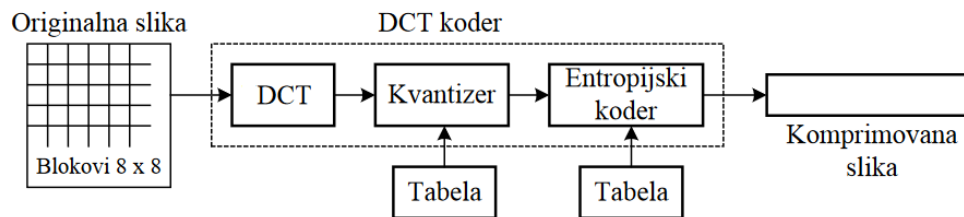


PROJEKTNI ZADATAK – JPEG ENKODER

Zadatak

U okviru zadatka potrebno je implementirati JPEG enkoder. Blok dijagram enkodera je dat na slici:



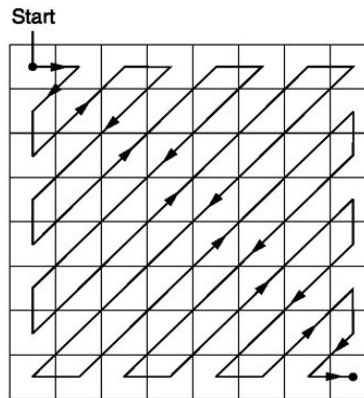
Prvi korak kod JPEG enkodera je transformacija slike iz RGB u YUV420 prostor boja. Nakon toga slika se deli na blokove veličine 8x8. Ukoliko veličina slike nije deljiva sa 16, vrši prethodno proširivanje slike.

Za svaki blok vrši se DCT transformacija. Obratite pažnju da priložena funkcija za DCT transformaciju vrši obradu nad ulaznim odbircima koji su označenog osmobitnog tipa. Pre poziva funkcije nad vrednosti Y komponente potrebno je konvertovati iz neoznačenog u označeni tip pomeranjem opsega, odnosno dodavanjem -128 na svaki koeficijent (Ovo je naprimera koji je rađen na vežbama bilo urađeno u samo funkciji za računanje DCT-a), Za U i V komponentu nije potrebno vršiti ovu konverziju pošto su već označenog tipa.

Zatim sledi kvantizacija DCT koeficijenata koristeći unapred pripremljenu matricu kvantizacije. Kvantizacija bloka koeficijenata veličine NxN vrši se tako što se svaki koeficijent **pomnoži** sa odgovarajućim faktorom kvantizacije, i potom se izvrši zaokruživanje rezultata kako bi se dobio ceo broj. Kvantizovane vrednosti nije potrebno množiti nakon kvantizacije kako bi se vratile u originalni opseg (kao na vežbama) jer se taj deo radi sa dekoderske strane.

Dodatak: Pre kvantizacije podesiti matricu kvantizacije sa **quality** parametrom korišenjem `void quantQuality()` funkcije.

Nakon kvantizacije koeficijenata, a pre primene entropijskog kodovanja potrebno je izvršiti konverziju 2D matrice u 1D niz. Ovo je potrebno uraditi primenom Zig-Zag transformacije.



Slika 1 - Zig zag transformacija za blok 8x8

Format izlazne datoteke

Izlazna datoteka formatirana je u skladu sa JPEG standardom. Uspešno komprimovanu sliku ćete moći da otvorite bilo kojim programom za pregledanje slika.

Sve potrebne funkcije za zapis podataka u izlaznu datoteku se nalaze u okviru klase *JPEGBitStreamWriter*

- `void writeHeader();` - Vršiti zapis zaglavlja datoteke po JFIF standard.
- `void writeQuantizationTables(const uint8_t LuminanceQuantTable[64], const uint8_t ChrominanceQuantTable[64]);` - Vršiti zapis prosleđenih kvantizacionih matrica. **Napomena: pre zapisa potrebno je izvršiti transformaciju prosleđenih tabela u 1D niz upotrebom Zig—Zag transformacije. Ovaj deo je potrebno implementirati.**
- `void writeImageInfo(int width, int height);` - Vršiti zapis informacija o samoj slici (dimenzije, broj komponenti i sl.)
- `void writeHuffmanTables();` - Vršiti zapisivanje Hafman rečnika potrebnog za dekodovanje
- `void finishStream();` - Zapisuje oznaku za završetak datoteke

ove metode su redom pozivane iz funkcije *performJPEGEncoding()*. Nakon poziva funkcije za zapis Hafman rečnika, sledi poziv funkcije za upis blokova za svaki kanal:

- `void writeBlockY(int16_t input[64])`
- `void writeBlockU(int16_t input[64])`
- `void writeBlockV(int16_t input[8*8])`

Redosled kojim je potrebno vršiti kompresiju (odnosno kojim se zapisuju komprimovani podaci u datoteku) je sledeći:

- 1) Iterirati kroz sliku prvo po širini, pa onda po visini. Odnosno tek nakon obrađenog čitavog reda preći u sledeći.
- 2) Komprimovati 4 susedna bloka podataka iz Y komponente (blok 16*16), i to tako što će se prvo obraditi gornji levi, pa gornji desni, donji levi pa donji desni.
- 3) Komprimovati 1 blok podataka U komponente koji se odnosi na 4 bloka iz prethodnog koraka
- 4) Komprimovati 1 blok podataka V komponente koji se odnosi na 4 bloka iz prethodnog koraka

Priprema za laboratorijske vežbe iz predmeta Osnovi algoritama i struktura DSP II
Projektni zadatak - JPEG enkoder

Na slici ispod je prikazan redosled obrade blokova za sliku dimenzije 64*32. Svaki kvadrat predstavlja blok veličine 8x8, dok broj u kvadratu predstavlja redni broj bloka u izlaznoj datoteci.

Y							
1	2	7	8	13	14	19	20
3	4	9	10	15	16	21	22
25	26	31	32	37	38	43	44
27	28	33	34	39	40	45	46

U			
5	11	17	23
29	35	41	47

V			
6	12	18	24
30	36	42	48

Kada završite implementaciju algoritma, potrebno je isprobati rešenje za različite ulazne slike. Odaberite za ulazne slike one koje su nekompresovane (najbolje bmp format). Potrebno je napraviti tabelu u formatu:

Ulazna slika	Dimenzije	Veličina ulazne slike [KiB]	Veličina izlazne slike [KiB]

Dodatak: Koristiti **quality** parametar. Izabrati jednu sliku i usporediti stepen kompresije za različite vrednosti ovog parametra:

Kvalitet	Veličina ulazne slike [KiB]	Veličina slike sa najboljim kvalitetom kompresije [KiB]	Veličina izlazne slike [KiB]

Ovaj projekat nosi 15 bodova.

Vremenski tok izrade projekta:

- 19. odnosno 21. aprila će na vežbama biti kontrolna tačka, na kojoj će se raditi izrada projekta uz mogućnost konsultovanja asistenata, a uz njihov pregled dosadašnjeg napretka.
- U nedelji 25. - 29. aprila će se u dogovorenom popodnevnom terminu vršiti odbrana zadatka pred komisijom asistenata, gde će biti proverena originalnost rešenja i razumevanje istog.