalin estimaatin välisille suuntavirheille kuvan 5.1 d) värivääristymistä.

Kuvan 5.2 d) värivääristymän laskettu keskipiste on kuvan alalaidassa. Manipuloidun alueen paikallisten estimaattien keskiarvon tulee poiketa vähintään 60 astetta globaalista estimaatista jotta alue merkittäisiin manipuloiduksi. Esimerkkikuvassa tämä ehto toteutuu vain vasemmalla olevan kopioidun alueen kohdalla. Oikealla olevan alueen tapauksessa paikalliset estimaatit poikkeavat raja-arvoa vähemmän ja oikeassa alukulmassa alue on liian pieni vaikuttaakseen paikallisten estimaattien keskiarvoon.

Seuraavaksi menetelmän toimivuutta testattiin kuvan 5.2 d) jälkikäsitellyille versioille. Keskiarvo- ja mediaanisuodatuksen jälkeen suurin osa kuvista ei enää ylittänyt vaadittua keskiarvogradienttia, jolloin suurinta osaa kuvasta ei luokiteltu. Manipuloidut alueet jätettiin nyt luokittelematta. Jäljelle jäävien paikallisten estimaattien laatu heikkeni huomattavasti, lisäten suuntaerojen suuruutta globaalin estimaatin kanssa. Keskiarvosuodatus vaikutti estimaatteihin jonkin verran mediaanisuodatusta enemmän. Kontrastin ja kirkkauden muutokset lisäsivät estimaattien virhettä jonkin verran, mutta eivät vaikuttaneet luokitustulokseen merkittävästi. Kuitenkin näiden operaatioiden jälkeen

jotkin kuvan alueet joutuvat dynaamisen alueen ääripäihin ja niiden keskiarvogradientti pienenee. Näin ollen kuvasta jää enemmän alueita luokittelematta. Samankaltainen tulos saavutettiin gammakorjauksen tapauksessa. Huomattavaa kuitenkin on, että esimerkkikuvan tapauksessa gammakorjaus kertoimella 0,7 itse asiassa vahvistaa mitatun väriväräristymän suuruutta kuvasta löytyvillä kirkkailla alueilla. Näillä alueilla gammakorjaus suurentaa intensiteetin vaihtelua ja käytetty algoritmi löytää suuremman estimaatin. Kuvan kääntäminen ja suurennos eivät vaikuta estimaattiin, mikäli suurennoksen aiheuttama muutos paikallisiin gradientin keskiarvoihin ei ole kovin suuri.

Mielenkiintoisin tulos saatiin kun kuvaan lisättiin valkoista kohinaa. Saaduilla estimaateilla ei enää ollut kuitenkaan mitään tekemistä alkuperäisten estimaattien kanssa. Paikalliset estimaatit olivat kuvassa vinoittaisia ja toisiinsa nähden samansuuntaisia, 90- tai 180-asteen kulmassa. Kuvassa 5.6 on havainnollistettu tilannetta. Ilmiö on mielenkiintoinen, sillä se toistuu sekä Photoshop-, että GIMP-kuvankäsittelyohjelmilla. Ilmiö johtuu todennäköisesti menetelmän tavasta etsiä paikallisia estimaatteja. Näin ollen menetelmällä voitiin ainakin esimerkkikuvan tapauksessa selvittää onko kuvaan lisätty kohinaa keinotekoisesti.



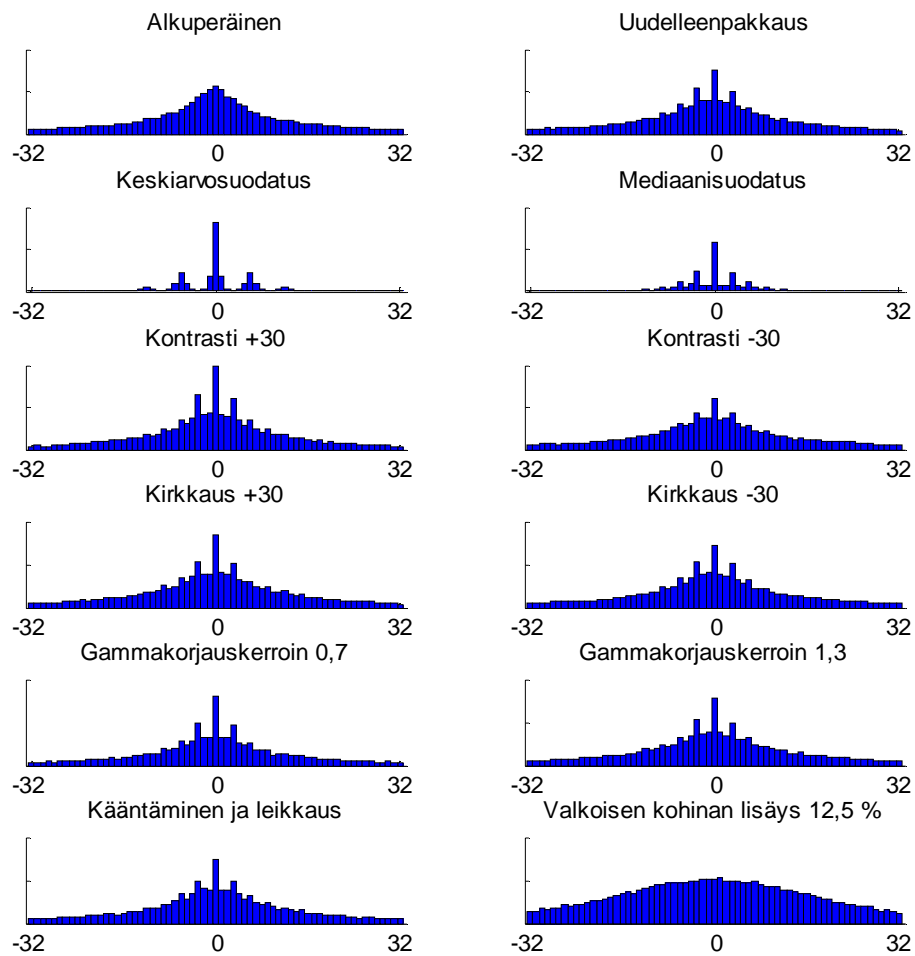
Kuva 5.6. Väriväräristymämenetelmän visualisoidut tulokset kuvalle 5.2 d), kun kuvaan on lisätty muokkauksen jälkeen valkoista kohinaa. Vihreät viivat kuvaavat globaalin väräristymäkeskipisteen (punainen neliö) kanssa yhdenmukaisia paikallisia estimaatteja. Punaiset viivat kuvaavat manipuloituiksi luokiteltuja alueita.

Uudelleenpakatut kuvat käyttäytyivät myös mielenkiintoisesti. Esimerkkikuvan tapauksessa kirkkauden lisäämisen tai kertoimella 1,3 gammakorjaamisen jälkeinen JPEG-pakkaus lisää luokiteltavien alueiden määrää. Tämä johtuu pakkauskohinan li-

sääntymisestä, joka luonnollisesti lisää lohkojen keskiarvogradienttia. Yleisesti ottaen testeissä ilmeni uudelleenpakkauksen heikentävän estimaattien laatua, mutta ei kuitenkaan aiemmin mainittuun luokitustulokseen vaikuttavalla tavalla. Valkoisen kohinan lisäämisen jälkeinen JPEG-pakkaus heikentää edellisessä kappaleessa havaitun ilmiön lähes olemattomiin, mutta estimaattien suunnat pysyivät kuitenkin samoina.

5.4 DCT-histogrammimenetelmä

Kappaleessa 4.3 toteutettu DCT-histogrammimenetelmä osoittautui tehokkaaksi JPEG-uudelleenpakkauksen havaitsemiseen. Testikuvista kaikki alle JPEG-laatuparametrilla 95 uudelleenpakatut kuvat osoittivat kuvan 4.1 kaltaisia merkkejä säännönmukaisista artefakteista. Yhdelläkään manipuloidulla ja/tai jälkikäsitellyllä kuvalla, jotka olivat kertaalleen pakattuja, ei havaittu merkkejä tämänkaltaisista artefakteista. Uudelleenpakattujen ja jälkikäsiteltyjen kuvien tapauksessa artefaktit olivat havaittavissa kaikissa muissa tapauksissa, paitsi kuvissa joihin oli lisätty valkoista kohinaa. Näissä tapauksissa artefakteja ei havaittu, mutta DCT-arvot olivat levittäytyneet huomattavasti laajemmalle arvoalueelle kuin alkuperäisissä kuvissa.



Kuva 5.7. DCT-lohkojen arvon $\omega_x, \omega_y = 1$ histogrammit. Ylhäältä alas ja vasemmalta oikealle: Alkuperäinen JPEG-kuva, toiseen kertaan JPEG-pakattu kuva, keskiarvosuodatus, mediaanisuo-datus, kontrastin muutos (+30 ja -30), kirkkauden muutos (+30 ja -30), gammakorjaus (0,7 ja 1,3), kuvan kääntäminen ja leikkaus, sekä valkoisen kohinan lisäys (12,5 %). Kuvat on pakattu toiseen kertaan GIMP-ohjelman laatuparametrilla 90.

Kuvassa 5.7 on havainnollistettuna esimerkikuvan 5.2 d) DCT-histogrammeista. Taulukossa ylärivillä vasemmalta oikealle ovat DCT-histogrammit alkuperäiselle kuvalle ja uudelleenpakatulle kuvalle. Muut histogrammit ovat kappaleen 5 johdannossa mainittuja kuvan jälkikäsitteilyoperaatioita. Kuvasta 5.7 havaitaan kaikkien operaatioiden, paitsi kohinan lisäyksen jälkeisissä DCT-histogrammeissa on selkeästi merkkejä säännömukaisista artefakteista. Huomattavaa on, että manipuloimattomien kuvien DCT-histogrammit eivät poikenneet merkittävässä määrin manipuloiduista kuvista. Kaikista testatuista kuvista esimerkikuvan 5.2 d) DCT-histogrammin artefaktit olivat heikoimpia. Muissa kuvissa artefakteja oli huomattavasti laajemmalla alueella ja ne näkyivät selkeämmin. Tämä johtuu siitä että kuva 5.2 d) sisältää muihin kuviin verrattuna varsin paljon kohinankaltaista sisältöä.

6 JOHTOPÄÄTÖKSET

Tässä työssä esiteltiin digitaalisen kameran kuvantamisprosessi ja erilaisia kuvamanipulointityyppejä. Tämän jälkeen esiteltiin joitakin kuvantamisprosessiin perustuvia kuvamanipuloinnihavaitsemismenetelmiä, sekä kopioidun alueen löytämiseen käytettäviä algoritmeja. Kuvamanipuloinnin havaitsemiseen ei ole nykyisellään olemassa yhtäkään täysin varmasti ja kaikissa mahdollisissa tilanteissa toimivaa menetelmää. Kaikilla esitellyillä menetelmillä on omat rajoitteensa ja käyttötilanteensa. Esitellyn teorian pohjalta toteutettiin kolme eri teoriapohjaa hyödyntävää menetelmää. Menetelmät perustuivat optiikan aiheuttamiin väriääritymiin, kuvasensorin epätasaiseen kykyyn muuntaa valoa sähköksi, sekä JPEG-uudelleenpakkauksen havaitsemiseen diskreetistä kosinimuunnoshistogrammista.

Väriääritymä- ja sensorikohinamenetelmät pyrkivät havaitsemaan manipuloidun alueen ja DCT-histogrammimenetelmä pyrki koko kuvan kattavan JPEG-uudelleenpakkauksen havaitsemiseen. Menetelmiä testattiin tyypillisimmillä kuvamanipulaatiotavoilla, sekä havainnollistettiin jälkikäsitteilyn vaikutusta menetelmän antamaan luokitustulokseen. Tämän jälkeen kaikki testatut kuvat pakattiin uudelleen JPEG-algoritmeilla, sekä käsiteltiin pääkohdat uudelleenpakkauksen vaikutuksista menetelmien toimintaan.

Sensoriestimaattimenetelmä havaitsi manipuloidut alueet parhaiten. Menetelmä havaitsi alueet, mikäli manipuloitu alue oli käytettyä ikkunakokoa huomattavasti suurempi, sekä sisällönpoistosuodatin onnistui tehtävässään riittävän hyvin. Myös opetusnäytteiden laadun havaittiin vaikuttavan suuresti luokitustulokseen. Parhaassa mahdollisessa tapauksessa sensoriestimaatin opetukseen käytettävien kuvien tulisi olla sävyiltään tasaisia ja sisältää informaatiota kaikilla värikanavilla. Luokittimen opetukseen käytettävien kuvien tulisi olla mahdollisimman samankaltaisia luokiteltavien kuvien kanssa. Lisäksi kaikkien opetuskuviin tulisi olla käsittelemättömiä ja luonnollisesti peräisin samasta kamerasta luokiteltavien kuvien kanssa. Nämä vaatimukset asettavat rajoituksia menetelmän käyttötilanteille. Menetelmää testattaessa sopivat kuvat valittiin kuitenkin laajemmasta valikoimasta, eikä kuvia otettu erikseen tarkoitusta varten. Tulokset olivat rohkaisevia menetelmän jatkokehitykselle. Seuraavaksi toteutusta voisi parantaa tutkimalla erilaisten sisällönpoistosuotimien tehokkuutta ja toteuttamalla parempi menetelmä samankaltaisten näytteiden havaitsemiseen. Parannuksia tulisi tehdä myös menetelmän nopeutta parantavaan esiluokittimeen, sekä luokitukseen kelpaamattomien lohkojen poiston tekevään algoritmiin.

Väriääritymämenetelmä toimi teorian kuvaamalla tavalla vain erikoistapauksissa, vaikka luokiteltavat kuvablokkot täyttivät menetelmän teoriaosuudessa asetetut rajoitteet.

Kuvalohkojen luokituskelpoisuuden määrittämiseen käytetyn mustavalkokuvan keskiarvogradientin osoitettiin soveltuvan tehtäväänsä heikosti. Tästä johtuen lohkon luokituskelpoisuuden voisi päättää nykyisen sijaan tarkastelemalla vain vertailtavista värikanavista muodostettua keskiarvogradienttia tai lohkon sisältämään taajuussisältöön perustuvaa analyysia. Testausvaiheessa menetelmän havaittiin käyttäytyvän helposti tulkittavalla tavalla, mikäli kuvaan oli lisätty valkoista kohinaa. Estimaatinhakualgoritmin toimintaa voisi yrittää parantaa lisäämällä yksittäisen lohkon intensiteettieroja, jolloin vääristymän approksimoivan algoritmin on osoitettu löytävän normiltaan suurempia estimaatteja paikalliselle vääristymälle. Myös erilaisten interpolointimenetelmien vaikutusta estimaatinhakualgoritmin toimintaan voisi tutkia tarkemmin.

Viimeiseksi DCT-histogrammin menetelmän osoitettiin olevan varma tapa tunnistaa JPEG-uudelleenpakkaus. Menetelmä epäonnistui säännönmukaisesti testikuvien uudelleenpakkauksen havaitsemisessa vain, mikäli kuva oli uudelleenpakattu erittäin korkealla laatuparametrilla tai kuvaan oli lisätty valkoista kohinaa. Yhdessäkään vain kertaalleen pakatusta testikuvasta ei löytynyt merkkejä uudelleenpakkauksesta. Seuraavaksi menetelmään voisi toteuttaa automaattisen luokittimen, sekä lohkoittain toimivan luokituksen muiden toteutettujen menetelmien tapaan.

Toteutettu sovellus tekee uusien menetelmien lisäämisestä valmiiseen toteutukseen yksinkertaista. Toteutus tarjoaa sovelluskehittäjälle helppokäyttöisen, graafisen käyttöliittymän menetelmän testausta varten, sekä mahdollisuuden tarkastella visuaalisesti menetelmän antamia tuloksia. Sovellus on tehty mahdollisimman alustariippumattomaksi ja kaikki sovelluksen käyttämät kirjastot ovat saatavilla Linux-, Windows- ja Mac OS X -käyttöjärjestelmille. Ennen sovelluksen laajempaa käyttöönottoa tulisi kuitenkin kiinnittää huomiota nyt toteutettujen menetelmien laskennalliseen tehokkuuteen, sekä sisäisen laskennan rinnakkaistamiseen. Tulevaisuudessa olisi mielenkiintoista toteuttaa ensin menetelmät väri-interpoloinnin epäyhdenmukaisuuden ja kopioidun alueen havaitsemiseen. Tämän jälkeen sovelluksen käyttöliittymää voisi kehittää ammattisovellukseen sopivaksi.

LÄHTEET

- [1] Bazhyna, A. Image Compression in Digital Cameras. Dissertation. Tampere 2008. Tampereen teknillinen yliopisto. Julkaisu – Tampere University of Technology. Publication 797. 91 p.
- [2] Trussel, H. & Vrhel, M. Fundamentals of Digital Imaging. Cambridge University Press 2008. 532 p.
- [3] Pat. US 3971065. Color imaging array. Bayer, B. 1976.
- [4] Pat. US 5965875. Color Separation in an Active Pixel Cell Imaging Array Using a Triple-Well Structure. Foveon Inc. (Merrill R.). Appl. No.:09/065,939, 12 October, 1999. 9 p.
- [5] El Gamal, A. Trends in CMOS image sensor technology and design. 2002. International Electron Devices Meeting. IEDM Digest. pp. 805–808.
- [6] Kao, W., Wang, S., Chen, L., & Lin, S.. Design considerations of color image processing pipeline for digital cameras, IEEE Transactions on Consumer Electronics 52(2007)4, pp. 1144–1152.
- [7] Gunturk, B., Glotzbach, J., Altunbasak, Y., Schafer, R. & Mersereau, R. Demosaicking: color plane interpolation in digital cameras, IEEE Signal Processing Magazine, vol. 22, no. 1, 2005, pp. 44–54.
- [8] IEC 61966-2-1 (1999-10). Multimedia systems and equipment – Color measurement and management – Part 2-1: Colour management – Default RGB colour space – sRGB. Geneva 1999, International Electrotechnical Commission. 51 p.
- [9] Adobe. The Adobe®RGB (1998) Color Image Encoding [WWW]. [viitattu 14.8.2010], 1998, 2005. Saatavissa: <http://www.adobe.com/digitalimag/pdfs/AdobeRGB1998.pdf>.
- [10] Wandell, B. Foundations of Vision. Sunderland, MA, USA 1995. Sinauer Associates. 443 p.
- [11] Theuwissen, A. Image Processing Chain in Digital Still Cameras. Symposium on VLSI Circuits, 2004. Digest of Technical Papers, Honolulu, USA, June, 2004. IEEE Solid-State Society. pp. 2–5.
- [12] Wallace, G. The JPEG still picture compression standard. IEEE Transactions on Consumer Electronics 38(1992)1, pp. 18–34.

- [13] CIPA DC-008-2010 / JEITA CP-3451B. Exchangeable image file format for digital for digital still cameras: Exif Version 2.3. Tokyo 2010, Camera & Imaging Products Association (CIPA). 190 p.
- [14] ISO 12234-2:2001. Electronic still-picture imaging -- Removable memory -- Part 2: TIFF/EP image data format. 2001, International Organization for Standardization. 60 p.
- [15] Adobe. Tiff: Specification for revision 6.0 [WWW]. [viitattu 14.8.2010]. 1992. Saatavissa: <http://partners.adobe.com/public/developer/en/tiff/TIFF6.pdf>.
- [16] Taubman, D. & Marcillin, M. JPEG2000: Image compression fundamentals, standards and practice. Boston 2002, Kluwer. 800 p.
- [17] Ahmed, N., Natarajan, T. & Rao, K. Discrete Cosine Transform. IEEE Transactions on Computers C-23(1974)1, pp. 90–93.
- [18] Gonzales R. & Woods R. Digital Image processing. Second edition. New Jersey 2001, Prentice Hall Inc. 793 p.
- [19] Popescu, A. Statistical Tools for Digital Image Forensics. Dissertation. Hanover, New Hampshire, USA 2005. Dartmouth College, Technical Report TR2005-531. 10 p. + appendix 1 p.
- [20] Huffman, D. A Method for the Construction of Minimum-Redundancy Codes. Proceedings of the IRE 40(1952)9, pp. 1098–1102.
- [21] Farid, H. Creating and detecting doctored and virtual images: Implications to the child pornography prevention act. 2004. Dartmouth College, Technical Report TR2004-518. 13 p.
- [22] Agence France-Presse [WWW]. [viitattu 6.10.2010]. Saatavissa: <http://www.afp.com/>.
- [23] Sepah News [WWW]. [viitattu 6.10.2010]. Saatavissa: <http://www.sepahnews.com/>.
- [24] Nizza, M. & Lyons, P. In an Iranian Image, a Missile Too Many [WWW]. Washington. New York Times: The Lede. July 10, 2008, 9:16 am. [viitattu 3.9.2010]. Saatavissa: <http://thelede.blogs.nytimes.com/2008/07/10/in-an-iranian-image-a-missile-too-many/?hp>.
- [25] Huang, H., Guo, W. & Zhang, Y. Detection of Copy-Move Forgery in Digital Images Using SIFT Algorithm. IEEE Pacific-Asia Workshop on Computational Intelligence and Industrial Application, Wuhan, December 19-20, 2008. pp. 272–276.

- [26] Farid, H. & Bravo, M. Photorealistic Rendering: How Realistic Is It? *Journal of Vision* [verkkolehti]. 7(2007)9 [viitattu 6.10.2010]. Saatavissa: <http://www.journalofvision.org>.
- [27] Hsu, Y. & Chang, S. Image Splicing Detection using Camera Response Function Consistency and Automatic Segmentation. *IEEE International Conference on Multimedia and Expo, Beijing 2007*, pp. 28–31.
- [28] Fang, Z., Wang, S. & Zhang, X. Image Splicing Detection Using Camera Characteristic Inconsistency. *International Conference on Multimedia Information Networking and Security, 2009, MINES '09. Hubei, 31 December 2009*, pp. 20–24.
- [29] Johnson, M. & Farid, H. Exposing Digital Forgeries in Complex Lightning Environments, *IEEE Transactions on Information Forensics and Security* 2(2007)3, pp. 450–461.
- [30] Johnson, M. & Farid, H. Exposing Digital Forgeries Through Chromatic Aberration. *ACM Multimedia and Security Workshop, Geneva, Switzerland, September 26–27, 2006. ACM*. pp. 48–55.
- [31] Young, H. & Freedman, R. *University physics with modern physics*. 11th edition. San Francisco 2004, Pearson Education. 1714 p.
- [32] Popescu, A. & Farid, H. Exposing digital forgeries in color filter array interpolated images. *IEEE Transactions on Signal Processing*, 53(2005)10, pp. 3948–3959.
- [33] Dempster, A., Laird, N. & Rubin, D. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)* 99(1977)1, pp. 1–38.
- [34] Lukáš, J., Fridrich J. & Goljan, M. Determining Digital Image Origin Using Sensor Imperfections. Binghamton 2005. Department of Electrical and Computer Engineering SUNY Binghamton. 12 p.
- [35] Lukáš, J., Fridrich, J. & Goljan, M. Detecting Digital Image Forgeries Using Sensor Pattern Noise. Binghamton 2006. Department of Electrical and Computer Engineering SUNY. 11 p.
- [36] Chen, M., Fridrich, J., Goljan, M. & Lukáš, J. Determining Image Origin and Integrity Using Sensor Noise. *IEEE Transactions on Information Forensics and Security* 3(2008)1, pp. 74–90.
- [37] Li, C. Source Camera Identification Using Enhanced Sensor Pattern Noise. *IEEE Transactions on Information Forensics and Security* 5(2010)2, pp. 280–287.

- [38] Holst, G. *CCD Arrays, Cameras, and Displays*. 2nd edition. Bellingham, USA, 1998. SPIE Optical Engineering in JCD Publishing. 378 p.
- [39] Healey, G. & Kondepudy, R. Radiometric CCD Camera Calibration and Noise Estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 16(1994)3, pp. 267–276.
- [40] Amerini, I., Caldelli, R., Cappellini, V., Picchioni, F. & Piva, A. Analysis of denoising filters for photo response non uniformity noise extraction in source camera identification. 16th International Conference on Digital Signal Processing, Santorini-Hellas, 18 August 2009. pp. 1–7.
- [41] Mihcak, K., Kozintsev, I. & Ramchandran, K. Spatially adaptive statistical modeling of wavelet image coefficients and its application to denoising. *IEEE International Conference on Acoustics, Speech, and Signal Processing*. Phoenix, Arizona, USA, 15-19 March, 1999. pp. 3253–3256.
- [42] Lowe, D. Object recognition from local scale-invariant features. *The Proceedings of the Seventh IEEE International Conference on Computer, Kerkyra, Greece*, 6 August, 2002. pp. 1150–1157.
- [43] Popescu, A. & Farid, H. Exposing Digital Forgeries by Detecting Duplicated Image Regions. Hanover 2004, Dartmouth College, Computer Science, Technical Report TR2004-515. 131 p.
- [44] Lowe, D. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision* 60(2004)2. pp. 91–110.
- [45] Beis, J. & Lowe, D. Shape Indexing Using Approximate Nearest-Neighbour Search in High Dimensional Spaces. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Puerto Rico, 17–19 June, 1997. pp. 1000-1006.
- [46] Qt – A cross-platform application and UI framework [WWW]. [viitattu 6.10.2010]. Saatavissa: <http://qt.nokia.com/>.
- [47] OpenCV Wiki [WWW]. [viitattu 6.10.2010]. Saatavissa: <http://opencv.willowgarage.com/wiki/>.
- [48] Toteutuksen lähdekoodi [ZIP-tiedosto]. [viitattu 6.10.2010]. Saatavissa: <http://ubuntuone.com/p/f8c/>.
- [49] Farid, H. & Simoncelli, E. Differentiation of discrete multidimensional signals. *IEEE Transactions on Image Processing* 13(2004)4, pp. 496–508.
- [50] Lim, J. *Two-Dimensional Signal and Image Processing*. New Jersey 1990, Prentice Hall. 694 p.

- [51] MATLAB R2010b Documentation. 2-D adaptive noise-removal filtering [WWW]. [viitattu 6.10.2010]. Saatavissa: <http://www.mathworks.com/help/toolbox/images/ref/wiener2.html>.
- [52] Mahdian, B. & Saic, S. Detecting double compressed JPEG images. 3rd International Conference on Crime Detection and Prevention, London, Great Britain, 3 December 2009. pp. 1–6.
- [53] Photoshop [WWW]. [viitattu 7.10.2010]. Saatavissa: <http://www.photoshop.com/>.
- [54] GIMP – The GNU Image Manipulation Program [WWW]. [viitattu 7.10.2010]. Saatavissa: <http://www.gimp.org/downloads/>.