2013/837

ISSN 1797-3457 (verkkojulkaisu)
ISBN (PDF) 978-951-25-2527-0

TIIVISTELMÄRAPORTTI (SUMMARY REPORT)

Grafiikkaprosessorilaskenta laajojen videoaineistojen käsittelyssä

Antti Lehmussola, Niko Mikkilä
Keskusrikospoliisi, rikostekninen laboratorio
PL 285, 01301 Vantaa

Tiivistelmä

Videotekniikan hyödyntäminen on jatkuvassa kasvussa. Haastavista kuvausolosuhteista johtuen taltioitu kuva-aineisto ei kuitenkaan aina vastaa sille asetettuja odotuksia. Tämä edellyttää tehokkaampien videokäsittelymenetelmien kehittämistä. Tässä hankkeessa tutkittiin superresoluutioalgoritmien suorituskyvyn parantamista näytönohjaimella tehtävän laskennan avulla. Hankkeessa onnistuttiin saavuttamaan viisinkertainen suorituskyvyn parannus valitulle superresoluutioalgoritmille. Saavutetulla tehokkuushyödyllä voidaan joko nopeuttaa videoiden käsittelyaikaa tai vastaavasti kasvattaa laskentatarkkuutta.

1. Johdanto

Digitaalisen videotekniikan kehityksen myötä niin rikostutkinta-, valvonta-, tiedustelu- kuin tarkkailukäytössäkin hyödynnetään yhä laajempia videoaineistoja. Videoaineisto voi olla peräisin useista eri lähteistä, kuten tavallisista kameroista, valvontakameroista tai matkapuhelimista. Myös uudenlaiset, videokuvausta käyttävät laitteet, kuten miehittämättömät tiedustelulennokit muodostavat yhä merkittävämmän lähteen videomateriaalille.

Kattavaa ja laadukasta videoaineistoa voidaan käyttää apuna esimerkiksi rikostutkinnassa tai tilannekuvan luonnissa. Valitettavan usein kuvausolosuhteet ovat kuitenkin huonot, eikä käytettävissä oleva kamera- ja taltiointitekniikka ole riittävää. Kun valoa on vain vähän, kohina peittää yksityiskohtia ja kameran sijainti, heikko resoluutio tai videon häviöllinen pakkaus tekee lopullisesta kuvamateriaalista käyttökelvotonta. Toisin elokuvat ja televisiosarjat antavat ymmärtää, todellisuudessa keinot heikkolaatuisten kuvien tai videon parantamiseen ovat varsin rajalliset. Lupaavimmat menetelmät liittyvät superresoluutioon, jossa yhdistellään informaatiota useista kuvista tai saman kuvan eri kohdista, poistetaan kohinaa ja terävöitetään kuvaa käänteissuodatuksella. Superresoluutio on laskennallisesti monimutkaista ja hyvin raskasta, joten videoaineistojen käsittely tavallisilla tietokoneilla on edelleen haastavaa.

1.1. Kuvanlaadun parantaminen superresoluutiomenetelmillä

Valokuvien ja videon laatuun vaikuttavat useat tekijät kuvausprosessin eri vaiheissa. Optiikan osalta valon kulkeminen ilmakehän läpi johtaa havaittavaan sumentumiseen ja vääristymiseen suurilla kuvausetäisyyksillä; kameran tai kohteen tahdistamattomasta liikkeestä aiheutuu helposti liike-epäterävyyttä, ellei valotusaika ole riittävän lyhyt, ja kameran linssin kompromissit tai väärä tarkennus saattavat sumentaa kuvaa. Jos optiikka on mitoitettu oikein, digitaalikuvauksessa CCD- tai CMOS-kennon kuvapisteiden lukumäärä ratkaisee kaapatun kuvan maksimiresoluution. Linssin ja kennon ominaisuudet, valon määrän ja valotusajan ohella vaikuttavat myös kuvassa näkyvän kohinan määrään. Viimeisenä laatuun vaikuttavana tekijänä videokuvauksessa ja erityisesti laajojen videoaineistojen taltioinnissa kuvaa prosessoidaan digitaalisesti tiiviimpään muotoon: resoluutiota saatetaan pienentää, tarpeettomat kuvajaksot leikataan automaattisesti pois ja video pakataan häviöllisesti. Tällä saavutetaan suuria säästöjä tallennustilassa, mutta laatu heikkenee yleensä huomattavasti alkuperäiseen raakakuvaan verrattuna.

Digitaalisen signaalinkäsittelyn kehittymisen ja laskentatehon eksponentiaalisen kasvun myötä käytettävissä on useita laskennallisia menetelmiä kuvanlaadun parantamiseksi: ilmakehän, kameran optiikan ja liikkeen aiheuttamaa epäterävyyttä voidaan poistaa käänteiskonvoluutiolla, kohinaa voidaan vaimentaa laskemalla vastaavien kuvapisteiden keskiarvoja tai mediaaneja useista kuvaotoksista ja kuvan erottelua voidaan parantaa kohdistamalla kuvien osia toisiinsa alipikselitasolla, eli yhtä kuvapistettä tarkemmin. Superresoluutio käsitetään usein kuvanlaadun parantamiseksi juuri alipikselitason toimenpiteillä, mutta myös muut edellä mainitut menetelmät parantavat kuvaa niin, että siitä on erotettavissa pienempiä yksityiskohtia. Niiden tarkkuus vain rajautuu alkuperäisen kuvan kuvapisteiden kokoon. Kirjallisuuden superresoluutiomenetelmissä hyödynnetään useimmiten kaikkia näitä menetelmiä, edeten kuvien alipikselitason yhteensovittamisesta kohinanpoistoon ja käänteiskonvoluutioon.

1.2. Grafiikkaprosessorilaskenta

Grafiikkaprosessori tai näytönohjain on tietokoneiden ja nykyisin esimerkiksi älypuhelinten komponentti, joka on erikoistunut näytöllä esitettävän grafiikan muodostamiseen. Erityisesti reaaliaikainen kolmiulotteinen grafiikka edellyttää suurten tietomäärien nopeaa käsittelyä vektori- ja matriisilaskutoimituksilla. Monet tietokonegrafiikkaan liittyvät laskentatehtävät ovat jaettavissa tuhansiin tai jopa miljooniin rinnakkaisiin, toisistaan riippumattomiin osiin. Perinteisen sarjamuotoisen, yhdellä prosessoriytimellä tehtävän laskennan sijaan näitä osatehtäviä voidaan ratkaista samanaikaisesti sadoilla grafiikkaprosessorin ytimillä. Vaikka grafiikkaprosessorin kukin ohjelmaa suorittava ydin on huomattavasti hitaampi ja yksinkertaisempi kuin pääprosessorin ydin, näiden ydinten kymmen- tai satakertaisella määrällä saavutetaan moninkertainen laskentanopeus.

Grafiikkaprosessorit ja niiden ohjelmointirajapinnat saavuttivat vuonna 2001 kehityksen pisteen, jossa niitä pystyttiin hyödyntämään 3D-grafiikan piirtämisen lisäksi myös muuhun laskentaan. Valmiiden geometriaan ja pintakuvioihin liittyvien funktioiden sijaan ohjelmoija pystyi kokoamaan suoritettavan ohjelman joukosta peruskomentoja ja laskutoimituksia samaan tapaan kuin perinteisissä prosessoriarkkitehtuureissa. Tämäkin ominaisuus oli alunperin suunnattu grafiikkalaskentaan, mutta ohjelmoitavuuden kautta grafiikkaprosessorien suurta laskentakapasiteettia alettiin pian hyödyntää kaikenlaisessa vaativassa laskennassa. Helpottaakseen monimutkaista grafiikkarajapintoihin kahlittua ohjelmointia, näytönohjainvalmistaja Nvidia julkaisi vuonna 2007 Cuda-ohjelmointialustan omille grafiikkaprosessoreilleen. Avoimempi ja useiden eri valmistajien hyväksymä OpenCL-standardi valmistui vuoden 2008 lopussa. Tällä hetkellä molempia alustoja tai ohjelmistokehyksiä käytetään laajasti eri alojen erikoisohjelmissa ja tieteellisessä laskennassa. Esimerkiksi maailman kaksi tehokkainta supertietokonetta, Kiinan Tianhe-2 ja Yhdysvaltojen Titan perustuvat kymmeniin tuhansiin Intel Xeon Phi ja Nvidia Tesla -rinnakkaisprosessorikortteihin. Tesla on arkkitehtuuriltaan samanlainen kuin Nvidian näytönohjaimet.

2. Tutkimuksen tavoite ja suunnitelma

Tutkimuksen päätavoite oli selvittää grafiikkaprosessorilaskennan soveltuvuutta vaativaan videoaineistojen käsittelyyn ja erityisesti kuvasarjojen superresoluutiorekonstruktioon. Tutkimuksen aikana oli tarkoitus toteuttaa prototyyppinä tarkka ja nopea superresoluutioohjelmisto, jossa on ainakin yhden superresoluutiomenetelmän toteutukset sekä perinteisille prosessoriarkkitehtuureille että grafiikkaprosessoreille OpenCL-kehitysalustan kautta. Tutkimus jaettiin seuraaviin osatavoitteisiin:

Arvioidaan videoaineiston käsittelyyn soveltuvia superresoluutiomenetelmiä tieteellisen kirjallisuuden ja saatavissa olevien toteutusten perusteella. Valitaan toteutettavaksi grafiikkaprosessorilaskennan ja saavutettavan kuvanlaadun kannalta hyvältä vaikuttava menetelmä.

Tutkitaan erilaisten liike-estimointimenetelmien vaikutusta superresoluutiorekonstruktioon. Yritetään saada kokeiltavaksi joitakin tarkimmista julkaistuista menetelmistä ja verrataan tuloksia esimerkiksi OpenCV-konenäkökirjaston Lucas-Kanade-menetelmällä laskettuihin liikevektoreihin.

Kuvataan testiaineistoksi monenlaista videota vaihdellen muun muassa kuvauskohteita, kameraa, etäisyyttä ja päivän- ja vuodenaikaa.

Toteutetaan valittu superresoluutiomenetelmä C-kielellä ja verrataan sen toimintaa muihin saatavissa oleviin superresoluutio-ohjelmiin. Jos tulokset vaikuttavat huonoilta, harkitaan vaihtoehtoisia menetelmiä. Säikeistetään C-kielinen versio käyttämään kaikkia pääprosessorin ytimiä ja pyritään optimoimaan muistin käyttöä.

Muokataan C-kielisestä toteutuksesta grafiikkaprosessoriversio OpenCL-kehitysalustan avulla. Kokeillaan ja kehitetään uusia ratkaisuja laskentaongelman rinnakkaistamiseksi.

Verrataan toteutusten nopeutta ja tarkkuutta muihin saatavissa oleviin superresoluutio-ohjelmiin ja perinteisiin interpolointi- ja uudelleennäytteistysmenetelmiin omalla testiaineistollamme.

3. Aineisto ja menetelmät

Tutkimuksessa toteutettavaksi superresoluutiomenetelmäksi valikoitui Fast and Robust Multiframe Super Resolution (Farsiu et al. 2004), jatkossa lyhyemmin FRSR, joka on yksi viitatuimmista superresoluutiota käsittelevistä julkaisuista. Artikkelin kirjoittaneen tutkimusryhmän johtaja Peyman Milanfar perusti MotionDSP-yrityksen vuonna 2005, jonka kehittämä Ikena-ohjelmistoperhe on tällä hetkellä markkinoiden johtava superresoluutioratkaisu. MotionDSP:n Ikena-videoforensiikkaohjelmaa käytettiin vertailukohteena tutkimuksen aikana.

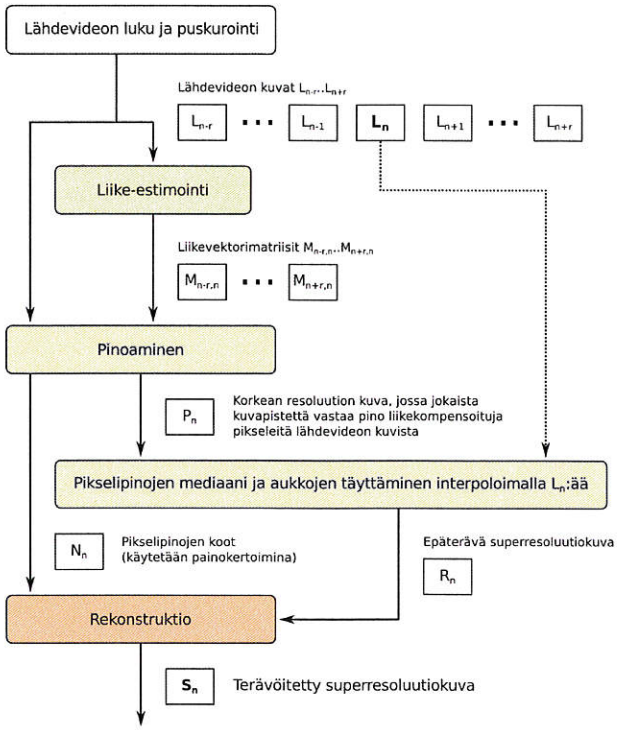
FRSR kuuluu multiframe-superresoluutiomenetelmiin, eli algoritmi saa syötteenä joukon kuvia, esimerkiksi videon 15 peräkkäistä kuvaa, sekä kaikkien kuvapisteiden suhteelliset siirtymät (liikevektorit) kuvien välillä. Varsinaista superresoluutiota varten siirtymät on annettava alipikselitarkkuudella. Jos tämän liike-estimoinnin tarkkuus on vain pikselitasoa, algoritmi toimii liikekompensoituna kohinanpoistajana ja terävöittäjänä. Tässä tutkimuksessa liike-estimointiin käytettiin pääsääntöisesti MDP-Flow2-menetelmää (Xu et al. 2012), joka oli vuoden 2013 alussa Middlebury optical flow evaluation -testin kärkisijoilla. Menetelmän valmis toteutus on myös vapaasti saatavissa Windows-ohjelmana.

Multiframe-superresoluutiolaskennan kaikki vaiheet on esitetty lohkokaaviona kuvassa 1. Syötekuvat luetaan videotiedostosta tai muusta kuvalähteestä ja puskuroidaan tietokoneen keskusmuistiin. Yksi kuvista (L_n) valitaan tarkennettavaksi referenssikuvaksi ja kaikkien muiden kuvien kuvapisteiden liike suhteessa siihen estimoidaan. Saatujen liikevektorimatriisien $M_{n-r,n} .. M_{n+r,n}$ sekä vastaavien alkuperäisten kuvien $L_{n-r} .. L_{n+r}$ perusteella muodostetaan superresoluutiokuvan esivaihe suurempaan kuvapistematriisiin. Tämä kuvapino P_n voi olla esimerkiksi kaksi tai neljä kertaa suurempi kuin L_n vaaka- ja pystysuunnassa. Kaikki syötteenä saadut kuvapisteet sijoitetaan liikevektorien määräämiin kuvapistepinoihin tai joukkioihin P_n :ssä. Jos yhdistämiseen käytetään tavallista keskiarvoa, pinojen sijaan riittää ylläpitää summaa ja lukumäärää. FRSR-menetelmässä suositaan kuitenkin päällekkäin aseteltujen kuvien mediaania, joka on vähemmän herkkä liike-estimoinnin virheille.

Kun kaikki kuvapisteet on sijoiteltu paikoilleen, lasketaan kunkin kuvapistepinon mediaani ja muodostetaan epäterävä superresoluutiokuva R_n sekä normalisointiin käytettävät kertoimet

N_n . Kuvaan R_n jää usein aukkoja, joiden täyttäminen nopeuttaa ja parantaa rekonstruktiota. Täyttämiseen voidaan käyttää alkuperäisen kuvan L_n interpoloitua versiota tai jotain monimutkaisempaa arviointia.

Lopuksi tehdään varsinainen superresoluutiorekonstruktio, jossa kuva R_n terävöitetään painotetulla käänteiskonvoluutiolla. Lähdeartikkeli keskittyy pääasiassa kolmen viimeisen vaiheen ja erityisesti rekonstruktion kuvaamiseen. Tuloksen kannalta olennainen liike-estimointi oletetaan suoritetuksi tarkoitukseen soveltuvalla menetelmällä.



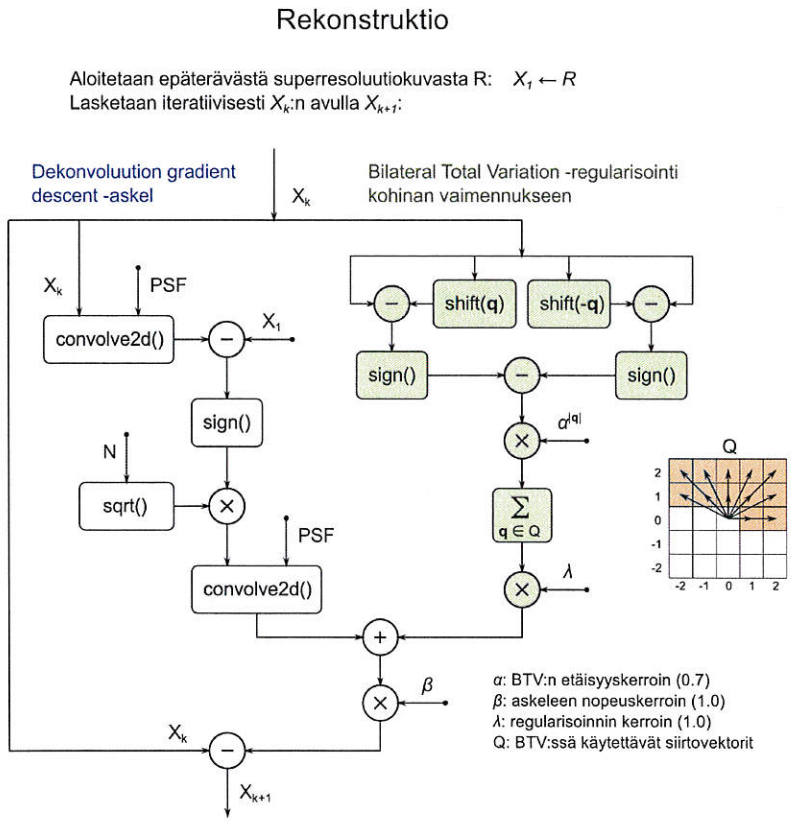
Kuva 1. Videon kuvasarjaa käsittelevän multiframe-superresoluution suoritusvaiheet ja niiden välillä siirrettävä kuvainformaatio

3.2. Superresoluutiorekonstruktio

Rekonstruktiovaiheessa epäterävää superresoluutiokuvaa terävöitetään iteratiivisesti parametrina annetun pisteenleviämisfunktion PSF perusteella. PSF :n avulla voidaan määritellä esimerkiksi optiikan ominaisuuksista johtuva yksittäisen kuvapisteen sumentuminen. Laskenta perustuu käänteiskonvoluutioon, jossa huomioidaan kuvapisteiden painoarvot N , ja jota regularisoidaan Bilateral Total Variation -menetelmällä (BTV). Bilateraalisuodatus on yleinen menetelmä kohinanpoistoon. Se säilyttää kuvassa esiintyvien reunojen ja suurten paikallisten kontrastierojen terävyyden, mutta tasoittaa hienojakoista kohinaa. Käytännössä rekonstruktio voidaan tehdä joissain tapauksissa kokonaan ilman BTV:tä, joka poistaa kuvasta kohinan mukana helposti myös pintojen yksityiskohtia.

Käänteiskonvoluutio-ongelman iteratiivinen ratkaiseminen onnistuu gradient descent -optimointimenetelmällä. Alussa syötteenä on epäterävä superresoluutiokuva X_1 . Tätä kuvaa pyritään muuttamaan vähitellen, iteraatioaskel kerrallaan sellaiseen suuntaan, että

muutettu kuva X_k konvoloituna funktiolla PSF muistuttaisi yhä enemmän kuvaa X_1. Jos tässä onnistuttaisiin täydellisesti jollain k, tiedettäisiin että optimointi on päätynyt globaaliin minimiin ja superresoluutiokuvaa sumentaneen funktion PSF vaikutus olisi onnistuttu poistamaan kokonaan. Todellisella kuvamateriaalilla onnistuminen ei kuitenkaan tarkoita täydellistä superresoluutiorekonstruktiota, koska pisteenleviämisfunktiota ei pystytä määrittämään täydellisesti. Jokainen iteraatioaskel perustuu myös arvaukseen, joka saattaa johtaa väärään suuntaan ja paikalliseen minimiin globaalin minimin sijaan. BTV-regularisoinnilla optimointia pyritään ohjaamaan kohti kohinattomampaa ja terävämpää ratkaisua.



Kuva 2. Superresoluutiorekonstruktio FRSR-menetelmällä

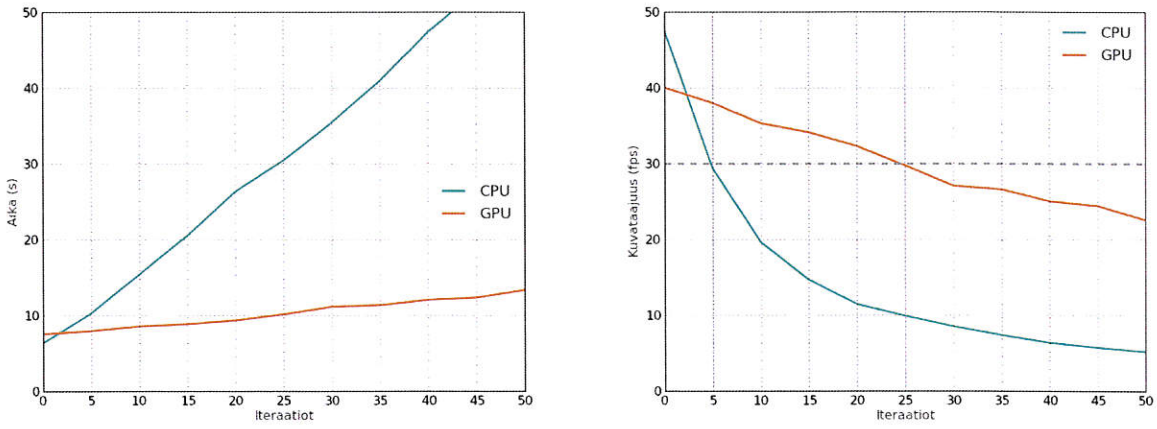
4. Tulokset ja pohdinta

Fast and Robust -superresoluutiomenetelmän C-kielinen toteutus optimoitiin käyttämään syötekuvien kuvapisteiden pinoamiseen ja mediaanin laskemiseen nopeaa kokonaislukuaritmetiikkaa. 8-bittisen tietotyypin käyttäminen säästää huomattavan määrän muistia, koska mediaanin laskentaa varten kaikkien syötekuvien kaikki kuvapisteet on tallennettava yhteen tietorakenteeseen. Saman lähdekuvan osien suhteellisen liikkeen vuoksi on varattava myös reilusti ylimääräistä tilaa. johtuen tarvittava kapasiteetti on superresoluutiokuvan pikselien lukumäärä kerrottuna. Tämän vaiheen grafiikkaprosessorioptimointi ei ehkä ole hyödyllistä, ellei myös liike-estimointia tehdä näytönohjaimella. Muuten liikevektorien siirto näytönohjaimen muistiin hitaan väylän kautta hidastaa koko laskentaa. Lisäksi kuvien yhdistäminen liikevektorien perusteella saattaa olla muistihakujen hajautumisen vuoksi yhtä nopeaa CPU:lla. Mediaanin laskenta sen sijaan olisi huomattavasti nopeampaa näytönohjaimella, mutta tässäkin vaiheessa siirrettävän informaation määrä on niin suuri, ettei operaatiosta ole hyötyä.

Rekonstruktion C-toteutuksessa hyödynnettiin viittausten paikallisuutta (locality of reference), eli nopeita siirtoja peräkkäisistä muistiosoitteista prosessorin välimuistien avustuksella. Esimerkiksi jokaisessa käänteiskonvoluutioaskeleessa joudutaan suorittamaan kaksi tavallista konvoluutio-operaatiota, joiden laskenta vie suurimman osan koko rekonstruktioon käytettävästä ajasta. Konvoluutiolaskennan nopeutta puolestaan rajoittaa pääasiassa muistin lukuoperaatioiden nopeus. Niiden optimointi välimuistiystävälliseksi vaikuttaa huomattavasti koko rekonstruktion nopeuteen. Tässä tutkimuksessa konvoluution pisteenleviämisfunktiona käytettiin tavallista kaksiulotteista gaussin funktiota, joka voidaan hajottaa häviöttömästi vaaka- ja pystysuuntaiseen yksiulotteiseen konvoluutioon. Tällöin tarvittavien laskutoimitusten määrä putoaa alle viidennekseen tavallisella 11x11 kuvapisteen kokoisella gaussin konvoluutiolla. (Tarvittava konvoluutiofunktio on juuri tätä kokoluokkaa, kun kuva suurennetaan nelinkertaiseksi molempien akselien suhteen). BTV-regularisoinnin laskentaan kuluu suunnilleen puolet yhden gaussin konvoluution vaatimasta ajasta.

OpenCL-rekonstruktiossa puskuroitiin neljä peräkkäistä rekonstruoitavaa kuvaa ja lomiteltiin niiden kuvapisteet peräkkäin (k1(1,1), k2(1,1), k3(1,1), k4(1,1), k1(2,1), k2(2,1), ...). Tällöin kaikki neljä kuvaa on mahdollista käsitellä samanaikaisesti näytönohjaimen prosessoriydinten 128-bittisillä SIMD-operaatioilla. Ohjelmassa hyödynnettiin myös OpenCL:n Image-tietotyyppiä, jolloin tekstuurivälimuisti nopeuttaa muistioperaatioita. Grafiikkaprosessorin tapauksessa gaussin funktion hajottaminen kahteen osaan on hieman mutkikkaampaa, eikä sitä voida tehdä käytännössä aivan yhtä tehokkaasti kuin CPU:lla. Optimointia hankaloittaa myös OpenCL-ohjelmien käynnistymisen ja säikeiden synkronoinnin viiveet, jolloin laskentaa ei voi pilkkoa kovin lyhytkestoisiin suorituksiin.

Rekonstruktion gradient descent -minimoinnin etenemisnopeuteen vaikuttavaa beta-parametria päädyttiin pienentämään iteroinnin edetessä kertoimella 0,5..0,9. Iterointi aloitetaan beta-parametrin arvolla 50 (tai 10, jos iteraatiossa suoritetaan vain muutamia askelia) ja päätetään, kun betan arvo on alle yksi. Tällä menetelmällä saavutetaan muutamassa kymmenessä iteraatioaskeleessa laadukkaampi lopputulos kuin sadoilla askelilla ilman etenemisnopeuden muuttamista. C- ja OpenCL-toteutusten suorituskykyä on vertailtu kuvassa 3.

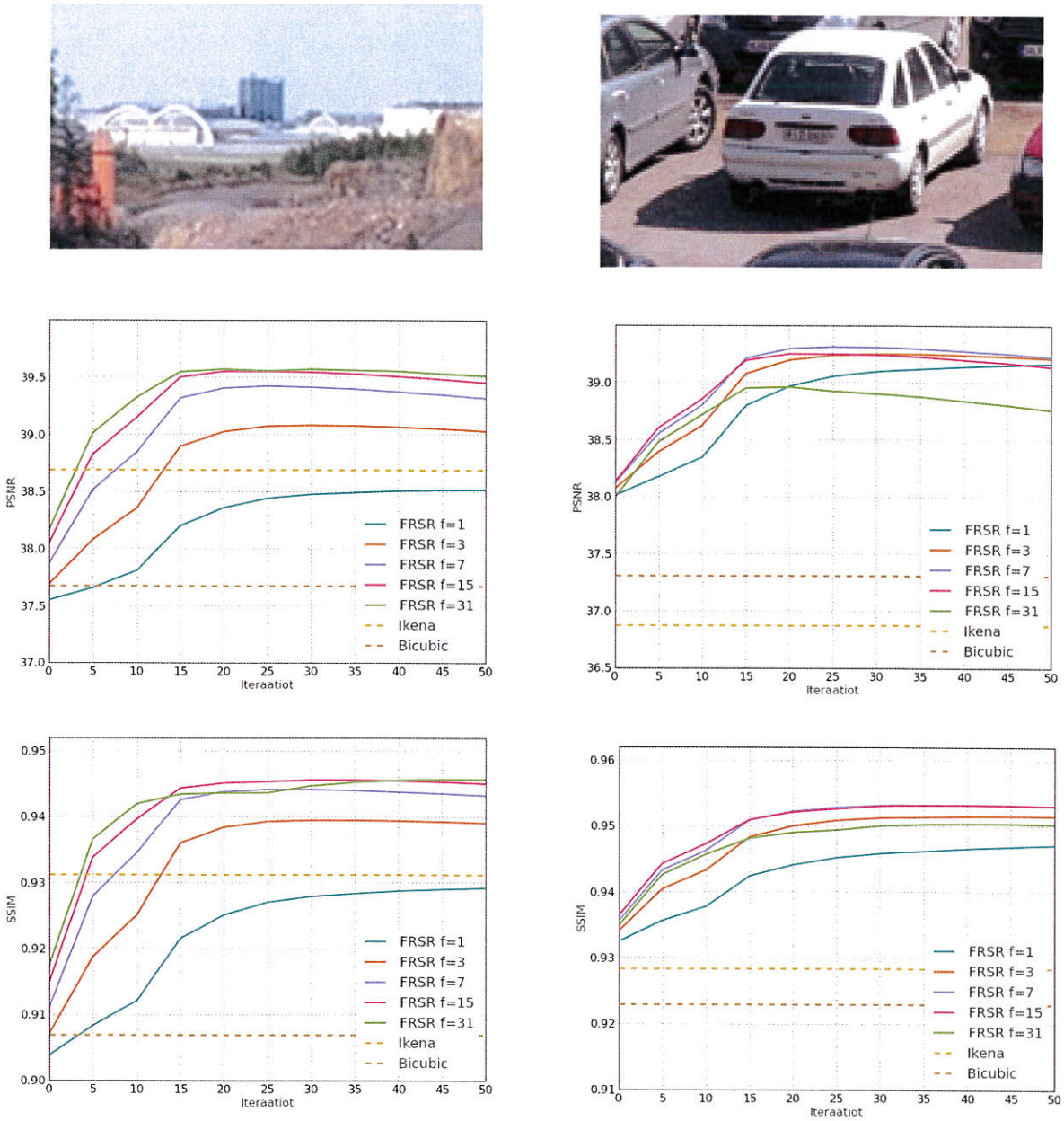


Kuva 3. C- ja OpenCL-toteutusten suoritusaikavertailu (vasen) osoittaa, miten iteraatioiden kasvaessa OpenCL-toteutus muuttuu merkittävästi nopeammaksi. Verrattaessa yhden sekunnin aikana käsiteltävien kuvien määrää (fps) (oikea), nähdään, että reaaliaikaista videota (30 fps) voidaan käsitellä C-toteutuksella ainoastaan viiden iteraation, kun taas OpenCL-toteutuksella 25 iteraatioin verran. Käsitellyn videon resoluutio oli 512x288 pikseliä.

Sekä C- että OpenCL-toteutuksessa kaikki rekonstruktiovaiheen laskenta tehtiin 32-bittisillä liukuluvuilla. Molempia ohjelmia olisi todennäköisesti mahdollista nopeuttaa 50 % - 100 % soveltamalla laskutoimituksiin 16-bittistä fixed-point-kokonaislukuaritmetiikkaa. CPU-laskennassa voisi lisäksi käyttää SSE2 tai AVX2 SIMD-operaatioita, mutta niillä saavutettava nopeushyöty rajoittunee keskusmuistin nopeuteen.

Koko CPU:lla suoritettava ohjelma säikeistettiin OpenMP-rinnakkaislaskentakirjaston avulla siten, että jokaisella prosessorin ytimellä käsitellään samanaikaisesti yhtä superresoluutiokuvaa. Kukin säie voi toimia eri laskentavaiheessa. GPU-toteutuksessa tämä johtaa useiden OpenCL-ohjelmien rinnakkaiseen ajamiseen, mikä peittää hitaista muistisiirroista aiheutuvia viiveitä, mutta kuluttaa myös moninkertaisen määrän näytönohjaimen muistia.

FRSR-menetelmän tarkkuutta mitattiin testivideoiden avulla. Alkuperäiset videot skaalattiin neljä kertaa pienemmiksi, jonka jälkeen niiden resoluutio kasvatettiin takaisin alkuperäisen kokoiseksi. Superresoluutiorekonstruktion ja alkuperäisen videon välillä laskettiin PSNR (peak signal-to-noise ratio)- ja SSIM (structural similarity index)-arvot, joista PSNR kuvaa analyyttisemmin kuvan laatua ja SSIM pyrkii mallintamaan ihmisen kykyä havainnoida kuvan laatua. Verrattaessa tuloksia kaupalliseen Ikena-ohjelmistoon ja tavalliseen kuvan skaalaukseen, voitiin havaita, että riittävällä iteraatioiden määrällä FRSR-menetelmä tuotti selkeästi paremman videon laadun. Kuvassa 4 on esitetty arvojen vertailu kahdelle testivideolle. Lisätuloksia on esitetty sivustolla http://www.forevid.org/matine.



Kuva 4. Kaksi esimerkkivideota (vasen ja oikea sarake), jotka on skaalattu neljä kertaa pienemmäksi, jonka jälkeen niiden resoluutio on kasvatettu vastaamaan alkuperäistä FRSR-menetelmällä (1,3,7,15 ja 31 kuvan ikkunalla), Ikena-ohjelmistolla ja normaalilla bicubic-skaalausmenetelmällä. PSNR- ja SSIM-mittarit kuvaavat, miten hyvin rekonstruoidut videot vastaavat alkuperäisiä. Riittävällä iteraatioiden määrällä FRSR-menetelmä tuottaa selkeästi parempi laatuista kuvaa, kuin verrokki menetelmät. Näytönohjainlaskennalla saavutettava hyöty mahdollistaa, useampien iteraatioiden käytön.

5. Loppupäätelmät

Tutkimushankkeessa voitiin osoittaa, että superresoluutioalgoritmeja voidaan nopeuttaa näytönohjaimella tehtävällä laskennalla. Hankkeen aikana toteutetun algoritmin suorituskyky oli noin viisi kertaa parempi, kuin C-kielisen toteutuksen. Saavutetulla tehokkuushyödyllä voidaan joko nopeuttaa videoiden käsittelyaikaa tai vastaavasti kasvattaa laskentatarkkuutta

Hankkeen tuloksena syntynyt ohjelma pyritään saamaan julkiseen jakeluun Forevid-ohjelmiston tulevissa versioissa ensi vuonna. Toteutus tulee olemaan tarkasti dokumentoitu ja tärkeät kuvanlaatuun vaikuttavat parametrit ovat käyttäjän säädettävissä.

Liike-estimoinnin hitaus ja nykyisten menetelmien heikko tarkkuus asettaa edelleen rajoitteita superresoluution hyödyntämiselle monissa sovelluskohteissa. Rekonstruktioon ja muuhun superresoluutiolaskentaan kuluva aika on häviävän pieni hetki verrattuna esimerkiksi tutkimuksessa käytetyn MDP-Flow2 -toteutuksen käyttämään aikaan. Saatavilla olevien nopeiden menetelmien tarkkuus puolestaan riittää superresoluutioon vain rajoitetuilla lähtöoletuksilla, kuten keskittymällä merkittyjen kuva-alueiden liikkeeseen.

Laadukkaan liike-estimointimenetelmän toteuttaminen kokonaan grafiikkaprosessorilaskentaa hyödyntäen olisi mielenkiintoista sekä superresoluution hyödyntämisen kannalta, että yleisemmin eri videonkäsittelymenetelmien nopeuttamiseksi. Liike-estimointia tarvitaan hyvin monenlaisissa sovelluksissa videoeditoreista konenäköön.

6. Hankkeen seuraajan lausunto raportista

Tutkimusryhmä on onnistunut tehtävässään ja projektissa on saavutettu lupaavia tuloksia. Tutkimuksen tulokset tuovat osaltaan ratkaisuja videomateriaalin käsittelyssä lähes aina törmättävään ongelmaan, eli vaikeuteen pystyä erottamaan riittävän tarkasti yksityiskohtia. On nähtävissä, että tutkimuksen tuloksena syntyvästä osaamisesta sekä teknisestä ratkaisusta tulevat hyötymään useat viranomaiset Suomessa. Näkisin hyväksi tutkimustyön jatkamisen, jotta loppukäyttäjää suoraan hyödyntäviä menetelmiä saataisiin edelleen kehitettyä.

Puolustusvoimien pääesikunnan edustaja