Ohjelman toteutus

Yleisrakenne

Ohjelma on jaettu useampaan moduliin, joista jotkut ovat riippuvaisia toisistaan. Funktioiden nimeämiskäytäntönä on käytetty tyyliä `moduuli_funktionimi`. Ohjelmassa on tekstipohjainen käyttöliittymä, joka käyttää compress-moduulin tarjoamia toimintoja kuvien lataukseen ja tallennukseen. Compress-moduuli käyttää lähes kaikkien muiden moduulien tarjoamia ominaisuuksia, sillä se sisältään pakkauksen korkean tason toimintalogiikan. Varsinaiset algoritmit (DCT, ZRL-pakkaus ja Huffman-koodaus) on toteutettu erillisissä moduuleissaan.

Moduulit

- util/bitbuf.h
  - bitin tarkkuudella toimiva dynaaminen taulukko
- util/stack.h
  - yksinkertainen dynaamisesti kasvava osoitinpino
- block.h
  - 8x8 pikselin ruutujen käsittely
- compress.h
  - korkean tason kuvanpakkausfunktiot
- dct.h
  - diskreetin kosinimuunnokset toteuttavat funktiot
- eks_math.h
  - C99-standardista puuttuvien matemaattisten funktioiden määrittelyt
- huffman.h
  - huffmanin-koodaukseen liittyvät funktiot
- jpeg.h
  - jpeg-kohtaiset vakiot
- stat/stat.h
  - testauksessa käytetyt tilastolliset toiminnot

- tiraimg.h
  - ohjelmakohtaiset vakiot
- trie.h
  - binääripuun toteutus

Saavutetut aika- ja tilavaativuudet

Pakkauksen tehokkuus ja laatu

Puutteet

Ohjelman sisäiset tietorakenteet ja tallennuksessa käytettävä tiedostoformaatti perustuvat oletukseen, että jokainen DCT-kerroin pystytään tallentamaan kvantisoinnin jälkeen 8-bitin mittaiseen muuttujaan. Matalamilla laatutasoilla kertoimet kvantisoidaan suuremmilla jakajilla, jolloin tallennettavien kertoimien itseisarvo pienenee, eikä tämä rakenteellinen rajoite koidu ongelmaksi. Kuitenkin käyttäessä korkeampaa laatua [1] artefaktit ovat ilmiselviä: kerroinmatriisin matalin taajuus (DC) on joissakin ruuduissa väärä, joka näkyy kirkkauden tai värisävyn suurena paikallisena vaihteluna.

Pakkausalgoritmi on myös varsin hidas, sillä trigonometristen funktioiden laskemista ei ole optimoitu millään tavalla. Jo yksinkertainen taulukointiratkaisu (mikä on käytössä useimmissa JPEG-toteutuksissa) toisi suuren hyödyn. Koko algoritmi on mahdollista toteuttaa myös pelkillä kokonaislukuoperaatioilla, mikä nopeuttaisi prosessointia huimasti.

[1] Tarkka arvo riippuu syötekuvasta, mutta yleensä virheitä alkaa esiintyä kun arvo on yli 80.