Tiralabran viikkoraportti #3 **30.5.2013** **Mikko Kangasmäki**
Kuvan pakkaus

Viime viikolla tajusin testejä kirjoittaessani, millainen sekasotku ohjelmani valmiskin osa oli. Aloin kokonaan alusta, mutta deadlineen ehättääkseni kirjoitin ohjelmasta vieläkin pahemman sekasotkun. Nyt aloitin maanantaina vielä uudelleen, ja tällä kertaa tuhlasin vähän aikaa ajatteluunkin.

Valmiin ohjelman tulisi nyt noudattaa kaavaa, jossa jokaiselle eri tiedon tallennusmuodolle on oma luokkansa, joka huolehtii näistä velvollisuuksista:
- kuvan tallentaminen tässä muodossa
- kuvan lukeminen tästä muodosta
- perusdatan muuttaminen tähän muotoon
- perusdataksi muuttaminen tästä muodosta.

Perusdatalla tarkoitetaan tässä kolmiulotteista byte-taulukkoa data[c][x][y], joka kuvailee kunkin pikselin perusvärien sävyn s.e. (x,y) on pikselin koordinaatit ja c ilmaisee, minkä perusvärin sävystä on kyse (sininen = 0, vihreä = 1, punainen = 2).

Havaitsin tässä esityksessä yhden ongelman: kun värit esitetään integer-muodossa (0,r,g,b), jokaista sävyä vastaa kokonaisluku 0-255. Kun ne muutetaan byteiksi, esimerkiksi laskutoimituksella

green = (byte) (rgbInteger >> 8),

tietoa ei mene hukkaan, mutta se vääristyy: Numerto 0,...,255 kuvautuvat numeroiksi 0,...,126 ,127, -128, -127,..., -2, -1. Näin vierekkäiset värisävyt jhoutuvat etäälle toisistaan, joten tiivistäminen ei tältä osin ota huomioon kuvan luonnollisia ominaisuuksia.

Tämän ongelman kiertämiseksi valitsen ensin värisävyä kuvaavan luvun 0,...,255, vähennän siitä 128, ja sitten vasta muutan byteksi. Tämä tekee koodista yleisesti ehkä vaikeampiselkoista, mutta pyrin kertomaan tämän kaikkialla javadocissa, missä oleellista.

Olen kokeillut jonkin verran itse waveletmuunnostakin. Koska dataa on suuria määriä (koekuvassani kuutisen miljoona pikseliä, eli n. 18 miljoonaa byteä), pyrin käyttämään tilaa säästeliäästi, vaikka se tekeekin koodista epäselvempää. Muunnos olisi helpompaa tallentaa kaksiulotteiseen taulukkoon, mutta siihen tulisi paljon hukkatilaa, joten olen käyttänyt yksiulotteista taulukkoa, mikä edellyttää vaikeampiselkoista indeksien käsittelyä.

Itse muunnos ei vielä toimi. Tai se toimii, mutta väärin. Sen sijaan tekemäni summataulukko, jonka indeksien pitäisi toimia samoin kuin muunnostaulukossa toimii. Tämä ei ole niinkään osaratkaisu muunnosongelmaan kuin siihen, että kuvan rivit eivät ole välttämättä kakkosen potenssin mittaisia.

Muunnoksessa ongelmaksi tuli, etten ole oikeasti harrastanut sellaisia oikeilla numeroilla tai diskreetissä maailmassa. Matematiikassa niiden käyttö on petollisen helppoa, koska vakioiden numeroarvoihin ei tarvinnut kiinnittää erityistä huomiota. Nyt ne ovat tärkein asia.

Koska muunnoksen kirjoittaminen suoraan datariville ei toiminut, aion perääntyä ja alkaa suosiolla kirjoittaa niiitä erikoistapaukseen, jossa rivin pituus on kakkosen potenssi.

Joitakin ongelmia voi tulla siitä, että muunnoksessa saatavat kertoimet eivät välttämättä ole kokonaislukuja. Pitäisi keksiä keino, jolla taataan, että muunnos pienentää datan määrää.