

Olen tällä viikolla aloittanut implementoimaan ohjelmaani. Opin paljon kuvanpakkauksesta ja diskreetistä kosinitransformaatiota. Olen nyt implementoinut DCT:n, mutta tahtoisin ymmärtää sen taustalla olevan matematiikan paremmin. **Epäselväksi on jäänyt, miten tällaiselle algoritmille laaditaan systemaattisia testejä. Siihen tarvitsisin ohjausta.** Palluroilla toiminnallisuudet, jotka on jo toteutettu.

- Perl-skripti kuva.pl, joka luo testikuvan. Testikuvassa on yksinkertaisia RGB-arvoja tavudatana.

PreProcessor -luokka:

- Tavudatasta syötteen lukeminen. Tähän meni paljon aikaa verrattuna siihen että koodia tuli lopulta vain pari riviä, sillä javassa ei pysty lukemaan suoraan Unsigned int -arvoja.
- Kuvan erottelu 8*8 -blokeiksi Blocks -taulukoon. Tällä hetkellä oletetaan että sekä kuvan korkeus että leveys ovat 8:lla jaollisia, jos ehdin niin lopuksi voin tutkia, miten algoritmi menisi jos eivät olisi.
- Konvertointi -ja vähennysfunktiot PreProcessor -luokkaan. Testikuvan arvot ovat RGB-dataa, joten ne täytyy konvertoida Y Cb Cr -muotoon, jolla DCT -algoritmi toimii. Sen lisäksi algoritmiin kuuluu että vähennetään luku 127, sillekin on oma funktio.

Transformer -luokka:

- 8*8 – blokkiin Diskreetin kosinitransformaation soveltaminen. Tämän tarkoituksena on käytännössä muuttaa pikselin yhden kanavan, joita on siis 3, yksi 8*8 -blokki tilallisesta ulottuvuudesta taajuusulottuvuuteen. [lähde 1]. Käytän tällä hetkellä epätehokkaampaa muunnosta, jos osaan niin voin myöhemmin muuttaa sen optimoiduksi versioksi.
- Myös käänteinen transformaatio implementoitu.
- Funktio, joka ottaa parametrinaan, kumpaa transformaatiota (käänteistä vai ei-käänteistä) sovelletaan ja tekee sen blocks -taulukon kaikille 8*8 -blokeille.
- Transformaation tekemisen jälkeen on Quantization -vaihe. Sen tarkoituksena on pyöristää arvoja niin, että lopulta tarvittaisiin mahdollisimman vähän bittejä tallennukseen. Käytän JPEGille optimoitua 8*8 Quantization -matriisia [lähde 2]. Eli siis pakkausvaiheessa jaetaan Quantization -matriisiin vastaavankohtaisella arvolla, purkamisvaiheessa kerrotaan.

Mitä seuraavaksi?

- Tiedostoonkirjoitus, testaamisen vuoksi. Eli siis teen kuvalle DCT:n ja sen jälkeen inverse-DCT:n ja kirjoitan .rgb- tiedostoon arvot. Siitä näen, miten muunnos toimii, eli kuinka paljon epäselvemmäksi se muuttaa kuvaa.
- Huffman -koodaukseen tarkemmin tutustuminen ja sen aloittaminen.

[1]: <http://unix4lyfe.org/dct/>

[2]: http://www.sersc.org/journals/IJFGCN/vol4_no3/10.pdf