

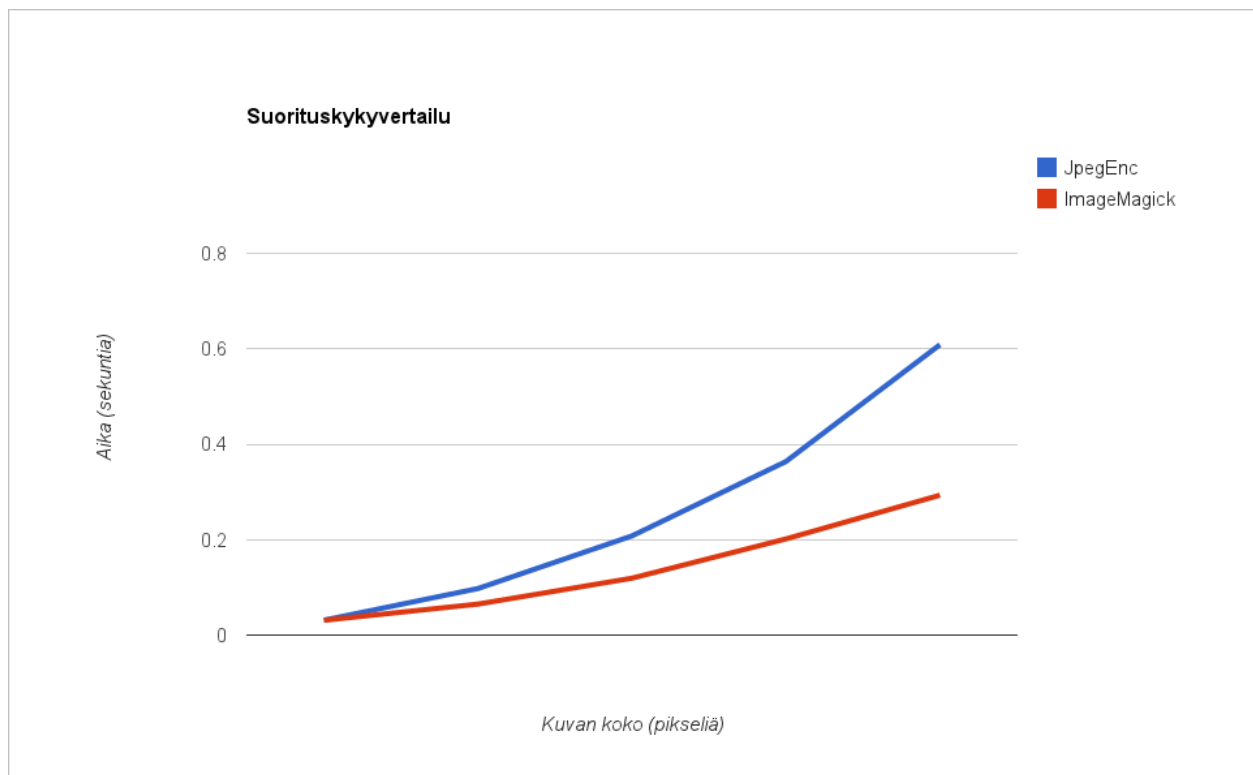
Testausdokumentti

Tietorakenteiden ja algoritmien harjoitustyö (vuodenvaihte 2013)

Matias Juntunen

Suorituskykytestaus

Ohjelman nopeutta testattiin virtuaalikoneella, joka sijaitsi Intelin 2.4-gigahertsisen Xeon E5645-suorittimen sisältävällä palvelimella. Verrokkiohjelmana toimi ImageMagick, joka käyttää JPEG-referenssikirjastoa libjpeg. Testikuvana toimi Wikimedia Commonsista ladattu maisemakuva, joka oli kooltaan 3236x1447 pikseliä.



Mittauksissa saatiin seuraavat tulokset:

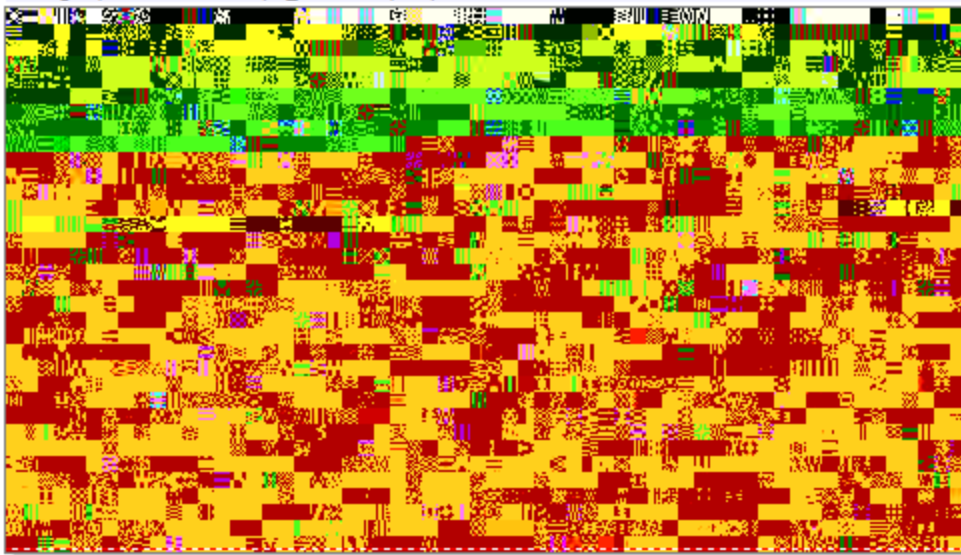
	JpegEnc	ImageMagick
3236x1447	0.609s	0.289s
2589x1158	0.364s	0.204s
1942x868	0.208s	0.122s
1294x579	0.097s	0.067s
647x289	0.031s	0.030s

Pahimmassa tapauksessa ohjelma oli siis hieman yli kaksi kertaa hitaampi kuin libjpegiä käyttävä ImageMagick. Ohjelma kuitenkin toimii odotetusti lineaarisessa ajassa kuvan koon suhteen. Pienillä kuvilla ohjelmat ovat tasoissa, koska eniten aikaa ei enää kulu kuvadatan pakkaamiseen, vaan muihin toimintoihin, kuten prosessin käynnistämiseen ja tiedostoista lukemiseen ja kirjoittamiseen. Suurempien kuvien suuri suorituskykyero selittyy kuitenkin sillä, että libjpegiä on kehitetty jo yli 20 vuotta useiden henkilöiden toimesta, ja siihen on ehditty tehdä lukuisia optimointeja.

Toiminnallinen testaus

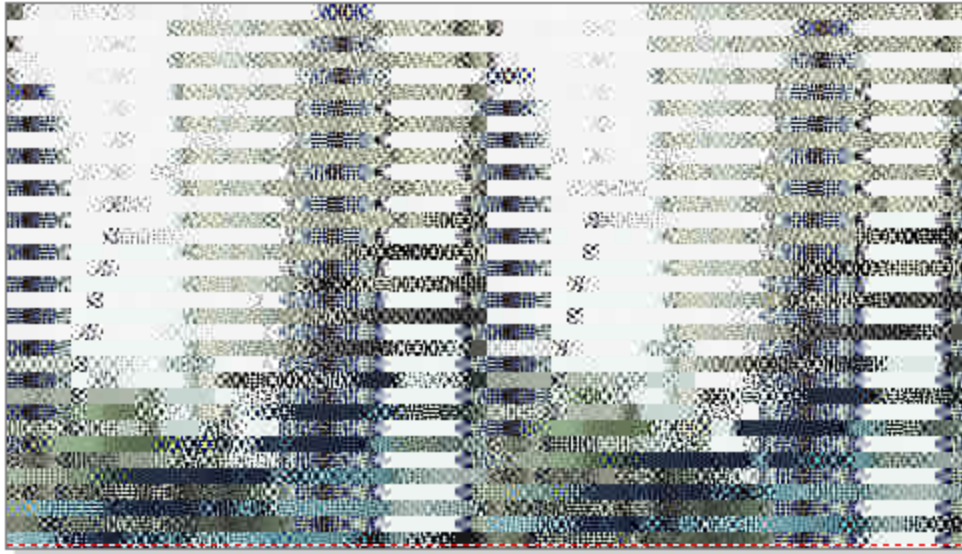
Ohjelman toiminnan testauksessa tuli esille useita ongelmia. Yksikkötestejä varten tarvitaan verrokkituloksia, mutta näiden tulosten hankkiminen on hankalaa. Olemassaolevien JPEG-pakkaajien lähdekoodin muokkaaminen ja tulosten hankkiminen tätä kautta olisi liian aikaavievää ja vaivalloista, joten ainoaksi vaihtoehdoksi jäi turvautua intuitioon ja saada ohjelma mahdollisimman nopeasti tuottamaan varsinaisia JPEG-tiedostoja. Joitain yksittäisiä osia ohjelmasta, kuten esimerkiksi matriisituloa ja bittitaulukoita oli kuitenkin helppo testata yksikkötesteillä. Pääasiallisesti ongelmien paikallistaminen vei melko paljon aikaa, mutta itse ongelman korjaaminen onnistui yleensä yhden rivin muokkaamisella.

Keskeneräisten JPEG-tiedostojen dekoddaamiseen käytettiin JPEGsnoop-ohjelmaa, joka ei tarvitse kokonaista JPEG-tiedostoa voidakseen purkaa siitä kuvadatan. Ensimmäisen kokeilun tulos näyttikin tältä:



Syynä kuvan ongelmaan oli se, että itse DCT-matriisin kertoimia ei koodattu lainkaan, vain niiden koko bitteinä ja niitä edeltävien nollien määrä.

Tämä oli kuitenkin helppo korjata ja tämän jälkeen kuva alkoi jo muistuttaa jossain määrin lähdekuvaa:



Kuvasta on selvästi tunnistettavissa tuttuja värejä, ja kuvan lohkojen uudelleenjärjestely paljastikin jotain itse kuvaa muistuttavaa. Tämän ongelman aiheutti väärin laskettu kuvalohkon sijainti kuvadatan joukossa. Ongelman korjaamisen jälkeen ongelmana oli vielä kuvan rakeisuus. Yksittäiset lohkot näyttivät jossain määrin oikealta, mutta itse kuvalohkon pikseleiden järjestys oli väärin. Tässäkin tapauksessa ongelma korjaantui yhden rivin muokkaamisella.



Lopullinen kuva näyttää tältä. Kuvassa näkyy selvästi kvantisaatiosta johtuvia pakkausartefakteja, joista JPEG-algoritmi tunnetaan.

Toiminnallisessa testauksessa kävi myös ilmi eri ohjelmien tapa kirjoittaa BMP-tiedostoja. Esimerkiksi Windowsin Paint-ohjelma tallentaa BMP-tiedostorakenteeseen kuvadatan koon,

mutta suosittu Paint.NET ei tee näin, vaan olettaa kuvaa lukevan ohjelman laskevan kuvadatan koon itse. Tämä aiheutti joitain ongelmia, jotka olisi voitu välttää käyttämällä valmista kirjastoa kuvadatan lataamiseen.