Toteutusdokumentti

Tietorakenteiden ja algoritmien harjoitustyö (vuodenvaihde 2013)

Matias Juntunen

Ohjelman rakenne

Ohjelman rakenne on pääasiallisesti jaettu kahteen hakemistoon, util ja jpeg.

Util-hakemisto

- **block.h/.cpp:** Kuvalohkojen rakenne ja kuvalohkon uudelleenjärjestäminen.
- **blocksplitter.h/.cpp:** Kuvalohkojen erottaminen tiukasti pakatusta kuvadatasta.
- **bmp.h/.cpp:** BMP-kuvien lataaminen.
- **colorutil.h/.cpp:** Kuvadatan väriavaruusmuunnos.
- **matrix.h:** Matriisitulon toteutus.
- **bitvector.h/.cpp:** Taulukko, jonka jokainen alkio vastaa yhtä bittiä.

Jpeg-hakemisto

- **fdct.h/.cpp:** Kuvalohkon diskreetin kosinitransformaation toteutus.
- **huffman.h/.cpp:** Huffman-koodauksen toteutus.
- **quantization.h/.cpp:** Kuvalohkon kvantisoinnin toteutus.
- **jpegfile.h/.cpp:** Pakatun datan kirjoittaminen tiedostoon. (ei varsinaisesti osa työtä.)

Algoritmin kulku

1. Ladataan kuvadata BMP-tiedostosta.
2. Muunnetaan kuvadata RGB-väriavaruudesta YCbCr-väriavaruuteen.
3. Erotetaan yksittäinen 8x8 pikselin kokoinen lohko jokaisesta kanavasta ja tämän jälkeen
    a. Lohkolle tehdään diskreetti kosinitransformaatio.
    b. Lohkon yksittäiset alkiot kvantisoidaan kvantisointimatriisin avulla. Kvantisointimatriisi valitaan lohkon esittämän kanavan (Y, Cb tai Cr) perusteella.
    c. Lohkon alkiot järjestellään uudelleen “zigzag”-kuvion mukaan peräkkäisten nollien määrän maksimoiseksi.
    d. Lohko Huffman-koodataan bittivirtaan. Huffman-koodauksessa käytetyt koodisanat valitaan lohkon esittämän kanavan perusteella (sama kuin kvantisointimatriisin kohdalla).
4. Varsinaisen JPEG-tiedoston rakenne kirjoitetaan tiedostoon.
5. Koodattu kuvadata kirjoitetaan tiedostoon.

Saavutetut aikavaativuudet

Kaikki käytetyt algoritmit toimivat lineaarisessa ajassa. Kaikki yksittäistä lohkoa käsittelevät algoritmit toimivat myös vakiotilassa, paitsi Huffman-koodaus, jonka tilavaativuus on lineaarinen lohkon koon ja sisällön suhteen.