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.

## fer\_h264.cpp

Ova datoteka sadrži ulaznu točku u program. U njoj postoje samo dvije funkcije – decode() i encode() koje pokreću redom dekoder odnosno koder.

### void decode()

Upravlja dekodiranjem ulaznog toka sintakse H.264. Otvara ulazni i izlazni tok. Pokreće funkcije za upravljanje dinamičkim kadrovima (resursima), što uglavnom uključuje inicijaliziranje i oslobađanje dinamički alocirane memorije. Najvažnija zadaća ove funkcije je dohvaćanje po jedne NAL-jedinice i njezino predavanje funkciji RBSP\_decode u kojoj se odvija dekodiranje same sintakse H.264.

## nal.cpp

Upravljanje NAL-jedinicama u koderu i dekoderu. Komunikacija s tokom podataka koji sadrži sintaksu H.264. U jednoj se NAL-jedinici nalazi jedan skup parametara ili jedna čitava slika – nije podržana podjela slika na odsječke.

### Globalne varijable i strukture podataka

typedef struct

{

bool forbidden\_zero\_bit;

unsigned int nal\_ref\_idc, nal\_unit\_type, NumBytesInRBSP;

unsigned char \*rbsp\_byte;

}

NALunit;

**NALunit**: Struktura koja sadrži sve značajne podatke o NAL-jedinici. Značenja i nazivi varijabli jednaki su onima u Normi (vidi poglavlje 7.4 u Normi).

**\*NALbytes**: niz koji sadrži oktete trenutne NAL-jedinice.

### void getNAL(unsigned long \*fPtr, NALunit &nu)

**fPtr**: kazaljka na trenutni oktet unutar ulaznog toka. Ova se kazaljka koristi kao parametar offset u pozivu STL funkcije fseek. fPtr se predaje kao referenca.

**nu**: primjerak strukture NALunit, sadrži izdvojene podatke iz NAL jedinice koja će se sljedeća dekodirati. nu se predaje kao referenca.

Funkcija pronalazi veličinu trenutne NAL-jedinice u ulaznome toku pozivom funkcija findNALstart odnosno findNALend. Učitava NAL-jedinicu u niz NALbytes te ju predaje funkciji parseNAL koja podatke razvrstava u strukturu nu. Po završetku funkcije getNAL u strukturi nu se nalaze značajni podatci iz trenutne NAL-jedinice. Ova se funkcija poziva isključivo u dekokderu.

### void parseNAL(unsigned int NumBytesInNALunit,

### NALunit &nal\_unit);

**NumBytesInNALunit**: broj učitanih okteta sadržanih u globalnome nizu NALbytes.

**nal\_unit**: struktura u koju se popunjavaju interpretirani podatci u kontekstu NAL-jedinice pročitani iz niza NALbytes. nal\_unit se predaje kao referenca.

Odvaja podatke o NAL-jedinici od podataka RBSP-a i upisuje ih u strukturu nal\_unit. Pronađeni kodovi za izbjegavanje emulacije (0x000003) se izbacuju prije prijepisa podataka u niz nal\_unit.rbsp\_byte. Ova se funkcija poziva isključivo u dekokderu.

### void writeNAL(NALunit nu)

**nu**: struktura koja predstavlja kodiranu NAL-jedinicu koja će se zapisati u izlazni tok.

nu je ulazna struktura te su svi podatci već u njoj zapisani. Ova funkcija ispisuje te podatke u odgovarajućemu formatu u izlazni tok. Kako bi se smanjila učestalost pristupanja datoteci, NAL-jedinice se privremeno zapisuju u međuspremnik streamBuffer veličine streamBufferSize pozivom funkcije writeToBuffer. Za inicijalizaciju međuspremnika streamBuffer potrebno je jednom, pri pokretanju programa pozvati funkciju InitNAL.