

PRVI DOMAĆI ZADATAK - KOMPRESIJA SLIKE

Zadatak

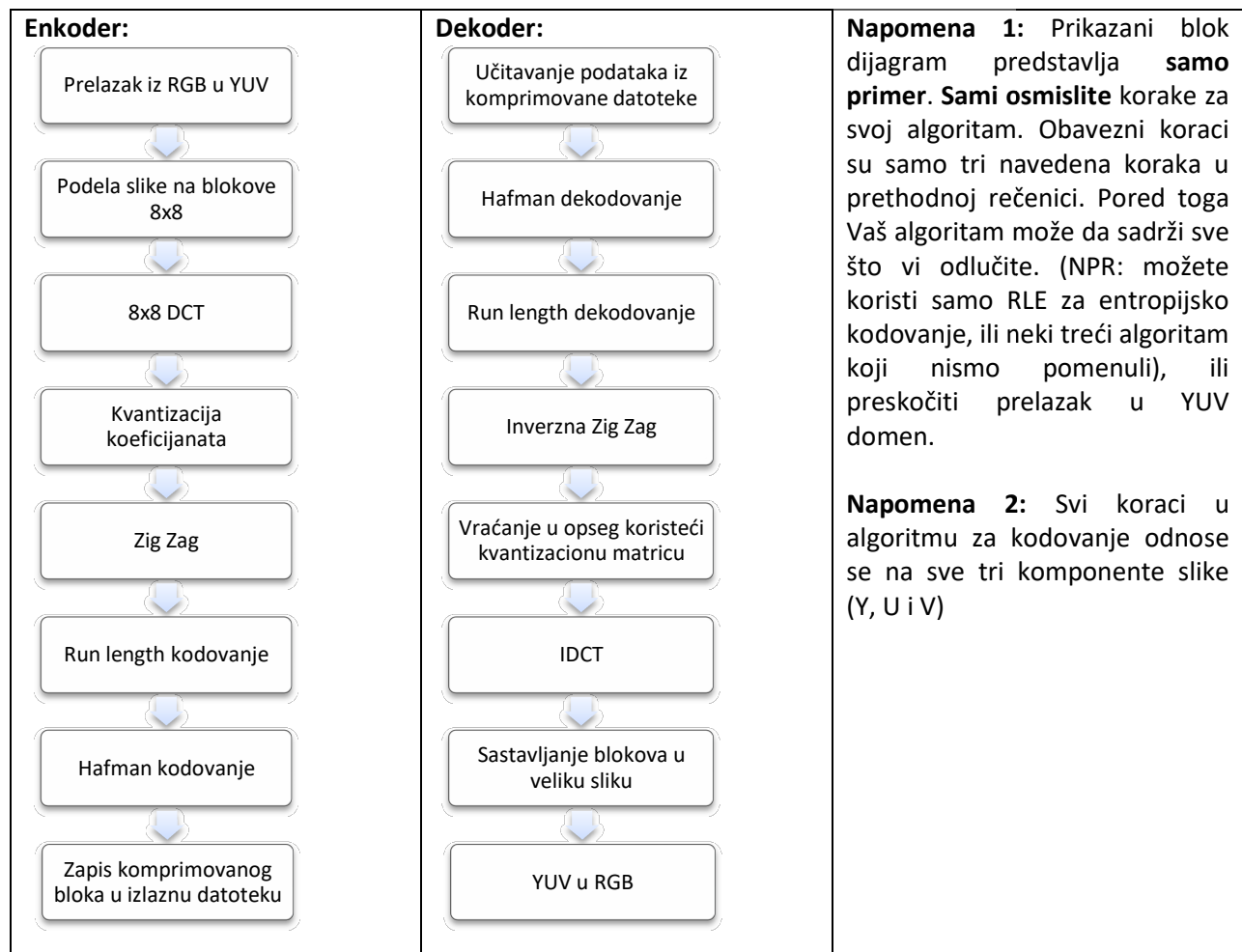
U okviru zadatka potrebno je **osmisлити** i **implementirati** algoritam za **enkodovanje** i **dekodovanje** slike.

Neophodno je da algoritam za enkodovanje sadrži:

- **Analizu slike (prelazak u spektralni domen), na koji god način odlučite da to radite**
- **Uklanjanje irelevantnih delova (kvantizacija spektralnih koeficijenata, odbacivanje nebitnih koeficijenata ili nešto treće)**
- **Neku formu entropijskog kodovanja**

Algoritam za dekodovanje treba da sadrži postupak inverzan dekodovanju.

U nastavku je dat primer jednog algoritma za kodovanje:



Pored algoritma, neophodno je osmisliti i **format komprimovane datoteke**. Datoteka treba da se sastoji iz dva dela: zaglavlje i podaci.

Zaglavlje treba da sadrži sve prateće podatke o slici. To mogu biti npr. dimenzije slike, kvantizacione matrice, rečnik u slučaju korišćenja Hafmana, veličina bloka kod DCT, itd. Drugim rečima, sve što je neophodno za proces dekodovanja.

Podaci sadrže komprimovane podatke, poređane u redosledu koji god odlučite (da li će ići prvo svi blokovi za jednu komponentu, pa drugu i na kraju treću, ili isprepleteno; da li ćete iterirati kroz sliku po visini ili po širini; itd.)

Pored ove dve stvari, potrebno je i da osmislite novu ekstenziju koju će imati datoteka koja sadrži komprimovanu sliku (kao što su .bmp, .jpg, .png).

Početni kod – postavka

Uz ovaj dokument priložena su dva projekta u Visual Studio okruženju: *Encoder* i *Dekoder*. Oba projekta predstavljaju konzolne aplikacije koje koriste QT okvir. QT okvir se koristi za potrebe učitavanja ulazne slike u fazi enkodovanja i zapisivanja izlazne slike u slučaju dekodovanja.

Oba projekta očekuju dva parametra na ulazu: Ulaznu datoteku i izlaznu datoteku.

Za ulaznu sliku u proces enkodovanja koristiti isključivo slike u nekomprimovanom formatu: .bmp. Isto važi i za izlaznu sliku iz dekodera.

Početni kod Enkodera učitava sliku, zapisuje zaglavlje koje sadrži početni simbol „RTRK“ (možete izmeniti početni simbol u svom formatu) i potom dimenzije slike kao dve 16-bitne vrednosti. Nakon toga, zapisuje se RGB slika u sirovom formatu (bez ikakve kompresije).

Dekoder učitava komprimovanu sliku, proverava zaglavlje, učitava veličinu slike, i potom učitava vrednosti piksela. Od pročitane matrice formira se izlazna, dekodovana slika.

U nastavku je dat pregled najbitnijih funkcija iz ova dva projekta.

Encoder	
Klasa <i>BitStreamWriter</i>	
<code>BitStreamWriter(QFile& output)</code>	Konstruktor klase. Prima otvorenu datoteku sa ulaznom komprimovanom datotekom.
<code>bool encode(uchar input[], int xSize, int ySize);</code>	Glavna funkcija klase. U okviru nje je potrebno realizovati proces enkodovanja. Za sada samo vrši zapis zaglavlja, dimenzija slike i vrednosti RGB piksela u sirovom formatu. Slika kao ulazni parametar prima matricu koja sadrži sliku u RGB888 formatu (<i>input</i>), i dimenzije slike (<i>xSize</i> i <i>ySize</i>).
<code>bool writeHeader()</code>	Funkcija koja zapisuje zaglavlje (oznaku „RTRK“) u izlaznu datoteku.
<code>bool writeImageInfo(int width, int height)</code>	Zapisuje dimenzije slike (dve 16-bitne vrednosti) u izlaznu datoteku.
<code>bool writeByte(uchar oneByte)</code>	Upisuje jednu osmobarbitnu vrednost u izlaznu datoteku.
<code>bool writeShort(ushort data)</code>	Upisuje jednu 16-bitnu vrednost u izlaznu datoteku.

<code>int writeData(uchar data[], int size)</code>	Upisuje zadati broj 8-bitnih vrednosti u izlaznu datoteku. Podaci se nalaze u nizu <i>data</i> , a parameter <i>size</i> predstavlja broj podataka. Povratna vrednost je broj uspešno zapisanih vrednosti.
Decoder	
Klasa <i>BitStreamReader</i>	
<code>BitStreamReader(QFile& input)</code>	Konstruktor klase. Prima otvorenu datoteku sa slikom (.bmp)
<code>bool decode(uchar* &output, int &xSize, int &ySize);</code>	Glavna funkcija klase. U okviru nje je potrebno realizovati proces dekodovanja. Za sada samo učitava zaglavlje, dimenzije slike i podatke. Sva tri parametra predstavljaju izlazne parametre (postavlja im se vrednost unutar funkcije). Nakon uspešnog izvršenja funkcije pokazivač <i>output</i> pokazuje na matricu koja sadrži sliku dimenzija <i>xSize*zSize</i> u RGB888 formatu.
<code>bool readHeader();</code>	Funkcija koja učitava zaglavlje (oznaku „RTRK“) iz ulazne datoteke.
<code>bool readImageInfo(int& width, int& height);</code>	Učitava dimenzije slike (dve 16-bitne vrednosti) iz ulazne datoteke.
<code>bool skip(int bytes);</code>	Preskače željeni broj bajta (osmootnih lokacija) iz ulazne datoteke.
<code>uchar readByte();</code>	Učitava jednu osmootbitnu vrednost iz ulazne datoteke
<code>ushort readShort();</code>	Učitava jednu 16-bitnu vrednost iz ulazne datoteke.
<code>int readData(uchar data[], int size);</code>	Učitava zadati broj 8-bitnih vrednosti i smešta u niz <i>data</i> . Potrebno je voditi računa da <i>data</i> ima dovoljno mesta da primi učitane podatke. Povratna vrednost je broj uspešno učitanih podataka.

Napomena: U projektima je naznačeno mesto koje Vam može poslužiti kao smernica odakle da počnete. Dozvoljeno je menjati početni kod na koji god način želite.

Očekivani rezultati

Nakon uspešne realizacije rešenja, potrebno je izvršiti testiranje sa nekoliko različitih slika. Prilikom pravljenja algoritma, cilj je da očuvate što veći kvalitet slike, uz što bolji dobitak kompresije.

Zbog toga, rezultat koji se od Vas očekuje za svaku testnu sliku koju ste odabrali je:

- Dobitak kompresije (količnik veličina slike pre i posle kompresije)
- Srednja kvadratna greška (MSE) između originalne slike i slike nakon dekodovanja
- Subjektivna ocena kvaliteta izlazne slike, uz opis oštećenja koja se javljaju na slici

Prve dve stavke potrebno je predstaviti u formi tabele.

Ukoliko su pojedine faze Vašeg algoritma parametrizovane (npr. broj koeficijenata B koji se odbacuje nakon DCT), isprobajte rešenje nad istom slikom, za različite parametre. Opišite uticaj parametra na kvalitet slike/stepen kompresije.

Nakon završetka implementacije i testiranja, potrebno je napisati dokumentaciju koristeći Šablon.doc. Šablon sadrži uputstvo za formatiranje teksta. U šablonu je naznačeno šta dokumentacija treba da sadrži. Dokumentacija treba da sadrži do 2 stranice (sve dodatne stranice nakon druge smeju da sadrže samo slike).

Završne napomene

1. Rešenje je potrebno okačiti na **Kanvas** najkasnije **do datuma odbrane u 00:00č**. Kao rešenje potrebno je okačiti oba projekta (Encoder i Decoder) i dokumentaciju u pdf formatu.
2. Dozvoljeno je korišćenje postojećih kodova sa vežbi ili ranijih projekata (npr. Kod za DCT, kod za Hafmanovo kodovanje iz prethodnog semestra, itd.)
3. Dozvoljeno je koristiti svu literaturu u potrazi za idejama. Predlažem udžbenike koje je profesor predložio na uvodnim predavanjima.
4. **Implementirajte algoritam korak po korak.** Npr. Ako u enkoderu uradite konverziju RGB u YUV, odmah nakon tog koraka zapišite sliku u izlaznu datoteku. Potom dodajte konverziju YUV u RGB u dekoderu i isprobajte rešenje. Tek kada ovo povežete da radi bez greške, pređite na implementaciju naredne faze algoritma. Tako ćete biti sigurni da uvek imate nešto što radi kako treba. **Nemojte da dozvolite sebi da dođete na odbranu sa rešenjem koje ne radi.**
5. Uz postavku Vam dajemo neke pomoćne dokumente koji Vam mogu biti od koristi.
6. Potrudite se da osmislite što bolje rešenje. Najefikasniji algoritam će biti nagrađen.