

JPEG2000-standardin hyödyntäminen virtuaalimikroskopiassa

Vilppu Tuominen

Tampereen yliopisto
Tietojenkäsittelytieteiden laitos
Tietojenkäsittelyoppi
Pro gradu -tutkielma
Ohjaajat: Martti Juhola ja
Jorma Isola
Maaliskuu 2008

Tampereen yliopisto

Tietojenkäsittelytieteiden laitos

Tietojenkäsittelyoppi

Vilppu Tuominen: JPEG2000-standardin hyödyntäminen virtuaalimikroskopiassa

Pro gradu -tutkielma, 73 sivua, 18 liitesivua

Maaliskuu 2008

Virtuaalimikroskopian (eli mikroskooppinäytteiden käsittelyn digitaalisessa muodossa) hyödyntäminen kliinisessä lääketieteessä sekä mikroskopian opetuksessa on ollut lähivuosien aikana nopeassa kasvussa. Tästä huolimatta alalle ei ole vielä muodostunut maailmanlaajuisesti hyväksyttyä ja yhtenäistä kuvaformaattia. Lupaava ehdokas tähän on JPEG2000-standardi, joka sisältää monia virtuaalimikroskopian kannalta suotuisia ominaisuuksia. Tämä pro gradu -tutkielma käsittelee JPEG2000-standardin hyödyntämistä virtuaalimikroskopiassa.

Mikroskooppinäytteiden osittaiskuvantamiseen liittyy seuraava merkittävä ongelmakehitys: syntyvien kuva-alkioiden yhdistäminen montaasikuvaksi siten, että alkioiden sauma-alueille ei muodostu näkyviä epäjatkuvuuksia. Tämä ongelma oli ratkaistava ennen mikroskooppinäytteiden rutiininomaista digitointia ja muuntamista JPEG2000-muotoisiksi virtuaalimikroskopiakuviksi (virtuaalinäytelaseiksi). Ongelman ratkaisua varten kehitettiin montaasinrakennusohjelmisto (LargeMontage), jonka yhdistämiseksi osaksi automaattista kuvantamisprosessia kehitettiin yleiskäyttöinen apuohjelmisto (DirObserver). Tämän jälkeen määriteltiin virtuaalimikroskopian kannalta optimaalinen JPEG2000-koodivirran parametrisointi, joka oli edellytys standardin tehokkaaseen hyödyntämiseen. Määrittelyn jälkeen luotiin virtuaalinäytelasien JPEG2000-pakkausohjelmisto (JVScmp), -katseluohjelmisto (JVSview) ja -kuvapalvelinohjelmisto (JVSserv). Ohjelmistojen käyttämä metatieto kuvattiin formaalisti XML-skeemalla (JVSchema). Lopuksi mitattiin JPEG2000-standardin pakkausnopeutta, jonka todettiin olevan riittävän nopea osaksi automaattista näytelasien kuvantamisjärjestelmää. Lisäksi standardissa määritellyn JPIP-tiedonsiirtoprotokollan osoitettiin soveltuvan laajamittaiseen kuvapalvelinkäyttöön.

Tutkielmassa käsitellyistä aiheista on julkaistu yksi tieteellinen artikkeli (Liite 1) ja yksi kokousabstrakti (Liite 2). Lisäksi kaikki tutkielmassa kuvatut ohjelmistot on julkaistu vapaaseen levitykseen tutkimuksen yhteyteen perustetulla web-sivustolla (<http://jvsmicroscope.uta.fi/>). Sivusto sisältää myös yleisölle avoimen JPIP-kuvapalvelimen, jonka sisältämiä virtuaalinäytelasikokoelmia on mahdollisuus katsella käyttäen JVSview-ohjelmistoa.

Avainsanat ja -sanonnat: virtuaalimikroskopia, digitaalinen patologia, telepatologia, virtuaalinäytelasi, JPEG2000, JPIP.

Alkusanat

Tämä tutkimus tehtiin Tampereen yliopiston Lääketieteellisen teknologian instituutissa, professori Jorma Isolan Syöpäbiologian tutkimusryhmässä. Haluan lämpimästi kiittää professori Isolaa työn ohjaamisesta ja mahdollisuudesta tehdä täysipäiväistä tutkimusta virtuaalimikroskopian parissa. Lisäksi haluan kiittää Tampereen yliopiston tietojenkäsittelytieteiden laitoksen professori Martti Juholaa sekä Jouni Mykkästä, joita ilman en olisi tätä työtä tekemässä.

Erityiskiitoksen haluan kuitenkin antaa kihlatulleni Elinalle, jonka tuki ja kärsivällisyys ovat suuresti vaikuttaneet päätökseeni jatkaa tutkimustyötä väitöskirjaksi asti.

Sisällys

1.	Johdanto.....	1
2.	Digitaalikuventamisesta.....	5
2.1.	Digitaalikuvan muodostus ja rakenne.....	5
2.2.	Kuvanpakkaus.....	8
3.	Virtuaalimikroskopiasta.....	11
3.1.	Näytteiden kuvantaminen.....	11
3.2.	Virtuaalinäytelasien katselu.....	13
3.3.	Virtuaalimikroskopian sovelluskohteet.....	14
3.4.	Kohti yhtenäistä kuvaformaattia.....	16
4.	JPEG2000-standardi.....	17
4.1.	Standardiperheen osat.....	18
4.2.	JP2-tiedostoformaatti.....	19
4.3.	Hyödyt virtuaalimikroskopian näkökulmasta.....	20
4.4.	JPEG2000-pakkausalgoritmin kuvaus.....	22
4.4.1.	Esikäsittely.....	23
4.4.2.	Pakkaus.....	27
4.4.3.	Koodivirran muodostus.....	32
4.5.	JPIP-tietoliikenneprotokolla.....	34
5.	Tavoitteet.....	37
6.	Menetelmät.....	38
6.1.	LargeMontage-ohjelmiston kehitys.....	38
6.2.	DirObserver-ohjelmiston kehitys.....	38
6.3.	Virtuaalinäytelasien kuvantaminen testauskäyttöön.....	38
6.4.	JPEG2000-koodivirran parametrien optimointi.....	39
6.5.	JVS-ohjelmistopakettin kehitys.....	39
6.6.	JPEG2000-suorituskykytestaus.....	39
7.	Tulokset.....	42
7.1.	LargeMontage-ohjelmisto.....	42
7.2.	DirObserver-ohjelmisto.....	46
7.3.	Optimaalinen JPEG2000-koodivirran parametrisointi.....	48
7.4.	JVS-ohjelmistopaketti.....	49
7.5.	JPEG2000-pakkauksen suorituskyky.....	55
7.6.	JPIP-kuvapalvelimen suorituskyky.....	56
8.	Pohdinta.....	57
8.1.	JPEG2000 – yhtenäinen kuvaformaatti virtuaalimikroskopiaan.....	57
8.2.	Malli JPEG2000-pohjaiselle virtuaalimikroskopiajärjestelmälle.....	60
8.3.	Jatkotutkimusten aiheita.....	61
9.	Yhteenvedo.....	65
	Viiteluettelo.....	66
	Liitteet	

1. Johdanto

Kuluneen kahden vuosikymmenen yleinen digitalisoituminen on vauhdittanut valomikroskopian kehitystä. Satoja vuosia teknisesti hyvin samankaltaisena säilynyt mikroskooppi on kokenut vallankumouksen näytteiden digitointimahdollisuuden myötä. Digitalisoitumisen alkuvaiheessa ilmestyivät videokuvaa hyödyntävät mikroskoopit, joiden tuottamasta kuvasignaalista voitiin kaapata yksittäisiä – ja yleensä heikkolaatuisia – kuvia tietokoneen näytölle katseltavaksi. Samaan aikaan käytettiin myös filmipohjaisia mikroskooppikameroita, joiden tuottamat analogiset kuvat voitiin muuntaa digitaaliseen muotoon filmidigitointilaitteiston avulla. Pian näiden jälkeen yleistyivät nykyisen kaltaiset digitaalikamerat, joiden avulla mikroskooppinäytteestä voidaan poimia haluttuja näkymiä digitaaliseen muotoon huomattavasti helpommin, ilman hidasta digitointivälvaihetta.

Digitaalikameroiden alati kasvava optinen erottelukyky mahdollistaa näytteiden entistä tarkemman kuvantamisen. Valomikroskopian kannalta nykyisten digitaalikameroiden erottelukyky on kuitenkin jo sellaisella tasolla, jota parantamalla voidaan saavuttaa kuvanlaadullisesti enää vain pieniä käytännön etuja. Tästä johtuen nykyinen kehitys onkin keskittynyt näytteiden koko pinta-alan digitointiin sekä kuvantamisnopeuden kasvattamiseen. Perinteisten, manuaalisten pöytämikroskooppien rinnalla kehityksen huippua edustavat usean sadan näytelasin vuorokausinopeuteen kykenevät automaattiset skannauslaitteistot (Kuva 1).



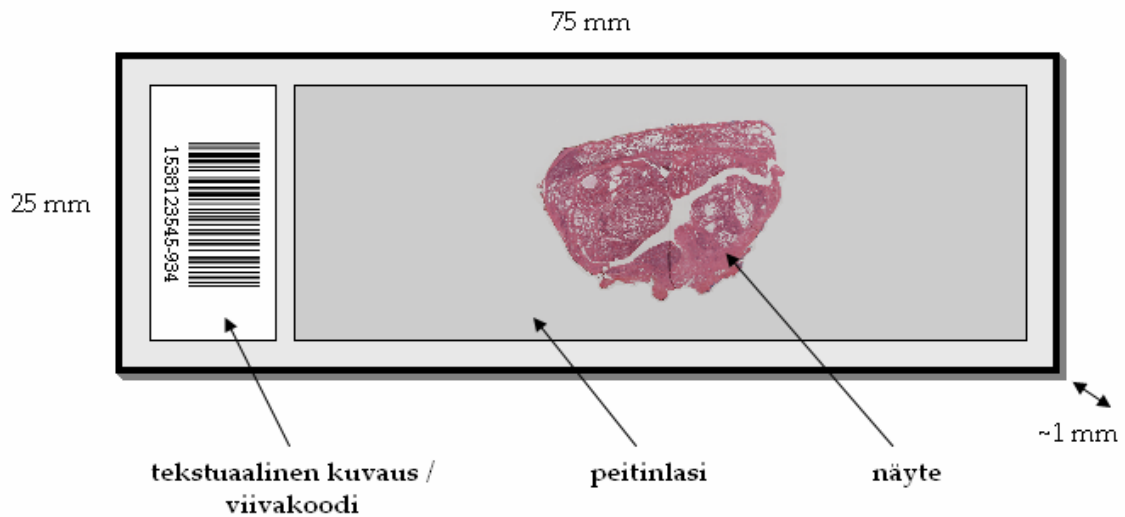
(a)



(b)

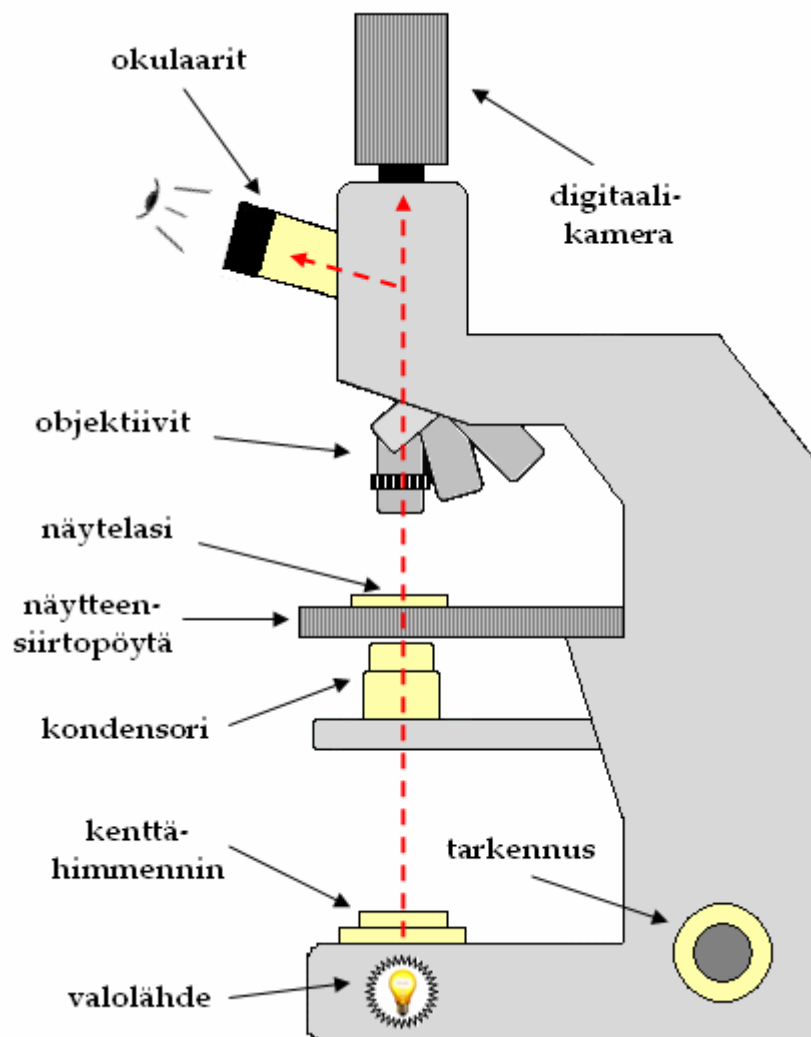
Kuva 1. Manuaalinen pöytämikroskooppi (a) (Axioskop, Carl Zeiss MicroImaging, USA); ja automaattinen näytelasiskanneri (b) (ScanScope® XT, Aperio Technologies, USA).

Valomikroskopia voidaan jakaa lukuisiin alahaaroihin, kuten fluoresenssi- ja konfokaalimikroskopiaan. Tämän tutkielman yhteydessä keskitytään lähinnä haaroista yleisimpään, kirkaskenttämikroskopiaan (bright field microscopy). Kirkaskenttämikroskopiassa näyte altistetaan valkoiselle valolle, joka läpäistessään näytteen muodostaa tarkasteltavan näkymän. Mikroskooppinäyte on tyyppillisesti ohut kudosleike (paksuus 4 μm), joka on asetettu läpinäkyvän lasilevyn päälle (Kuva 2). Pohjalasin ja näytteen päälle kiinnitetään ohut peitinlasilevy, joka suojaa näytettä ja luo tasaisen pinnan. Näytelasin kantaan kirjataan identifioiva tunnistus, yleensä joko tekstuaalisesti tai viivakoodilla. Standardoituja näytelasikokoja on muutamia, joista yleisin on 75 \times 25 mm (3" \times 1"), jonka korkeus on tyyppillisesti noin yhden millimetrin.



Kuva 2. Standardikokoinen (3" \times 1") mikroskooppinäytelasi.

Tavanomaisen, alapuolelta valaistun kirkaskenttämikroskoopin rakenne on esitetty Kuvassa 3. Valolähteenä käytetään useimmiten halogeenilamppua, jonka ylöspäin säteilemän valon määrää kontrolloidaan kenttähimmentimellä. Kondensori kokoaa saapuvan valon pienelle pinta-alalle, joka kohdennetaan näytteensiirtopöydän päällä sijaitsevaan näytelasiin. Näytteensiirtopöytää liikutetaan X- ja Y-suunnassa joko manuaalisesti tai moottoreiden avulla, joita kontrolloidaan erillisellä ohjaussauvalla. Tarkennus, joka tavanomaisesti vastaa näytteensiirtopöydän Z-suuntaista liikettä, voidaan suorittaa automaattisesti kuvantamisohjelmiston avustamana tai manuaalisesti mikroskoopin kyljessä olevalla säätöruuvilla. Mikroskoopin käytettävissä olevat objektiivit (useimmiten 4 – 6 kappaletta) on kerätty renkaaseen, jota pyörittämällä voidaan nopeasti vaihtaa käytettyä suurennusta. Näytelasin näkymä suurennetaan käytetyllä objektiivilla, josta se johdetaan joko silmillä katseltaviin okulaareihin tai digitaalikameran valokennoihin.



Kuva 3. Tavanomaisen, alapuolelta valaistun kirkaskenttämikroskoopin rakenne.

Riippumatta tarkasteltavasta mikroskooppinäytteen tyypistä (esimerkiksi kudosisäily- tai solunäyte), näytteen käsittelyä digitaalisessa muodossa voidaan yleisesti kutsua virtuaaliseksi mikroskopiaksi, eli **virtuaalimikroskopiaksi** (virtual microscopy) [Weinstein, 2005]. Vastaavasti fyysisen mikroskooppinäytelasin digitaalista vastinetta (kuvatiedostoa) kutsutaan **virtuaalinäytelasiksi** (virtual slide). Vaikka virtuaalimikroskopia-termi kattaakin periaatteessa kaikki mikroskopian haarat, lääketieteen kirjallisuudessa se on suhteellisen vakiintunut tarkoittamaan patologian, sytologian ja hematologian yhteydessä tapahtuvaa biologisten näytteiden mikroskopointia. Myös termejä digitaalinen patologia sekä telepatologia käytetään kirjallisuudessa monin paikoin synonyymeinä virtuaalimikroskopialle. Terminologiasta huolimatta samat mikroskopian peruseräkkeet ja -ongelmat esiintyvät kaikilla sen osa-alueilla ja sovelluskohteilla, ja monia menetelmiä onkin mahdollista käyttää yhdenmukaisesti riippumatta

tarkasteltavan näytteen tyypistä. Tämän tutkielman yhteydessä virtuaalimikroskopia-termillä viitataan pääosin patologia-perustaiseen mikroskopiaan.

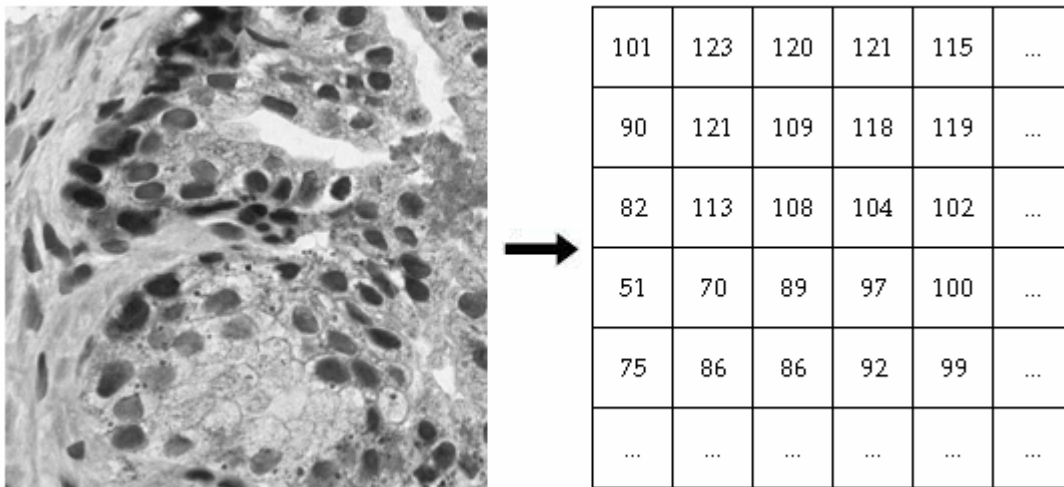
Mikroskooppinäytteen koko pinta-alan digitoiminen riittäväällä tarkkuudella tuottaa poikkeuksellisen suuren määrän informaatiota (usein jopa satoja gigatavuja näytettä kohden). Tästä johtuen virtuaalimikroskopian käytännön soveltamiseen vaaditaan teknisiä erityisratkaisuja, jotka poikkeavat huomattavasti menetelmistä, joita käytetään muilla lääketieteellisen kuvantamisen alueilla (esimerkiksi radiologia). Näytteen kuvantaminen, käsittely, automaattinen analysointi, jakelu käyttäjille sekä arkistointi asettavat kaikki oman erityishaasteensa, joihin tässä pro gradu -tutkielmassa (jatkossa ”tutkielma”) pyritään löytämään ratkaisuja. Tutkimustyö on tehty professori Jorma Isolan Syöpäbiologian tutkimusryhmässä (Lääketieteellisen teknologian instituutti, Tampereen yliopisto), johon liityin vuonna 2005. Käytetyt tutkimusmenetelmät ovat luonteeltaan konstruktivisia ja käytännönläheisiä, perustuen ohjelmistojen kehittämiseen. Tutkielman neljä ensimmäistä lukua ovat johdannollisia, joissa lukija perehdytetään tutkielman kannalta olennaisiin digitaalikuvantamisen käsitteisiin, esitellään virtuaalimikroskopian sovelluskohteita ja ongelmia, sekä esitellään JPEG2000-standardi, joka soveltuu hyvin virtuaalimikroskopian erityishaasteisiin. Viidennessä luvussa eritellään tutkimuksen tarkat tavoitteet. Kuudes luku käsittelee käytettyjä tutkimusmenetelmiä, jonka jälkeen esitetään saavutetut tulokset ja kuvataan tuotetut ohjelmistot. Lopuksi pohditaan tuloksien vaikutuksia ja käsitellään jatkotutkimusten mahdollisuuksia.

Osa tutkielmassa kuvatuista tuloksista on julkaistu tieteellisenä artikkelina Journal of Digital Imaging -lehdessä (Liite 1). Kyseistä artikkelia, tätä pro gradu -tutkielmaa sekä kaikkia näissä kuvattuja ohjelmistoja tullaan käyttämään osana tulevaa väitöskirjaani.

2. Digitaalikuventamisesta

2.1. Digitaalikuvan muodostus ja rakenne

Digitaalikuventamisessa jatkuva, analoginen kuvasignaali muunnetaan diskreetiksi, digitaalseksi signaaliksi käyttäen digitointilaitteistoa, esimerkiksi digitaalikameraa (Kuva 4). Digitaalikamerassa muuntaminen tapahtuu tyypillisesti käyttämällä CCD-kennoa (Charge-Coupled Device), joka mittaa sen antureihin saapuvaa valoenergian määrää halutulla näytteenottotaajuudella. Mitattu valoenergian määrä kvantisoidaan kokonaisluvuiksi (toisinaan myös liukuluvuiksi) käyttäen A/D-muunninta (analogi-digitaali), joka on tyypillisesti sisäänrakennettuna kameran laitteistoon. Lopputuloksena syntyy digitaalikuva I , joka on kaksiulotteinen $M \times N$ -matriisi, jonka alkioina ovat kuvapisteen eli pikselin. Pikseliin viitataan koordinaatistopisteellä (x,y) , missä parametri x vastaa matriisin riviä ja parametri y vastaa matriisin saraketta. Kuvan ensimmäistä pikseliä vastaava koordinaatiston origo ($x = 0, y = 0$) sijaitsee tavanomaisesti kuvan vasemmassa yläreunassa, akselien positiivisen suunnan ollessa vastavasti alas ja oikealle.

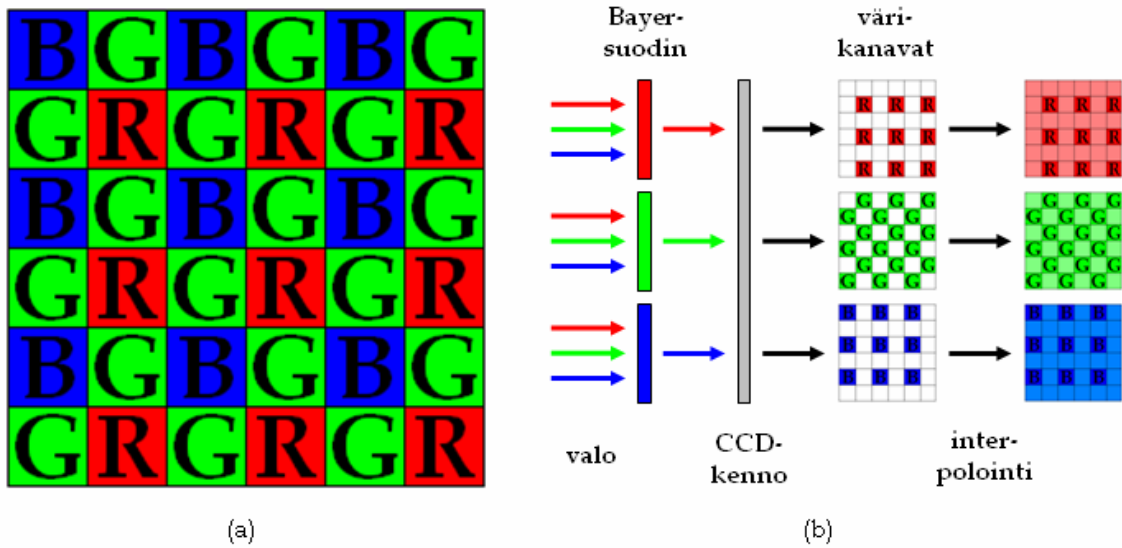


Kuva 4. Jatkuvan, analogisen kuvasignaalin muuntaminen diskreettiin, digitaaliseen muotoon.

Pikselillä ei ole ennalta määrättyä fyysistä muotoa, kokoa eikä väriä. Yleensä pikseli on kuitenkin neliö, eikä kuvan sisällä esiinny kuin yhden kokoisia pikseleitä. Pikselillä on arvo, joka, riippuen digitaalikuvan tyypistä, voidaan tulkita eri tavoin. Luonnollisia kohteita esittävässä kuvissa se ilmaisee kyseisen kohdan valon intensiteetin (kirkkauden). Intensiteettiä esitetään käyttäen **harmaasävykuvaa** (grayscale image), jossa ne ilmaistaan eri harmaan sävyillä,

alkaen mustasta ja päätyen valkoiseen. Käytössä olevan intensiteettiarvojen alueen, eli toisin sanoen toistettavissa olevien kirkkaustasojen lukumäärän, määrää kuvan **bittisyvyys** (bit-depth). Yleisesti bittisyvyydellä tarkoitetaan toistettavissa olevien sävyjen lukumäärää, joista kaikki eivät välttämättä ole harmaan sävyjä. Bittisyvyys ilmaistaan bittien lukumäärällä per pikseli ja se on yhtenäinen koko kuvan alueella. Esimerkiksi 8:n bitin syvyisen kuvan (eli 8-bittisen kuvan) pikseli voi saada arvon $[0...255]$ (2^8). Pikseleiden **dynaamisella alueella** (dynamic range) tarkoitetaan tämän tutkielman yhteydessä myös bittisyvyyttä, vaikka dynaaminen alue mielletäänkin usein vain digitointilaitteiston ominaisuudeksi. Harmaasävykuvan erikoistapaus on **binäärikuva** (bi-level image), joka sisältää vain kahta intensiteettiarvoa (2^1), esimerkiksi mustaa ja valkoista.

Digitaaliset **värikuvat** muodostuvat useista niin kutsutuista **värikanavista**, joista jokainen määrittää eri valon aallonpituuksien intensiteetin kuvan alueella. Näin ollen värikanavan voidaan ajatella vastaavan sisällöltään harmaasävykuvaa. Värikanava on niin kutsutun **(kuva)komponentin** erikoistapaus, jonka sisältöä ei ole rajattu ainoastaan väri-informaatioon. Värikuvaa esitettäessä esimerkiksi tietokoneen näyttölaitteella, lopullisen pikselin väri muodostuu värikanavien intensiteettiarvojen kombinaationa. Tavanomaisin värikuva pohjautuu RGB-värimalliin, jossa kuva koostuu kolmesta 8-bittisestä värikanavasta (punaisesta, vihreästä ja sinisestä), eli kuvan pikseli $I(x,y)$ muodostuu kolmesta intensiteettiarvosta: $I(x_R,y_R)$, $I(x_G,y_G)$ ja $I(x_B,y_B)$. RGB-värikuva muodostetaan CCD-digitaalikameralla useimmiten käyttäen niin kutsuttua Bayer-suodinta [Bayer, 1976]. Siinä digitaalikameran CCD-kennon antureiden päälle asetetaan suodinmatriisi, joka läpäisee valikoidusti valon eri aallonpituuksia (Kuva 5). Suodinmatriisi on kuvioinniltaan 25 % punainen, 50 % vihreä ja 25 % sininen, ottaen näin huomioon ihmissilmän suuremman herkkyyden vihreälle valolle. Lopputuloksena syntyy kolme identtisen kokoista värikanavaa, joiden pikseleistä vain osalla on mitattu intensiteettiarvo. Mittaamattomien naapuripikseleiden intensiteettiarvot interpoloidaan mitatuista pikseleistä.



Kuva 5. Bayer-suotimen kuviointi (a) ja RGB-värikuvan muodostaminen CCD-digitaalikameran ja Bayer-suotimen avulla (b).

Taulukossa 1 on esitetty yleisesti käytössä olevien digitaalikuvien bittisyvyudet sekä niiden esimerkkikäyttökohteet. Yhden bitin syvyyttä käytetään useimmiten mustavalkokuvissa. Tavanomaiset harmaasävykuvat ovat 8-bittisiä, ja tätä korkeampia harmaasävy-bittisyvyyksiä käytetään esimerkiksi radiologian tietokonekerroskuvauksessa (Computed Tomography, CT) ja korkean dynaamiseen alueen kuvantamisessa (High Dynamic Range Imaging, HDRI). Värikuvat ovat useimmiten 24-bittisiä RGB-kuvia, mutta myös esimerkiksi painotuotteissa käytetty 32-bittinen CMYK-kuva on yleinen. Eräänlaisena harmaasävykuvien poikkeuksena ovat niin kutsutut indeksoidut värikuvat. Niissä eri harmaan sävyt vastaavat ennalta määriteltäviä väriarvoja. Väriarvot sijaitsevat hakutaulukossa (lookup table, LUT), johon jokainen kuvan pikseli viittaa. Näin ollen esimerkiksi vain 8:n bitin syvyydellä voidaan saavuttaa hyvinkin luonnollisen näköinen värikuva.

Taulukko 1. Yleisesti käytössä olevat digitaalikuvien bittisyvydet ja niiden esimerkkikäyttökohteet.

Bittisyvyys (bittinä per pikseli)	Sävyjen lukumäärä	Esimerkkikäyttökohte
1	2 (2^1)	Mustavalkokuva
4	16 (2^4)	Indeksoitu värikuva
8	256 (2^8)	Tavanomainen harmaasävykuva
16	65 536 (2^{16})	CT-harmaasävykuva
24	~1,7 miljoonaa (2^{24})	RGB-värikuva
32	~4,3 miljardia (2^{32})	HDRI-harmaasävykuva, CMYK-värikuva

Digitaalikuvien **resoluutiolla** tarkoitetaan yleisesti joko kuvan kokoa suhteessa reaali maailmaan, joka ilmaistaan pikseleinä per mittayksikkö (esimerkiksi 250 pikseliä per μm), tai kuvan kokoa pikseleinä, joka ilmaistaan kuvan leveytenä kerrottuna sen korkeudella (esimerkiksi 100×300 pikseliä). Ellei toisins mainita, tämän tutkielman yhteydessä resoluutiolla viitataan jälkimmäiseen tapaan.

2.2. Kuvanpakkaus

Digitaalikuvien pakkaus voidaan jakaa kahteen osaan: 1) häviöttömään ja 2) häviölliseen pakkaukseen. Häviöttömän pakkauksen tavoitteena on tallentaa alkuperäinen kuva mahdollisimman tiiviisti siten, että alkuperäinen kuva voidaan palauttaa ilman kuvainformaation hävikkiä. Häviöllisen pakkauksen tavoitteena on puolestaan tallentaa kuva häviötöntä pakkausta tiiviimmin siten, että hyväksytään määrätyn suuruinen kuvainformaatio hävikki. Häviöllisesti pakatun kuvan informaatio vääristyy, eikä sen palauttaminen täysin alkuperäisenä ole mahdollista. Käytännössä, esimerkiksi luonnollisia kohteita esittävien värikuvien yhteydessä, ihmisen näköaisti kykenee erottamaan häviöllisesti pakatussa värikuvassa virheitä vasta suhteellisen korkeilla pakkausasteilla. Häviöllisesti pakattuja kuvia, jotka näyttävät ihmiskatsojalle häviöttömästi pakatuilta, kutsutaan "visuaalisesti häviöttömiksi".

Kuten mitä tahansa algoritmia, pakkausalgoritmin tehokkuutta voidaan mitata useilla mittareilla. Kaksi yleisintä tehokkuuden mittaria ovat **pakkaussuhde** (compression ratio) ja **bittisuhde** (bit-rate). Pakkaussuhde määritellään seuraavasti:

$$\text{pakkaussuhde} = \frac{\text{alkuperäisen kuvan koko}}{\text{pakatun kuvan koko}}. \quad (2.1)$$

Jos esimerkiksi 2 048 tavua vaativa digitaalikuva pakataan 256 tavuun, pakkaussuhteeksi tulee tällöin 8:1. Bittisuhde mitataan bitteinä per pikseli (bpp) ja se määritellään seuraavasti:

$$\text{bittisuhde} = \frac{\text{kuvan koko (bitteinä)}}{MN}, \quad (2.2)$$

missö M on kuvan leveys pikseleinä ja N on kuvan korkeus pikseleinä. Häviöllisesti pakattujen, luonnollisia kohteita kuvaavien RGB-värikuvien (24 bpp) tyypillinen bittisuhde on alle 1 bpp, häviöttömän pakkauksen yltäessä vain noin puoleen alkuperäisestä koosta [Taubman and Marcellin, 2002]. Kuvien sisältämä informaatio vaikuttaa kuitenkin merkittävästi pakkaustehokkuuteen, sillä esimerkiksi suuret, yhtenäistä värisävyä sisältävät alueet voidaan pakata erittäin tehokkaasti.

Häviöllisesti pakatun kuvan laatua voidaan mitata joko subjektiivisin tai objektiivisin menetelmin. Subjektiiviseen laadun mittaukseen käytetään usein yksinkertaista Mean Opinion Score (MOS) -menetelmää, jonka avulla usean arvioijan mielipide keskiarvoistetaan [Ebrahimi *et. al.*, 2004]. Menetelmä on suhteellisen väljä ja arvioinneissa voidaan esimerkiksi käyttää skaalaa välillä 1 (huonoin) ja 5 (paras). Menetelmän heikkoutena on luonnollisesti sen työläys. Kaksi yleisintä, objektiivista digitaalikuva laadun mittaria, ovat keskineliövirheen neliöjuuri (Root Mean Squared Error, RMSE) sekä signaali-kohina-suhteen huippuarvo (Peak Signal to Noise Ratio, PSNR) [Sayood, 2006]. Keskineliövirheen neliöjuuri määritellään seuraavasti:

$$RMSE = \sqrt{\frac{1}{MN} \sum_{x=1}^M \sum_{y=1}^N [I(x, y) - I'(x, y)]^2}, \quad (2.3)$$

missä M on kuvan leveys pikseleinä, N on kuvan korkeus pikseleinä, I on alkuperäinen kuva, I' on häviöllisesti pakattu kuva ja (x,y) viittaa kuvan pikseliar-

voon rivillä x , sarakkeessa y . Signaali-kohina-suhteen huippuarvo käyttää hyväkseen keskineliövirheen neliöjuurta, ja se on määritelty desibeleinä (dB) seuraavasti:

$$PSNR = 20 \log_{10} \frac{(2^D - 1)}{RMSE}, \quad (2.4)$$

missä D on käytetty bittisyvyys (tavanomaisen harmaasävykuvan tai komponentin tapauksessa 8). Tyypillisen, hyvälaatuisen kuvan PSNR-arvo on vähintään 30 dB [Taubman and Marcellin, 2002].

Pakkauksen lopputuloksena syntyy pakattu **bittivirta** (bit-stream). Pakkausta voidaan kutsua myös **koodaukseksi** (coding) ja bittivirran muodostavaa osapuolta **enkooderiksi** (encoder). Bittivirran purkavaa osapuolta voidaan nimittää vastaavasti **dekooderiksi** (decoder). Kuvanpakkausstandardeille on tunnusomaista, että ainoastaan bittivirran syntaksi ja dekooderi määritellään. Standardi ei ota kantaa siihen, miten enkooderi on suunniteltu. Laitteisto tai ohjelmisto on standardin mukainen silloin, kun sen tuottama bittivirta noudattaa täsmälleen määriteltyä rakennetta ja se voidaan purkaa standardissa määritellyn dekooderin avulla. Näin ollen useat kuvanpakkausstandardit ovatkin oikeastaan kuvanpurkustandardeita. On kuitenkin yleistä, että standardin yhteydessä viitataan joihinkin enkoodereiden referenssitoteutuksiin, jolloin standardi on helpommin ymmärrettävässä ja lähestyttävässä muodossa.

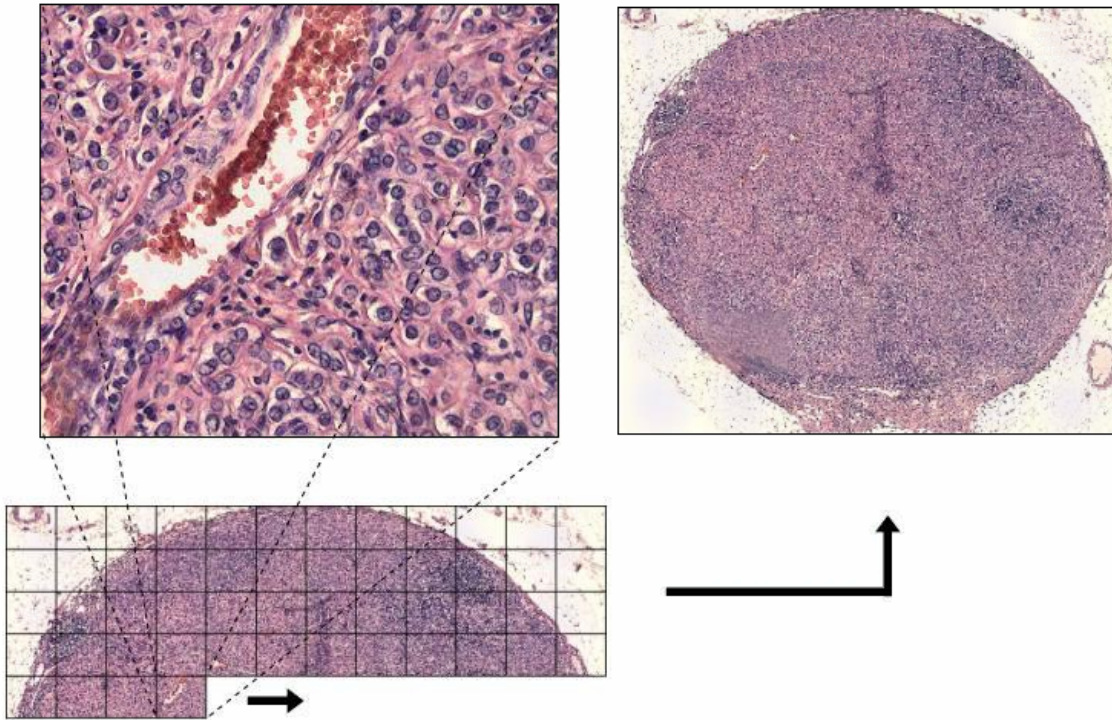
3. Virtuaalimikroskopiasta

Radiologia, joka sisältää muun muassa röntgen-, tietokonekerros- ja magneettikuvauksen, on perinteisesti nähty lääketieteellisen kuvantamisen edelläkävijänä. Useimmilla radiologian tietojärjestelmillä (Radiology Information Systems, RIS) on tarkasti kuvatut, standardeihin perustuvat laitteisto- ja ohjelmistoliitynnät, jotka mahdollistavat informaation tehokkaan jakamisen eri järjestelmien ja kuvantamislaitteistojen välillä. Laajasti hyväksytyn standardoinnin myötä monet käytetyistä tekniikoista ovat kypsyneet hyvin tehokkaaseen kuvantamiseen sekä kuvien käsittelyyn, jakeluun ja arkistointiin. Koska radiologiassa käytetty kuvakoko on yleisesti korkeintaan vain muutamia kymmeniä megatavuja, kuvatiedostojen käsittely olemassa olevilla tietokone- ja verkkotekniikoilla onkin ollut suhteellisen helppoa jo useiden vuosien ajan.

Patologian, sytologian ja hematologian yhteydessä mikroskooppinäytteiden sisältämä valtava informaatiomäärä on hidastanut ja vaikeuttanut niiden digitalisoitumista. Tilanne on parantumassa, sillä nykyinen tietotekniikan taso mahdollistaa mikroskooppinäytteiden koko pinta-alan kuvantamisen virtuaalinäytelaseiksi riittävällä optisella tarkkuudella.

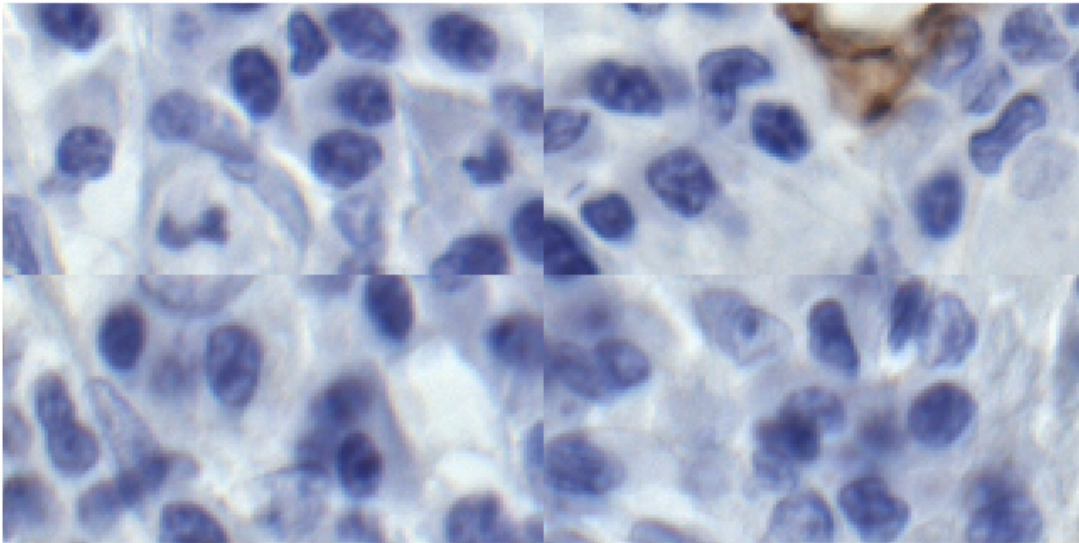
3.1. Näytteiden kuvantaminen

Jotta virtuaalinäytelasien katselu saadaan muistuttamaan mahdollisimman tarkasti oikeaa mikroskopointia, näytteet on kuvannettava riittävän korkealla optisella tarkkuudella. Tarkkuuksia välillä 0.20 ja 0.40 μm per pikseli pidetään yleisesti tähän riittävänä [Rojo *et al.*, 2006]. Käyttämällä näin korkeita tarkkuuksia, mikroskoopin näkökenttä ei kata yhtäaikaisesti kuin murto-osan kuvannettavan näytteen pinta-alasta, joten kuvantamisprosessi on suoritettava osissa. Näytteestä kuvannetaan systemaattisesti yhden näkökentän otoksia hyödyntäen mikroskoopin moottoroitua näytteesiirtopöytää, jonka liikkeitä ohjataan automaattisesti kuvantamisohjelmiston avulla. Kuvantamisen tuotoksena syntyy usein tuhansia yksittäisiä kuva-alkioita (tile), jotka jälkikäsitellään ja yhdistetään (stitching) ohjelmallisesti suureksi montaasikuvaksi (Kuva 6). Myös muita osittaiskuvantamiseen tarkoitettuja menetelmiä käytetään, kuten esimerkiksi line scanning -kuvantamistapa (Aperio Technologies, USA), jossa näytelasi ositetaan pystysuuntaisiin "raitoihin" (stripe).



Kuva 6. Mikroskooppinäytteen kuvantaminen osissa, jonka tuloksena syntyy viittä kuva-alkioista muodostetaan yhtenäinen montaasikuva. Kuva esittää 40 \times -objektiivilla kuvannettua kudosta, joka on värjätty hematoksylin-eosinmenetelmällä.

Yleinen osittaiskuvantamiseen liittyvä ongelma on kuva-alkioiden sauma-alueiden epäjatkuvuudet, jotka luovat montaasikuvaan selkeästi näkyviä artefakteja (Kuva 7). Epäjatkuvuudet johtuvat esimerkiksi näytteesiirtopöydän liikkeen epätarkkuudesta. Tämän korjaamiseksi voidaan käyttää esimerkiksi sauma-alueita pehmentäviä kuvasuotimia tai hyödyntää kuva-analyysialgoritmeja, jotka etsivät rinnakkaisten kuva-alkioiden reuna-alueilta yhteneväisyyksiä, joiden perusteella alkioita voidaan siirtää tai kiertää.



Kuva 7. Neljän kuva-alkion sauma-alueiden näkyvät epäjatkuvuudet monttaasikuvassa. Kuva esittää 40×-objektiivilla kuvannettua kudosisleikettä, joka on värjätty hematoksylin-menetelmällä.

Tavanomaisten mikroskooppinäytteiden maksimikoon ollessa noin 20×30 mm, virtuaalinäytelasitiedosto voi koostua jopa 40:stä gigatavusta pakkaamattomasta kuvainformaatiosta (käyttäen kolmea 8-bittistä värikanavaa). Mikäli virtuaalinäytelasit ovat sytologisia tai hematologisia näytteitä, niiden kuvantaminen on suoritettava tavanomaista korkeammalla optisella tarkkuudella (käyttäen öljy-40×-, öljy-60×- tai öljy-100×-objektiivia) [Burthem *et al.*, 2005]. Tämä moninkertaistaa tuotetun kuvainformaation määrän suhteessa tavanomaiseen kuvantamiseen. Lisäksi jos kuvannetaan useampia tarkennustasoja (Z-akselin suuntaisesti), muodostuva kuvainformaatio edelleen moninkertaistuu tarkennustasojen lukumäärän kerrannaisena, jonka seurauksena lopullisen virtuaalinäytelasitiedoston koko voi olla peräti teratavuja.

3.2. Virtuaalinäytelasien katselu

Virtuaalinäytelasitiedostoja voidaan käsitellä esimerkiksi tietokoneen kiintolevyllä, mukana kannettavalta USB-muistilta tai – käytännöllisimmin – verkkoyhteyden yli [Lundin *et al.*, 2004b]. Tiedostojen suuresta koosta johtuen niiden katseluun ei ole mahdollista käyttää tavanomaisia kuvankäsittelyohjelmistoja (kuten Adobe Photoshop®), jotka edellyttävät koko kuvainformaation lukemisen tietokoneen keskusmuistiin. Tästä syystä kaikki virtuaalinäytelasien katseluun suunnitellut ohjelmistot soveltavat niin kutsuttua on demand -periaatetta: katseluohjelmisto lataa, purkaa ja näyttää virtuaalinäytelasista vain käyttäjän haluaman alueen, käyttäjän haluamalla tarkkuudella. Verkkoyhteyden yli ta-

pahtuvassa katselussa, jossa on erityisen tärkeää välittää vain olennainen kuvainformaatio, on demand -periaate tunnetaan yleisesti nimellä streaming-periaate.

Virtuaalinäytelasin mielekäs katselu edellyttää erillisen yleisnäkömökkunan käyttöä, jossa näytteen koko pinta-ala esitetään käyttäen hyvin pientä tarkkuustasoa. Yleisnäkömökkunan avulla käyttäjä voi nopeasti havaita, mitä osaa näytteestä pääikkunan näkömökkuna edustaa. Navigointi näytteen sisällä voi tapahtua tietokoneen hiiren sekä näppäimistön avulla tai vaihtoehtoisesti käyttäen erityismuotoiltuja osoitinlaitteita [Rojo *et al.*, 2006]. Tehokas navigointi edellyttää mahdollisuutta ”raahata” (pan) näytettä sekä tarkastella sitä eri tarkkuustasoilla. Tarkkuustasot voivat kuvastaa mikroskooppi-objektiivien suurennuksia (esimerkiksi 20× tai 40×). Alkuperäinen mikroskooppinäyte on tällöin luonnollisesti kuvannettava tarkkuudella, joka on suurempi tai yhtä suuri kuin katselun yhteydessä käytettävä suurennus.

3.3. Virtuaalimikroskopian sovelluskohteet

Mikroskopisteilta saatu palaute osoittaa, että virtuaalimikroskopiaa voidaan pitää huomattavana parannuksena varhaisempiin digitaalisen mikroskopian menetelmiin sekä mikroskooppeihin, joissa näytettä tarkastellaan kahden tai useamman okulaarin avulla [Furness, 2007]. Virtuaalimikroskopia tuo selkeitä hyötyjä muun muassa seuraaville sovellusalueille:

- tieteellinen tutkimustyö,
- mikroskopian koulutus ja esitelmöinti,
- ainutkertaisten näytteiden digitaalinen arkistointi,
- kliininen laboriodiagnostiikka,
- kansalliset ja kansainväliset konferenssit,
- sairaaloiden välinen konsultaatio sekä
- laadunarviointitoiminta.

Tieteellisessä tutkimustyössä virtuaalinäytelaseja voidaan tutkia automaattisilla tai puoli-automattisilla kuva-analyysialgoritmeilla [Konsti *et al.*, 2008]. Kahta tai useampaa näytettä on mahdollista katsella rinnakkain tietokoneen näytöllä vertailua ja standardointia varten [Helin *et al.*, 2006]. Tutkimusjulkaisuihin voidaan painettujen, ja usein heikkolaatuisten, mikroskooppinäyttekuvien sijasta liittää linkki julkisella kuvapalvelimella sijaitsevaan virtuaalinäytelasiin [Lundin *et al.*, 2004a]. Näin lukijalle tarjotaan mahdollisuus tutustua näytteeseen kokonaisuudessaan, hyödyntäen eri suurennuksia.

Kumar *et al.* [2006] osoittavat virtuaalimikroskopian soveltuvan hyvin opetuksellisiin tarkoituksiin. Nykyisin mikroskopian koulutus perustuu monil-

ta osin edelleen näytekuvia sisältäviin oppikirjoihin ja diaprojektorilla esitettävään kuviin. Jotta näytekuviin saadaan opetuksen kannalta mielekäästä sisältöä, ne on esitettävä sellaisella tarkkuustasolla, joka mahdollistaa tarkasteltavien kohteiden selkeään erottelun. Tämän seurauksena kuvat esittävät yleensä vain hyvin pientä osaa koko näytteestä ja näin ollen ne vastaavat huonosti todellista mikroskopiaa, jossa tunnetusti haasteellisinta on näytteen diagnostisten solumuutosten havaitseminen. Virtuaalimikroskopiaa hyödyntämällä reaaliaikainen mikroskooppikuva voidaan esittää opiskelijoille dataprojektorin välityksellä. Vaihtoehtoisesti reaaliaikaista mikroskooppikuvaa voidaan myös välittää esimerkiksi mikrotietokonealueen opiskelijapäätteille, jotka on liitetty yhteiseen lähiverkkoon. Opetuksen tueksi voidaan perustaa keskitetty kuvapalvelin, johon on koottu opetettavan aihepiirin keskeisimmät kuvasarjat. Opintojaksojen tenttitilaisuudet voidaan toteuttaa vuorovaikutteisesti liittämällä kirjallisen kokeen yhteyteen virtuaalinäytelaseja ja niihin liittyviä diagnosointitehtäviä.

Virtuaalimikroskopia mahdollistaa näytteiden tallentamisen digitaaliseen muotoon julkiseksi kuva-arkistoksi, jonka avulla näytelasien ”lainaaminen” ei ole aikaan ja paikkaan sidottua [Waterhouse *et al.*, 2008]. Fyysisten näytelasien ollessa ainutkertaisia biologisia kohteita, niiden monistaminen ei ole mahdollista. Tästä syystä näytelaseja joudutaan lainaamaan yhteisistä arkistoista, jonka seurauksena niiden saatavuus heikkenee, ja ilman tiukasti kontrolloituja sääntöjä, näytelasien kulloinenkin sijainti voi olla epäselvä. Välimatkojen kasvaessa joudutaan turvautumaan postin välityksellä tapahtuvaan lainaamiseen, jolloin näytelasien kierrättäminen muuttuu hyvin aikaavieväksi ja epäluotettavaksi.

Sairaalan tai patologian laitoksen sisäisessä verkossa virtuaalinäytelaseja voidaan hyödyntää esimerkiksi potilastapauksia käsittelevissä kokouksissa ja näytelasiseminaareissa [Lundin *et al.*, 2004a]. Kokousten yhteydessä tarkasteltavat näytteet kiertävät tavanomaisesti laitoksen sisällä kaikkien osallistujien kesken. Näytteiden kierrätys voidaan korvata virtuaalinäytteillä, joita osallistujien on mahdollista katsella henkilökohtaisilla työasemillaan. Lisäksi kaikkia tarkasteltavaan potilastapaukseen liittyviä näytteitä, esimerkiksi kasvaimesta tehtyjä erikoisvärjäyksiä, voidaan tutkia samanaikaisesti, ja jopa päällekkäin.

Tarjoamalla mahdollisuus virtuaalinäytelasien katseluun internetin yli, niitä voidaan tehokkaasti hyödyntää kansallisissa ja kansainvälisissä konferensseissa, sairaaloiden välisissä konsultaatioissa sekä laboratorioiden välisissä laadunhallintaohjelmissa [Helin *et al.*, 2005]. Mikroskooppitutkimuksen laadun- ja tarkkailun yhteydessä laadunvalvontaorganisaatio lähettää laboratoriolle preparoidut ja mahdollisimman hyvin vakioidut mikroskooppinäytteet diagnosointia varten. Tämän avulla pyritään selvittämään eri laboratorioiden näytteiden tulkintojen sekä luokittelujen yhdenmukaisuutta. Nykyisen laadunvalvonta-

toiminnan suuri ongelma on identtisen näyttemateriaalin monistaminen kaikille osallistujille. Se on erittäin haasteellista ja monen näytetyypin osalta jopa mahdotonta. Tavanomaisesti esimerkiksi kasvainnäytteestä on jouduttu leikkaamaan useita perättäisiä leikkeitä, jotka voivat olla hyvin samankaltaisia, mutta ei koskaan identtisiä. Vaihtoehtoisesti näytteitä on voitu kierrättää osallistuvien laboratorioden kesken postin välityksellä. Soveltamalla virtuaalimikroskopiaa, mikroskooppinäytteiden identtisyys on taattu, näytteiden kierrättäminen osallistujien kesken voidaan välttää sekä osallistajat voivat nopeasti syöttää tulostiedot web-selaimen kautta.

3.4. Kohti yhtenäistä kuvaformaattia

Huolimatta virtuaalimikroskopian suosion vahvasta kasvusta viime vuosien aikana, maailmanlaajuisesti hyväksytyä ja yhtenäistä kuvaformaattia ei alalle ole vielä vakiintunut. Käytettäessä tavanomaisiin formaatteihin (kuten JPEG) perustuvia virtuaalinäytelaseja, kuvainformaatio joudutaan jakamaan tuhansiksi hyvin pieniksi kuvatiedostoiksi, johtuen formaattien sisältämistä pikseli- ja tiedostokokorajoituksista. Pienistä kuvatiedostoista lähetetään asiakkaalle yhtäaikaaisesti vain katseluikkunan peittämän näytealueen kattavat tiedostot (esimerkiksi Zoomify-järjestelmä [Zoomify, 2008]). Monet kaupalliset näytelasis-kannerit (esimerkiksi Mirax, Carl Zeiss MicroImaging, USA; dotSlide, Olympus UK, Iso-Britannia; ScanScope®, Aperio Technologies, USA) noudattavat tätä periaatetta niiden web-pohjaisissa virtuaalimikroskopiasovelluksissa. Useat kaupalliset organisaatiot ovat myös kehittäneet omia kuvaformaattejaan, jotka ovat luonteeltaan suljettuja ja yhteensopimattomia muiden kilpailijoiden formaattien kanssa. Kilpailusta johtuen on epätodennäköistä, että teollisuudessa kehitetyistä yksinoikeudellisista ja suljetuista kuvaformaateista tulee maailmanlaajuisesti hyväksytyä formaattia virtuaalimikroskopian alalla. Myöskään tavanomaisten kuvaformaattien rajoituksiin mukautuminen – niiden korjaamisen asemesta – ei ole kestävä ratkaisu.

Lupaava ehdokas avoimeksi ja maailmanlaajuisesti hyväksytyksi kuvaformaattiksi virtuaalimikroskopiassa on JPEG-standardin seuraaja, jonka nimeksi on annettu JPEG2000.

4. JPEG2000-standardi

Joint Photographic Experts Group (JPEG) -komitea [JPEG, 2008] on digitaalikuvi-
vien pakkauksen standardointiin keskittynyt järjestö. Komitean perustajina
toimivat kansainväliset standardointiorganisaatiot International Organization
for Standardization (ISO) sekä International Telecommunications Union (ITU).
Nykyisin komitea koostuu useista yksityisen sektorin yrityksistä sekä julkisen
sektorin akateemisista organisaatioista. Komitean tunnetuin kuvanpakkaus-
standardi on vuonna 1992 julkaistu JPEG (ISO/IEC 10918-1 | ITU-T Recom-
mendation T.81), josta on sittemmin muodostunut yleisin kuvaformaatti luon-
nollisia kohteita esittäville digitaalikuville. JPEG-termillä tarkoitetaan yleisesti
ainoastaan standardin yhtä osaa: niin kutsuttua baseline-JPEG-algoritmia, joka
perustuu perättäiseen diskreettiin kosinimuunnokseen (sequential DCT).

Vuonna 2000 julkaistu, JPEG-formaatin seuraajaksi suunniteltu JPEG2000
[Taubman and Marcellin, 2002; Skodras *et al.*, 2001; Rabbani and Joshi, 2002] on
standardiperhe, joka hyödyntää aallokemuunnokseen (wavelet transformation)
perustuvaa pakkausta. Perheen keskeisin osa (Part 1) on julkaistu standardina
ISO/IEC 15444-1 | ITU-T Recommendation T.800. JPEG2000-standardin kehi-
tyksen lähtökohtana on ollut ajatus kuvatiedon pakkaamisesta vain kerran,
jonka jälkeen se voidaan purkaa monin eri tavoin, moninaiisiin tarpeisiin.
JPEG2000-kuvaformaattia käytetäänkin useilla eri tekniikan alueilla, kuten 3D-
kuvantamisessa [Schelkens *et al.*, 2006], videonpakkauksessa [Fossel *et al.*, 2003]
sekä satelliitti- ja ilmakuvantamisessa [Liu *et al.*, 2005]. Virtuaalimikroskopian
näkökulmasta satelliitti- ja ilmakuvantaminen on hyvin lähellä virtuaalinäyte-
lasien katselua ja käsittelyä. Lähtökohtaisesti molemmissa tapauksissa käsitel-
lään erittäin suuria kuvapinta-aloja, joiden hyvin suuret tietomäärät edellyttä-
vät erikoisratkaisuja.

JPEG2000-standardin yhteydessä käytettävä termi **koodivirta** (code-
stream) viittaa sekä koodattuun kuvatietoon että erinäisiin markkereihin (mar-
ker) ja markkerilohkoihin (marker segments), jotka sisältävät muun muassa tie-
toa kuvan koosta sekä käytetyistä koodausparametreista. Näin ollen ne toimi-
vat tavanomaisten tiedostoformaattien otsaketietojen kaltaisina apuvälineinä.
Yksi koodivirta vastaa yhtä JPEG2000-pakattua kuvaa, joka voi koostua useista
kuvakomponenteista, eikä koodivirta tarvitse varsinaista tiedostoformaattia
ympäriinsä. Yleensä se kuitenkin upotetaan standardissa kuvattuihin tiedosto-
formaatteihin (kuten JP2), jotka mahdollistavat monipuolisemman kuvaan liit-
tyvän oheistiedon käytön.

4.1. Standardiperheen osat

JPEG2000-standardiperhe koostuu useista osista (Part), joista tämän tekstin kirjoitushetkellä osa kymmenen on vielä standardointiprosessin alaisuudessa. Standardiperheen osat ovat seuraavat:

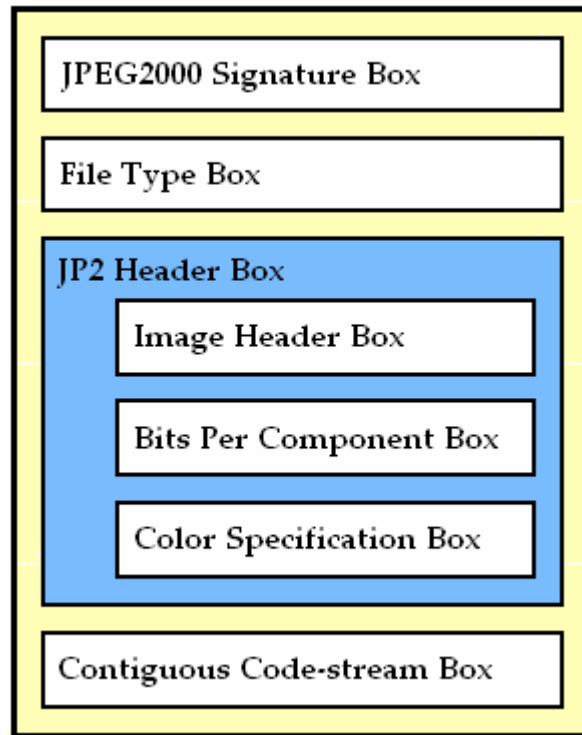
- Part 1 – Core coding system. Ensimmäinen osa sisältää JPEG2000-koodivirran syntaksin ja säännöt sekä tiedostoformaatin (JP2), jonka sisälle koodivirta voidaan upottaa. Osa on julkaistu standardina ISO/IEC 15444-1 | ITU-T Recommendation T.800. Koodivirran syntaksi esitellään tarkemmin kohdassa 4.4. ja JP2-tiedostoformaatti kohdassa 4.2.
- Part 2 – Extensions. Toinen osa laajentaa ensimmäisen osan ominaisuuksia, esitellen muun muassa toisen tiedostoformaatin (JPX), jonka sisälle voidaan upottaa useita koodivirtoja. Tätä ominaisuutta voidaan hyödyntää esimerkiksi eri tarkennustasoilla kuvannetuissa mikroskooppinäytteissä, joista on muodostettu kuvapino. Lisäksi toinen osa mahdollistaa monipuolisen tekstuaalisen ja graafisen metatiedon yhdistämisen haluttuihin kuva-alueisiin. Osa on julkaistu standardina ISO/IEC 15444-2 | ITU-T Recommendation T.801.
- Part 3 – Motion JPEG 2000. Kolmas osa esittelee videotiedostoformaatin (MJ2), jossa sarja kuvia pakataan käyttäen samaa standardin ensimmäisen osan kuvaamaa arkkitehtuuria. Osa on julkaistu standardina ISO/IEC 15444-3 | ITU-T Recommendation T.802.
- Part 4 – Conformance testing. Neljäs osa käsittelee ohjelmistojen ja laitteistojen yhteensopivuuden testausta standardin ensimmäisen osan suhteen. Osa on julkaistu standardina ISO/IEC 15444-4 | ITU-T Recommendation T.803.
- Part 5 – Reference software. Viides osa sisältää standardin ensimmäisen osan toteuttavat referenssiohjelmistopakettit, joiden lähdekoodi on asetettu vapaasti saataville. Toinen paketeista on kirjoitettu C-kielellä ja toinen Java-kielellä. Osa on luonteeltaan vapaamuotoinen, mutta on tästä huolimatta julkaistu standardina ISO/IEC 15444-5 | ITU-T Recommendation T.804.
- Part 6 – Compound image file format. Kuudes osa esittelee tiedostoformaatin (JPM), jonka tarkoituksena on tarjota yhtenäinen koosteformaatti erityyppisille kuville ja tekstin kuvantamiselle, jotka voidaan organisoida painokelpoiseen muotoon. JPM-formaatti tarjoaa mahdollisuuden käyttää JPEG2000-pakkauksen lisäksi myös muita pakkausalgoritmeja. Osa on julkaistu standardina ISO/IEC 15444-6.
- Part 7 – Seitsemäs osa on hylätty.

- Part 8 – Secure JPEG 2000. Kahdeksas osa käsittelee turvallisuuteen ja tietoturvaan liittyviä piirteitä JPEG2000-standardin kannalta. Osa tarjoaa mahdollisuuden muun muassa JPEG2000-pakattujen kuvien salaukseen, kuvatiedon eheyden tarkastamiseen sekä tekijänoikeudellisten tietojen piilottamiseen kuvatiedon sekaan. Osa on julkaistu standardina ISO/IEC 15444-8 | ITU-T Recommendation T.807.
- Part 9 – Interactivity tools, APIs and protocols. Yhdeksäs osa esittelee JPEG2000-kuvien tietoverkkovälitykseen suunnitellun tietoliikenneprotokollan JPEG2000 Interactive Protocol (JPIP). Osa on julkaistu standardina ISO/IEC 15444-9 | ITU-T Recommendation T.808 ja se esitellään tarkemmin kohdassa 4.5.
- Part 10 – Extensions for three-dimensional data. Kymmenes osa on suunniteltu 3D-kuvantamista ajatellen ja on tämän tutkielman kirjoitushetkellä standardointiprosessin viimeisessä vaiheessa ennen varsinaista julkaisua standardina ISO/IEC 15444-10 | ITU-T Recommendation T.809.
- Part 11 – Wireless. Yhdestoista osa käsittelee JPEG2000-kuvien välittämistä virheille alttiiden, langattomien verkkoyhteyksien yli. Osa on julkaistu standardina ISO/IEC 15444-11 | ITU-T Recommendation T.810.
- Part 12 – ISO base media file format. Kahdestoista osa on JPEG- ja Motion Picture Experts Group (MPEG) -komiteoiden yhteisstandardi, joka luo pohjan yhtenäiselle, liikkuvan kuvan pakkaamiseen tarkoitetulle tiedostoformaatile. Osa on julkaistu standardina ISO/IEC 15444-12.

4.2. JP2-tiedostoformaatti

JPEG2000-standardin ensimmäinen osa esittelee valinnaisen JP2-tiedostoformaatin (tavanomainen tiedostopäätte ".jp2"), johon yksi JPEG2000-koodivirta voidaan upottaa. Käytännössä suurin osa JPEG2000-kuvista on JP2-muotoisia. JP2-tiedostoformaatti mahdollistaa muun muassa metatiedon, esimerkiksi virtuaalinäytelasin näytetyypin, upottamisen JP2-otsaketietoihin. JP2-tiedoston rakentuu perättäisistä niin kutsutuista **laatikoista** (box), joista osa on valinnaisia ja osa pakollisia. Lisäksi laatikoiden järjestys on osittain ennalta määritetty. Laatikot voivat olla tyypiltään ylilaatikoita (superbox), jotka sisältävät toisia laatikoita. Näin laatikoiden käytöllä voidaan muodostaa JP2-tiedostoon sisäkkäisiä rakenteita. Kuvassa 8 on esitetty JP2-tiedoston pakolliset laatikot ja niiden järjestys. Laatikot tunnistetaan erityisellä neljän tavun mittaisella avainsanalla. Esimerkiksi avainsana "jp2c" vastaa laatikkoa Contiguous Code-stream Box, joka sisältää koodivirran. Pakollisista laatikoista JPEG2000 Signature Box ilmaisee, että kyseessä on JPEG2000-perheen tiedostoformaatti.

ti. Tämän jälkeen tuleva File Type Box puolestaan kertoo kyseessä olevan JP2-formaatti. JP2 Header Box on yllälaatikko, joka sisältää esimerkiksi kuvan leveys- ja korkeustiedot, komponenttikohtaisen bittisyvyyden sekä kuvan värimallin (kuten sRGB). Lopuksi Contiguous Code-stream Box sisältää itse JPEG2000-koodivirran. Standardi mahdollistaa myös käyttäjien omien laatikoiden luomisen ja upottamisen tiedostoformaattiin.



Kuva 8. JP2-tiedoston pakolliset laatikot ja niiden järjestys.

4.3. Hyödyt virtuaalimikroskopian näkökulmasta

Kohdassa 4.1. esitellyistä JPEG2000-standardiperheen osista keskeisiä tämän tutkielman osalta ovat ensimmäinen, toinen sekä yhdeksäs. Niissä on kuvattu virtuaalimikroskopian kannalta useita merkittäviä standardiperheen ominaisuuksia, joihin kuuluu muun muassa:

- tuki suurille kuvako'uille, joiden ylärajana on $(2^{32} - 1) \times (2^{32} - 1)$ pikseliä, joka teoriassa mahdollistaa eksatavu-kokoluokan virtuaalinäytelasit,
- kuvatiedon satunnainen hajakäsittely (spatial random access),
- skaalautuvuus resoluution ja kuvanlaadun perusteella,
- yhtenäinen pakkausarkkitehtuuri häviöttömään ja häviölliseen pakkaukseen,
- tehokas häviöllinen pakkaus, etenkin hyvin alhaisilla bittisuhteilla,

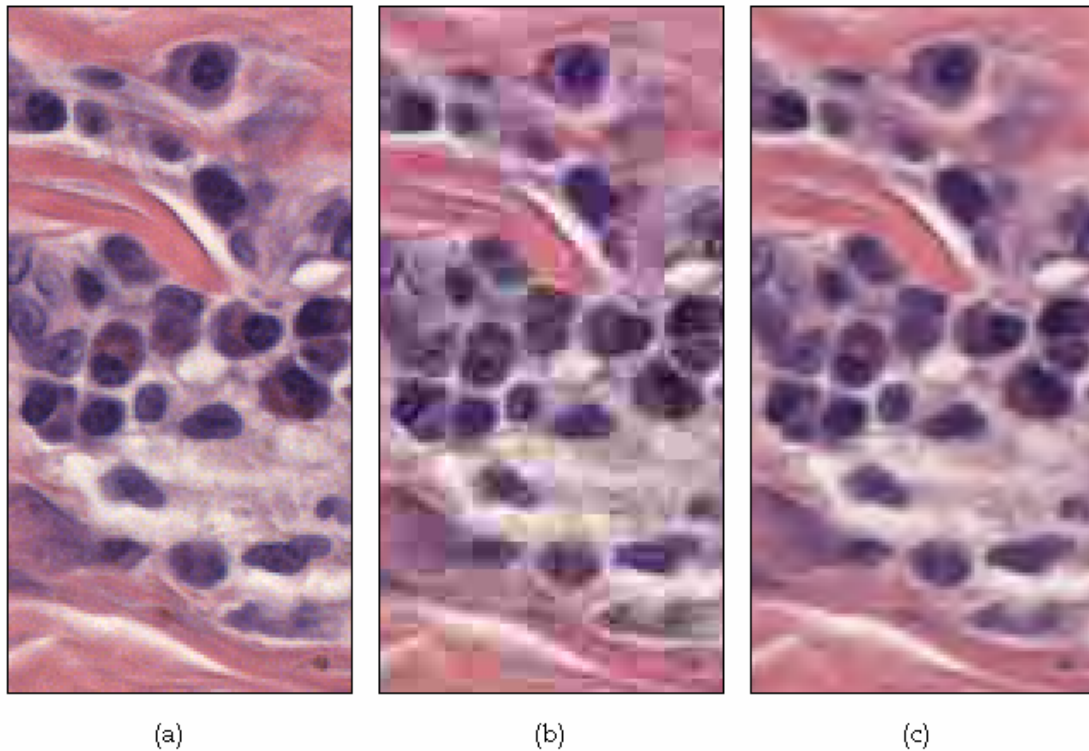
- metatiedon, eli kuvaan liittyvän oheistiedon, upottaminen standardin määrittelemien tiedostoformaattien yhteyteen,
- koodivirran rakenteen muuntelu ilman tarvetta uudelleenpurkamiseen ja -pakkaukseen sekä
- kuvatiedon välittäminen tietoverkkojen yli käyttäen virheensietokykyistä tiedonsiirtoprotokollaa.

Standardi sisältää myös muita hyödyllisiä ominaisuuksia, kuten: kuvakomponenttien suuri määrä, jonka yläraja on 16 384 (2^{14}) kappaletta; kuvakomponenttien korkea dynaaminen alue (bittisyvyys), joka voi olla jopa yli 16 bittiä, tavanomaisen komponentin dynaamisen alueen ollessa 8 bittiä; mahdollisuus harmaasävy- ja värikuvien lisäksi myös binäärikuvien käsittelyyn käyttäen samaa pakkausarkkitehtuuria; haluttujen kuva-alueiden pakkaaminen eri pakkaussuhteilla (region of interest coding); muodostuvan koodivirran koon ennaltamäärittely (rate control), jolloin esimerkiksi voidaan tehokkaasti hyödyntää kaikki käytettävissä oleva tila tai kaistanleveys; sekä turvallisuuteen liittyvät tekijät, kuten kuvaan upotettavat tekijänoikeudelliset, näkymättömät tunnisteet. Edellä mainitut ominaisuudet eivät kuitenkaan ole keskeisiä virtuaalimikroskopian perustarpeiden kannalta, eikä niitä käsitellä tässä tutkielmassa.

Poiketen yleisestä lääketieteellisen kuvantamisen käytännöstä, häviöttömän pakkauksen käyttö ei ole edellytys virtuaalimikroskopiassa. JPEG2000-pakkauksen ideaalitulanteessa alkuperäinen kuva pakataan kertaalleen häviöttömästi, ja muodostuvasta koodivirrasta puretaan tarpeen vaatiessa joko alkuperäinen tai häviöllistä pakkausta vastaava kuva. Käytännössä häviöttöntä pakkausta ei kuitenkaan aina voida käyttää, johtuen sen suuresta, jopa moninkymmenkertaisesta tilavaatimuksesta verrattuna häviölliseen pakkaukseen. Virtuaalinäytelasien pakkaamattoman koon ollessa jopa satoja gigatavuja, kymmenien tuhansien lasien arkiston tallentaminen häviöttömässä muodossa on nykytekniikalla vielä lähes mahdotonta. Häviöllistä pakkausta käyttämällä virtuaalinäytelasit voidaan tehokkaasti pakata "visuaalisesti häviöttömiksi" siten, että ainoastaan ihmissilmän kannalta olennainen informaatio säilytetään ja suuri osa redundantista kuvainformaatiosta hylätään.

JPEG2000 tarjoaa yhtenäisen pakkausarkkitehtuurin riippumatta siitä, käytetäänkö häviöttöntä tai häviöllistä pakkausta. Käyttämällä kuvanlaadun vertailumenetelmänä signaali-kohina-suhteen huippuarvoa (PSNR), JPEG2000-pakkauksen on todettu olevan baseline-JPEG-pakkausta tehokkaampi kaikilla pakkaussuhteilla [Skodras *et al.*, 2001; Rabbani and Joshi, 2002; Santa-Cruz *et al.*, 2002]. Mutta jos laadunarviointi pohjautuu ihmisten havainnointiin, JPEG2000-pakkauksen paremmuus on ilmeistä vasta pakkaussuhteilla, jotka ylittävät 20:1 [Ebrahimi *et al.*, 2004]. Kuvassa 9 on esitetty vertailu korkeasta häviöllisestä

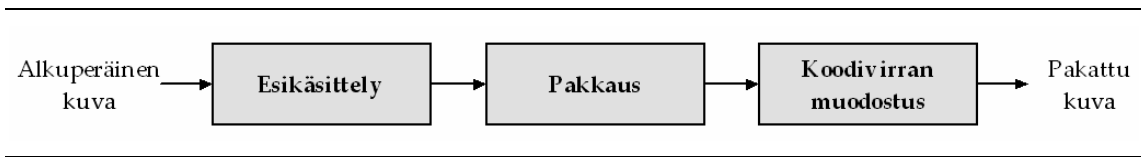
pakkauksesta (noin 60:1) JPEG2000- ja baseline-JPEG-pakkausalgoritmien välillä. Kuvasta käy ilmi korkean JPEG-pakkauksen aiheuttama ”pikselöityminen”, vastaavan JPEG2000-pakkauksen visuaalisen lopputuloksen ollessa pehmeämpi.



Kuva 9. Häviöllisen JPEG2000-pakkauksen vertailu baseline-JPEG-pakkaukseen. Alkuperäinen, pakkaamaton kuva (a); JPEG-pakattu, 0.39 bpp (b); ja JPEG2000-pakattu, 0.39 bpp (c). Kuva esittää 40 \times -objektiivilla kuvannettua kudოსleikettä, joka on värjätty hematoksylin-eosin-menetelmällä.

4.4. JPEG2000-pakkausalgoritmin kuvaus

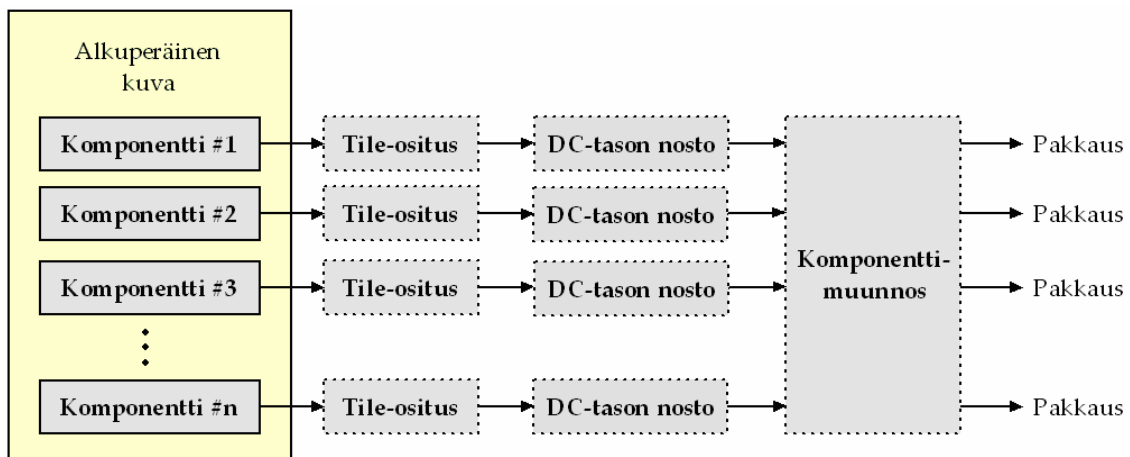
JPEG2000-pakkausalgoritmi voidaan jakaa kolmeen perusosaan: 1) Esikäsittely, 2) Pakkaus ja 3) Koodivirran muodostus. Algoritmin yleiskuvaus on esitetty Kuvassa 10. Seuraavissa kolmessa alakohdassa jokainen perusosa esitellään omiana kokonaisuutenaan. Esittelyjen yhteydessä huomioidaan myös eri koodausparametrien valintaan liittyviä tekijöitä (kuten parametrien vaikutus kuvanlaatuun). Kattava JPEG2000-pakkausalgoritmin kuvaus löytyy JPEG2000-perusteoksesta [Taubman and Marcellin, 2002] ja hieman yksinkertaistetumpi esitys julkaisuista [Skodras *et al.*, 2001] ja [Rabbani and Joshi, 2002].



Kuva 10. JPEG2000-pakkausalgoritmin yleiskuvaus.

4.4.1. Esikäsittely

Esikäsittely-osan (Kuva 11) ensimmäinen vaihe on alkuperäisen kuvan jakaminen komponentteihinsa (yleensä värikanavat). Jakamisen jälkeen jokainen komponentti voidaan valinnaisesti edelleen osittaa suorakulmisiin kuvaalkioihin (tile). Tämän jälkeen – tarpeen vaatiessa – muodostuvien alkioiden (tai komponentin, jos ositusta ei suoriteta) pikseliarvojen dynaaminen alue keskitetään nollan ympärille (DC-tason nosto). Esikäsittely-osan lopuksi suoritetaan valinnainen komponenttimuunnos, jonka avulla pyritään vähentämään komponenttien välillä esiintyviä korrelaatioita.



Kuva 11. JPEG2000-pakkausalgoritmin ensimmäinen osa: Esikäsittely. Katkoviivalla merkityt vaiheet ovat valinnaisia.

Tavanomaisessa RGB-värikuvassa on kolme kuvakomponenttia (punainen, vihreä ja sininen värikanava). JPEG2000 ei kuitenkaan ota kantaa siihen, mikä komponenttien funktio on. Edellä mainitut värikanavat ovat vain yksi mahdollinen vaihtoehto, eikä komponenttien lukumäärää ole rajattu kolmeen. Standardi mahdollistaa jopa 2^{14} (16 384) komponentin käytön, joka on hyödyllistä esimerkiksi fluoresenssimikroskooppinäytteiden ja monikanavakuvantamisen (multispectral imaging) yhteydessä. Komponentteja voidaan käyttää myös esimerkiksi alfa-kanavan (läpinäkyvyysaste) tai kuvan päälle piirrettävän grafiikan tallentamiseen.

Kuvan komponentit voidaan valinnaisesti osittaa perättäisiin, suorakulmaisiin kuva-alkioihin (tile). Operaation voidaan ajatella olevan käänteinen aiemmin esitettyyn mikroskooppinäytteen osittaiskuvantamiseen nähden (Kuva 6). Alkiot ovat JPEG2000-pakkausalgoritmin atominen yksikkö, eli jokaisen komponentin jokainen alkio käsitellään erillisenä kokonaisuutenaan, jotka algoritmin lopuksi liitetään yhdeksi bittivirraksi. Näin ollen kuva-alkioihin osittamisen avulla erittäin suuret kuvat voidaan käsitellä pieninä osakokonaisuuksina. Alkioita iteroimalla kuvasta voidaan nopeasti käsitellä satunnaisesti valittua aluetta. Alkioiden koko voi olla mielivaltaisen, mutta kuitenkin korkeintaan yhtä suuri kuin komponentin koko. Lisäksi kaikkien alkioiden on oltava keskenään samankokoisia. Koska alkiot käsitellään erillisinä yksikköinä, niiden raja-alueille muodostuu suurilla pakkausasteilla selkeästi näkyviä artefakteja. Tämä ilmiö on yhteinen kaikkien osittaisesti käsittelevien (block-based) pakkausalgoritmien kesken (esimerkiksi JPEG-pakkaus, Kuva 9). Mitä pienempää alkion kokoa käytetään, sitä suurempi on raja-alueiden artefaktien määrä, ja näin ollen heikompi kuvanlaatu (Taulukko 2) [Rabbani and Joshi, 2002]. Vastaavasti parhaaseen kuvanlaatuun voidaan päästä ainoastaan jättämällä ositus suorittamatta tai käyttämällä häviötöntä pakkausta. Yksinkertaisuuden vuoksi esittelen tässä tutkielmassa JPEG2000-algoritmista ainoastaan sellaisen version, jossa komponenttien osittamista ei suoriteta. Näin ollen, ellei toisin mainita, algoritmi suoritetaan aina komponenttikohtaisesti, eli komponentti on itsessään yksi suuri kuva-alkio. Riippuen JPEG2000-enkooderin toteutuksesta, erittäin suurten kuvien yhteydessä alkioihin ositus voi olla välttämätöntä. Satunnaiseen hajakäsittelyyn JPEG2000 tarjoaa myös vaihtoehdoisen tavan (perustuen niin kutsuttuihin piireihin), joka esitellään myöhemmin alakohdassa 4.4.3.

Taulukko 2. Kuva-alkion (tile) koon vaikutus kuvanlaatuun häviöllisessä pakkauksessa. Mukailtu lähteestä [Rabbani and Joshi, 2002].

Bittisuhde (bittä / pikseli)	PSNR-keskiarvo (dB)			
	Ei tile-ositusta	512×512	256×256	128×128
0.0625	22.82	22.73 (−0.09)	22.50 (−0.32)	21.79 (−1.03)
0.125	24.84	24.77 (−0.07)	24.59 (−0.25)	24.06 (−0.78)
0.25	27.61	27.55 (−0.06)	27.41 (−0.20)	26.96 (−0.65)
0.5	31.35	31.30 (−0.05)	31.19 (−0.16)	30.82 (−0.53)
1.0	36.22	36.19 (−0.03)	36.11 (−0.11)	35.85 (−0.37)
2.0	42.42	42.40 (−0.02)	42.34 (−0.08)	42.16 (−0.26)

Koska komponentin pikselien intensiteetti-arvot on tallennettu tavanomaisesti etumerkittömällä kokonaisluvulla, laskennan helpottamiseksi suoritetaan niin kutsuttu DC-tason nosto (Direct Current level shifting), joka keskittää dynaamisen alueen nollan ympärille. Tämä saavutetaan muuntamalla kaikkien intensiteetti-arvojen $I(x, y)$ esitystapa kahden komplementtiluku-pohjaiseksi $I'(x, y)$ seuraavasti:

$$I'(x, y) \leftarrow I(x, y) - 2^{D-1}, \quad (4.1)$$

missä D on käytetty bittisyvyys. Jos intensiteetti-arvot esitetään jo valmiiksi etumerkillisillä kokonaisluvulla, DC-tason nostoa ei suoriteta. Noston suorittaminen on kuitenkin tavanomaista, varsinkin digitoitujen mikroskooppinäytteiden yhteydessä.

Esikäsittely-osan viimeisenä, valinnaisena vaiheena on komponenttien välisten korrelaatioiden vähentäminen suorittamalla niin kutsuttu komponenttimuunnos. Komponenttimuunnoksen avulla redundantti kuvainformaatio vähenee ja näin ollen koko algoritmin pakkaustehokkuus kasvaa. Muunnos tunnetaan myös yleisesti nimellä värimuunnos, sillä JPEG2000-standardin ensimmäinen osa määrittelee komponenttimuunnoksen ainoastaan kolmelle ensimmäiselle komponentille, joiden oletetaan vastaavan RGB-värikanavia. Standardin toinen osa kuitenkin laajentaa tätä koskemaan useampaa komponenttia, jolloin värimuunnos-termi on harhaanjohtava. Esittelen tässä kuitenkin ainoastaan standardin ensimmäisen osan tarjoamat kolmen komponentin RGB-värimuunnokset, joita on kaksi: 1) palautuva värimuunnos (Reversible Color Transformation, RCT) ja 2) palautumaton värimuunnos (Irreversible Color Transformation, ICT). Värimuunnosten vaikutus häviöttömän pakkauksen bittisuhteeseen sekä häviöllisen pakkauksen kuvanlaatuun on esitetty Taulukossa 3 [Skodras *et al.*, 2001].

Taulukko 3. Komponenttimuunnoksen vaikutus häviöttömän pakkauksen bittisuhteeseen (RCT-värimuunnos) ja häviöllisen pakkauksen kuvanlaatuun (ICT-värimuunnos). Lähde: [Skodras *et al.*, 2001].

	Ilman muunnosta	Muunnoksen kanssa
Häviötön pakkaus	16.88 bpp	14.78 bpp
Häviöllinen pakkaus (0.25 bpp)	25.67 dB	26.49 dB

Palautuva RCT-muunnos on nimensä mukaisesti palautettavissa, eli muunnoksen yhteydessä ei menetetä informaatiota. Tästä johtuen muunnosta voidaan käyttää sekä häviöttömään että häviölliseen JPEG2000-pakkaukseen. RCT-muunnos vastaa RGB–YUV-värimallimuunnosta. YUV-värimalli koostuu yhdestä luminanssi-komponentista (Y) sekä kahdesta krominanssi-komponentista (U ja V). Luminanssi-komponentti edustaa pikselien valovoimaa ja krominanssi-komponentit pikselien väri-informaatiota. RCT-muunnos on kuvattu seuraavasti:

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} \left[\frac{R + 2G + B}{4} \right] \\ R - G \\ B - G \end{bmatrix}. \quad (4.2)$$

Käänteinen RCT-muunnos on vastaavasti kuvattu seuraavasti:

$$\begin{bmatrix} G \\ R \\ B \end{bmatrix} = \begin{bmatrix} Y - \left[\frac{U + V}{4} \right] \\ V + G \\ U + G \end{bmatrix}. \quad (4.3)$$

Palautumattoman ICT-muunnoksen yhteydessä informaatiota menetetään, johtuen kokonaislukujen muuntamisesta liukuluvuiksi. Se ei näin ollen sovellu häviöttömään JPEG2000-pakkaukseen, vaan sitä käytetään ainoastaan häviöllisen pakkauksen yhteydessä. ICT-muunnos vastaa osittain baseline-JPEG-algoritmissakin käytettyä RGB–YCbCr-värimallimuunnosta. YCbCr-värimalli on hyvin samankaltainen YUV-värimallin kanssa, sillä myös se koostuu yhdestä luminanssi-komponentista (Y) sekä kahdesta krominanssi-komponentista (Cb ja Cr). Toisin kuin baseline-JPEG-algoritmin tapauksessa, ICT-muunnoksen yhteydessä krominanssi-komponenttien alinäytteistys (subsampling) ei ole suositeltavaa, sillä myöhemmin esiteltävän DWT-muunnoksen avulla päästään samankaltaiseen lopputulokseen. Approksimoitu ICT-muunnos on kuvattu seuraavasti:

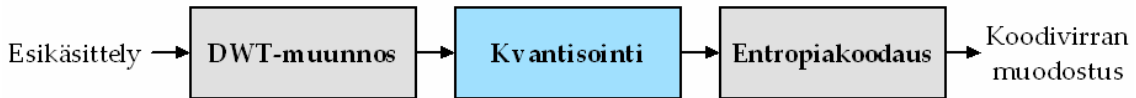
$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 0.299000 & 0.587000 & 0.114000 \\ -0.168736 & -0.331264 & 0.500000 \\ 0.500000 & -0.418688 & -0.081312 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}. \quad (4.4)$$

Käänteinen, approksimoitu ICT-muunnos on vastaavasti kuvattu seuraavasti:

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1.000000 & 0.000000 & 1.402000 \\ 1.000000 & -0.344136 & -0.714136 \\ 1.000000 & 1.772000 & 0.000000 \end{bmatrix} \begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix}. \quad (4.5)$$

4.4.2. Pakkaus

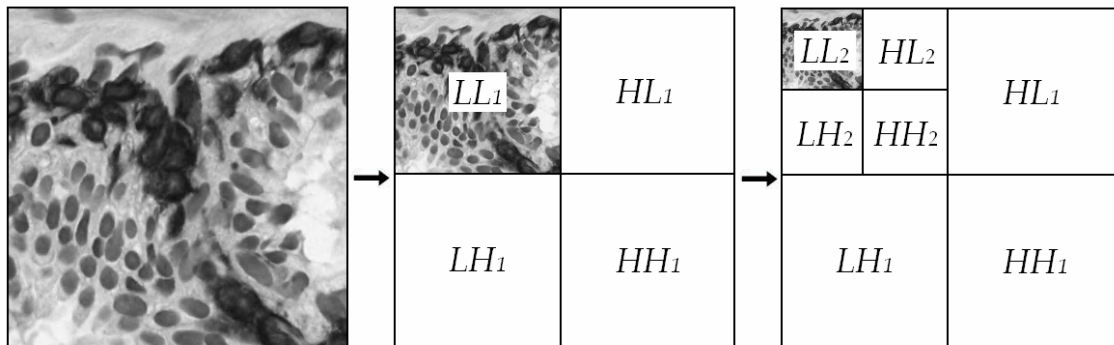
Pakkaus-osan (Kuva 12) ensimmäinen vaihe on diskreetti aallokemuunnos (Discrete Wavelet Transform, DWT), joka muuntaa Esikäsittely-osasta saadun kuvakomponentin spatiaalisesta kohdealueesta (spatial domain) DWT-kohdealueeseen, jossa komponentti hajotetaan sen eri resoluutiotasoihin, jotka edelleen jakaantuvat alikaistoihin (subband). Tämän jälkeen, ainoastaan häviöllisen pakkauksen yhteydessä, jokainen alikaista kvantisoidaan. Kvantisoinnin avulla saavutetaan tiiviimpi pakkaus tarkkuuden kustannuksella. Pakkausosan lopuksi kuvakomponentille suoritetaan entropiakoodaus, joka tiivistää muunnetun informaation häviöttömästi. Wavelet-teoria sekä DWT-muunnoksen kattava kuvaaminen ovat tämän tutkielman laajuuden ulkopuolella, ja seuraavassa esitelläänkin vain DWT-muunnoksen pääpiirteet. Lisäksi entropiakoodauksessa käytetyt kompleksiset koodausmenetelmät kuvataan vain lyhyesti.



Kuva 12. JPEG2000-pakkausalgoritmin toinen osa: Pakkaus. Kvantisointi-vaihe sisältää informaation hävikkiä (ainoastaan häviöllisen pakkauksen yhteydessä).

DWT-muunnoksessa kuvakomponentti analysoidaan ja hajotetaan eri resoluutiotasoihin, jotka edelleen jaetaan alikaistoihin. DWT-muunnos voidaan suorittaa n kertaa, eli n -tasoisena. Ensimmäisellä tasolla ($n = 1$) komponentti hajotetaan neljään alikaistaan: LL_1 , HL_1 , LH_1 sekä HH_1 . Alikaistoista LL_1 sisältää kuvan matalat taajuudet ja se vastaa alkuperäisen kuvan 2:1 alinäytteistettyä versiota, eli kuvan resoluutio puolitetaan. Kolme muuta alikaistaa sisältävät korkeita tajuuksia, joista käy ilmi kuvassa esiintyvät paikalliset epäjatkuvuudet, jotka vastaavat kuva-alueella sijaitsevia reunoja ja yksityiskohtia. DWT-muunnosta voidaan jatkaa toiselle tasolle ($n = 2$), jolloin kaista LL_1 jaetaan edelleen neljään alikaistaan: LL_2 , HL_2 , LH_2 sekä HH_2 . Tämän seurauksena LL_2 -kanava pitää sisällään kuvan, jonka resoluutio on jälleen puoliintunut edelliseen tasoon nähden. DWT-muunnosta voidaan jatkaa tasolle n , jolloin viimeisellä tasolla muodostetaan alikaistat LL_n , HL_n , LH_n ja HH_n . Näin ollen n -tasoisella DWT-

muunnoksella pakattu JPEG2000-kuva sisältää $(n + 1)$ eri resoluutiotasoa. JPEG2000-standardin ensimmäinen osa rajoittaa DWT-muunnosten tasojen lukumäärän 32:een. Kuvassa 13 on esitetty kaksitasoinen DWT-muunnos.



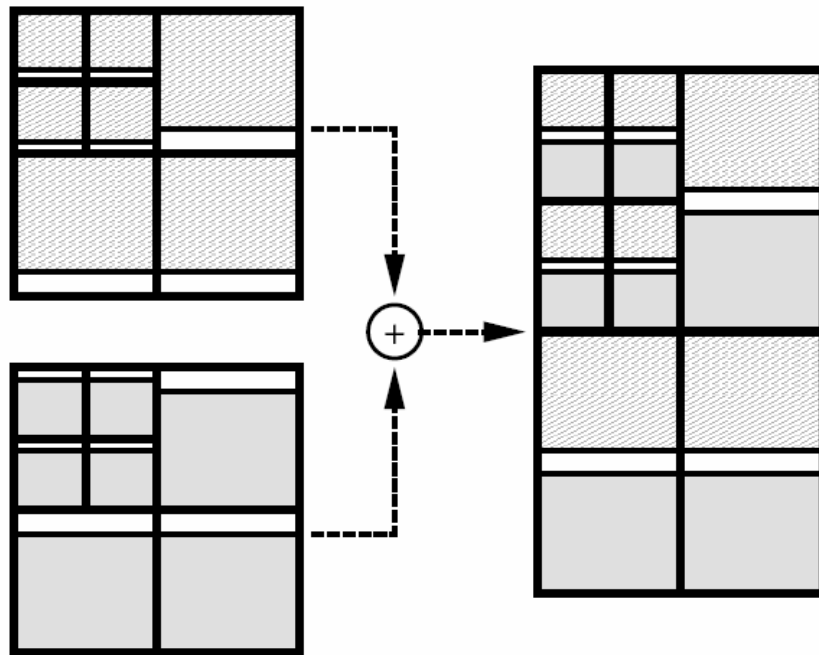
Kuva 13. Kaksitasoinen DWT-muunnos, joka hajottaa kuvan eri resoluutiotasoihin, jotka edelleen jaetaan alikaistoihin.

JPEG2000 mahdollistaa kahden wavelet-suotimen käytön DWT-muunnoksessa: 1) palautuva Le Gall (5-tap low-pass, 3-tap high-pass) -splini-suodin [Gall and Tabatabai, 1988] ja 2) palautumaton, biortogonaali Daubechies (9-tap low-pass, 7-tap high-pass) -splini-suodin [Antonini *et al.*, 1992]. Le Gall -suodinta voidaan käyttää sekä häviöttömään että häviölliseen pakkaukseen, siinä missä Daubechies-suodinta voidaan käyttää ainoastaan häviölliseen pakkaukseen. Daubechies-suodin tuottaa kuvanlaadullisesti paremman lopputuloksen, mutta on laskennallisesti vaativampi (Taulukko 4) [Rabbani and Joshi, 2002]. Laskennallinen kompleksisuus johtuu siitä, että Daubechies-suodin käyttää liukulukuaritmetiikkaa, Le Gall -suotimen hyödyntäessä kokonaislukupohjaista aritmetiikkaa.

Taulukko 3. Palautuvan Le Gall -suotimen sekä palautumattoman Daubechies-suotimen vaikutus kuvanlaatuun. Lähde: [Rabbani and Joshi, 2002].

Bittisuhde (bit-tä / pikseli)	PSNR-keskiarvo (dB)	
	Palautumaton (Daubechies)	Palautuva (Le Gall)
0.0625	22.82	22.37 (-0.45)
0.125	24.84	24.37 (-0.47)
0.25	27.61	27.04 (-0.57)
0.5	31.35	30.74 (-0.61)
1.0	36.22	35.48 (-0.74)
2.0	42.42	41.33 (-1.09)

DWT-muunnoksen yhteydessä muodostuvia alikanavia ja resoluutiotasoja on mahdollista käsitellä aivan kuten tavanomaisia kuvia. Esimerkiksi aiemmin esitelty osittaiskuvantaminen (Kuva 6), ja siihen liittyvä montaasikuvan rakentaminen voidaan suorittaa DWT-muunnoksen yhteydessä (Kuva 14) [Ortiz *et al.*, 2007]. Tästä on kaksi hyötyä: 1) kuvantamisprosessin kokonaissuoritus aika nopeutuu, sillä yksittäisistä kuva-alkioista ei tarvitse rakentaa väliaikaista montaasikuvaa, josta pakataan lopullinen JPEG2000-kuva ja 2) kuva-alkioiden välisen sauma-alueiden epäjatkuvuudet pehmentyvät, johtuen DWT-muunnoksen luonteesta käsitellä koko kuvan pinta-alaa yhtenä kokonaisuutena.



Kuva 14. Kuva-alkioiden yhdistäminen DWT-muunnoksen yhteydessä. Lähde: [Ortiz *et al.*, 2007].

DWT-muunnoksen jälkeen, mutta ainoastaan häviöllisen pakkauksen yhteydessä, siirrytään kvantisointi-vaiheeseen, joka on vastuussa koko pakkausprosessin suurimmasta informaatiohävikistä. Kvantisointi-vaiheessa DWT-muunnoksessa syntyneiden alikanavien näytteiden (niin kutsutut wavelet-kertoimet – wavelet coefficient) tarkkuutta vähennetään tiiviimmän pakkauksen saavuttamiseksi. JPEG2000-standardin ensimmäisessä osassa tämä suoritetaan käyttämällä skalaarikvantisointia. Kvantisointiaskeleen koko Δ_b ilmaistaan suhteessa alikanavan b dynaamiseen alueeseen, joka puolestaan määräytyy alkuperäisen kuvakomponentin bittisyvyyden perusteella. Mitä suurempi askelkoko valitaan, sitä suurempi on tarkkuuden menetys ja vastaavasti tehokkaampi pakkaus. Standardi mahdollistaa usean kvantisointiaskelkoon käytön eri alikanavien yhteydessä, mutta sallii kuitenkin vain yhden per alkanava. Kvantisointiaskeleen koon dynaaminen muuttaminen onkin yksi mahdollinen

tapa, jolla voidaan kontrolloida muodostuvan kuvan lopullista bittisuhdetta (rate control). Kvantisoinnin lopputuloksena jokainen wavelet-kerroin muuntuu kokonaisluvuksi, riippumatta sen aiemmasta esitysmuodosta. Alikanavan b wavelet-kertoimen $c_b(x, y)$ kvantisoitu kokonaislukuarvo $q_b(x, y)$ voidaan laskea seuraavasti:

$$q_b(x, y) = \text{sign}[c_b(x, y)] \left\lfloor \frac{|c_b(x, y)|}{\Delta_b} \right\rfloor. \quad (4.6)$$

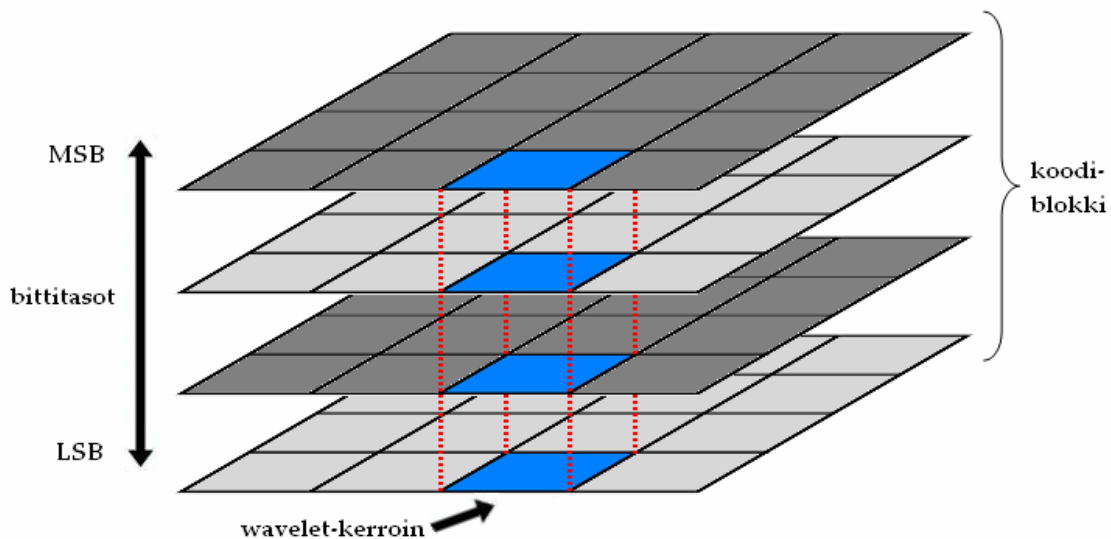
Pakkaus-osan viimeinen vaihe on entropiakoodaus, jossa alikanavien wavelet-kertoimet enkoodataan lopullista bittivirtaa varten. Enkoodauksen yhteydessä jokainen alikanava jaetaan edelleen pienempiin, suorakulmaisiiin **koodiblokkeihin** (code-block). Koodiblokkien leveys voi olla kooltaan 2^x ja korkeus 2^y , missä x ja y ovat $[2...10]$, ja leveyden ja korkeuden summan on oltava vähintään 12. JPEG2000-standardin koodivirtaa rajoittavista yhteensopivuusprofiileista (conformance profiles) ensimmäinen (Profile-0) määrää koodiblokin kooksi joko 32×32 tai 64×64 . Tätä pienemmät koodiblokit aiheuttavat kuvanlaadullisesti heikomman lopputuloksen (Taulukko 4) [Rabbani and Joshi, 2002].

Taulukko 4. Koodiblokin koon vaikutus kuvanlaatuun. Lähde: [Rabbani and Joshi, 2002].

Bittisuhde (bittiä / pikseli)	PSNR-keskiarvo (dB)			
	64 × 64	32 × 32	16 × 16	8 × 8
0.0625	22.82	22.78 (-0.04)	22.62 (-0.20)	22.27 (-0.55)
0.125	24.84	24.78 (-0.06)	24.57 (-0.27)	24.13 (-0.71)
0.25	27.61	27.52 (-0.09)	27.23 (-0.38)	26.63 (-0.98)
0.5	31.35	31.22 (-0.13)	30.84 (-0.51)	30.04 (-1.31)
1.0	36.22	36.09 (-0.13)	35.68 (-0.54)	34.70 (-1.52)
2.0	42.42	42.28 (-0.14)	41.83 (-0.59)	40.70 (-1.72)

Entropiakoodauksessa koodiblokit enkoodataan käyttäen bittitasokoodausta (Bit-Plane Coding, BPC) sekä aritmeettista koodausta. Standardiin on valittu BPC-koodaukseen monimutkainen EBCOT-algoritmi (Embedded Block Coding with Optimized Truncation) [Taubman, 2000] ja aritmeettiseen koodaukseen MQ-kooderi, jota hyödynnetään esimerkiksi JBIG2-standardissa [ISO/IEC 14492, 2000]. Entropiakoodauksessa koodiblokit käsitellään omina yksikköi-

nään, jolloin ne eivät riipu ympäristöstä ja näin ollen saavutetaan esimerkiksi tehokkaampi satunnainen hajakäsittely. Nimensä mukaisesti entropiakoodausosan aloittavassa bittitasokoodauksessa alikanava jaetaan tasoihin, joiden tarkkuus on yksi bitti. Näin ollen niiden voidaan ajatella vastaavan binäärikuvaa, jonka pikselin arvona on joko 0 tai 1. Eli jos alikanavan, johon koodiblokki kuuluu, bittisyvyys on D , sen alueella olevat koodiblokit jaetaan D tasoon. Tämän jälkeen jokainen bittitaso enkoodataan vuorollaan käyttäen kolmea EBCOT-algoritmin vaihetta, joissa bittitasojen kontekstista, eli wavelet-kertoimien muodostamasta naapurustosta, etsitään keskinäisiä riippuvuuksia. Prosessi aloitetaan tasosta, joka vastaa wavelet-kertoimien eniten merkitsevää bittiä (MSB) ja iteroidaan aina vähiten merkitsevään bittiin (LSB) (Kuva 15). Konteksti-informaatio syötetään MQ-kooderille, joka luo koodiblokikohtaisen, pakatun bittivirran.



Kuva 15. Bittitasokoodauksen periaate, jossa koodiblokki ja sen sisältämät wavelet-kertoimet jaetaan yhden bitin tasoihin. Tasot käsitellään yksi kerrallaan, alkaen eniten merkitsevästä bitistä (MSB) ja päättyen vähiten merkitsevään bittiin (LSB).

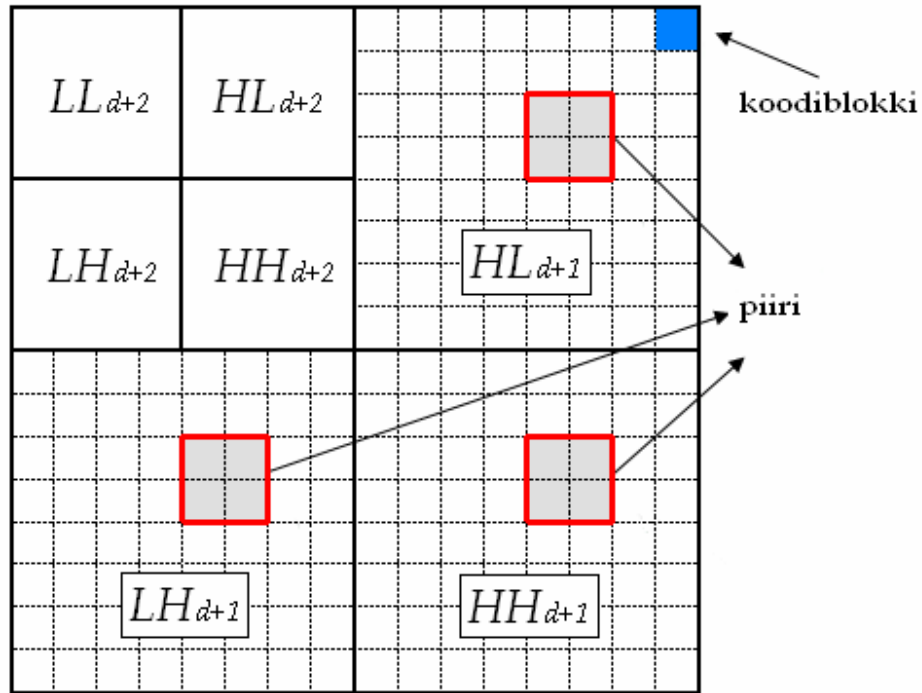
Siinä missä JPEG2000-kuvan skaalautuvuus resoluution perusteella saavutetaan DWT-muunnoksen ominaisluonteesta, entropiakoodauksen lopputulokseksi luodut, koodiblokkikohtaiset bittivirrat mahdollistavat kuvan skaalautuvuuden laadun perusteella. EBCOT-algoritmin yhteydessä koodiblokeille muodostetaan niin kutsuttuja katkaisupisteitä (truncation point), jotka toimivat eri **laatukerrosten** (quality layer) erottelevina tekijöinä. Laatukerrosten avulla alkuperäisestä kuvasta voidaan purkaa laadultaan erilaisia versioita. Tämä mahdollistaa esimerkiksi hyvin karkean esityksen käyttämisen, jolloin kuvasta puretaan vain hyvin pieni määrä informaatiota, joka voidaan nopeasti välittää

esimerkiksi rajoitettua kaistanleveyttä noudattavan verkkoyhteyden yli. Jotta skaalautuvuusominaisuudet sekä satunnainen hajakäsittely ovat mahdollisia, lopullisen JPEG2000-koodivirran rakenne on organisoitava huolellisesti.

4.4.3. Koodivirran muodostus

Lopullinen JPEG2000-koodivirta muodostetaan yhdistämällä entropiaenkoodauksessa pakatut, koodiblokkikohtaiset bittivirrat yhteinäiseksi bittivirraksi. Koodivirran sisällä koodiblokkikohtaiset bittivirrat järjestetään halutulla tavalla ja koodivirtaan liitetään tarvittavat, sekä valinnaiset, markkerit ja markkerilohkot. Koodivirran organisointiin JPEG2000 tarjoaa koodiblokkien ja laatukerrosten lisäksi kaksi muuta käsitettä: **piiri** (precinct) ja **paketti** (packet).

Piirit ovat DWT-kohdealueen vastine spatiaalisen kohdealueen tavalle ja kaa JPEG2000-kuva tile-osituksen avulla, joka mahdollistaa kuvan satunnaisen hajakäsittelyn. Kuten tile-osituksen, myös piirien käyttö on valinnaista. Koska koodiblokki on määritelty DWT-kohdealueella alikanavakohtaisesti, kuvan spatiaalisen alueen pikseliarvot muodostuvat eri alikanavien koodiblokkien yhdistelmän perusteella, joten yhdellä koodiblokilla ei ole suoraa vastaavuutta kuvan spatiaalisen alueen kanssa. Vastaavuus voidaan luoda käyttämällä piirejä. Piiri yhdistää DWT-resoluutiotason LL_d alikanavien HL_{d+1} , LH_{d+1} ja HH_{d+1} koodiblokit spatiaalisesti korreloiviin ryhmiin (Kuva 16). Yksi piiri voi koostua useista koodiblokeista ja yksi koodiblokki kuuluu ainoastaan yhteen piiriin. Piirin leveys on rajoitettu olevan 2^n , missä n on kokonaisluku ja $1 \leq n \leq 15$. Piirin korkeus on vastaavasti rajoitettu olevan 2^n , missä n on kokonaisluku ja $1 \leq n \leq 15$. Eri resoluutiotasoilla voidaan käyttää eri kokoisia piirejä, ja jos piirin koko on pienempi kuin käytettävän koodiblokin koko, koodiblokin kokoa rajataan kyseisellä resoluutiotasolla piirin koon mukaiseksi. Tehokkaan hajakäsittelyn kannalta piirin koko on oltava riittävän pieni, sillä jos se on hyvin suuri, piirin vaikutus ulottuu suureen spatiaaliseen kuvapinta-alaan, ja pientä aluetta käsiteltäessä saatetaan joutua purkamaan hyvinkin paljon ylimääräisiä koodiblokkeja. Vastaavasti jos piirin koko on hyvin pieni (kuten 4×4), satunnaisen hajakäsittelyn tehokkuus kärsii, sillä tavanomaisen katseluikkunan alue (esimerkiksi 1280×1024 pikseliä) vastaa satoja piirejä. Lisäksi tämä rajoittaa koodiblokin kokoa. Tyypillinen piirin koko on joko vastaava kuin koodiblokin koko tai hieman suurempi.

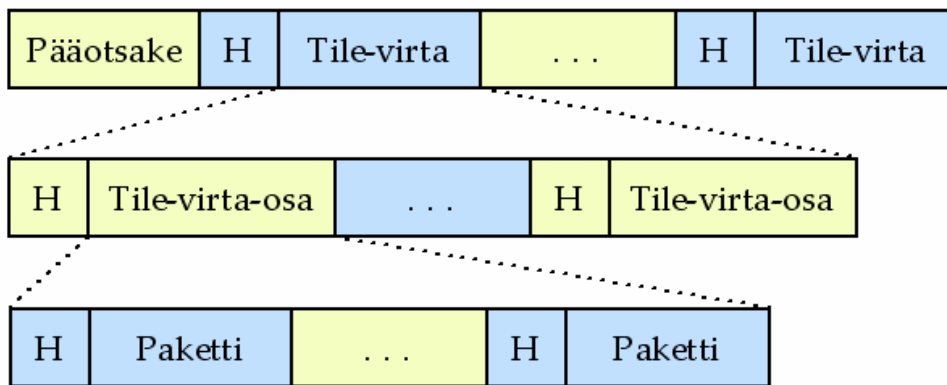


Kuva 16. DWT-resoluutiotason LL_d alikanavien HL_{d+1} , LH_{d+1} ja HH_{d+1} jakaantumisen piireihin, ja edelleen koodiblokkeihin.

JPEG2000-koodivirta muodostuu perättäisistä paketeista, jotka ovat koodivirran perusyksikkö. Paketti $P_{c,r,p,l}$ vastaa yhden kuvakomponentin c , yhden resoluutiotason r , yhden piirin p sisältämien koodiblokkien yhtä laatukerrosta l . Paketit organisoidaan koodivirran sisällä haluttuun progressiojärjestykseen (progression order), johon standardin ensimmäinen osa tarjoaa viisi mahdollista tapaa (joissa "layer" vastaa laatukerrosta ja "position" vastaa piiriä): 1) Layer-Resolution-Component-Position (LRCP), 2) Resolution-Layer-Component-Position (RLCP), 3) Resolution-Position-Component-Layer (RPCL), 4) Position-Component-Resolution-Layer (PCRL) ja 5) Component-Position-Resolution-Layer (CPRL). Progressiojärjestys organisoii paketit sisäkkäisiin silmukkarakenteisiin, jotka nopeuttavat joidenkin ominaisuuksien käsittelyä toisten kustannuksella. Esimerkiksi progressiojärjestys LRCP tarjoaa nopeimman iteroinnin laatukerrostojen perusteella (laatuskaalautuvuus), mutta hitaimman spatiaalisen alueen hajakäsittelyn. Vastaavasti progressiojärjestys PCRL tarjoaa nopeimman spatiaalisen alueen hajakäsittelyn, mutta hitaimman iteroinnin laatukerrostojen perusteella. Progressiojärjestystä voidaan vaihdella tile-kohtaisesti.

JPEG2000-koodivirta muodostuu yhdestä pääotsakkeesta (main header), joka sisältää koodivirran yleisluontoista tietoa, sekä yhdestä tai useammasta tile-virrasta (tile-stream), joita yksinkertaisimmassa tapauksessa (ilman tile-ositusta) on yksi kappale. Yksi koodivirta voi sisältää korkeintaan 65 535 tile-

virtaa. Jokainen tile-virta sisältää oman tile-otsakkeen, joka sisältää kyseistä tile-osa koskevaa tietoa. Jokainen tile-virta jaetaan edelleen yhteen tai useampaan tile-virta-osaan (tile-part), jotka sallivat yhtenäisen progressiojärjestyksen käytön koko kuvan alueella, vaikka käytettäisiinkin tile-ositusta. Yksi tile-virta voi sisältää korkeintaan 255 tile-virta-osa. Vastaavasti tile-virta-osat koostuvat omista otsakkeistaan (tile-part header), jotka muodostetaan kuten tile-otsakkeet, mutta sisältävät rajoitetun määrän tietoa. Tile-virta-osien pituus tavuina kuvataan 32-bittisellä tarkkuudella (neljän tavun etumerkitön kokonaisluku), jolloin sen sisältämä pakattu tieto on rajoitettu neljään gigatavuun. Lopulta jokainen tile-virta-osa jakaantuu edelleen paketteihin (myös ei yhtään pakettia sisältävät tile-virta-osat sallitaan). Paketeilla on niin ikään oma otsake, joka sisältää esimerkiksi tiedon paketin koosta. Paketin oma otsaketieto mahdollistaa muun muassa tehokkaan virheensietokyvyn, jolloin tiedonsiirrossa aiheutunut virheen vaikutusala on hyvin rajattu. Koodivirran rakenne on esitetty Kuvassa 17.



Kuva 17. JPEG2000-koodivirran rakenne. Kirjain H tarkoittaa otsaketietoa.

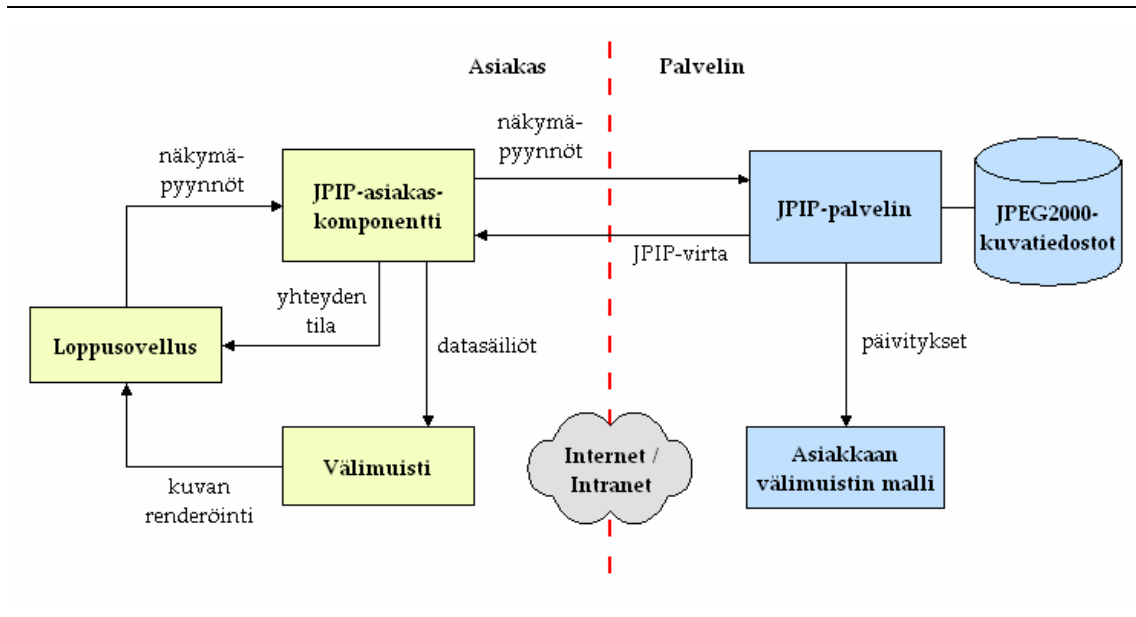
Koodivirran kaikki otsaketiedot koostuvat markkereista ja markkerilohkoista. Markkerit ovat kahden tavun mittaisia; esimerkiksi $FF4F_h$ vastaa niin kutsuttua Start Of Code-stream -markkeria (SOC), joka ilmoittaa koodivirran alkamiskohdan. Markkerilohkot puolestaan koostuvat yhdestä markkerista sekä siihen liittyvästä parametrilistasta. Eri otsakkeilla (siis pääotsake, tile-virran otsake, tile-virta-osan otsake ja paketin otsake) on oma, rajattu joukko käytettävissä olevia markkereita ja markkerisegmenttejä. Näitä on lukuisia, eikä niitä kuvata tässä yhteydessä tarkemmin. Pohjimmiltaan JPEG2000-koodivirta rakentuu siis paketeista, markkereista ja markkerilohkoista.

4.5. JPIP-tietoliikenneprotokolla

JPEG2000-standardiperheen yhdeksäs osa sisältää JPIP (JPEG2000 Interactive Protocol) -tietoliikenneprotokollan kuvaksen, joka on tarkoitettu asiakas-

palvelin-pohjaiseen JPEG2000-kuvien välittämiseen tietoverkkojen yli [Taubman and Prandolini, 2003]. JPIP hyödyntää pääasiallisesti tiedonsiirtoon olemassa olevaa HTTP/1.1.-protokollaa, mutta myös TCP- ja UDP-protokollia voidaan käyttää. JPIP jakaa JPEG2000-koodivirran elementit (esimerkiksi pääotsake, tile-osat, piirit ja metatieto) niin kutsuttuihin datasäiliöihin (data-bin), jotka voidaan välittää palvelimelta asiakkaalle mielivaltaisessa järjestyksessä. JPIP-palvelimen lähettämä kuvainformaatio tunnustetaan MIME-mediatyyppillä "image/jpp-stream", jos kyseessä on piiri-pohjainen kuva ja MIME-mediatyyppillä "image/jpt-stream", jos kyseessä on tile-pohjainen kuva. Näitä kutsutaan yhteisellä nimellä JPIP-virroiksi. JPIP-virta koostuu sarjasta JPIP-viestejä, joista jokainen sisältää osia (tai kaiken) yhden datasäiliön sisällöstä. JPIP-virrat ovat itsenäisiä esityksiä kuvan eri osista ja niitä voidaan vapaasti yhdistää muodostamaan uusia JPIP-virtoja. JPIP-virtojen ja datasäiliöiden itsenäisyyden ansiosta tietoverkkojen luontainen epävakaisuus ei ole ongelma JPIP-pohjaisessa kuvatiedon välityksessä.

JPIP käyttää välimuistipohjaista tiedonvälitysarkkitehtuuria, jossa asiakkaan välimuistin lisäksi myös palvelin sisältää asiakaskohtaisen välimuistin mallin (Kuva 18). Asiakaskohtaista välimuistin mallia päivitetään ajoittain vastaamaan asiakkaan välimuistin tilaa. Välimuisti voi olla pysyväluontoinen, toisin sanoen se voidaan tallentaa esimerkiksi kiintolevyille, mutta se voi myös olla tilapäinen, keskusmuistissa pidettävä muistipuskuri. Asiakkaalla on lisäksi mahdollisuus itse ilmoittaa palvelimelle välimuistin tilan muutoksista, jolloin palvelin päivittää mallia vastaamaan uutta tilaa. Myös palvelimen täytyy huolehtia siitä, että kuvatiedoston muutoksen yhteydessä kyseiseen kuvaan liittyvät välimuistien mallit tyhjennetään ja tieto muutoksesta välitetään asiakkaille. Välimuistiperustaisen arkkitehtuurin tehokkuuden takaamiseksi asiakaspalvelin-toiminnallisuus on eristetty varsinaisesta JPEG2000-koodivirran purkamisesta. Asiakkaan välimuistin voidaan ajatella olevan aluksi tyhjä runko palvelimella sijaitsevalle JPEG2000-kuvalle. Loppusovellus, esimerkiksi virtuaalinäytelasien katselusovellus, välittää JPIP-asiakaskomponentille katseluikkunan näkymiä, jotka välitetään edelleen JPIP-palvelimelle. Palvelin vastaa näkymäpyyntöihin muodostamalla JPIP-virtoja, jotka sisältävät kaiken, vain osan tai ei lainkaan pyydetyn näkymän kuvainformaatiosta. Päätös tästä tehdään asiakkaan välimuistin mallin perusteella, josta käy ilmi, mitä osia kuvasta asiakkaalla on jo tallennettuna. Muodostuvat JPIP-virrat lähetetään JPIP-asiakaskomponentille, joka tallentaa ne asiakkaan välimuistiin, jolloin tyhjä kuvan runko täyttyy paloittain. Loppusovellukselle ilmoitetaan uuden kuvainformaation saapumisesta, jonka perusteella sovellus voi renderöidä välimuistista halutun näkymän näyttölaitteelle.



Kuva 18. Välimuistia ja JPIP-protokollaa hyödyntävä asiakas-palvelin-arkkitehtuuri JPEG2000-kuvien välittämiseen.

5. Tavoitteet

Tämä tutkielma rakentuu kolmen vuoden tutkimustyön tuloksista, jotka perustuvat pääosin käytännön ohjelmistojen toteutukseen. Tutkielman tiivistetyt tavoitteet sekä kehitettyjen ohjelmistojen nimet ovat seuraavat:

1. *Osittaiskuvantamisessa muodostuvien kuva-alkioiden yhdistäminen suureksi montaasikuvaksi siten, että alkioiden sauma-alueilla ei ole selkeästi näkyvää epäjatkuvuutta (**LargeMontage**).*
2. *Montaasikuvan luomisen yhdistäminen kuvantamislaitteistoon siten, että prosessi saadaan täysin automatisoitua (**DirObserver**). Yhdistäminen on tarpeen tehdä mahdollisimman yleiskäyttöiseksi ja laajennettavaksi, jotta jatkossa kehitettävät ohjelmistot voidaan liittää osaksi automatisoitua prosessia.*
3. *Virtuaalimikroskopian kannalta optimaalisen JPEG2000-koodivirran parametrisoinnin määrittely. Koska JPEG2000-standardia käytetään hyvin moninlaisiin tarkoituksiin, sen käyttö virtuaalimikroskopiassa edellyttää sopivien koodivirtaparametrien määrittelemistä.*
4. *JPEG2000-standardia hyödyntävien ohjelmistojen kehitys virtuaalimikroskopian tarpeisiin (**JVS-ohjelmistopaketti**). Ohjelmistopaketti koostuu kolmesta erillisestä JPEG2000-sovelluksesta: (i) virtuaalinäytelasien pakkausohjelmisto (**JVScomp**), (ii) virtuaalinäytelasien katseluohjelmisto (**JVSview**) sekä (iii) virtuaalinäytelasien kuvapalvelinohjelmisto (**JVSserv**). Lisäksi ohjelmistopakettin käyttämän XML-muotoisen metatiedon rakenne kuvataan formaalisti XML-skeemana (**JVSchema**).*
5. *JPEG2000-standardin suorituskyvyn testaus. Testauksella tutkitaan onko JPEG2000-pakkaus riittävän nopeaa, jotta se voidaan liittää osaksi automaattista näytelasien kuvantamista. Lisäksi tutkitaan JPIP-protokollan soveltuvuutta laajamittaiseen virtuaalinäytelasien kuvapalvelinkäyttöön.*

6. Menetelmät

6.1. LargeMontage-ohjelmiston kehitys

LargeMontage-ohjelmisto toteutettiin Java-ohjelmointikielellä (versio 1.5.0) [Sun, 2008a], jolloin sitä voidaan käyttää lähes kaikilla tavanomaisilla käyttöjärjestelmillä. LargeMontage hyödyntää Java Advanced Imaging (versio 1.3.3-alpha) [Sun, 2008b] sekä Java Advanced Imaging Image I/O Tools (versio 1.1-alpha) [Sun, 2008c] -kirjastoja, jotka sisältävät useita monipuolisia kuvankäsittely- ja kuva-analyysityökaluja. LargeMontage on suunniteltu itsenäiseksi, komentorivipohjaiseksi sovellukseksi, mutta sitä voidaan käyttää myös lisäosana (plugin) vapaaseen ImageJ-kuva-analyysiohjelmistoon [Rasband, 2008]. Lisäksi LargeMontage hyödyntää ImageJ-ohjelmiston TurboReg-lisäosaa [Thévenaz, 2008] kuva-alkioiden automaattisessa asemoinnissa. Ohjelmiston kehityksen apuvälineenä käytettiin Eclipse-ohjelmointiympäristöä [Eclipse, 2008].

6.2. DirObserver-ohjelmiston kehitys

DirObserver-ohjelmisto toteutettiin C++-ohjelmointikielellä käyttäen Microsoft Foundation Classes (MFC) -kirjastoa [Microsoft, 2008a]. Ohjelmisto on suunniteltu 32-bittisille Microsoft Windows®-käyttöjärjestelmille. Ohjelmiston kehityksen apuvälineenä käytettiin Microsoft Visual Studio 2005 -ohjelmointiympäristöä [Microsoft, 2008b].

6.3. Virtuaalinäytelasien kuvantaminen testauskäyttöön

Testauskäyttöön tarkoitetut virtuaalinäytelasit kuvannettiin käyttäen Zeiss Axioskop 40 -mikroskooppia (Carl Zeiss MicroImaging, USA), joka oli varustettu 10×-, 20×- ja öljy-40×-objektiiveilla sekä moottoroidulla näytteensiirtopöydällä (Märzhäuser Wetzlar, Saksa). Kuvat digitoitiin CCD-värikameralla (QICAM Fast, Qimaging, Kanada; kolme 8-bitin värikanavaa; resoluutio $1\,388 \times 1\,036$ pikseliä; pikselin koko $4.65 \mu\text{m}$). Kuvantamisprosessia kontrolloitiin Surveyor-ohjelmistolla (Objective Imaging, Iso-Britannia, ohjelmistoversio 5.03) käyttäen Windows® XP -työasemaa. Surveyor-ohjelmisto vastasi näytteensiirtopöydän ja tarkennuksen ohjaamisesta. Tavanomaisten histologisten näytteiden lisäksi sytologisia näytteitä sisältäviä virtuaalinäytelaseja kuvannettiin käyttäen usean tarkennustason kuvantamistapaa (Z-Stack). Muodostuneet kuva-alkiot tallennettiin pakkaamattomana käyttäen BMP-tiedostoformaattia ja niistä luotiin montaasikuva joko käyttäen Surveyor-ohjelmiston sisäänrakennettua toimintoa tai edellä kuvattua LargeMontage-ohjelmistoa.

6.4. JPEG2000-koodivirran parametrien optimointi

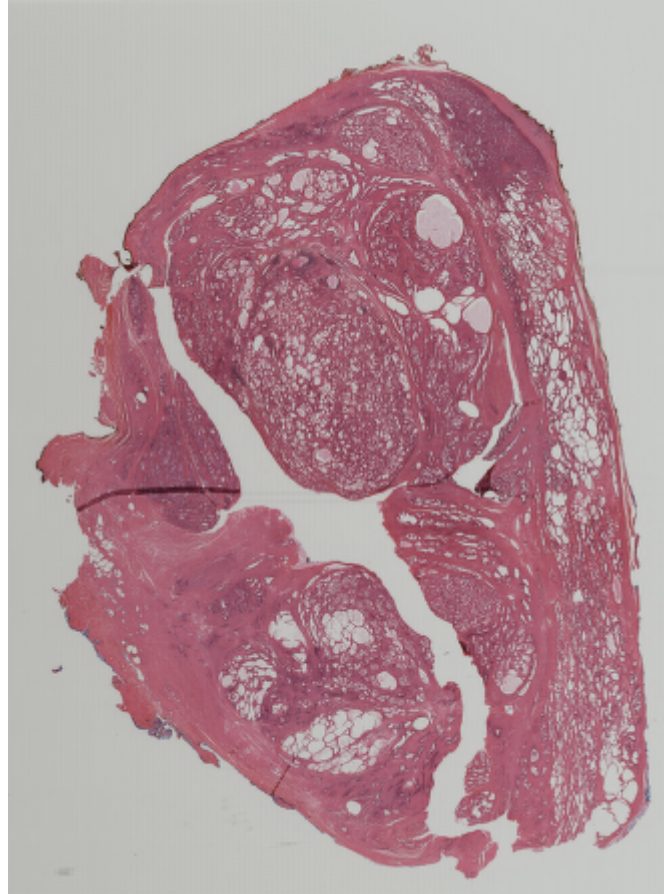
JPEG2000-koodivirran parametrien optimointi perustui teoreettisiin tietoihin standardin ominaisuuksista, visuaaliseen arviointiin kuvan latautumisessa, vuorovaikutusasteeseen navigoinnin yhteydessä (raahaus ja tarkkuustason vaihto) sekä pakkausprosessin vaatimaan aikaan. Testaukset suoritettiin käyttäen hyvin monipuolisia Kakadu `kdu_compress` (versio 5.2.2) sekä `kdu_show` (versio 5.2.2) -ohjelmistoja [Taubman, 2008].

6.5. JVS-ohjelmistopakettien kehitys

Kolme erillistä ohjelmistoa (JVScmp, JVSview ja JVSserv) sisältävän JVS-ohjelmistopaketti toteutettiin C++-ohjelmointikielellä. Ohjelmistopaketti on suunniteltu 32-bittisille Microsoft Windows®-käyttöjärjestelmille, mutta sitä voidaan käyttää myös 64-bittisillä Microsoft Windows®-käyttöjärjestelmillä. JVScmp hyödyntää ECW JPEG 2000 SDK -ohjelmistokehystä [ERM, 2008] sekä `libjpeg`- [IJG, 2008] ja `LibTIFF`-kirjastoa [Warmerdam, 2008]. JVSview sekä JVSserv perustuvat Kakadu JPEG2000 Framework -ohjelmistokehykseen [Taubman, 2008] ja käyttävät Microsoft Foundation Classes (MFC) -kirjastoa [Microsoft, 2008a]. Ohjelmistopakettien kehityksen apuvälineenä käytettiin Microsoft Visual Studio 2005 -ohjelmointiympäristöä [Microsoft, 2008b].

6.6. JPEG2000-suorituskykytestaus

JPEG2000-pakkaustestaukset sekä JPIP-nopeustestaukset suoritettiin Windows® XP -työasemalla, joka oli varustettu tuplaydinprosessorilla, 3 gigatavulla keskusmuistia, kahdella ATA-kiintolevyllä (ilman RAID-tukea) sekä 100 megabitin verkkoyhteydellä. Testauksissa käytetyn virtuaalinäytelasin yleisnäkymä on esitetty Kuvassa 19. Se on kooltaan $42\,865 \times 57\,222$ pikseliä ja se muodostuu kolmesta 8:n bitin värikanavasta, jolloin sen vaatima tila on noin 6.9 gigatavua. Virtuaalinäytelasi tallennettiin pakkaamattomana binäärimuotoiseksi PPM-tiedostoksi (Portable Pixmap). Lisäksi virtuaalinäytelasista luotiin joukko pienemmän resoluution versioita, puolittamalla resoluutio jokaisella askeleella.



Kuva 19. Yleisnäkymä JPEG2000-testauksissa käytetystä virtuaalinäytelasista. Kuva esittää 40 \times -objektiivilla kuvannettua eturauhasnäytettä, joka on värjätty hematoksylin-eosin-menetelmällä.

JPEG2000-pakkauksen nopeutta optimaalisilla koodivirran parametreilla mitattiin käyttäen itse kehitettyä JVScomp-pakkausohjelmistoa. Pakkausprosessin aikana valvottiin prosessorikuormitusta, keskusmuistin käyttöä, kiintolevyn kuormitusta ja sen keskimääräisiä luku- ja kirjoitusjonojen pituuksia (read and write queue lengths).

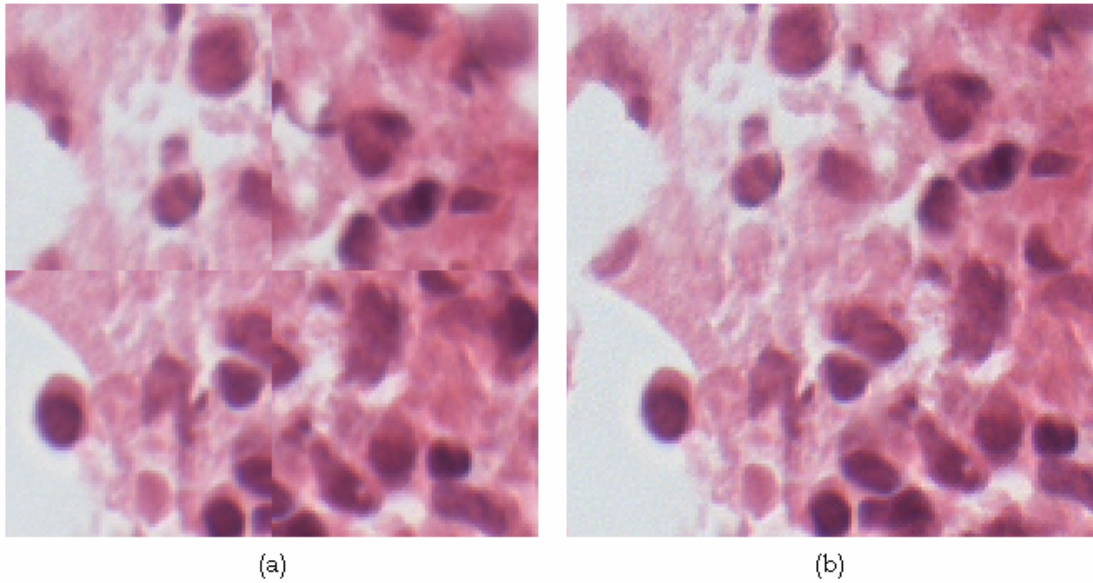
Kuvien jakelua JPIP-protokollan avulla mitattiin käyttäen itse kehitettyä JVSserv-ohjelmistoa, jota lisäksi verrattiin vaihtoehtoista kuvapalvelintekniikkaa hyödyntävään Zoomifyer EZTM -ohjelmistoon (versio 3.0) [Zoomify, 2008]. Siinä missä JVSserv toimii itsenäisenä palvelinsovelluksena, Zoomifyer käyttää ulkoista HTTP-palvelinta kuvatiedon välittämiseen. Suositettu Apache HTTP -palvelin (versio 2.2.4) [Apache, 2008] valittiin tähän tarkoitukseen. Mittaus ja vertailu suoritettiin käyttäen simulaatiota, jossa 10 lähiverkkoon liitettyä työasemaa kukin simuloivat 10:tä asiakasta, muodostaen näin 100:n asiakkaan joukon. Jokaisella asiakkaalla oli oma testikuva, joka kopioitiin päänäytteestä (Kuva 19). Palvelimen ja asiakkaiden välimuistit olivat asetettu pois käytöstä. Simulaatioiden aikana tarkkailtiin palvelimen kiintolevyn lukujonon pituutta, pro-

sensorikuormitusta, verkkoyhteyden kuormitusta, keskusmuistin käyttöä sekä arvioitiin subjektiivisesti asiakkaiden katselukokemuksen vuorovaikutusasetta.

7. Tulokset

7.1. LargeMontage-ohjelmisto

LargeMontage-ohjelmisto kehitettiin osittaiskuvantamisessa syntyvien kuva-alkioiden yhdistämiseen montaasikuvaksi siten, että kuva-alkioiden saumojen väliset epäjatkuvuudet eivät ole visuaalisesti havaittavissa. Yhdestä kuvannettavasta mikroskooppinäytteestä syntyy tuhansia kuva-alkioita, jotka mikroskooppia kontrolloiva ohjelmisto tallentaa tilapäisesti kiintolevyille. Kyseistä hakemistoa käyttäen LargeMontage lukee kuva-alkiot rasterijärjestyksessä (siis kuva-alkiorivi kerrallaan, näytteen vasemmasta yläreunasta alkaen) ja tuottaa näistä yhden suuren montaasikuvan. Kuva-alkioille määritellään joko kiinteät vaaka- ja pystysuuntaiset päällekkäisyydet (overlap) pikseleinä tai vaihtoehtoisesti hyödynnetään algoritmia, joka kohdistaa kuva-alkiot automaattisesti toisiinsa nähden [Thévenaz, 1998]. Prosessoinnin yhteydessä kuva-alkiot voidaan käsitellä unsharp mask -suotimella, joka monissa tapauksissa parantaa virtuaalinäytelasin laatua terävöimällä kuvaa, sitä kuitenkin liikaa vääristämättä. Lisäksi kuva-alkioiden sauma-alueita voidaan pehmentää käyttäen convolve-suodinta. Pehmennys on hyödyllinen silloin kun kuva-alkioiden asemointi ei onnistu täydellisesti ja sauma-alueilla esiintyy näkyviä epäjatkuvuuksia. Kuvassa 20 on esitetty neljän kuva-alkion sauma-alueilla esiintyvä epäjatkuvuus, joka johtuu virheellisestä alkioden asemoinnista, sekä vastaava kuva-alue prosessoituna LargeMontage-ohjelmistolla.



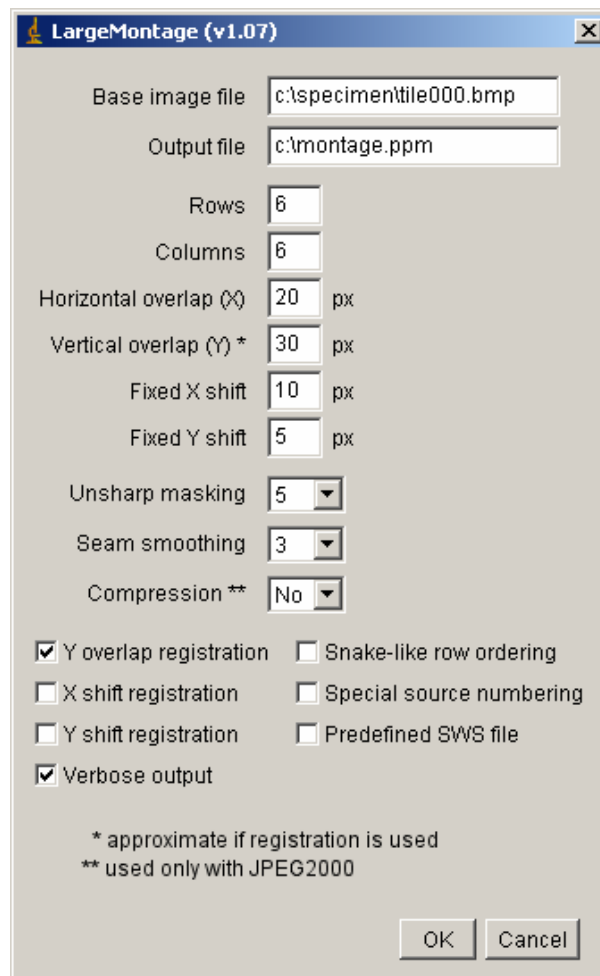
Kuva 20. Virheellisen asemoinnin aiheuttama epäjatkuvuus neljän kuva-alkion sauma-alueilla montaasikuvassa (a) ja LargeMontage-ohjelmistolla prosessoitu vastaava alue, jossa epäjatkuvuus on visuaalisesti häivytetty (b).

Ohjelmisto voi vastaanottaa syötteenä useita kuva-alkioiden tiedostomuotoja (kuten JPEG, BMP ja PNG) sekä kykenee tuottamaan montaasikuvan useissa tiedostomuodoissa (kuten PNM-formaattiperhe ja TIFF). PNM-formaattiperhe koostuu muodoista Portable Bitmap (PBM) -muodosta binaarikuville, Portable Graymap (PGM) -muodosta harmaasävykuville ja Portable Pixmap (PPM) -muodosta värikuville. PNM mahdollistaa yli neljän gigatavun tiedostokoon, joka on yleinen raja tavallisten kuvatiedostomuotojen (kuten TIFF) yhteydessä. Kuvan kokoa koskevat rajoitukset johtuvat yleensä 32-bittisestä rakenteesta tiedoston tavu- tai pikselikoon määrittelyn yhteydessä. Ohjelma kykenee tuottamaan myös JPEG2000-muotoisen montaasikuvan, mutta sen parametrisointi on suurilta osin JAI-kirjaston kiinteästi määrittelemä, eikä se ole optimaalinen virtuaalimikroskopian kannalta. Näin ollen, etenkin suurien virtuaalinäytelasien yhteydessä, montaasikuva on suositeltavaa luoda esimerkiksi PPM-muotoon, jonka jälkeen se voidaan pakata JPEG2000-muotoon erillisohjelman avulla (esimerkiksi JVScomp-ohjelmisto, joka esitellään myöhemmin).

LargeMontage-ohjelmistoa voidaan käyttää joko komentoriviltä tai graafisen käyttöliittymän avulla (Kuva 21). Lisäksi sitä voidaan kutsua sekä ImageJ-lisäosien että -makrojen kautta. Käyttäjä määrittelee ohjelmalle

- ensimmäisen kuva-alkion sijainnin,
- muodostuvan montaasikuvan sijainnin ja tiedostomuotoa,

- kuva-alkioiden lukumäärän rivi- ja sarakekohtaisesti,
- kuva-alkioiden vaaka- ja pystysuuntaisen päällekkäisyyden (overlap) pikseleinä,
- kuva-alkioiden vaaka- ja pystysuuntaisen kumulatiivisen siirtymän (shift) pikseleinä,
- unsharp mask -suotimen voimakkuuden,
- kuva-alkioiden sauma-alueiden pehmennyksessä käytettävän convolve-suotimen voimakkuuden,
- häviöllisen JPEG2000-pakkauksen voimakkuuden,
- automaattisen asemointialgoritmin sovelluskohteet (kuva-alkioiden pystysuuntainen päällekkäisyys ja vaaka- sekä pystysuuntainen kumulatiivinen siirtymä),
- käytettävän kuva-alkioiden rivijärjestyksen sekä
- muita pienempiä ohjelman suoritukseen liittyviä yksityiskohtia.



Kuva 21. LargeMontage-ohjelmiston graafinen käyttöliittymä.

Kuva-alkioiden nimeämiseen hakemiston sisällä ohjelmisto tarjoaa kaksi mahdollisuutta: 1) tiedostonimen loppuosa on numeroitu juoksevasti välillä $[0..n]$ (esimerkiksi "tile1.bmp", "tile2.bmp", "tile3.bmp" ja niin edelleen) tai 2) kuva-alkion tiedostonimi on muotoa "tileXXXYYY.bmp", missä XXX on kolmimerkkinen sarakkeen numero ja YYY on kolmimerkkinen rivin numero, ensimmäisen alkion ollessa 000000. Jos käytetään automaattista kuva-alkioiden asemointia, pystysuuntainen (eli kuva-alkioiden vaakasaumojen välinen) päällekkäisyysarvo on vain approksimaatio, jota asemointialgoritmi käyttää hyödykseen. Approksimointitarkkuus riippuu suurelta osin kuvan sisältämästä informaatiosta, mutta riittävän tarkan lopputuloksen saavuttamiseksi sen on oltava muutaman kymmenen pikselin kokoluokkaa. Jos automaattista asemointia ei käytetä, pystysuuntaista päällekkäisyyttä käytetään kiinteänä arvona. Vaakasuuntainen (eli kuva-alkioiden pystysaumojen välinen) päällekkäisyys on aina kiinteä arvo. Kumulatiivinen siirtymä on korjausarvo, joka lisätään jokaiseen käsiteltävään kuva-alkioon. Esimerkiksi vaakasuuntaisen kumulatiivisen siirtymän ollessa 5 pikseliä, rivin ensimmäistä kuva-alkiota siirretään (perusaseoinnin lisäksi) 5 pikseliä alaspäin, rivin toista alkia edelleen 5 pikseliä alaspäin ja niin edelleen. Kumulatiivinen siirtymä auttaa kompensoimaan mikroskoopin näytteensiirtopöydän mekaanista epätarkkuutta, josta aiheutuu ylimääräistä X- ja Y- suuntaista liikettä. Kuva-alkiot luetaan oletusarvoisesti rasterijärjestyksessä, mutta ohjelma tukee myös vuorotonta ("snake-like") rivijärjestystä (Kuva 22), jota käytetään minimoimaan mikroskooppipöydän liikkeiden epätarkkuudesta johtuvaa poikkeamaa, joka vaikuttaa sauma-alueiden päällekkäisyyteen.

1	2	3	4
5	6	7	8
9	10	11	12
13

(a)

1	2	3	4
8	7	6	5
9	10	11	12
...	13

(b)

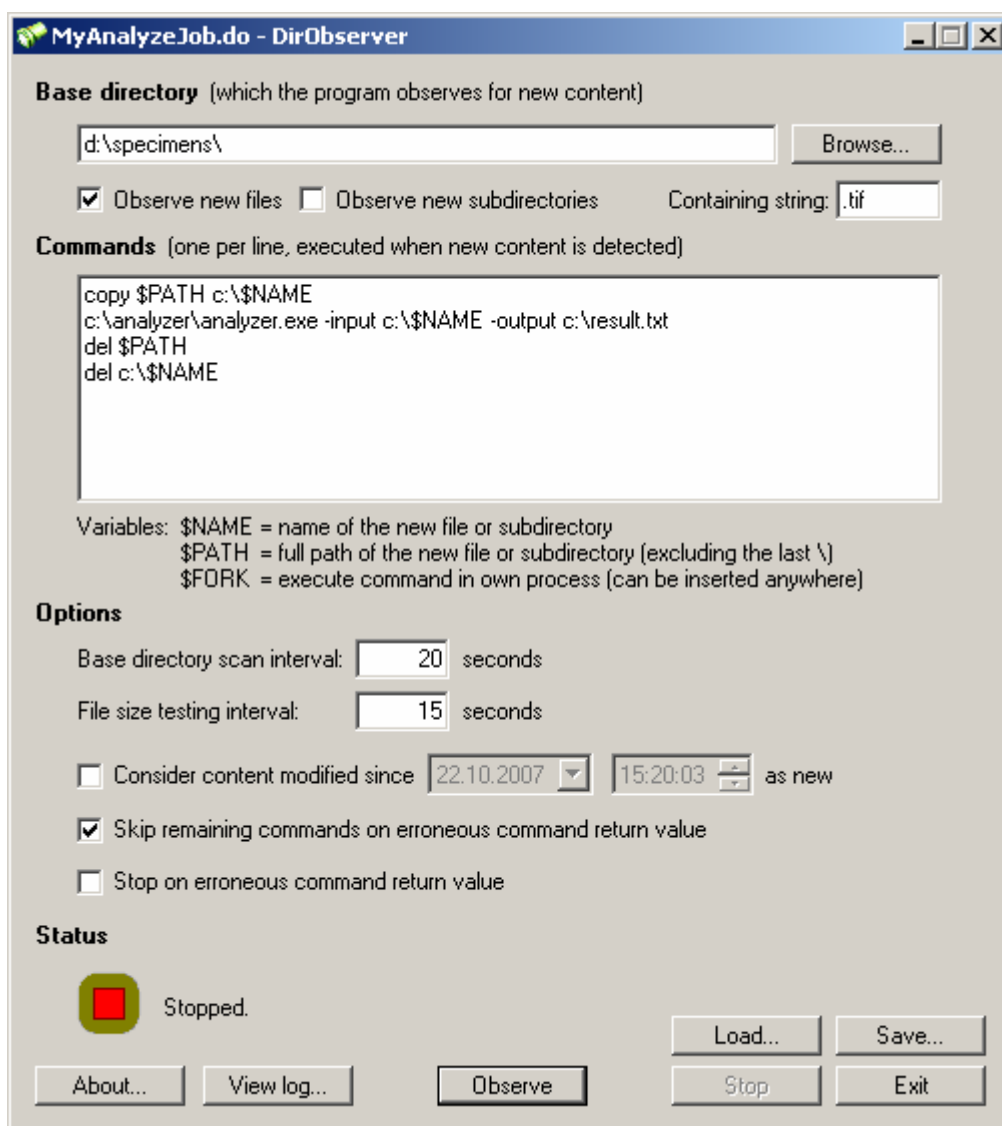
Kuva 22. Kuva-alkioiden rasteririvijärjestys (a) ja vuorottainen rivijärjestys (b).

LargeMontage on julkaistu GNU General Public License (GPL) -lisenssin [FSF, 2008] alaisuudessa ja ohjelman JAR-paketti, joka sisältää käännetyt luokkatiekiedostot sekä lähdekoodin, on saatavilla osoitteessa <http://iki.fi/vtuo/software/largemontage/>. Ohjelman verkkosivuilla on myös

saatavilla referenssiaineistoja, joiden avulla ohjelman toimintaa on mahdollista testata.

7.2. DirObserver-ohjelmisto

DirObserver-ohjelmisto kehitettiin montaasikuvan luomisen yhdistämiseen kuvantamislaitteistoon siten, että prosessi saadaan automatisoitua. Ohjelmistoa voidaan lisäksi hyödyntää muihinkin tarkoituksiin, esimerkiksi kuvantamisen jälkeen suoritettaviin jatkokäsittelyihin (esimerkiksi JPEG2000-pakkaus). Ohjelman graafinen käyttöliittymä on esitetty Kuvassa 23.



Kuva 23. DirObserver-ohjelmiston graafinen käyttöliittymä.

DirObserver tarkkailee hakemistopuiden sisältöjen muutoksia ja reagoi niihin suorittamalla käyttäjän määrittelemiä komentosarjoja (Algoritmi 1). Käyttäjä määrittelee tarkkailtavan hakemiston, jonka sisältöä voidaan tarkastella tiedosto- ja alihakemistokohtaisesti. Lisäksi tätä voidaan edelleen tarkentaa siten, että tiedostojen tai alihakemiston nimien on sisällettävä haluttu merkkijo-

non osa. Komentosarja kirjoitetaan yksi komento per yksi rivi -muotoon, joiden yhteydessä voidaan käyttää apumuuttujia $\$NAME$ (uuden, havaitun tiedoston tai alihakemiston nimi ilman hakemistopolkua), $\$PATH$ (uuden, havaitun tiedoston tai alihakemiston nimi hakemistopolun kera) ja $\$FORK$ (komennon suorittaminen uudessa prosessissa). Komennon suorittamisella eri prosessissa voidaan tehokkaasti hyödyntää moniydinprosessoreilla varustettua ajoympäristöä. Päähakemiston tarkkailu suoritetaan joka n . sekunti, missä n on käyttäjän määrittelemä aika. Ohjelman havaitessa uuden tiedoston (alihakemiston tapauksessa sen sisältämän uusimman tiedoston), jonka viimeisin muutos aika ei ole muuttunut sitten edellisen tarkkailukierroksen, tiedoston kokoa tarkkaillaan n . sekunnin ajan, missä n on käyttäjän määrittelemä aika. Jos tämän ajan kuluessa tiedoston koko pysyy samana, DirObserver suorittaa käyttäjän määrittelemän komentosarjan. Käyttäjällä on mahdollisuus ilmoittaa ohjelmalle aloitusaika, jonka perusteella ohjelma tulkitsee onko tiedosto tai alihakemisto uusi (aiheuttaen komentosarjan suorituksen) vai vanha (komentosarjaa ei suoriteta). Lisäksi käyttäjä voi komentovirheen ilmaantuessa määrätä ohjelman pysäyttämään komentosarjan suorituksen tai hyppäämään joidenkin kommentojen yli ennen seuraavaa tarkkailukierrosta. Lopuksi ohjelma tarjoaa loki-ominaisuuden sekä profiilien lataus- ja tallennusmahdollisuuden.

Algoritmi 1 DirObserver-ohjelmiston suorittama päähakemiston tarkkailualgoritmi.

```

1: if käyttäjä määritellyt sisällön uutuuden päivämäärän perusteella then
2:   aloitusAika ← käyttäjän määrittelemä aloitusaika
3: else
4:   aloitusAika ← järjestelmän aika
5: valmisLista ← päähakemiston tiedostot ja alihakemistot, joiden viimeinen muutos aika on
   vanhempi kuin aloitusAika
6: while joka  $n$ . sekunti, missä  $n$  on käyttäjän määrittelemä päähakemiston tarkkailuaika do
7:   tarkkailulista ← päähakemiston tiedostot ja alihakemistot, joiden viimeinen muutos aika
   on uudempi tai yhtä suuri kuin aloitusAika
8:   for jokainen tiedosto tai alihakemisto  $x \in$  tarkkailulista do
9:     if  $x$  on muuttunut sitten edellisen tarkkailukierroksen then
10:      if tiedostoon (tai alihakemiston uusimpaan tiedostoon) ei kirjoiteta  $m$ 
      sekuntiin, missä  $m$  on käyttäjän määrittelemä tarkistusaika then
11:        suorita käyttäjän määrittelemä komentosarja
12:        poista  $x$  tarkkailulistalta
13:      valmisLista ←  $x$ 

```

DirObserver on julkaistu GNU General Public License (GPL) -lisenssin [FSF, 2008] alaisuudessa ja ohjelman binääritiedostot (eli suorituskelpoiset tiedostot) sekä lähdekoodi on saatavilla osoitteessa <http://iki.fi/vtuo/software/dirobserver/>.

7.3. Optimaalinen JPEG2000-koodivirran parametrisointi

JPEG2000-standardin tehokkaan soveltamisen edellytyksenä virtuaalimikroskopiassa on optimaalisesti parametroitu JPEG2000-koodivirta. Tämän parametrisoinnin tulokset on tiivistetty Taulukossa 5.

Taulukko 5. JPEG2000-koodivirran optimaalinen parametrisointi virtuaalimikroskopian tarpeisiin.

JPEG2000-koodivirran parametri	Parametrin arvo
Pakkaussuhde (häviöllinen)	25:1 – 30:1
Wavelet-suodin	Palautumaton
DWT-muunnostasojen lukumäärä	10
Tile-ositus	Ei tarpeen
Progressiojärjestys	RPCL
Tile-virta-osat	1
Piirin koko	128 × 128
Koodiblokin koko	64 × 64
Laatukerrokset	1
PLT-markkerilohkot	Käytetään

Virtuaalimikroskopiassa häviöllisen pakkauksen käyttö on pakollista, joutuksen suuresta kuvainformaation määrästä. Gao *et al.* [2003] ovat osoittaneet, että korkean pakkaussuhteen omaavia, häviöllisesti pakattuja kuvia voidaan käyttää patologian diagnostisissa tutkimuksissa. Pakkaussuhteiden 25:1 – 30:1 käyttö virtuaalinäytelasin alueilla, jotka sisälsivät solu- tai kudosalueita, tuottivat lopputuloksen, jossa kuvan alkuperäisellä resoluutiolla ei ollut silmin havaittavia, häiritseviä artefakteja. Eri JPEG2000-pakkaussuhteiden vertailukuvat löytyvät tutkimuksen yhteyteen perustetuilta web-sivuilta (<http://jvsmicroscope.uta.fi/examples/>).

Kahden JPEG2000-standardin mahdollistaman wavelet-suotimen (palautumaton Daubechies ja palautuva Le Gall) välillä paremmaksi vaihtoehdoksi valittiin palautumaton suodin, sen tuottaessa paremman visuaalisen lopputuloksen. Jotta virtuaalimikroskopiaan olennaisesti liittyvä virtuaalinäytelasin yleisnäkymä olisi mahdollinen, DWT-muunnostasojen lukumäärän on oltava

riittävän suuri, jotta kuvasta saadaan purettua riittävän pienen resoluution versio. Yleisnäkyresoluution ollessa kokoluokkaa 256×256 pikseliä, sopivan DWT-muunnostasojen lukumäärän 150 megapikselin virtuaalinäytelasille (450 megatavua pakkaamattonta kuvainformaatiota) arvioitiin olevan 8, jonka jälkeen kuvainformaation nelinkertaistuessa muunnostasojen lukumäärä nousee yhdellä. Näin ollen päädyttiin sopivaan DWT-muunnostasojen lukumäärään 10, joka riittää poikkeuksellisen suurillekin virtuaalinäytelaseille, vaikka pienempien näytelasien tapauksessa se luo joitakin turhia resoluutiotasoja.

Koska JPEG2000-standardi tarjoaa tehokkaan satunnaisen hajakäsittelyn piirien avulla, tile-osituksen käyttöä ei nähty tarpeellisena, johtuen sen aiheuttamista haitoista kuvanlaatuun. Standardin mahdollistaessa 255 tile-virta-osan (kokorajoitus 4 GB) käytön yhden tile-virran sisällä, ilman tile-ositusta luodun kuvan maksimikoko on $4GB * 255 \approx 1TB$, joka vastaa kymmenien teratavujen kokoista pakkaamatonta kuvaa. Yhden tile-virta-osan katsottiin olevan riittävä suurillekin virtuaalinäytelaseille, mutta lukumäärä voidaan kuitenkin tarpeen vaatiessa kasvattaa. Progressiojärjestys Resolution-Position-Component-Layer (RPCL) havaittiin suotuisimmaksi järjestykseksi sen tarjotessa nopeimman resoluutiotaso-saannin yleisnäkyäikkunaa varten, joka ladataan ja näytetään virtuaalinäytelasin avauksen yhteydessä. Lisäksi se tarjoaa nopean satunnaisen hajakäsittelyn piirien avulla. Koska progressiojärjestyksen vaihtelusta saman kuvan sisällä ei ole virtuaalimikroskopian kannalta hyötyä, yhden tile-virta-osan katsottiin olevan tältäkin kannalta riittävä. JPIP-palvelimen näkökulmasta optimaalinen piirin koko vastaa koodiblokin kokoa, mutta tämä aiheuttaa lokaalin katselun yhteydessä satunnaisen hajakäsittelyn hidastumista. JPIP-palvelin pystyy kuitenkin dynaamisesti muuntamaan alkuperäisen koodivirran piirin kokoa ilman tehokkuushäviötä [Taubman and Prandolini, 2003]. Näin ollen sopivaksi piirin kooksi valittiin standardin teknisten ominaisuuksien perusteella 128×128 ja sopivaksi koodiblokin kooksi 64×64 . Nopeiden verkkoyhteyksien ollessa edellytys virtuaalimikroskopian soveltamisessa, useiden laatu-kerrosten käytön hyöty havaittiin ainoastaan hyvin hitailla JPIP-yhteyksillä. Satunnaisen hajakäsittelyn tehostamiseksi tile-virta-osan otsakkeeseen on suotuisaa sijoittaa pakettien pituudet määrittelevä PLT-markkerilohko. Jos pakattu kuva ylittää neljä gigatavua, jonka seurauksena tile-virta-osien lukumäärää on nostettava, on lisäksi hyödyllistä sijoittaa tile-virta-osien pituudet määrittävä TLM-markkerilohko koodivirran pääotsakkeen sisään.

7.4. JVS-ohjelmistopaketti

JVS-ohjelmistopaketti kehitettiin demonstroimaan JPEG2000-standardin soveltuvuutta virtuaalimikroskopiaan. Ohjelmistopaketti on pääosin suunniteltu Windows® 2000 ja XP -käyttöjärjestelmille, mutta se on myös Windows Vista™ -

yhteensopiva. Virtuaalinäytelasien JPEG2000-pakkausohjelmisto JVScomp on julkaistu vapaan lähdekoodin lisenssin alaisuudessa ja JVSview-katseluohjelmiston sekä JVSserv-palvelinohjelmiston binääritiedostot (eli suorituskelpoiset tiedostot) on julkaistu vapaaseen, ei-kaupalliseen käyttöön. Lisäksi ohjelmistopakettin rinnalle kehitettiin XML-skeema JVSschema, joka kuvaa formaalisti JVS-ohjelmistopakettin käyttämän metatiedon rakenteen. Ohjelmistopaketti on saatavilla tutkimuksen yhteyteen perustetuilta web-sivuilta (<http://jvsmicroscope.uta.fi/>). Web-sivusto sisältää myös julkisen JPIP-palvelimen sekä useita virtuaalinäytelasikokoelmia, joiden katselu onnistuu JVSview-ohjelmiston avulla.

JVScomp. Tämä ohjelmisto tuottaa optimaalisesti parametrisoituja JPEG2000-virtuaalinäytelaseja, joita voidaan katsella ja jakaa JVSview- sekä JVSserv-ohjelmistojen avulla (Taulukko 6). JVScomp on komentorivipohjainen ja se noudattelee kohdassa 7.3. esitettyjä optimaalisia koodivirran parametreja. Ohjelmisto kykenee vastaanottamaan useita eri syötetiedostoformaatteja (kuten PPM, BMP ja JPEG) ja se sisältää tuen tuplaydinproessori-ympäristöille, jotka tunnustetaan automaattisesti. Vaikka muodostuvan koodivirran koon ennalta määrittäminen on hyödyllinen JPEG2000-standardin ominaisuus, se ei ole virtuaalimikroskopian kannalta merkityksellinen, jossa tavoitteena on yleensä mahdollisimman pieni koodivirran koko annettuun laatuun nähden. Tämän vuoksi JVScomp on suunniteltu suorittamaan JPEG2000-pakkaus mahdollisimman tehokkaasti noudattaen käyttäjän määrittelemää lopullisen pakkaus-suhteen arviota (target compression ratio). Virtuaalinäytelasien tapauksessa tämä arvio on pakkaussuhde, jota käytetään solu- ja kudosalueilla. Vastaavasti jos virtuaalinäytelasi sisältää paljon alueita, jotka ovat erityisen herkkiä pakkaukselle (esimerkiksi homogeeniset taustalasiaalueet, jotka eivät sisällä kudosta), JVScomp käyttää näillä alueilla tiiviimpää pakkaussuhdetta. Näin ollen saavutetaan suurempi kokonaispakkaussuhde sekä pienempi tiedostokoko.

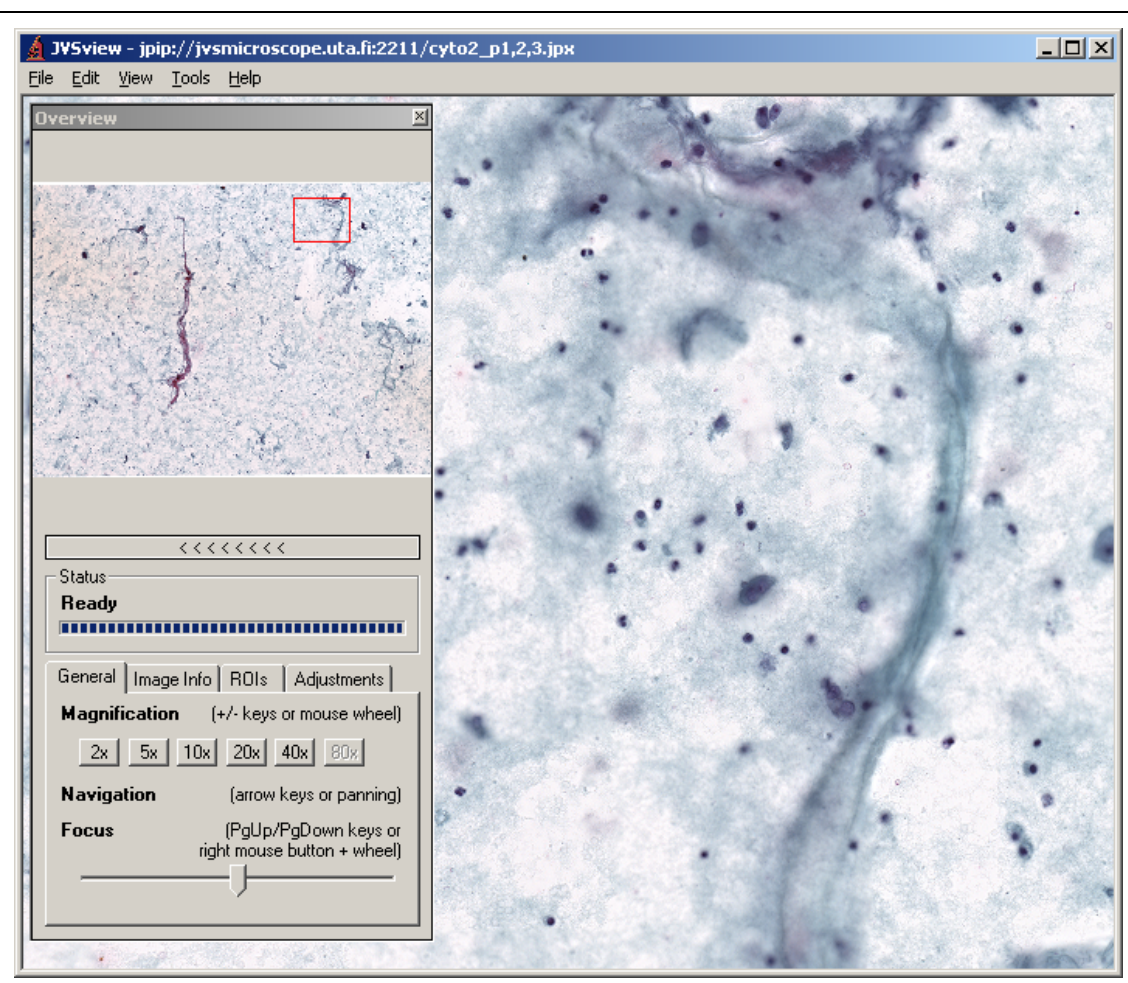
Taulukko 6. JVScomp JPEG2000-virtuaalinäytelasien pakkausohjelmiston keskeisimmät ominaisuudet.

Ominaisuuksien lista
Optimoitu JPEG2000-koodivirran parametrisointi virtuaalimikroskopian tarpeisiin
Pakkaustehokkuus maksimoitu ilman ennakoitavaa tiedostokokoa
Tuki tuplaydinprosessoreille
Tuki useille eri syötetiedostomaateille (kuten PPM, BMP ja JPEG)

JVSview. Tämä ohjelmisto on JPEG2000-virtuaalinäytelasien katselusovellus, joka toimii identtisesti sekä paikallisten näytelasien katselun että verkon yli tapahtuvan JPIP-pohjaisen katselun yhteydessä (Taulukko 7). Ohjelma on graafinen sovellus, jonka käyttöliittymä on esitetty Kuvassa 24. *JVSview* tarjoaa käyttäjälle yleisnäkömökkunan, joka näyttää virtuaalinäytelasista pienen tarkkuustason version. Yleisnäkömän rinnalla on päänäköm, joka esittää (yleensä) vain pientä kuvan aluetta, jonka sijainti koko näytteen kontekstissa esitetään yleisnäkömökkunassa punaisella suorakaiteella. Ohjelmiston tarjoamat tarkkuustasot kuvastavat eri mikroskooppiobjektiveja. Käyttäjä voi navigoida näytettä hiirellä "raahaamalla", käyttämällä nuolinäppäimiä tai valitsemalla haluttu alue yleisnäkömökkunasta. Ohjelmisto tarjoaa käyttäjälle mahdollisuuden säätää kuvan kirkkautta, kontrastia sekä värikylläisyyttä. Virtuaalinäytelasi voidaan esittää täyden ruudun katselutilassa, jonka avulla voidaan maksimoida käytettävissä oleva kuvapinta-ala. Lisäksi *JVSview* tarjoaa usean näyttölaitteen tuen, jonka avulla täyden ruudun katselutila levitetään kaikille käytössä oleville näyttölaitteille. Näytteen katselun vuorovaikutteisuus riippuu käytetyn laitteiston tehokkuudesta. Riittävän nopeaan katseluun käy tavanomainen yhden prosessorin työasema, mutta *JVSview* tukee myös moniydinprosessoreilla varustettuja laitteistokokoonpanoja, jolloin JPEG2000-proessointi nopeutuu.

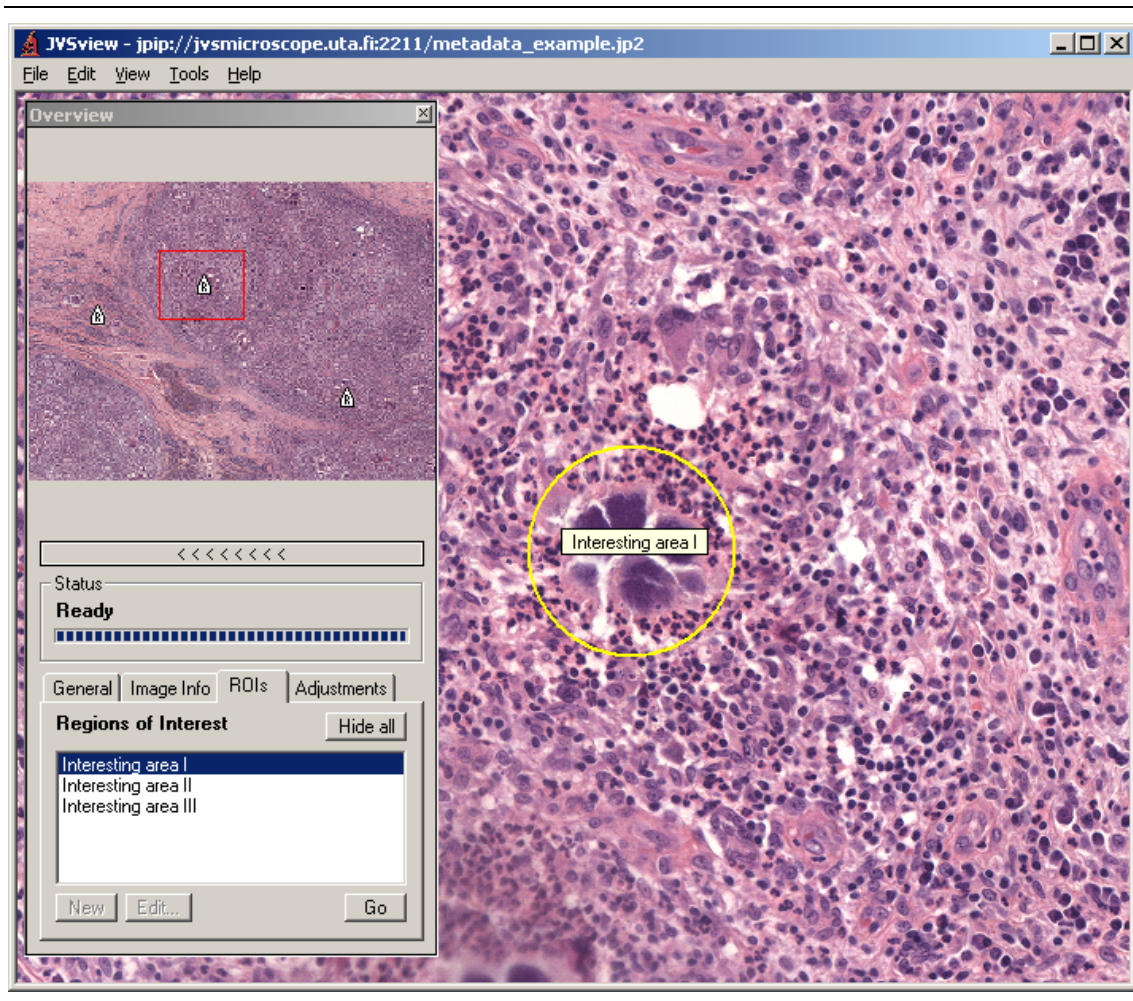
Taulukko 7. JVSview JPEG2000-virtuaalinäytelasien katseluohjelmiston keskeisimmät ominaisuudet.

Ominaisuuksien lista
Kuvien paikallinen katselu & etäkatselu hyödyntäen JPIP-protokollaa
Koko näytteen kattava yleisnäkömääkkuna
Mikroskooppiobjektiveja vastaavat tarkkuustasot
Navigointi hiirellä sekä näppäimistöllä
Kirkkauden, kontrastin ja värikylläisyyden säätömahdollisuus
Täyden ruudun katselutilassa usean näyttölaitteen tuki
Tuki useille kuvatasoille (kuten tarkennustasot)
Kuvaan liittyvän oheistiedon ja haluttujen kuva-alueiden (ROI) tallentaminen XML-metatietona
Toiminnallinen linkki ulkoisen kuva-analyysiohjelmiston kanssa (ImageJ)



Kuva 24. JVSview JPEG2000-virtuaalinäytelasien katseluohjelmiston graafinen käyttöliittymä.

JPEG2000-standardin käyttö mahdollistaa monia uusia virtuaalimikroskopian sovelluskohteita, joiden esittelemiseksi JVSview-ohjelmistoon on sisällytetty tarkennustaso-ominaisuuden. Tämän avulla käyttäjä voi vapaasti vaihdella eri soluja tai kudosta esittävien kuvakerrosten välillä, jotka ovat kuvanettu eri tarkennustasoilla (katso esimerkkejä web-sivuilta <http://jvsmicroscope.uta.fi/examples/>). Tarkennustasojen käyttö tapahtuu siten, että kaksi tasoa näytetään päällekkäin osittain läpinäkyvinä ja niiden läpinäkyvyysuhdetta muutetaan käyttäjän manipuloidessa liukukosketinta (esimerkiksi kun tarkennustaso a esitetään 25 % läpinäkyvyydellä, niin tällöin tarkennustaso b esitetään 75 % läpinäkyvyydellä). Lisäksi JVSview tarjoaa mahdollisuuden upottaa virtuaalinäytelasitiedostojen otsakkeisiin tekstuaalista metatietoa, joka voi sisältää esimerkiksi kuvantamistarkkuuden, kyseessä olevan elimen nimen, histopatologisen diagnoosin sekä käytetyn värjäysmenetelmän. Metatieto on upotettu JP2- tai JPX-tiedostoformaattiin XML-muotoisena, jonka rakenne on kuvattu formaalisti käyttäen JVS-ohjelmistopakettin rinnalle kehitettyä XML-skeemaa JVSschema (<http://jvsmicroscope.uta.fi/jvsschema/>). Metatietoa voidaan lisätä ja muokata käyttäen JVSview-ohjelmiston sisältämiä työkaluja. Metatiedon käyttö on tarkoitettu pääasiallisesti näytekoelmiin, joihin ei liity potilastietoja (esimerkiksi opetuskäyttöön tarkoitettut kokoelmat). Metatietoa hyödyntämällä JVSview tarjoaa mahdollisuuden määrittellä ja tallentaa virtuaalinäytelasiin mielenkiintoisia kuva-alueita (Region Of Interest, ROI), jotka määrittellään halutun kokoisella ja värisellä ellipsillä sekä tekstillä, joka näytetään hiiren cursorin ollessa kyseisen alueen päällä (Kuva 25). Lisäksi virtuaalinäytelasin sisältämät ROI-alueet esitetään käyttäjälle yleisnäkymäikkunassa graafisin symbolein ja tekstuaalisena listana, joilla käyttäjä voi nopeasti siirtyä halutulle alueelle. JVSview sisältää toiminnallisen linkin ImageJ-kuvananalyysiohjelmiston [Rasband, 2008] kanssa. Tämä tarjoaa helpon tavan analysoida kuvan pituuksia ja pinta-aloja sekä suorittaa esimerkiksi partikkelilaskentaa (kuten kuvassa näkyviä soluja ja tumia) käyttäen Cell Counter -lisäosaa [De Vos, 2008].

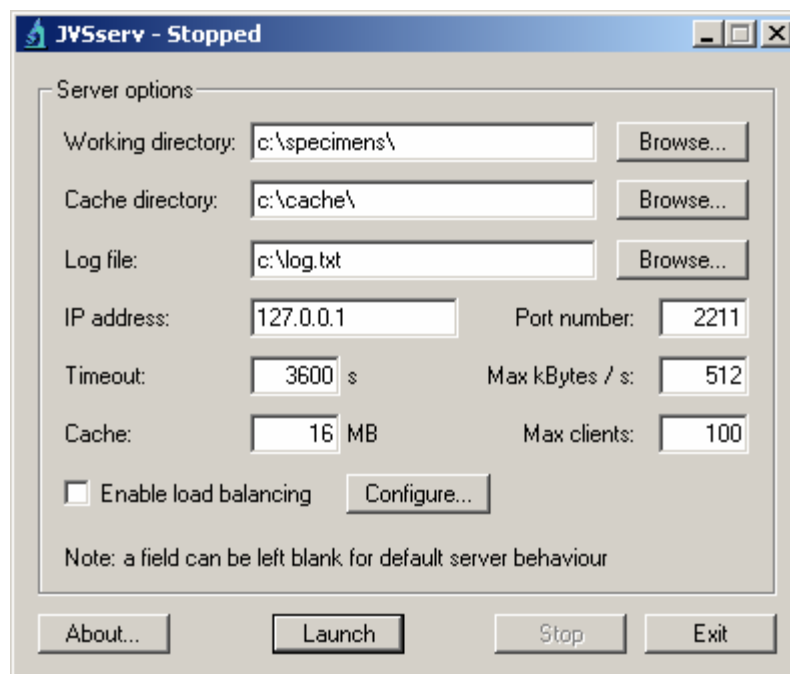


Kuva 25. JVSview-ohjelmiston tarjoama mielenkiintoisten näkymäalueiden (Region Of Interest, ROI) määrittely ja tallennus käyttäen metatietoa.

JVSServ. Tämä ohjelmisto on JPEG2000-virtuaalinäytelasien kuvapalvelin, joka käyttää tiedonsiirrossa JPIP-protokollaa (Taulukko 8). Palvelinta voidaan hallinnoida sekä komentoriviltä että graafisen käyttöliittymän avulla (Kuva 26). JVSServ perustuu kdu_server-ohjelmistoon [Taubman, 2008], jota on muutettu lisäämällä graafinen käyttöliittymä, poistamalla kuvan kokoon liittyviä rajoituksia sekä muuttamalla palvelimen viestinvälitysjärjestelmää. Tästä syystä JVSServ-ohjelmistoa tulisi käyttää ainoastaan JVS-ohjelmistopakettien yhteydessä. JVSServ tukee useita yhtäaikaista asiakasyhteyksiä sekä käyttää tavanomaista jonopohjaista (First In, First Out – FIFO) välimuistimallia, jonka avulla palvelimen välimuistissa säilytetään eniten käsitelty kuvainformaatio. Ohjelmisto tukee lisäksi kuormanjakamista (load balancing), jonka avulla pääpalvelimeen kohdistuva työmäärä voidaan jakaa useiden alipalvelimien kesken, täten mahdollistaen järjestelmän kokonaiskäyttäjäkapasiteetin nostamisen.

Taulukko 8. JVSserv JPEG2000-virtuaalinäytelasien kuvapalvelinohjelmiston keskeisimmät ominaisuudet.

Ominaisuuksien lista
Virtuaalinäytelasien välitys tietoverkkojen yli käyttäen JPIP-protokollaa
Palvelimen hallinointi sekä komentoriviltä että graafisen käyttöliittymän avulla
Tuki usealle yhtäaikaaiselle asiakkaalle
Kuormanjako (load balancing)



Kuva 26. JVSserv-palvelinohjelmiston graafinen käyttöliittymä.

7.5. JPEG2000-pakkauksen suorituskyky

JPEG2000-pakkauksen suorituskykyä mitattiin, jotta voitiin varmistua sen sopivuudesta automaattisen mikroskooppikuvantamisen yhteyteen. Pakkaustestauksessa käytetyn virtuaalinäytelasin (Kuva 19), ja siitä johdettujen pienempien versioiden perusteella, JPEG2000-pakkauksen vaatima aika on karkeasti lineaarinen suhteessa alkuperäisen kuvan kokoon. Keskusmuistin käyttö prosessin aikana oli pientä: 7 gigatavun näyte aiheutti JVScomp-ohjelmistoa käytettäessä vain noin 70 megatavun muistikuormituksen. Pakkausnopeudessa JVScomp ylsi käytetyllä tuplaydinkokoonpanolla 50 gigatavuun per tunti. Rajoittava tekijä oli prosessorinopeus, sillä lähdekiintolevy, johon suurin osa levyoperaati-

oista kohdistuu, olisi pystynyt tehokkaasti käsittelemään yli kaksi kertaa enemmän lukupyynnöitä.

7.6. JPIP-kuvapalvelimen suorituskyky

JPIP-protokollan kuvanjakelunopeutta arvioitiin käyttämällä JVSserv-kuvapalvelinohjelmistoa. Lisäksi tätä verrattiin yleisesti käytössä olevaan Zoomifyer EZTM -kuvapalvelinohjelmistoon. Molemmilla ohjelmistoilla virtuaalinäytelasien jakelun rajoittavana tekijänä oli kiintolevyjen lukunopeus; prosessorikuormitus oli tasaisesti alle 50 %, kuten myös keskusmuistin käyttö. Verkkoyhteyden (100 Mbit/s) käyttöaste oli vain noin 25 % kokonaiskaistanleveydestä*. Asiakkaiden lukumäärän noustessa, keskimääräinen kiintolevyjen lukujonojen pituus ylitti nopeasti tehokkaan rajan (rajana käytettiin arvoa 2.0).

Käyttämällä virtuaalinäytelasia, joka oli kopioitu jokaisella asiakkaalla, JVSserv pystyi tehokkaasti palvelemaan noin 30:tä samanaikaista asiakasta, Zoomifyer-ohjelmiston (Apache HTTP -palvelimen välityksellä) kyetessä palvelemaan tehokkaasti noin 20 yhtäaikaista asiakasta. Tämä eroavaisuuden voi selittää sillä, että Zoomifyer jakaa virtuaalinäytelasin tuhansiin pieniin JPEG-kuvatiedostoihin, joiden lukeminen aiheuttaa ylimääräistä levykuormitusta. Jos testauksissa olisi käytetty välimuistia, sekä JVSserv että Zoomifyer olisivat hyötäneet asiakkaista, jotka katselevat samaa virtuaalinäytelasia. Jos oletetaan, että jotkin asiakkaat katselevat samaa näytelasia, ja että jokainen asiakas sietää hie-man katseluvivettä, JVSserv-ohjelmiston voidaan arvioida pystyvän palvelemaan noin 50:tä samanaikaista asiakasta. Lisäksi JVSserv sisältää kuormanjakominaisuuden, jota voidaan käyttää jakamaan asiakkaat usealle eri alipalvelimelle. Näin ollen, käyttämällä neljää tai viittä alipalvelintä, tavanomaisella laitteistokokoonpanolla ja 100 megabitin verkkoyhteydellä varustetun JVSserv-palvelimen voidaan olettaa pystyvän palvelemaan noin 200 – 300 yhtäaikaista asiakasta.

* Koska verkkoyhteyksien todellinen kaistanleveys on harvoin lähellä sen teoreettista maksiminopeutta, testauksissa käytetty kokonaiskaistanleveys mitattiin kokeellisesti, ja näin ollen se vastasi verkon todellista maksiminopeutta.

8. Pohdinta

Liittyessäni professori Isolan tutkimusryhmään vuonna 2005, tavoitteena oli kehittää ratkaisuja virtuaalimikroskopiaan liittyviin ongelmakohtiin (luku 3). Ensimmäinen selkeä pullonkaula näytelasien kuvantamisprosessissa oli ositaiskuvantamisessa syntyvien kuva-alkioiden yhdistäminen kuvamontaasiksi siten, että montaasin sauma-alueille ei muodostu näkyviä epäjatkuvuuksia. Mikään tunnettu ja avoin ohjelmisto ei pystynyt suorittamaan tätä riittävän laadukkaasti sitten, että se olisi yhtäaikaisesti kyennyt myös käsittelemään virtuaalimikroskopiassa käytettyjä erittäin suuria kuvakokoja. Tätä tarkoitusta varten kehitettiin LargeMontage, jonka todettiin oman tutkimusryhmän rutiinikäytössä tuottavan riittävän hyvälaatuisia kuvamontaaseja, riittävällä nopeudella. Ohjelmisto – mukaanlukien sen lähdekoodi – asetettiin vapaasti saataville ja se onkin kasvattanut suosiotaan ImageJ-ohjelmiston web-sivujen ansiosta, jossa ylläpidetään listausta kaikista julkisista ImageJ-lisäosista. Nykyisin ohjelmiston käyttö oman tutkimusryhmämme sisällä on vähentynyt, sillä kaupallisten mikrokooppiskannereiden kuvantamisvirheiden määrä on laskenut sekä kuvantamisohjelmistojen tuottamat montaasikuvat ovat jo riittävän laadukkaita.

LargeMontage-ohjelmiston kehittämisen jälkeen kuvantamisprosessi haluttiin saada automatisoitua siten, että kuvamontaasin luominen tapahtuu ilman käyttäjältä vaadittavaa manuaalista välivaihetta. Tätä varten kehitettiin DirObserver-ohjelmisto, joka tulisi toimimaan kuvantamisprosessin keskeisenä hallinnoijana: ohjelmisto tarkkailee syntyviä kuva-alkioita, joiden valmistuttua se aloittaa kuvamontaasin luomisen kutsumalla LargeMontage-ohjelmistoa ja lopuksi huolehtii kuva-alkioiden sekä muiden tilapäistietojen siivoamisesta. Ohjelmisto suunniteltiin joustavaksi ja laajennettavaksi, sillä jo kehityksen aikana suunniteltiin yhteinäisen kuvaformaatin kartoittamista virtuaalimikroskopian käyttöön, joka vaatisi kuvantamisprosessiin lisää välivaiheita. DirObserver todettiin oman tutkimusryhmämme rutiinikäytössä toimivaksi ja se julkaistiin vapaan lähdekoodin kera avoimeen levitykseen. On huomattavaa, että ohjelmisto ei ole erityisesti rajattu vain mikroskopian kuvantamissovelluksiin, vaan sitä voidaan käyttää ohjaamaan myös muunlaisia prosesseja.

8.1. JPEG2000 – yhtenäinen kuvaformaatti virtuaalimikroskopiaan

Nykyisin monet virtuaalinäytelasien skannerivalmistajat käyttävät suljettuja kuvaformaatteja, jotka estävät virtuaalinäytelasien siirtämisen eri valmistajien järjestelmien välillä. Lisäksi valmistajien tuki näille suljetuille, epästandardeille formateille voi olla hyvin lyhytikäinen. Näistä syistä avoimeen standardiin perustuva kuvaformaatti on ehdoton edellytys suurimittaisten virtuaalinäyte-

lasikokoelmien perustamiseksi. Kokoelmat kasvavat ja niitä hyödynnetään useiden vuosien, jopa vuosikymmenten ajan. Koska automaattisten näytelasis-kannereiden tekninen kehitys tulee jatkumaan nopeaa tahtia, on hyödyllistä erottaa itse kuvantamisprosessi virtuaalinäytelasien arkistoinnista ja jakelusta. Näin varmistetaan, että vanhoilla laitteistoilla kuvannetut virtuaalinäytelasit ovat yhteensopivia uusien laitteistojen kanssa. Kaikki edellä mainitut vaatimukset täyttyvät JPEG2000-standardin yhteydessä, joka on monipuolisilta ominaisuuksiltaan ideaalinen valinta virtuaalimikroskopian yhtenäiseksi kuvaformaatiksi. Jotta JPEG2000-standardia voidaan tehokkaasti hyödyntää virtuaalimikroskopiassa, on JPEG2000-koodivirran parametrisointi oltava optimaalinen suurille kuvako'uille. Optimaalinen koodivirtaparametrisointi määriteltiin ja sitä käytetään tutkimusryhmässämme kuvannetuissa virtuaalinäytelaseissa, joissa se on osoittautunut toimivaksi.

Optimaalisten JPEG2000-koodivirran parametrien määrittämisen jälkeen kehitettiin niihin perustuva ohjelmistopaketti, joka soveltuu virtuaalimikroskopian käyttötarkoituksiin. Ohjelmistopaketti sai nimekseen JVS (JPEG2000 Virtual Slide), joka koostuu virtuaalinäytelasien JPEG2000-pakkausohjelmasta JVScomp, katseluohjelmasta JVSview sekä kuvapalvelinohjelmasta JVSserv. Ohjelmistopaketin käyttämä metatieto kuvattiin formaalisti JVSschema-nimisen XML-skeeman avulla. Paketti julkaistiin kokonaisuudessaan vapaaseen käyttöön, mutta ainoastaan JVScomp asetettiin vapaan lähdekoodin alaisuuteen, sillä, johtuen käytetyn ohjelmistokehityksen lisensointiehdosta, JVSview- sekä JVSserv-ohjelmistojen lähdekoodia ei ollut mahdollista julkaista. Ohjelmistopaketti on todettu toimivaksi ja sitä hyödynnetään tutkimusryhmässämme rutiininomaisesti. Lisäksi tutkimuksen yhteyteen avattiin julkinen JVSserv-JPIP-palvelimen (<http://jvsmicroscope.uta.fi/>), joka sisältää virtuaalinäytelasikoelmia, jotka on luotu käyttäen JVScomp-ohjelmistoa, ja joiden katselu onnistuu käyttäen JVSview-ohjelmistoa.

Virtuaalinäytelasien pakkaus JPEG2000-muotoon on laskennallisesti vaativa prosessi. Kuten JPEG2000-suorituskykytestauksemme osoittavat, rajoittava tekijä tavanomaisella tuplaydinprosessorilla varustetulla työasemalla on itse prosessorin nopeus. Kuitenkin, koska JPEG2000-pakkausohjelmistot voidaan suunnitella hyödyntämään moniydinprosessoreita, kiintolevyjen lukunopeuden maksimiraja saavutetaan nopeasti, ja pakkausprosessi muuttuu kiintolevyn lukunopeudesta riippuvaiseksi. Hyödyntämällä peilattua RAID-tallennustekniikkaa, jossa useiden toisiinsa liitettyjen kiintolevyjen sisältö on identtinen, lukunopeutta ja haku-aikaa voidaan kasvattaa kiintolevyjen lukumäärän kerrannaisena. Näin ollen voidaan arvioida, että neliydinprosessoria hyödyntävä JPEG2000-pakkausohjelma kykenee pakkaamaan yli 100 gigatavua

kuvadataa per tunti, joka vie vähemmän aikaa kuin mitä nykyisiltä kuvantamislaitteistoilta kestää digitoida 20 × 20 mm -kokoinen mikroskooppinäyte, käyttäen tarkkuutta 0.23 μm per pikseli [Rojo *et al.*, 2006]. Tästä syystä näyttäisi siltä, että JPEG2000-pakkauksen laskennalliset vaatimukset virtuaalimikroskoopiassa voidaan täyttää käyttäen ohjelmistopohjaisia ratkaisuja, eikä laitteistokiihdytettyjä toteutuksia (esimerkiksi VLSI-mikropiirit [Acharya, 2006]) ole tarpeen käyttää.

JPEG2000 eroaa edeltäjästään, JPEG-standardista, monin tavoin. Virtuaalimikroskopian kannalta tärkein edistysaskel on JPEG2000-standardin käytännössä rajaton kuvakoko (sekä pikseli- että tavukoko) ja mahdollisuus kuvan satunnaiseen hajakäsittelyyn. Lisäksi virtuaalinäytelaseissa käytetty yleisnäkyä voidaan purkaa JPEG2000-koodivirrasta hyvin nopeasti, sillä eri mikroskoopiobjektiiveja vastaavat tarkkuustasot ovat DWT-muunnoksen luontaisia oheistuotoksia. Näin ollen, koosta riippumatta, yksi JPEG2000-tiedosto voi sisältää koko virtuaalinäytelasin. Käyttämällä JPX-tiedostoformaattia, yhden tiedoston sisälle voidaan upottaa useita koodivirtoja, jotka voivat vastata esimerkiksi eri tarkennustasoja. Kuvanlaadullisesti JPEG2000 on ylivoimainen JPEG-formaattiin nähden, etenkin häviöttömän pakkauksen suurilla pakkaussuhteilla. Pakkaamaton, 7-gigatavuinen, 50 000 × 50 000 pikselin kokoinen virtuaalinäytelasi voidaan pakata 300-megatavuiseksi JPEG2000-kuvaksi riittävän hyvällä laadulla. Jos näyte sisältää suuria taustalasiaalueita, pakatun kuvan koko pienenee merkittävästi, ilman kuvanlaadun heikentymistä solu- ja kudosalueilla.

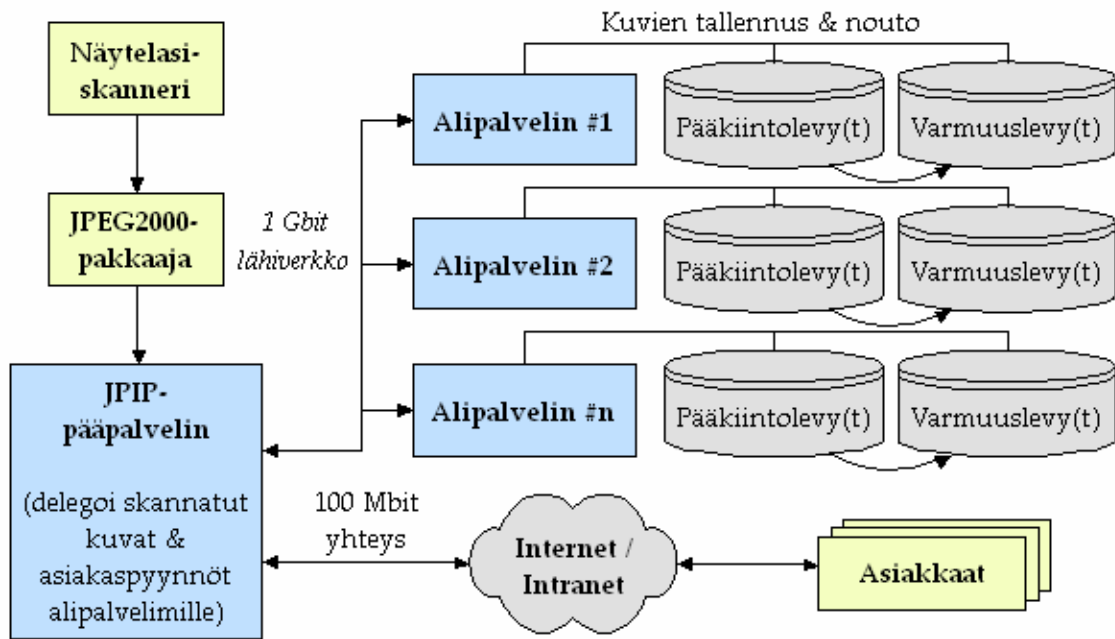
Hiljattain esiteltyä BigTIFF-laajennusta [Aware, 2008] yleisesti käytettyyn TIFF-tiedostoformaattiin on ehdotettu mahdolliseksi virtuaalimikroskopian kuvaformaattiksi. Se ei sisällä TIFF-formaatin neljän gigatavun tiedostokokorajoitusta, mahdollistaen näin erittäin suurten kuvien käsittelyn. BigTIFF ei itsessään määrittele uutta pakkausalgoritmia, vaan se voi sisältää esimerkiksi useita JPEG- tai LZW-pakattuja kuvabittivirtoja, jotka on organisoitu tile-ositusta käyttäen. Käyttämällä JPEG-pakkausta BigTIFF-formaatin sisällä menetetään häviöllisen JPEG2000-pakkauksen tuomat hyödyt. JPEG2000-pakkauksen sisällyttäminen BigTIFF-formaatin sisään on kyseenalaista, sillä JPEG2000-koodivirran rakenne itsessään mahdollistaa, muiden ominaisuuksien ohessa, tehokkaan satunnaisen hajakäsittelyn. Lisäksi JPEG2000-standardi määrittelee JPIP-protokollan kuvien jakamiseen tietoverkkojen yli, jollaista BigTIFF ei kirjoitushetkellä vielä sisällä.

Johtuen JVS-ohjelmistopakettien esimerkinomaisesta luonteesta, tietoverkkojen yli tapahtuvaa liikennöintiä ei salata. Jos tarvetta salatulle yhteydelle kuitenkin ilmenee, ohjelmistopakettien yhteydessä on mahdollista hyödyntää stan-

dardoituja, suojattuja tietoliikenneprotokollia (kuten SSH tai TLS) ohjaamaan suojaamatonta JPIP-liikennettä niiden läpi. Esimerkiksi TLS-protokollaa voidaan käyttää Virtual Private Networking (VPN) -tekniikan kanssa, jossa JPIP-palvelimen rinnalle perustetaan VPN-palvelin ja JPEG2000-katseluohjelmaasiakkaan rinnalle perustetaan VPN-asiakas. VPN-asiakas toimii välittäjänä JPEG2000-asiakkaalle, toimittaen katseluohjelman pyynnöt salatun yhteyden yli VPN-palvelimelle, ja sieltä edelleen varsinaiselle JPIP-palvelimelle, sekä päinvastoin.

8.2. Malli JPEG2000-pohjaiselle virtuaalimikroskopiajärjestelmälle

Kuten tutkielman tuloksista käy ilmi, JPEG2000 tarjoaa ratkaisut moniin, olennaisiin virtuaalimikroskopian ongelmiin. Kuvassa 27 on esitetty malli helposti päivitettävissä olevalle JPEG2000-pohjaiselle virtuaalimikroskopiajärjestelmälle, joka hyödyntää kuormanjakoa. Järjestelmä koostuu yhdestä JPIP-pääpalvelimesta ja siihen liittyvistä useista alipalvelimista. Pääpalvelin on varustettu kahdella verkkoliitynnällä: 100-megabittinen asiakasyhteys (joko internet tai intranet) ja 1-gigabittinen lähiverkko (LAN) tiedonvälitykseen pääpalvelimen ja alipalvelinten välillä. Virtuaalinäytelasit jaetaan järjestelmän alipalvelinten kiintolevyille ja pääpalvelin sisältää tietokannan, josta käy ilmi virtuaalinäytelasien sijainti alipalvelinten kiintolevyillä. Tietokantaa voidaan käyttää myös muunlaisen tiedon tallentamiseen (esimerkiksi aiemmin kuvattu metatieto). Pääpalvelin delegoi saapuvat asiakasyhteydet eteenpäin alipalvelimelle, joka sisältää halutun näytelasin. Ideaalinen alipalvelin on edullinen työasema, joka on varustettu kolmella kiintolevyllä (yksi levy kuville, yksi niiden varmuuskopioille ja yksi käyttöjärjestelmälle sekä ohjelmistoille). Varmuuskopiointiin käytettävä levy on itsenäinen levy, joka aktivoidaan esimerkiksi vain kerran yössä suoritettavan varmuuskopioinnin yhteydessä. Tällä hetkellä saatavilla olevat yhden teratavun kiintolevyt kykenevät varastoimaan noin 3 000 – 4 000 virtuaalinäytelasia. Kun alipalvelimen kiintolevykapasiteetti täyttyy, uusi palvelin voidaan helposti liittää osaksi järjestelmää. Näin ollen tämä on tehokas ja edullinen laajentamisstrategia, joka voidaan toteuttaa osittain pitkän aikajakson kuluessa. Lisäksi järjestelmän modulaarisuus mahdollistaa osittaisen alipalvelinten alasajon, ilman että se vaikuttaa järjestelmän toimintaan kokonaisuutena.



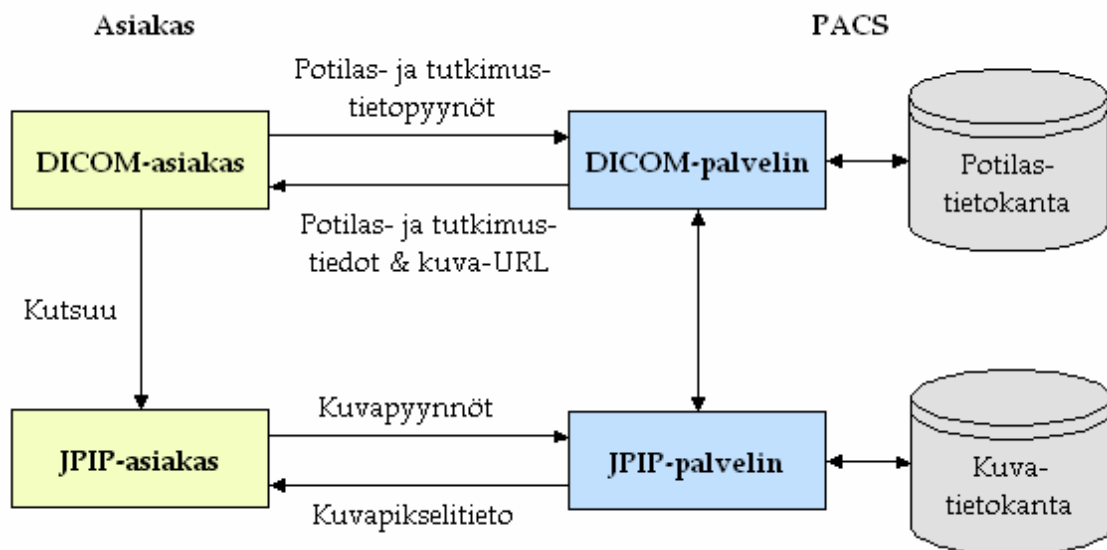
Kuva 27. Malli JPEG2000-pohjaiselle virtuaalimikroskopiajärjestelmälle.

8.3. Jatkotutkimusten aiheita

Virtuaalimikroskopian laajamittaisen käyttöönoton edellytyksenä on yhteensovitus Digital Imaging and Communications in Medicine (DICOM) -standardin [DICOM, 2008a] kanssa, jota pidetään tärkeimpänä spesifikaationa kliinisen lääketieteen digitaalkuville. Kirjoitushetkellä DICOM-standardiin on sisällytetty joitakin osia JPEG2000-standardista: ensimmäinen osa (Part 1) on liitetty DICOM-standardiin Supplement 61 (JPEG 2000 Transfer Syntaxes) -lisäyksen [DICOM, 2008b] myötä sekä toinen osa (Part 2) Supplement 105 (JPEG 2000 Part 2 Multi-component Transfer Syntaxes) [DICOM, 2008c] -lisäyksen myötä. Hiljattain Supplement 106 (JPEG 2000 Interactive Protocol) -lisäys [DICOM, 2008d] liitti standardiin myös JPIP-protokollan. DICOM-pohjaisessa Picture Archiving and Communication System (PACS) -järjestelmässä JPIP-protokollaa voidaan käyttää välittämään virtuaalinäytelasin kuvapikselitieto erillään potilas- ja tutkimustiedosta.

Käytännön kliinisessä toteutuksessa virtuaalinäytelasit vievät hyvin todennäköisesti suurimman osan kaikesta järjestelmän tallennuskapasiteetista, muiden kuvantamisalueiden osuuden ollessa hyvin pieni. JPIP-toiminnallisuus voidaan helposti liittää osaksi olemassa olevaa PACS-järjestelmää siten, että virtuaalinäytelasit sisältävä JPIP-palvelin asennetaan tavanomaisen DICOM-palvelimen rinnalle. DICOM-palvelin toimii kuten ennenkin, sisältäen potilas- ja tutkimustiedon yhdistettynä vastaavaan kuvatietoon ja palvelen esimerkiksi radiologian asiakkaita tavanomaiseen tapaan. Jos asiakas pyytää esimerkiksi

potilaan mikroskooppikudosnäytettä, DICOM-palvelin lähettää potilas- ja tutkimustiedot tavanomaiseen tapaan, mutta pikselitiedon sijaan lähetetäänkin Uniform Resource Locator (URL) -merkkijono, joka sisältää viittauksen kuvapikselitiedon toimittajaan (eli JPIP-palvelimeen) sekä kuvatiedoston nimen tai vastaavan tietokantatunnuksen. Tiedoston nimi tai tietokantatunnus voi olla mielivaltaisesti annettu, joka ei suoraan liity potilastietoihin. Tietojen saapuessa DICOM-asiakkaalle, virtuaalinäytelasi avataan ulkoisen – tai DICOM-asiakkaaseen sisäänrakennetun – JPIP-asiakasohjelmiston avulla. Virtuaalinäytelasin katselu aiheuttaa suoraa liikennöintiä JPIP-asiakkaan ja -palvelimen välillä, joka ei ole sidoksissa DICOM-asiakkaan ja -palvelimen keskinäiseen yhteyteen. Edellä kuvatun järjestelmän periaate on esitetty Kuvassa 28.



Kuva 28. JPEG2000- ja JPIP-toiminnallisuuden yhdistäminen DICOM-pohjaiseen PACS-järjestelmään.

Olemme professori Isolan kanssa mukana DICOM-standardin patologian työryhmässä (WG-26: Pathology) [DICOM, 2008e], jonka tavoitteena on laajentaa standardia tukemaan patologian käsitteistöä ja näytekuvia (useimmiten virtuaalinäytelaseja). Tähän liittyen olemme kehittämässä JVSdicom-nimistä ohjelmistoa, joka mahdollistaa JPEG2000-muotoisten virtuaalinäytelasiin käytön DICOM-standardin yhteydessä. JVSdicom koostuu DICOM-palvelimesta ja DICOM-asiakkaasta, jotka osaavat hyödyntää kuvatiedon siirtoon JPIP-prokollaa edellä kuvatun periaatteen mukaisesti. JVSdicom sisältää tavanomaisen potilas- ja tutkimustietojen hakutoiminnon sekä mahdollisuuden käsitellä kuvia ImageJ-kuva-analyysiohjelmiston avulla. Kun JVSdicom vastaanottaa virtuaalinäytelasin, joka sisältää JPIP-viittauksen, se näyttää potilas- ja tutkimustiedot omassa ikkunassaan ja kutsuu ulkoista JVSview-ohjelmistoa avaamaan

näytelasikuvan JPIP-yhteyden avulla toiseen ikkunaan. Näin esimerkiksi kahden näyttölaitteen avulla toisella ruudulla voidaan selata potilastietoja ja toisella ruudulla voidaan katsella vastaavaa näytelasikuvaa.

Olemme myös mukana projektissa, joka hankki syksyllä 2007 Pirkanmaan sairaanhoitopiirin Laboratoriokeskukseen automaattisen näytelasien skannauslaitteiston (ScanScope® XT, Aperio Technologies, USA), jonka vuorokausikapasiteetti on 120 näytelasia. Tavoitteena on liittää skannauslaitteisto osaksi rutiininomaista kuvantamisprosessia, jossa kaikki näytelasit digitoidaan ennen fyysistä arkistointia. Jotta kuvantaminen voidaan toteuttaa automatisoidusti, näytelasit on identifioitava viivakoodilla, jonka skannauslaitteisto lukee automaattisesti. Viivakoodien käyttö aloitettiin Laboratoriokeskuksessa vuoden 2007 loppupuolella. Itse kuvantamisprosessissa tullaan hyödyntämään DirObserver-sekä JVS-ohjelmistoja: DirObserver ja JVScomp huolehtivat, yhdessä skannausohjelmiston kanssa, näytelasien kuvantamisesta ja muuntamisesta JPEG2000-muotoon; JVSserv vastaa virtuaalinäytelasien jakamisesta Tampereen yliopistollisen sairaalan sisäverkkoon; ja JVSview tarjoaa asiakastyöasemille näytelasien katselumahdollisuuden. Järjestelmää tullaan alkuvaiheessa käyttämään ilman suoraa liitosta potilastietojärjestelmiin, mutta tulevaisuudessa se saatetaan yhdistää sairaalan DICOM-PACS-tietojärjestelmään edellä kuvattujen periaatteiden mukaisesti.

Julkisten organisaatioiden, kuten sairaalat ja oppilaitokset, tietoturvapoliittikka on usein hyvin tiukka. Ulkopuolisten, suorituskelpoisten tiedostojen ajaminen tietojärjestelmään kuuluvilla työasemilla on usein kielletty, Windows®-käyttöjärjestelmän rekisteriin kirjoittaminen on usein hyvin rajattua tai sen sisältö palautetaan alkutilaan käyttäjän kirjautuessa ulos, web-selainten lisäosien asennus on mahdotonta ja niin edelleen. Nämä rajoitukset koskevat usein myös vuokrattavien konferenssi- ja kokoustilojen kiinteitä, esitelmöintiin käytettäviä työasemia. Virtuaalimikroskopian näytelasiseminaarien yleinen ongelma on esitelmöijien vaihtelevat virtuaalinäytelasiformaatit, joiden näyttämiseen vaaditaan erityisohjelmisto tai web-selaimen lisäosa. Tämän kaltaisissa tilanteissa olisikin suotavaa, jos virtuaalinäytelaseja voitaisiin katsella web-selaimen avulla, ilman lisäosien tai ulkoisten ohjelmistojen asennusta. Nykyisellä tekniikalla ainoa mielekäs ratkaisu on käyttää JavaScript-ohjelmointikieltä, joka on sisällytetty käytännössä kaikkiin yleisimpiin web-selaimiin. Sen avulla voidaan suunnitella niin kutsuttuja Dynamic HTML (DHTML) -sivustoja, joiden sisältö generoidaan ja päivitetään dynaamisesti, käyttäjän vuorovaikutuksen ohjaamana. Hyvänä esimerkkinä tästä on Google Maps -palvelu [Google, 2008], jonka periaatteet ovat lähellä virtuaalimikroskopiaa. Vastaavanlaisen tekniikan soveltaminen virtuaalimikroskopiassa on vain ajan kysymys. Tutkimusryhmämme

seuraavana tavoitteena virtuaalimikroskopian osalta on kehittää JPEG2000-standardiin ja JPIP-protokollaan perustuva DHTML-kuvapalvelin, joka hyödyntää karttasovelluksiin tarkoitettua, avoimen lähdekoodin OpenLayers-nimistä JavaScript-kirjastoa [OpenLayers, 2008].

9. Yhteenveto

Tässä tutkielmassa käsiteltiin virtuaalimikroskopian ongelmakohtia ja niiden ratkaisuja, jotka edesauttavat virtuaalimikroskopian laajamittaista käyttöä. Osittaisen mikroskooppikuvantamisen avuksi kehitettiin ohjelmisto, joka kykenee luomaan montaasikuvia ilman näkyviä sauma-alueiden epäjatkuvuuksia. Kuvantamisprosessin automatisointia varten luotiin yleiskäyttöinen apuohjelmisto, jonka avulla prosessiin voidaan liittää ulkopuolisia ohjelmistoja. Avoimen JPEG2000-standardin osoitettiin soveltuvan hyvin virtuaalimikroskopian tarpeisiin: se mahdollistaa automaattisten näytelasiskannereiden tuottamisen suurten kuvatiedostojen pakkaamisen, katselemisen, arkistoinnin ja jakelun tietoverkkojen yli. Tehokkaan hyödyntämisen edellytyksenä oli virtuaalinäytelasien JPEG2000-koodivirran parametrisoinnin optimaalinen määrittely. Määrittelyn pohjalta kehitettiin ohjelmistopaketti, joka koostuu JPEG2000-muotoisten virtuaalinäytelasien pakkaus-, katselu- ja kuvapalvelinohjelmistoista. Kaikki tutkielmassa kuvatut ohjelmistot on julkaistu vapaaseen levitykseen ja niitä hyödynnetään rutiininomaisesti tutkimusryhmämme käytössä.

Viiteluettelo

- [Acharya, 2006] T. Acharya, VLSI Algorithms and architectures for JPEG2000. *Ubiquity* 7, 35 (2006), 1–42.
- [Antonini *et al.*, 1992] M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies, Image coding using the wavelet transform. *IEEE Trans. Image Processing* 1, 2 (1992), 205–220.
- [Apache, 2008] The Apache Software Foundation, The Apache HTTP Server Project. Available at <http://httpd.apache.org/>. Checked 12.3.2008.
- [Aware, 2008] Aware Systems, The BigTIFF File Format Proposal. Available at <http://www.awaresystems.be/imaging/tiff/bigtiff.html>. Checked 12.3.2008.
- [Bayer, 1976] B. E. Bayer, Color imaging array. United States Patent 3971065 (1976).
- [Burthem *et al.*, 2005] J. Burthem, M. Brereton, J. Ardern, L. Hickman, L. Seal, A. Serrant, C.V. Hutchinson, E. Wells, P. McTaggart, B. De la Salle, J. Parker-Williams, and K. Hyde, The use of digital 'virtual slides' in the quality assessment of haematological morphology: results of a pilot exercise involving UK NEQAS(H) participants. *Br. J. Haematol.* 130, 2 (2005), 293–296.
- [De Vos, 2008] K. De Vos, ImageJ Cell Counter plugin. Available at <http://rsbweb.nih.gov/ij/plugins/cell-counter.html>. Checked 12.3.2008.
- [DICOM, 2008a] Digital Imaging and Communications in Medicine (DICOM). Available at <http://medical.nema.org/dicom/2007/>. Checked 12.3.2008.
- [DICOM, 2008b] Digital Imaging and Communications in Medicine (DICOM), Supplement 61: JPEG 2000 Transfer Syntaxes. Available at ftp://medical.nema.org/medical/dicom/final/sup61_ft.pdf. Checked 12.3.2008.
- [DICOM, 2008c] Digital Imaging and Communications in Medicine (DICOM), Supplement 105: JPEG 2000 Part 2 Multi-component Transfer Syntaxes. Available at ftp://medical.nema.org/medical/dicom/final/sup105_ft.pdf. Checked 12.3.2008.
- [DICOM, 2008d] Digital Imaging and Communications in Medicine (DICOM), Supplement 106: JPEG 2000 Interactive Protocol. Available at ftp://medical.nema.org/medical/dicom/final/sup106_ft.pdf. Checked 12.3.2008.
- [DICOM, 2008e] Digital Imaging and Communications in Medicine (DICOM), Strategic Document & Principal Contacts. Available at <http://medical.nema.org/dicom/geninfo/Strategy.pdf>. Checked 12.3.2008.
- [ERM, 2008] Earth Resource Mapping, ER Mapper Image Compression SDK. Available at <http://www.ermapper.com/>. Checked 12.3.2008.

- [Ebrahimi *et al.*, 2004] F. Ebrahimi, M. Chamik, and S. Winkler, JPEG vs. JPEG2000: An Objective Comparison of Image Encoding Quality. In: *Proc. of SPIE* **5558** (2004), 300–308.
- [Eclipse, 2008] Eclipse Foundation: Eclipse development platform. Available at <http://www.eclipse.org/>. Checked 12.3.2008.
- [Fossel *et al.*, 2003] S. Fossel, G. Fottinger, and J. Mohr, Motion JPEG2000 for high quality video systems. *IEEE Trans. Cons. Electr.* **49**, 4 (2003), 787–791.
- [FSF, 2008] Free Software Foundation: The GNU General Public License. Available at <http://www.gnu.org/copyleft/gpl.html>. Checked 12.3.2008.
- [Furness, 2007] P. Furness, A randomized controlled trial of the diagnostic accuracy of internet-based telepathology compared with conventional microscopy. *Histopathology* **50**, 2 (2007), 266–273.
- [Gall and Tabatabai, 1988] D. L. Gall and A. Tabatabai, Subband Coding of Digital Images Using Symmetric Short Kernel Filters and Arithmetic Coding Techniques. In: *Proc. of IEEE International Conference Acoustics, Speech, and Signal Processing* **2** (1988), 761–764.
- [Gao *et al.*, 2003] M. Gao, P. Bridgman, and S. Kumar, Computer Aided Prostate Cancer Diagnosis Using Image Enhancement and JPEG2000. In: *Proc. of SPIE* **5203** (2003), 323–334.
- [Google, 2008] Google Inc., Google Maps. Available at <http://maps.google.com/>. Checked 12.3.2008.
- [Helin *et al.*, 2005] H. Helin, M. Lundin, J. Lundin, P. Martikainen, T. Tammela, H. Helin, T. van der Kwast, and J. Isola, Web-based virtual microscopy in teaching and standardizing Gleason grading. *Hum. Pathol.* **36**, 4 (2005), 381–386.
- [Helin *et al.*, 2006] H. O. Helin, M. Lundin, M. Laakso, J. Lundin, H. J. Helin, and J. Isola, Virtual Microscopy in Prostate Histopathology: Simultaneous Viewing of Biopsies Stained Sequentially With Hematoxylin and Eosin, and α -Methylacyl-Coenzyme A Racemase/p63 Immunohistochemistry. *J. Urol.* **175**, 2 (2006), 495–499.
- [IJG, 2008] Independent JPEG Group, libjpeg library. Available at <http://www.iijg.org/>. Checked 12.3.2008.
- [ISO/IEC 14492, 2000] Information technology – Lossy/lossless coding of bi-level images. ISO/IEC 14492, 2002.
- [JPEG, 2008] The Joint Photographic Experts Group website. Available at <http://www.jpeg.org/>. Checked 12.3.2008.
- [Konsti *et al.*, 2008] J. Konsti, J. Lundin, M. Jumppanen, M. Lundin, A. Viitanen, and J. Isola, A public-domain image processing tool for automated quanti-

- fication of fluorescence in situ hybridisation signals. *J. Clin. Pathol.* **61**, 3 (2008), 278–282.
- [Kumar *et al.*, 2006] R. K. Kumar, B. Freeman, G. M. Velan, and P. J. De Permentier, Integrating histology and histopathology teaching in practical classes using virtual slides. *Anat. Rec. B New Anat.* **289**, 4 (2006), 128–133.
- [Liu *et al.*, 2005] J. K. Liu, H. C. Wu, and T. Y. Shih, Effects of JPEG2000 on the Information and Geometry Content of Aerial Photo Compression. *Photogrammetric Engineering & Remote Sensing* **71**, 2 (2005), 157–168.
- [Lundin *et al.*, 2004a] M. Lundin, J. Lundin, and J. Isola, Virtual microscopy. *J. Clin. Pathol.* **57**, 12 (2004), 1250–1251.
- [Lundin *et al.*, 2004b] M. Lundin, J. Lundin, H. Helin, and J. Isola, A digital atlas of breast histopathology: an application of web based virtual microscopy. *J. Clin. Pathol.* **57**, 12 (2004), 1288–1291.
- [Microsoft, 2008a] Microsoft Corporation, Microsoft Foundation Classes library. Available at <http://www.microsoft.com/>. Checked 12.3.2008.
- [Microsoft, 2008b] Microsoft Corporation, Visual Studio Development System. Available at <http://www.microsoft.com/visualstudio/>. Checked 12.3.2008.
- [OpenLayers, 2008] OpenLayers JavaScript library. Available at <http://www.openlayers.org/>. Checked 17.3.2008.
- [Ortiz *et al.*, 2007] J. P. Ortiz, V. G. Ruiz, and I. Garcia, Efficient Virtual Slide Telepathology Systems with JPEG2000. In: *Conf. Proc. IEEE Eng. Med. Biol. Soc. 2007* (2007), 880–883.
- [Rabbani and Joshi, 2002] M. Rabbani and R. Joshi, An overview of the JPEG 2000 still image compression standard. *Signal Process.: Image Comm.* **17**, 1 (2002), 3–48.
- [Rasband, 2008] W. S. Rasband, ImageJ image analysis software. Available at <http://rsb.info.nih.gov/ij/>. Checked 12.3.2008.
- [Rojo *et al.*, 2006] M. G. Rojo, G. B. García, C. P. Mateos, J. G. García, M. C. Vicente, Critical Comparison of 31 Commercially Available Digital Slide Systems in Pathology. *Int. J. Surg. Pathol* **14**, 4 (2006), 285–305.
- [Santa-Cruz *et al.*, 2002] D. Santa-Cruz, R. Grosbois, T. Ebrahimi, JPEG 2000 performance evaluation and assessment. *Signal Process.: Image Comm.* **17**, 1 (2002), 113–130.
- [Sayood, 2006] K. Sayood, *Introduction to Data Compression (third edition)*. Morgan Kaufmann, 2006.
- [Schelkens *et al.*, 2006] P. Schelkens, A. Munteanu, and J. Cornelis, Wavelet coding of volumetric medical datasets. *IEEE Trans. Med. Imag.* **22**, 3 (2003), 441–458.

- [Skodras *et al.*, 2001] A. Skodras, C. Christopoulos, and T. Ebrahimi, The JPEG 2000 Still Image Compression Standard. *IEEE Signal Proc. Mag.* **18**, 5 (2001), 36–58.
- [Sun, 2008a] Sun Microsystems, Java programming language. Available at <http://java.sun.com/>. Checked 12.3.2008.
- [Sun, 2008b] Sun Microsystems, Java Advanced Imaging API. Available at <https://jai.dev.java.net/>. Checked 12.3.2008.
- [Sun, 2008c] Sun Microsystems, Java Advanced Imaging Image I/O Tools. Available at <https://jai-imageio.dev.java.net/>. Checked 12.3.2008.
- [Taubman, 2000] D. S. Taubman, High Performance Scalable Image Compression with EBCOT. *IEEE Trans. Image Processing* **9**, 7 (2000), 1158–1170.
- [Taubman and Marcellin, 2002] D. S. Taubman and M. W. Marcellin, *JPEG2000: Image Compression Fundamentals, Standards and Practice*. Springer, 2002.
- [Taubman and Prandolini, 2003] D. S. Taubman and R. Prandolini, Architecture, Philosophy and Performance of JPIP: Internet Protocol Standard for JPEG2000. In: *Proc. of SPIE* **5150** (2003), 791–805.
- [Taubman, 2008] D. S. Taubman, Kakadu JPEG2000 Framework. Available at <http://www.kakadusoftware.com/>. Checked 12.3.2008.
- [Thévenaz, 1998] P. Thévenaz, U.E. Ruttimann, and M. Unser, A Pyramid Approach to Subpixel Registration Based on Intensity. *IEEE Trans. Image Processing* **7**, 1 (1998), 27–41.
- [Thévenaz, 2008] P. Thévenaz, An ImageJ plugin for the automatic alignment of a source image or a stack to a target image. Available at <http://bigwww.epfl.ch/thevenaz/turboreg/>. Checked 13.3.2008.
- [Warmerdam, 2008] F. Warmerdam, A. Kiselev, M. Welles, and D. Kelly, LibTIFF – TIFF Library and Utilities. Available at <http://www.libtiff.org/>. Checked 12.3.2008.
- [Waterhouse *et al.*, 2008] M. Waterhouse, D. Treanor, H. Dee, N. Griffin, Virtual pathology at the University of Leeds. Available at <http://www.virtualpathology.leeds.ac.uk/>. Checked 13.3.2008.
- [Weinstein, 2005] R. S. Weinstein, Innovations in medical imaging and virtual microscopy. *Hum. Pathol.* **36**, 4 (2005), 317–319.
- [Zoomify, 2008] Zoomify Inc., Zoomify image server software. Available at <http://www.zoomify.com/>. Checked 13.3.2008.

The Application of JPEG2000 in Virtual Microscopy

Vilppu J. Tuominen and Jorma Isola, Institute of Medical Technology, University of Tampere & Tampere University Hospital, Tampere, Finland

This article is originally published in *Journal of Digital Imaging* and it is available at www.springerlink.com (<http://dx.doi.org/10.1007/s10278-007-9090-z>).

ABSTRACT

Virtual microscopy (i.e., the viewing of entire microscope specimens on a computer display) is becoming widely applied in microscopy teaching and clinical laboratory medicine. Despite rapidly increasing use, virtual microscopy currently lacks of a universally accepted image format. A promising candidate is JPEG2000, which has potential advantages for handling gigabyte-sized virtual slides. To date, no JPEG2000-based software has been specifically suited for virtual microscopy. To study the utility of JPEG2000 in virtual microscopy, we first optimized JPEG2000 code-stream parameters for virtual slide viewing (i.e., fast navigation, zooming, and use of an overview window). Compression using ratios 25:1–30:1 with the irreversible wavelet filter were found to provide the best compromise between file size and image quality. Optimal code-stream pa-

rameters also consisted of 10 wavelet decomposition levels, progression order Resolution-Position-Component-Layer (RPCL), a precinct size of 128×128, and code-block size of 64×64. Tiling and the use of multiple quality layers were deemed unnecessary. A compression application (JVScomp) was developed for creating optimally parameterized JPEG2000 virtual slides. A viewing application (JVSview) was developed specifically for virtual microscopy, offering all of the basic viewing functions. JVSview also supports viewing of focus stacks, embedding of textual descriptions, and defining regions of interest as metadata. Combined with our server application (JVSserv), virtual slides can be viewed over networks by employing the JPEG2000 Interactive Protocol (JPIP). The software can be tested using virtual slide examples located on our public JPIP server (<http://jvsmicroscope.uta.fi/>). The software package is freely

downloadable and usable for noncommercial purposes.

Keywords: JPEG2000, JPIP, telepathology, digital pathology, virtual slide

BACKGROUND

Virtual slides are digitized versions of whole microscope specimens that can be viewed on a computer display.¹ Virtual slide image files can be accessed from the computer's hard disk, transportable media (e.g., a USB memory stick or a DVD), or, more practically, over networks.² Within the internal network of a hospital or pathology department, virtual slides can be used for case meetings, slide seminars, and didactic live audience presentations.³ By allowing access over the internet, virtual slides can be used more widely for national and international conferences and in interlaboratory quality assurance programmes.⁴ Feedback from microscopists shows that virtual microscopy can be regarded as a significant improvement over digital snapshot images and the use of multi-headed microscopes.^{5,6}

To create virtual slides that resemble real microscopy viewing experiences, specimens must be scanned at high optical resolution. Image resolutions from 0.20

to 0.40 μm per pixel are considered necessary for sufficient image quality.⁷ Since microscope specimens are often up to 20 \times 30 mm in size, a virtual slide can contain up to 40 gigabytes of uncompressed image data (with three 8-bit color channels). When virtual slides are applied in cytopathology or hematology, a higher optical magnification (scanning with oil-40 \times , oil-60 \times , or oil-100 \times lens) is often needed.⁸ This multiplies the amount of image data produced when compared to standard scanning. Moreover, when several focus planes (along Z-axis) are digitized, the resulting data amount is further multiplied by the number of planes. Due to the large size of the virtual slide files, it is not possible to use conventional image viewing software (such as Photoshop[®]), which requires the image data to be loaded entirely into computer's memory (RAM). For this reason, the virtual slide viewing systems described to date apply the on-demand principle, that is, the viewing software decodes and displays only a user-requested area of the virtual slide at the requested resolution. When using JPEG-compressed images, this requires splitting the virtual slide image data into tens of thousands of very small image files, which are uploaded to the client on demand (e.g., Zoomify).⁹ Proprietary image formats provided with commercial slide scanners (e.g., Mirax by Carl Zeiss MicroImaging, USA; dotSlide by Olympus

UK, UK; and ScanScope[®] by Aperio Technologies, USA) also follow this principle in their web viewing solutions. Due to competition, it is unlikely that any of the proprietary industry-based image formats will become universally accepted in virtual microscopy.

A promising candidate for a universal image format for virtual microscopy is JPEG2000, which is a family of standards, developed by the Joint Photographic Experts Group.¹⁰ It is based on a wavelet-based compression scheme that is defined in the core part of the standard series (published as International Standard ISO/IEC 15444-1 | ITU-T Rec. T.800).¹¹ JPEG2000 provides many features that support scalable and interactive access to large-sized images such as virtual slides. These include an efficient and unified compression architecture, especially at low bit-rates, resolution and quality scalability, region of interest coding, spatial random access, and effective error resiliency.¹²⁻¹⁴ Currently, the JPEG2000 standard consists of 11 parts, which describe techniques that are useful in various areas of imaging. Part 1 (Core Coding System),¹¹ Part 2 (Extensions),¹⁵ and Part 9 (Interactivity Tools, APIs and Protocols)¹⁶ are the most essential for virtual microscopy. The first part specifies the JPEG2000 code-stream syntax and the JP2 file format (“jp2” as the common file extension), which offers,

for example, the possibility of adding textual metadata into the image header. The second part specifies the JPX file format (“jpx” and “jpf” as the common file extensions), which contains more advanced features, such as building stacks of images (e.g., for specimens scanned at multiple focal depths), and associating textual and graphical metadata with spatial regions. The ninth part defines the JPEG2000 Interactive Protocol (JPIP), which provides a client-server architecture for network-based remote viewing of JPEG2000 family images.¹⁷ Currently, web browsers do not have direct, built-in support for JPIP. For this reason, internet-based viewing of virtual slides over JPIP requires external viewing software, which is, for example, invoked automatically from the browser upon clicking on a virtual slide URL link.

Although JPEG2000 has theoretical advantages for application in virtual microscopy, no efficient application software has been described so far. The computational demands of virtual microscopy images differ to a large extent from that of radiology and other medical imaging because of the larger size of the image files (up to tens of gigabytes). In this study, we first searched for an optimal parameterization of JPEG2000 for virtual microscopy and constructed a JPEG2000-based software package enti-

tled JVS (for JPEG2000 Virtual Slide), consisting of image compression, viewing, and network server applications. The performance of JPEG2000 was evaluated using JPIP image serving speed and compression execution time as outcome measures. In addition, the performance of our network server application (JVSserv) was compared against the commonly used virtual slide server solution Zoomify.⁹ Finally, we present a model for a simple, scalable, and inexpensive JPEG2000-based virtual microscopy system.

MATERIALS AND METHODS

Image Acquisition

We acquired test images by using a Zeiss Axioskop40 microscope (Carl Zeiss MicroImaging, USA) which was equipped with 10 \times , 20 \times and oil-40 \times objectives and a motorized specimen stage (Märzhäuser Wetzlar GmbH, Germany). The images were captured with a CCD color camera (QICAM Fast, QImaging, Canada; 24-bit color depth; resolution 1,388 \times 1,036 pixels, pixel size 4.65 μ m). The camera was attached to the microscope with a 1 \times phototube. Image acquisition was controlled by the Surveyor imaging system (Objective Imaging, UK, software version 5.03) running on a

standard Windows[®] XP workstation. The Surveyor software controls for stage and focus movements matched with automated image acquisition. In addition to standard histological samples, cytological slides were scanned using a multiple focus layer image acquisition option (Z-stack). Image tiles (up to 5,000 per scanned slide) were primarily saved in an uncompressed file format (BMP) and stitched as a contiguous montage either using the built-in function of Surveyor, or using the LargeMontage plugin¹⁸ of the publicly available image analysis software ImageJ.¹⁹

JPEG2000 Code-Stream Parameter Optimization

The JPEG2000 code-stream parameter optimization was based on theoretical assumptions of the JPEG2000 code-stream properties, visual assessment of the time needed to initially load the image, the time needed for image refreshment during magnification change and navigation within the specimen, and to lesser extent, the compression execution time. Tests were performed using the highly configurable Kakadu kdu_compress and kdu_show applications, version 5.2.2.²⁰

JPEG2000 Performance Evaluation

The performance of JPEG2000 was evaluated using JPIP image serving speed and compression execution time as outcome measures. The tests were carried out on a Windows[®] XP workstation, equipped with a dual-core processor, 3 gigabytes of RAM, two ATA hard disks, and a 100 Mbit network link. The used main test virtual slide image can be seen on our website (<http://jvsmicroscope.uta.fi/examples/>).

The image had 42,865×57,222 pixels, three 8-bit color channels, totaling 6.9 gigabytes uncompressed, and it was stored as a binary encoded PPM file. An additional set of test images was created by extracting sub-resolutions from the original image, halving the resolution in each step.

The compression tests were performed using our JVScomp application. During these tests, we monitored the processor load, RAM usage, disk usage percentage, and the average disk read and write queue lengths. The image serving performance of JPIP was evaluated with our JVSserv application, in comparison to the Zoomifyer EZ[™], version 3.0.⁹ While JVSserv is a standalone server application, Zoomifyer requires an external HTTP server underneath it. The Apache HTTP server, version 2.2.4,²¹ was se-

lected for this. The evaluation and comparison were performed by simulation, in which 10 local area network workstations were each simulating 10 clients, yielding a 100-client pool. Each client had their own test image, which was duplicated from the main test image. Server and client caches were disabled. During these simulations, we monitored the server's average disk read queue length, processor load, network bandwidth usage, RAM usage, and subjectively evaluated the clients' viewing interactivity.

Software Package Development

The demonstrational software package was written with the C++ programming language and built for 32-bit Windows[®] platforms, but can be run under a 64-bit Windows[®] platform as well. The package consists of three applications: JVScomp—a compression application for creating optimally parameterized JPEG2000 virtual slides; JVSview—a viewer application capable of viewing both local and remote JPEG2000 virtual slides; and JVSserv—a JPIP server for remote serving of JPEG2000 virtual slides. JVScomp is built on the ECW JPEG 2000 SDK,²² while JVSview and JVSserv are built on the Kakadu JPEG2000 Framework.²⁰

RESULTS

JPEG2000 Code-Stream Parameter Optimization

To fully utilize the advantages of JPEG2000 in virtual microscopy, we first searched for an optimal combination of JPEG2000 code-stream parameters. The findings are summarized in Table 1. In virtual microscopy, lossy compression is mandatory due to the large amount of image data. It is shown that highly compressed JPEG2000 images can be used for pathological diagnostic studies.²³ We found compression ratios ranging from 25:1 to 30:1 in areas containing cells or tissue sections to produce virtual slides without disturbing visible image compression artifacts at the original image resolution. A comparison of the image quality between different JPEG2000 lossy compression ratios is demonstrated on our website (<http://jvsmicroscope.uta.fi/examples/>).

Table 1. JPEG2000 Code-Stream Parameters Found Optimal for Virtual Microscopy

JPEG2000 Code-Stream Parameter	Parameter Value
Compression ratio (lossy)	25:1 to 30:1
Wavelet filter	Irreversible
Wavelet decomposition levels	10
Tiling	Not needed
Tile-parts	1
Precinct size	128 × 128
Code-block size	64 × 64
Progression order	RPCL
Quality layers	1
PLT pointer marker segments	Inserted always

Of the two alternative wavelet filters (reversible and irreversible), the irreversible filter yielded somewhat better visual quality, but at the expense of compression speed. The Discrete Wavelet Transform (DWT) process decomposes the original image into several low-resolution versions. To extract an overview image, we found 10 DWT levels to be sufficient for even the largest virtual slides, although this creates redundant resolution levels for smaller slides. Tiling and the use of multiple tile-parts were not found to be useful in virtual slides. Technically, an untiled JPEG2000 image actually contains a single tile whose size is identical to the image size. We found precincts to be a more useful alternative for partitioning the JPEG2000 code-stream for random access. A precinct size of 128×128 was a suitable average for all resolution levels. Precincts are further divided into code-blocks, the optimal size of which was 64×64. The progression order defines how packets—the basic segments of a JPEG2000 codestream—are prioritized. We found the Resolution-Position-Component-Layer (RPCL) order most useful, since it allowed the fastest access to different image resolutions, as well as fast spatial random access. No particular advantages were found for multiple quality layers, since the gradual image quality improvement after navigation or zooming became noticeable only with

very slow-speed JPIP connections. The JPEG2000 codestream headers are composed of so-called markers and marker segments, which are used, for example, to specify the image size and to offer pointers to the code-stream. To further increase the codestream random access, we found it advantageous to use packet length pointer marker segments within the tile-part header (PLT).

JPEG2000 Compression Performance

We performed various compression speed tests to ensure that JPEG2000 can be used within a real virtual microscopy scanning system. The duration of the compression process was linearly dependent on the original virtual slide image size. RAM usage was relatively low, as during the compression of a 7-gigabyte virtual slide the memory footprint of JVScomp constantly remained around 70 megabytes. JVScomp reached a compression speed of 50 gigabytes per hour on a dual-processor workstation. The limiting factor was the processor performance, as the source hard disk could have effectively processed over twice the number of read requests.

JPIP Image Serving Performance

We evaluated the serving performance of the JPIP protocol using our JVSserv ap-

plication. In addition, a comparison was made to the commonly used Zoomifyer EZ™ virtual slide server solution. On both systems, virtual slide serving was limited by the speed of the hard disk; the processor load was constantly under 50%, as was the RAM allocation. The network bandwidth (100 Mbit/s in our tests) was utilized only about 25% of the maximum transmission rate. As the number of clients increased, the average disk read queue lengths quickly reached the effective performance limits (an average of 2.0 was considered to be a suitable limit).

With the main test virtual slide, which was duplicated for every client, JVSserv could effectively handle about 30 simultaneous clients, while the Zoomifyer (run on top of the Apache HTTP server) had its limit at around 20 simultaneous clients. The difference can be explained by the fact that Zoomifyer splits the image into tens of thousands of small JPEG files, which the HTTP server (Apache) then serves to the clients. Because of caching policies, both servers would have benefited for clients that browse the same image simultaneously. Thus, assuming that some clients are usually viewing the same image, JVSserv could easily serve 50 simultaneous clients.

Furthermore, JVSserv features a load balancing function, which can be used to

distribute clients to several sub-servers. Thus, we estimate that a standard JVSserv workstation behind a 100 Mbit/s network link could handle 200 to 300 simultaneous clients when supplemented with 4 to 5 sub-servers.

Software Package Description

We developed the demonstrational JVS software package to illustrate the utility of JPEG2000 in virtual microscopy. The package is designed primarily for Windows[®] 2000 and XP, and is Windows Vista[™] compatible. JVScomp is licensed under the GNU General Public License, while the binaries (i.e., the actual executable programs) for JVSview and JVSserv are released for free, non-commercial usage. The software package can be obtained from our website (<http://jvsmicroscope.uta.fi/>). Our website also features a public JPIP server with several virtual slide collections, which can be viewed using JVSview.

JVSview. This software is a JPEG2000 virtual slide viewer that works identically both in local storage and JPIP-based virtual slide viewing (Table 2). JVSview displays an overview of the entire sample and a detailed microscopic view. The magnifications have been set to mimic commonly used microscope objectives. The user can navigate around the sample using either the arrow keys,

mouse panning, or by clicking on a location in the macroscopic view. The application allows the user to interactively adjust brightness, contrast and color saturation of the main window. Multiple display devices can be used in full screen viewing mode. Viewing interactivity is dependent on the processor performance of the client workstation. Acceptable speed can already be achieved with conventional single-processor computers, although JVSview has an automatic support for multiprocessor environments.

Table 2. Main Features of the JVSview JPEG2000 Viewer Application

List of Features
Local image viewing & remote image viewing over JPIP
Overview window representing the entire specimen
Magnifications resembling common microscope objectives
Interactive panning with mouse or keyboard
Image brightness, contrast & colour saturation adjustments
Multi-display full screen viewing
Support for multiple image layers (e.g., focus layers)
Storing information and regions of interest as XML metadata
A functional link with image analysis software (ImageJ)

To illustrate new functionalities of virtual microscopy, made possible by advanced features of JPEG2000, we included a focus feature, which enables switching between alternative image layers, each representing the cells or tissue scanned at different focal depth (see the examples on our website <http://jvsmicroscope.uta.fi/examples/>).

We also included the possibility to embed short textual data conveniently within the header of the virtual slide file. The textual data, called metadata, can

contain information on the scanning resolution, organ and histopathologic diagnosis, and the staining method, for example. The metadata is written in XML format and its structure is described formally with an XML schema;²⁴ it can be added and edited within JVSview. The metadata is primarily for the purposes of image collections, where the associated patient data is irrelevant (e.g., those used in teaching). JVSview also allows specifying and storing regions of interest (ROIs), which can then be retrieved from a ROI list by the clients. JVSview also provides a functional link with ImageJ,¹⁹ which is an efficient, feature-rich, public domain image analysis application. The link to ImageJ is an easy way to measure lengths and areas as well as to count particles (e.g., cells and nuclei) using a click-and-count feature.

JVSserv. This software is a JPEG2000 virtual slide server application, which utilizes the JPIP protocol as its data transfer technique (Table 3). It is based on the *kdu_server* application by the Kakadu Software.²⁰ It has been modified by adding a simple graphical user interface, and by modifying image sample count restrictions and the messaging system. Therefore, JVSserv should only be used with the JVSview viewer application. JVSserv supports multiple simultaneous client connections and uses a standard FIFO (First In, First Out) caching

policy, which is used to maintain the most frequently accessed data in memory. JVSserv also supports load balancing, which means sharing the main server's workload to several sub-servers, thereby allowing the system to handle more simultaneous clients.

Table 3. Main Features of the JVSserv JPEG2000 Server Application

List of Features
Remote image serving over JPIP
Graphical user interface for administration
Support for multiple simultaneous clients
Multiple server clustering (load balancing)

JVScomp. This software is a JPEG2000 virtual slide compression application capable of creating virtual slides that are optimized for viewing and serving with JVSview and JVSserv (Table 4). It follows the parameterization guidelines described earlier, and it is capable of compressing various input file formats, such as PPM, BMP, and JPEG. It supports dual-processor workstation environments, which are detected automatically. JVScomp utilizes the ECW JPEG 2000 SDK compression algorithm,²² which employs a rate control policy that yields as efficient a compression as possible. If an image contains areas that are substantially responsive to compression, for example, homogenous areas of virtual slide glass background with no tissue section, a higher compression ratio is applied to these areas. Thus, a greater overall com-

pression ratio and a smaller file size can be achieved.

Table 4. Main Features of the JVScomp JPEG2000 Compression Application

List of Features
Optimised parameterisation for virtual microscopy
Efficient rate control policy for virtual microscopy
Support for dual-processor environments
Support for various input file formats

DISCUSSION

Today many manufacturers of virtual slide scanners use proprietary image formats, which preclude image exchange between different systems. Support of proprietary non-standard image formats may also be short-lived. For these reasons, the use of an open image standard is a prerequisite for large-scale virtual slide collections that are supposed to be accumulated and utilized for years or even decades. Since technical development is likely to continue rapidly in slide scanners, it is advantageous if image acquisition can be decoupled from image archival and delivery. This ensures that virtual slides generated with older generation equipment are compatible with newer ones. All these conditions are met with the JPEG2000 image format, which, based on its advanced features, is in many ways an ideal choice for virtual microscopy. The present study defined optimal codestream parameters, which

are essential for fast viewing of JPEG2000 virtual slides.

Compression of virtual slide image data into JPEG2000 format is a computationally intensive process. As shown in our experiments, the limiting factor with a conventional dual-core processor environment is the processor performance. However, given that JPEG2000 compression software can be designed to efficiently utilize new multiprocessor environments, the hard disk reading performance eventually becomes critical. By utilizing the mirrored RAID data storage scheme, in which multiple hard disks are combined together and data content is duplicated to all disks, the reading performance can be effectively multiplied. Thus, we estimate that a JPEG2000 compression application using a quad-core processor is capable of compressing over 100 gigabytes of image data per hour, which is faster than the time it takes to scan a 20×20-mm tissue section with a 0.23 μm per pixel resolution using current scanners.⁷ Therefore, it seems that the computational demands of JPEG2000 image compression can be fulfilled with a software-based solution, and there is probably no need for external JPEG2000 processing hardware.

In general, JPEG2000 differs from its predecessor JPEG in many important

ways. With regards to virtual microscopy, the most important improvement over JPEG is the possibility to retrieve any part of the image with random spatial accessing. Moreover, due to the nature of the JPEG2000 data structure, different image resolutions (magnification levels in virtual slides) can be retrieved without compromising viewing speed. A single JPEG2000 file can thus contain the entire virtual slide data, and even data from multiple focus layers, when using the JPX file format. It has been extensively documented that when using the average Peak Signal-to-Noise Ratio (PSNR) image quality metric, JPEG2000 outperforms JPEG at all compression ratios.^{13,14,25} If the quality assessment is based on human observations, the superiority of JPEG2000 becomes evident at compression ratios of 20:1 or higher.²⁶ In practice, an 8-gigabyte uncompressed 50,000×50,000 pixel virtual slide can be readily compressed into a 300-megabyte JPEG2000 image. If the slide contains vast glass background areas, the corresponding file size can be substantially reduced, without compromising the overall image quality of the cells and tissues to be imaged.

Recently, an expansion proposal for the common TIFF file format, entitled BigTIFF,²⁷ has been suggested as a candidate image format for virtual microscopy. It circumvents the 4-gigabyte file

size limitation of TIFF, thereby enabling the manipulation of very large images. BigTIFF does not define a compression scheme of its own. It can contain, for example, several JPEG or LZW compressed image code-streams structured in a tile-like fashion. However, by using JPEG compressed BigTIFF, the benefits of JPEG2000 lossy compression are lost. Incorporating a JPEG2000 codestream within BigTIFF is questionable, since the internal structuring of the JPEG2000 code-stream already enables, among many other advantages, an efficient spatial random access. In addition, JPEG2000 specifies the JPIP protocol for image remote viewing, which has not been described for BigTIFF so far.

Due to its demonstrational nature, the current version of JVS software package does not include any encryption scheme for transmitted image data. However, should a need for encrypted network transmission arise, it is possible to use a standard secure network protocol, such as SSH or TLS, to tunnel the unencrypted JPIP traffic through it. For example, use of the TLS protocol with Virtual Private Networking (VPN) requires setting up a VPN server alongside the JPIP server in the server end, and a VPN client alongside the JPEG2000 viewer in the client end. The VPN client would then act as a proxy for the JPEG2000 viewer, redirecting the viewer's requests

through the network to the VPN server, and from there to the actual JPIP server, and vice versa.

A Model System for JPEG2000-Based Virtual Microscopy

As shown in our results, JPEG2000 already meets the functional needs of virtual microscopy viewing. We have described a model for an easily upgradeable JPEG2000-based virtual microscopy system that utilizes load balancing (Fig. 1). The system consists of a main JPIP server with two network interfaces: a 100-Mbit link for client accesses (via internet or intranet) and a 1-Gbit local area network (LAN) connection for data traffic between main server and sub-server(s). The images are distributed on the hard disks of the sub-servers. The main server includes a database of the image locations within the sub-server hard disks. The database can also store

other relevant information (similar to metadata described earlier). The main server delegates incoming client image requests onwards to the sub-servers according to the image location data. An ideally suited sub-server is an inexpensive workstation equipped with three hard disks (one disk for the images, one disk for their backup, and one disk for system software). The backup disk is an independent hard disk, activated only during a nightly backup. Currently available one terabyte hard disks can contain approximately 3,000 to 4,000 virtual slides. As the sub-server hard disks reach their storage capacity, a new sub-server can be easily inserted into the system. Thus, this represents an efficient and inexpensive upscaling strategy that can be done gradually over time. In addition, the modularity of the system allows parts of the system to be offline without affecting the functionality as a whole.

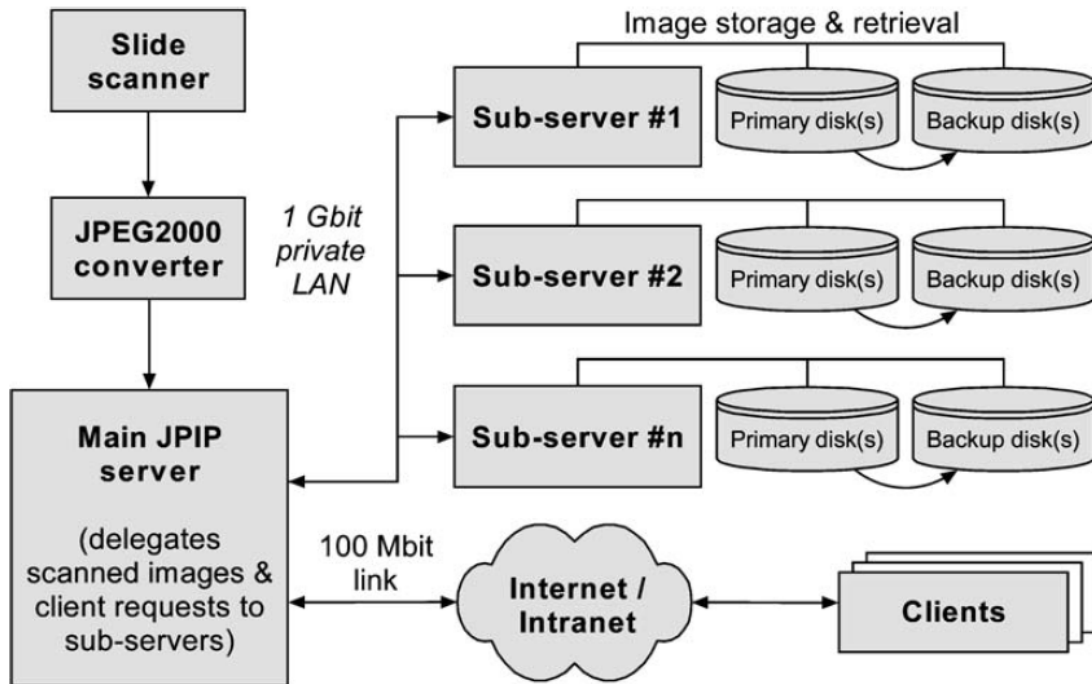


Fig 1. A model system for JPEG2000-based virtual microscopy.

Future Prospects

Large-scale clinical utilization of virtual microscopy requires compatibility with the Digital Imaging and Communications in Medicine (DICOM) standard,²⁸ which is currently recognized as the standard specification for image archival in clinical medicine. At the present time, DICOM includes some parts of the JPEG2000 standard, the foremost being Part 1, which is included in Supplement 61 (JPEG 2000 Transfer Syntaxes).²⁹ Recently, Supplement 106 (JPEG 2000 Interactive Protocol)³⁰ introduced the JPIP protocol as a method of delivering the image pixel data apart from the patient data. A joint international effort

towards consolidating pathology concepts and virtual slide imaging with DICOM is currently underway by the DICOM working group for pathology imaging (WG-26).³¹ Thus, we assume that the next step towards full implementation of JPEG2000 in clinical pathology and virtual slide imaging requires integration with DICOM.

CONCLUSION

We conclude that JPEG2000 is a well-suited image format for virtual microscopy. It enables compressing, viewing, and serving the large image files pro-

duced by the modern microscope slide scanners. We developed a demonstrational software package for JPEG2000 compression, viewing, and JPIP-based network serving, all suited for the needs of virtual microscopy. The package is freely downloadable and usable for non-commercial purposes.

ACKNOWLEDGEMENT

This study was financially supported by the Finnish Cancer Foundation, Sigrid Juselius Foundation, and an EVO-grant from Tampere University Hospital.

REFERENCES

1. Weinstein RS: Innovations in medical imaging and virtual microscopy. *Hum Pathol* 36(4):317–319, 2005
2. Lundin M, Lundin J, Helin H, Isola J: A digital atlas of breast histopathology: an application of web based virtual microscopy. *J Clin Pathol* 57(12):1288–1291, 2004
3. Kumar RK, Freeman B, Velan GM, De Permentier PJ: Integrating histology and histopathology teaching in practical classes using virtual slides. *Anat Rec B New Anat* 289(4):128–133, 2006
4. Lundin M, Lundin J, Isola J: Virtual microscopy. *J Clin Pathol* 57(12):1250–1251, 2004
5. Helin H, Lundin M, Lundin J, Martikainen P, Tammela T, Helin H, van der Kwast T, Isola J: Web-based virtual microscopy in teaching and standardizing Gleason grading. *Hum Pathol* 36(4):381–386, 2005
6. Furness P: A randomized controlled trial of the diagnostic accuracy of internet-based telepathology compared with conventional microscopy. *Histopathology* 50(2):266–273, 2007
7. Rojo MG, García GB, Mateos CP, García JG, Vicente MC: Critical Comparison of 31 Commercially Available Digital Slide Systems in Pathology. *Int J Surg Pathol* 14(4):285–305, 2006
8. Burthem J, Brereton M, Ardern J, Hickman L, Seal L, Serrant A, Hutchinson CV, Wells E, McTaggart P, De la Salle B, Parker–Williams J, Hyde K: The use of digital 'virtual slides' in the quality assessment of haematological morphology: results of a pilot exercise involving UK NEQAS(H) participants. *Br J Haematol* 130(2):293–296, 2005
9. Zoomify, Inc., USA: Zoomify image server software. Available at <http://www.zoomify.com/>. Accessed 12 July 2007
10. The Joint Photographic Experts Group website. Available at

<http://www.jpeg.org/>. Accessed 12 July 2007

11. Information technology—JPEG 2000 image coding system: Core coding system. ISO/IEC 15444-1, 2000 | ITU-T Recommendation T.800, 2002

12. Taubman DS, Marcellin MW: JPEG2000: Image Compression Fundamentals, Standards and Practice: Springer, 2002

13. Skodras A, Christopoulos C, Ebrahimi T: The JPEG 2000 Still Image Compression Standard. IEEE Signal Proc Mag 18(5):36–58, 2001

14. Rabbani M, Joshi R: An overview of the JPEG 2000 still image compression standard. Signal Process: Image Comm 17 (1):3–48, 2002

15. Information technology—JPEG 2000 image coding system: Extensions. ISO/IEC 15444-2, 2002 | ITU-T Recommendation T.801, 2002

16. Information technology—JPEG 2000 image coding system: Interactivity tools, APIs and protocols. ISO/IEC 15444-9, 2005 | ITU-T Recommendation T.808, 2005

17. Taubman DS, Prandolini R: Architecture, Philosophy and Performance of JPIP: Internet Protocol Standard for JPEG2000. Proc SPIE 5150:791–805, 2003

18. Tuominen VJ, Isola J, University of Tampere, Finland: LargeMontage—a Java application for creating large image montages. Available at

<http://iki.fi/vtuo/software/largemontage/>. Accessed 12 July 2007

19. Rasband WS, U.S. National Institutes of Health, USA: ImageJ image analysis software. Available at <http://rsb.info.nih.gov/ij/>. Accessed 12 July 2007

20. Taubman DS, The University of New South Wales, Australia: Kakadu JPEG2000 Framework. Available at <http://www.kakadusoftware.com/>. Accessed 12 July 2007

21. The Apache Software Foundation, USA: The Apache HTTP Server Project. Available at <http://httpd.apache.org/>. Accessed 12 July 2007

22. Earth Resource Mapping Ltd., Australia: ER Mapper Image Compression SDK. Available at <http://www.ermapper.com/>. Accessed 12 July 2007

23. Gao M, Bridgman P, Kumar S: Computer Aided Prostate Cancer Diagnosis Using Image Enhancement and JPEG2000. Proc SPIE 5203:323–334, 2003

24. Tuominen VJ, Isola J, University of Tampere, Finland: JVSSchema: a metadata XML schema for the JVS software. Available at <http://jvsmicroscope.uta.fi/jvsschema/>. Accessed 12 July 2007

25. Santa-Cruz D, Grosbois R, Ebrahimi T: JPEG 2000 performance evaluation and assessment. Signal Process: Image Comm 17(1):113–130, 2002

26. Ebrahimi F, Chamik M, Winkler S: JPEG vs. JPEG2000: An Objective Comparison of Image Encoding Quality. Proc SPIE 5558:300–308, 2004
27. Aware Systems, Belgium: The Big-TIFF File Format Proposal. Available at <http://www.awaresystems.be/imaging/tiff/bigtiff.html>. Accessed 10 October 2007
28. Digital Imaging and Communications in Medicine (DICOM). Available at <http://medical.nema.org/dicom/2007/>. Accessed 12 July 2007
29. Digital Imaging and Communications in Medicine (DICOM), Supplement 61: JPEG 2000 Transfer Syntaxes. Available at ftp://medical.nema.org/medical/dicom/final/sup61_ft.pdf. Accessed 12 July 2007
30. Digital Imaging and Communications in Medicine (DICOM), Supplement 106: JPEG 2000 Interactive Protocol. Available at ftp://medical.nema.org/medical/dicom/final/sup106_ft.pdf. Accessed 12 July 2007
31. Digital Imaging and Communications in Medicine (DICOM), Strategic Document & Principal Contacts. Available at <http://medical.nema.org/dicom/geninfo/Strategy.pdf>. Accessed 10 October 2007

A software package demonstrating the utility of JPEG2000 and JPIP as an efficient means of using virtual slides in DICOM

Submitted abstract for 9th European Congress on Telepathology & 3rd International Congress on Virtual Microscopy (May, 15th to 17th 2008, Toledo, Spain)

Authors: Vilppu J. Tuominen, Jorma Isola

Main topic: Virtual Microscopy applications

Abstract: Large-scale utilization of virtual microscopy requires compatibility with DICOM, which is currently recognized as the standard specification for digital imaging in clinical medicine. Direct application of conventional DICOM-based systems in virtual microscopy is currently impossible, mainly because virtual slides (also known as whole-slide images) exceed the specification's image object size limit (2 gigabytes). One solution is to utilize the JPEG2000 standard, a well-suited image format for virtual microscopy, which is partly included in DICOM with Supplements 61 (lossless and lossy compression), 105 (multi-component transformations), and 106 (the JPEG2000 Interactive Protocol, JPIP). By using the JPIP protocol, image pixel data can be transmitted apart from patient data, and thus the DICOM image size limits can be overcome. Moreover, since the image and patient data are not connected, virtual slides are readily interchangeable with non-DICOM systems.

Although the possibility to use the JPIP protocol has been described in the DICOM standard specification, we are not aware of any open software solutions or libraries supporting it. In this study, we developed a software package to demonstrate the utility of JPEG2000 virtual slides and the JPIP protocol in a DICOM-based Picture Archiving and Communication System (PACS). First, we modified an open source DICOM library (OFFIS DICOM Toolkit) by adding the JPIP support as described in the Supplement 106. The library was used as a basis for our software package, consisting of a virtual slide construction application (JVScomp) and a DICOM client-server application (JVSdicom). JVScomp is our existing JPEG2000 compression application, which we extended to include

DICOM functionality. In addition to the actual JPEG2000 image file, JVScomp generates a DICOM file, which contains medical information, image properties, and a JPIP reference to the JPEG2000 image. Accompanied with our existing JPIP server (JVSserv), the server half of JVSdicom provides a proof-of-principle of a DICOM server capable of transmitting DICOM JPEG2000 virtual slides with the JPIP protocol. The client half of JVSdicom interacts with the server as a conventional DICOM client, but it can also invoke an external JPEG2000 virtual slide viewing application when a JPIP reference is received. As a default external viewing application, the software package features our existing JPEG2000 viewer (JVSview). JVSdicom is designed to be readily usable with commercial-grade DICOM servers. The software package can be obtained for free from our website (<http://jvsmicroscope.uta.fi/>), which also features a demonstrational DICOM JPIP server.

To our knowledge, the software package described in this study is the first practical solution to overcome the limitations of DICOM in virtual microscopy. Compared to other approaches, such as splitting virtual slide into thousands of small image tiles, which are stored as a DICOM Series (an ongoing DICOM WG-26 project), JPEG2000 (with JPIP) is a well-working alternative, and its utility in DICOM should be explored in detail.