H.264-Bringer:

Dokumentacija

## Predgovor

U ovoj dokumentaciji opisani su postupci programiranja kodeka H.264 isključivo s tehničkog stajališta. Prethodno čitanju ove dokumentacije, kao teorijsku pozadinu preporuča se proučiti (ne nužno pročitati) diplomske radove koji opisuju ovaj kodek te u njima spomenutu literaturu.

# Važne globalne varijable

typedef struct {

int Lwidth,Lheight;

int Cwidth,Cheight;

unsigned char \*L, \*C[2];

} frame\_type;

**frame\_type**: Sadrži najvažnije podatke o slici. Članovi strukture su:

* **\*L** – niz luminantnih uzoraka slike (Y). Kardinalnost niza jednaka je broju piksela u slici, odnosno W\*H, gdje su W i H redom širina i visina slike u pikselima.
* **\*C[2]**–dva niza krominantnih uzoraka slike (Cb i Cr). C[0] odgovara komponenti Cb, a C[1] komponenti Cr. Kardinalnost svakog niza odgovara kardinalnosti krominantnih uzoraka u slici prema poduzorkovanju 4:2:0, odnosno (W/2)\*(H/2) , gdje su W i H redom širina i visina slike u pikselima.
* **Lwidth, Lheight, Cwidht, Cheight** – redom broj luminantnih uzoraka po širini i visini slike te broj krominantnih uzoraka po širini i visini slike. Vrijede jednakosti Cwidth = Lwidth / 2 i Cheight = Lheight / 2. Lwidth i Lheight su uvijek višekratnici broja 16, a Cwidth i Cheight su uvijek višekratnici broja 8. Ukoliko ulazne slike u koder nisu ovakvih dimenzija, one se obrezuju pri učitavanju.

**frame\_type frame**:Sadrži najvažnije podatke o slici koja se trenutno obrađuje.

**int CurrMbAddr** – indeks makrobloka koji se trenutno obrađuje. Raspon vrijednosti koje ova varijabla može poprimiti je [0, N–1], gdje je N broj makroblokova u slici.

## Inttransform.cpp

Upravljanje transformacijom i skaliranjem blokova ostataka dobivenih kao razlika stvarnih i previđenih vrijednosti.

### void transformDecoding4x4LumaResidual

### (int LumaLevel[16][16], int predL[16][16],

### int luma4x4BlkIdx, int QPy)

**LumaLevel**: Blokovi ostataka za cijeli makroblok

**predL**: predviđene vrijednosti za cijeli makroblok

**luma4x4BlkIdx**: položaj bloka unutar makrobloka

**QPy**: koeficijent kvantizacije

Provodi transformaciju i skaliranje blokova ostataka. Svi podaci obrađuju se unutar ove funkcije bez obzira jesu li dobiveni unutarslikovnim ili međuslikovnim predviđanjem i pripadaju li komponenti osvjetljenja ili boje. Nakon provođenja transformacije i skaliranja (funkcija scaleAndTransform4x4Residual), dobivene je vrijednosti potrebno rekonstruirati za što se koristi funkcija pictureConstruction4x4Luma.

### void transformDecodingIntra\_16x16Luma

### (int Intra16x16DCLevel[16], int Intra16x16ACLevel[16][16],int predL[16][16], int QPy)

**Intra16x16DCLevel**: DC koeficijenti

**Intra16x16ACLevel**: AC koeficijenti za trenutačni makroblok

**predL**: predviđene vrijednosti za trenutačni makroblok

**QPy**: koeficijent kvantizacije

Implementira transformaciju i skaliranje DC koeficijenata dobivenih unutarslikovnim predviđanjem pozivanjem funkcije InverseDCLumaIntra. Nakon obrade, koeficijenti se vraćaju u originalne blokove i provodi se transformacija i skaliranje blokova ostataka dopunjenih s DC koeficijentima (scaleAndTransform4x4Residual)

### void transformDecodingChroma

### (int ChromaDCLevel[4], int ChromaACLevel[4][16], int predC[8][8], int QPy, bool Cb)

**ChromaDCLevel**: DC koeficijenti komponente boje

**ChromaACLevel**: AC koeficijenti komponente boje

**predC**: predviđene vrijednosti

**QPy**: koeficijent kvantizacije

**Cb**: oznaka komponente boje

Funkcija obavlja transformaciju DC koeficijenata i nakon toga blokova ostataka. Sve je identično kao i u funkciji koja obavlja transformaciju DC koeficijenata makroblokova dobivenih unutarslikovnim predviđanjem.

## scaleTransform.cpp

Implementacija funkcija koje stvarno obrađuju podatke, za razliku od prethodno opisanih funkcija koje zapravo samo pozivaju funkcije implementirane u scaleTransform.cpp. Potrebno je napomenuti da su sve funkcije napisane u skladu s normom. Drugim riječima, svaka funkcija implementira jedno od poglavlja transformacije i kvantizacije opisanih u normi.

### int LevelScale[6][4][4]

Globalno polje (pregledna tablica) koje sadrži unaprijed izračunate faktore koji se koriste prilikom skaliranja.

### void inverseTransform4x4(int d[4][4], int r[4][4])

**d**: blok ulaznih podataka

**r**: blok izlaznih podataka (rekonstruirani)

Inverzna transformacija za blokove ostataka implementirana točno onako kako je definirana u normi.

### void inverseTransformDCLumaIntraFast

### (int c[4][4], int f[4][4])

**c**: blok ulaznih (kodiranih) DC koeficijenata

**f**: izlani podaci (dekodirani) DC koeficijenti

Dekodira DC koeficijente koji se nakon toga vraćaju u blokove ostataka i dalje obrađuju.

### void transformDCChromaFast (int c[2][2], int f[2][2])

**c**: blok ulazni DC koeficijenata

**f**: izlani (dekodirani) DC koeficijenti

Dekodira DC koeficijente komponente boje koji se vraćaju u blokove ostataka.

### void scaleResidualBlock

### (int c[4][4], int d[4][4], int qP, bool intra16x16OrChroma)

**c**: ulazni podatci (kvantizirani)

**d**: izlazni podatci (skalirani)

**qP**: kvantizacijski koeficijent

**intra16x16OrChroma**: oznaka radi li se o komponenti osvjetljenja ili unutarslikovnom predviđanju

Funkcija ostvaruje skaliranje blokova ostataka i u slučaju da se radi o unutarslikovnom bloku ili o bloku komponente boje prvi se element ne skalira.

### void scaleLumaDCIntra(int f[4][4], int qP, int dcY[4][4])

**f**: ulazni podaci

**qP**: kvantizacijski koeficijent

**dcY**: skalirani DC koeficijenti

Ova funkcija implementira skaliranje DC koeficijenata blokova dobivenih unutarslikovnim predviđanjem.

### void scaleChromaDC(int f[2][2], int qP, int dcC[2][2])

**f**: ulazni podaci (dobiveni inverznom transformacijom)

**qP**: kvantizacijski koeficijent

**dcC**: skalirani DC koeficijenti

Implementacija skaliranja DC koeficijenata koji pripadaju komponentama boje.

## quantizationTransform.cpp

Ova datoteka sadrži implementaciju kompletne transformacije i kvantizacije potrebne za realizaciju kodera.

### int LevelQuantize[6][4][4]

Globalno polje (pregledna tablica) koje sadrži unaprijed izračunate faktore potrebne za efikasnu implementaciju kvantizacije.

### void forwardTransform4x4(int r[4][4], int d[4][4])

**r**: blok ostataka kojeg je potrebno kodirati

**d**: transformirani blok

Transformacija bloka ostataka dobivenih kao razlika originalnih i predviđenih vrijednosti. Ova transformacija obrađuje svaki blok veličine 4x4 elemenata unutar videa.

### void forwardTransformDCLumaIntra(int f[4][4], int c[4][4])

**f**: polje ulaznih DC koeficijenata (unutarslikovno predviđanje)

**c**: transformirani DC koeficijenti

### void forwardTransformDCChroma (int f[2][2], int c[2][2])

**f**: polje ulazni DC koeficijenata (komponenta boje)

**c**: transformirani DC koeficijenti

### void quantisationResidualBlock

### (int d[4][4], int c[4][4], int qP, bool Intra16x16OrChroma)

**d**: ulazni (transformirani) podaci

**c**: izlazni (potpuno kodirani) podaci

**qP**: kvantizacijski koeficijent

**Intra16x16OrChroma**: oznaka radi li se o unutarslikovnom predviđanju ili komponenti boje

### void quantisationLumaDCIntra

### (int f[4][4], int qP, int c[4][4])

**f**: ulazni (transformirani) DC koeficijenti – unutarslikovno predviđanje

**qP**: kvantizacijski koeficijent

**c**: izlazni (kvantizirani, u potpunosti kodirani) podaci

### void quantisationChromaDC

### (int f[2][2], int qP, int c[2][2])

**f**: ulazni (transformirani) DC koeficijenti – komponenta boje

**qP**: kvantizacijski koeficijent

**c**: izlazni (kvantizirani, u potpunosti kodirani) podaci

### void transformScan

### (int c[4][4], int list[16], bool Intra16x16AC)

C: ulazno dvodimenzionalno polje dimenzija 4x4

List: izlazna lista (jednodimenzionalno polje)

Intra16x16AC: oznaka radi li se o AC koeficijentima za unutarslikovno predviđanje (u tom se slučaju ne popunjava prvi element liste jer je to DC koeficijent)

Ova funkcija provodi zig-zag preraspodjelu podataka iz dvodimenzionalnog polja u listu koja se šalje u entropijski koder.

### void quantizationTransform

### (int predL[16][16], int predCb[8][8],

### int predCr[8][8], bool reconstruct)

predL: predviđene vrijednosti za komponentu osvjetljenja za trenutačni makroblok

predCb: predviđene vrijednosti za prvu komponentu boje

predCr: predviđene vrijednosti za drugu komponentu boje

reconstruct: oznaka radi li se odmah i rekonstrukcija kodiranih blokova

Ova funkcija je zapravo omotač (engl. *wrapper*) svih procesa transformacije i kvantizacije u koderu, tj. sučelje prema ostalim modulima kodera. Unutar ove funkcije pozivaju se sve ostale funkcije ovisno o vrsti podataka, tj. predviđanja koje se koristi za trenutačni makroblok.