

Informacija

- Osnovna ideja: pronaći i ukloniti redundanciju u podacima
- Svojstvena (vlastita informacija)
 - Kvantitativna mjera za količinu informacije

$$i(A) = \log_b (1/P(A)) = - \log_b P(A)$$

gdje je: $P(A)$ vjerojatnost pojavljivanja događaja A

- primjer: za $P(A)=1$ $i(A)=0$
 za $P(A)=0$ $i(A)=\infty$

Informacija

- $i(A)$ nam formalno definira ono što je logično:
 - ako se nešto događa često (sa velikom vjerojatnošću) tada nam takav događaj ne donosi puno informacija (npr. suma brojeva na dvije kockice je 7)
 - rijetki događaji donose puno informacija (npr. suma brojeva na dvije kockice je 2)
- ako izaberemo \log_2 tada je količina informacija izražena u bitovima

Entropija

- $S = \{a_1, a_2, a_3, \dots, a_N\}$ – skup svih mogućih simbola sa vjerojatnostima $P(a_1), P(a_2), \dots, P(a_N)$

$$\mathbf{H(S)} = \sum P(a_n) i(a_n) = - \sum P(a_n) \log_b P(a_n)$$

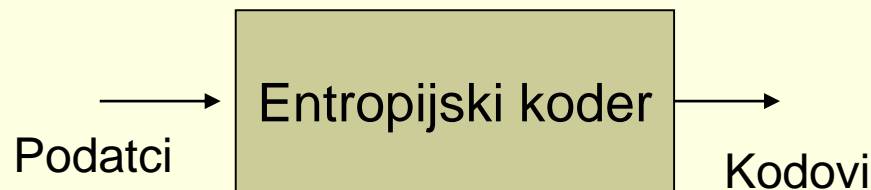
- H predstavlja *prosječni* vlastiti sadržaj informacije za izvor S i naziva se Entropija.
- Gornji izraz predstavlja pojednostavljenje stvarnog izraza za entropiju i naziva se entropija nultog reda i biti će dovoljno dobra aproksimacija za naša predavanja

Entropija

- Ako izaberemo \log_2 tada vrijednost entropije definira **najmanji prosječan broj bitova** potreban za kodiranje pojedinog simbola iz ulaznog niza
- Stvarnu entropiju slučajnog izvora za općeniti slučaj nikada ne možemo doznati

MAS – Osnove kompresije

- Kompresija podataka:



- Entropijski koder (statističko kodiranje): uklanja statističku redundanciju iz toka podataka

MAS – Entropijsko kodiranje

- Entropijsko kodiranje:
 - Huffmanovo kodiranje
 - RLE (engl. run length encoding)
 - Aritmetičko kodiranje
 - Shannon-Fano, Golomb-Rice, ...

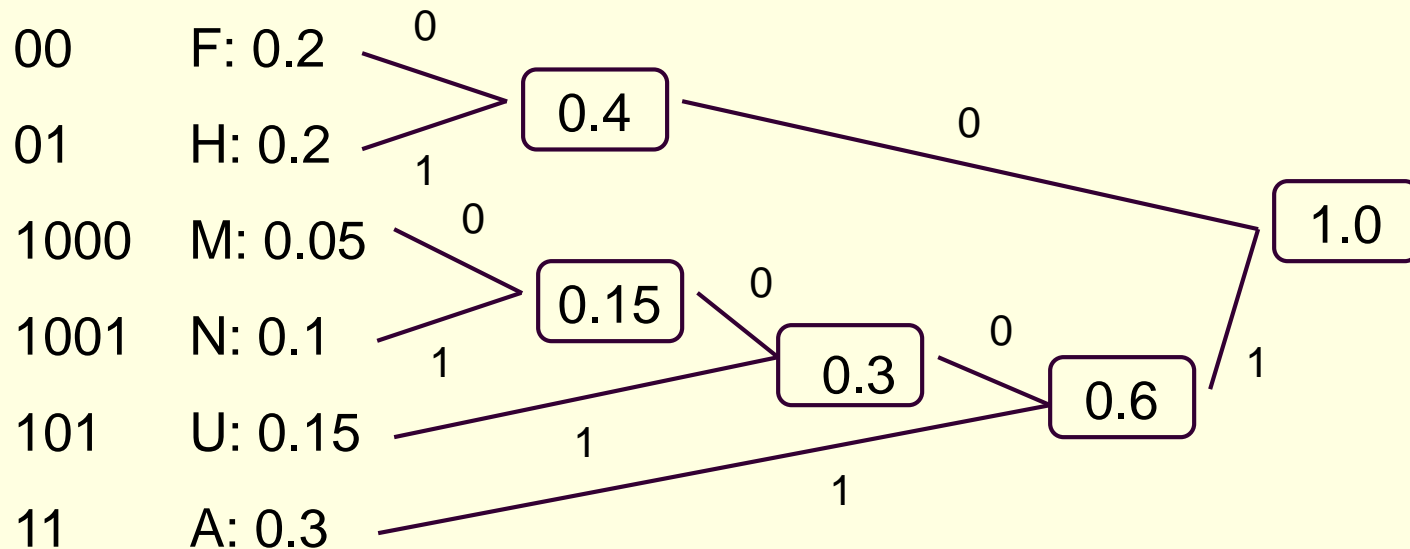


MAS – Osnove kompresije: Huffmanovo kodiranje

- Huffman, 1952:
 - Huffman-ovo kodiranje
 - Optimalni cjelobrojni kodovi (najbliži entropiji modela)
 - Prefiks kodovi (nijedan kod nije početak nekog drugog koda)
 - Često korištena metoda

MAS – Huffmanovo kodiranje

- Jednostavna prezentacija pomoću binarnog stabla



“HUFFMAN” = 0110100001000111001

56 bitova

19 bitova

MAS – Huffmanovo kodiranje

- Generiranje Huffmanovih kodova:
 - Statički – dva prolaza po podatcima, prvi skupljanje vjerojatnosti, drugi kodiranje, moraju se prenijeti i huffmanovi kodovi
 - Adaptivno – jedan prolaz, model (Huffmanovo stablo) se adaptivno izgrađuje tijekom prijelaza po podatcima, računski zahtjevnije jer i kompresor i dekompresor adaptivno izgrađuju model

MAS – Osnove kompresije: RLE kodiranje

- Zasniva se na uzastopnom ponavljanju jednog simbola
 - Jednostavna izvedba
 - Primjer:
 - linije očitane sa dokumenta u fax uređaju
- 1 linija (75 dpi, 8")=600 bita=75 B

[illegible]

17,24,3,211,22,188,77,54,4 = 9 B

Osnovni modeli

- Fizikalni model
 - ako poznajemo fizikalna svojstva izvora
 - obično je ovo prekompleksan model te se u takvim slučajevima pokušava koristiti neki alternativni model

Osnovni modeli

- Vjerojatnosni model
 - često korišten model, osnova za neke vrlo efikasne modele
 - pretpostavka je da su svi simboli generirani iz izvora potpuno neovisni
- Dvije osnovne varijante
 - ‘slijepi’ model: dodatno pretpostavljamo da je za sve simbole vjerojatnost pojavljivanja ista
 - statistički model: na neki način izračunati učestalost pojavljivanja simbola te na temelju toga definirati vjerojatnosti
- Problem: mogućnost POVEĆANJA ENTROPIJE!!

Osnovni modeli

- Markovljev model (A.A. Markov 1856-1922)
 - ovim modelom opisuje se izvor ‘s pamćenjem’ tj izvor kod kojeg vjerojatnost pojavljivanja određenog simbola ovisi o svim simbolima koji su se pojavili prethodno u nizu
 - tako za Markovljev model prvog reda vrijedi
$$P(A_i|A_{i-1}, A_{i-2}, A_{i-3}, \dots) = P(A_i|A_{i-1})$$
 - ovaj model je izuzetno efikasan za određene tipove podataka (npr model “predviđanja” spada u ovaj model)
 - ponovo postoji opasnost od povećanja entropije!!

MAS – neki modeli

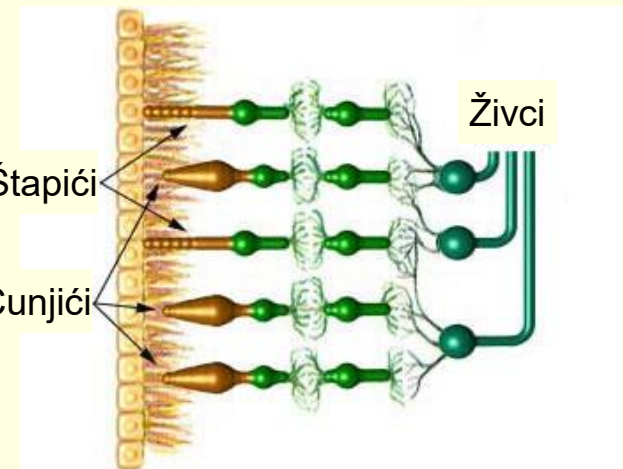
- Primjeri modela:
 - Slikovni podatci – prostorna korelacija u 2D prostoru, HVS
 - Video podatci – prostorna i vremenska korelacija podataka, HVS
 - Financijski podatci – korelacija u 1D, predviđanje

Model za slikovne podatke

- Podaci bi se mogli obraditi (kompresirati) i bez upotrebe posebno odabranog modela
- Npr. ZIP, RAR, ...
- Iz iskustva znamo da time nećemo ostvariti značajne omjere kompresije
- Upotrebljava se model koji omogućuje znatno veće omjere kompresije

Model zasnovan na našem vizualnom sustavu

- fotoreceptori mrežnice
 - štapići (*rods*) i čunjići (*cones*)
 - sadrže kemijske tvari osjetljive na svjetlost
- čunjići i štapići nisu jednako osjetljivi na cijeli spektar vidljive svjetlosti
 - različite vrste monokromatskog svjetla podražuju ili čunjiće ili štapiće
- Ljudski vizualni sustav – HVS



HVS – spektralna osjetljivost

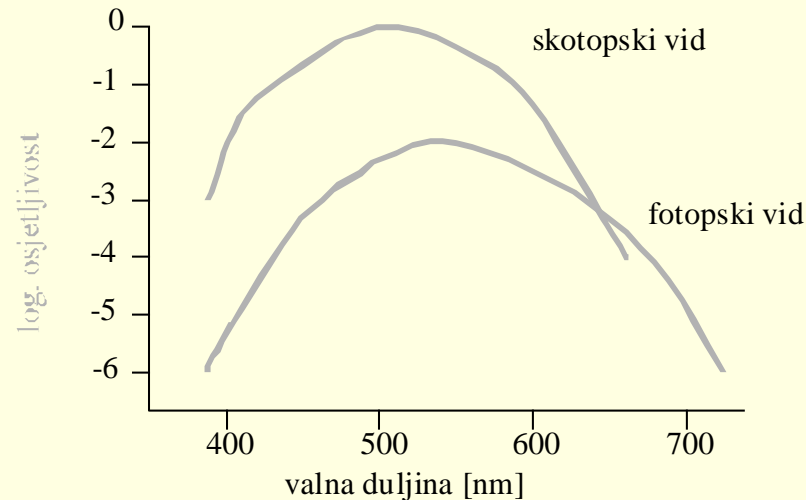
■ štapići

- osjetljivi na svjetlo i pri niskim razinama luminancije ispod 1cd/m^2 ("noćno" gledanje ili skotopski vid)
- mogu razlikovati samo promjene u luminanciji, a nisu osjetljivi na boju

■ čunjići

- doprinose osjetu i razlikovanju boja, a postaju aktivni pri višim razinama luminancije
- kod razina luminancije između 1cd/m^2 i 100cd/m^2 aktivni su i štapići i čunjići (fotopski vid)
- pri razinama luminancije većim od 100cd/m^2 štapići postaju zasićeni i aktivni su samo čunjići

HVS – spektralna osjetljivost



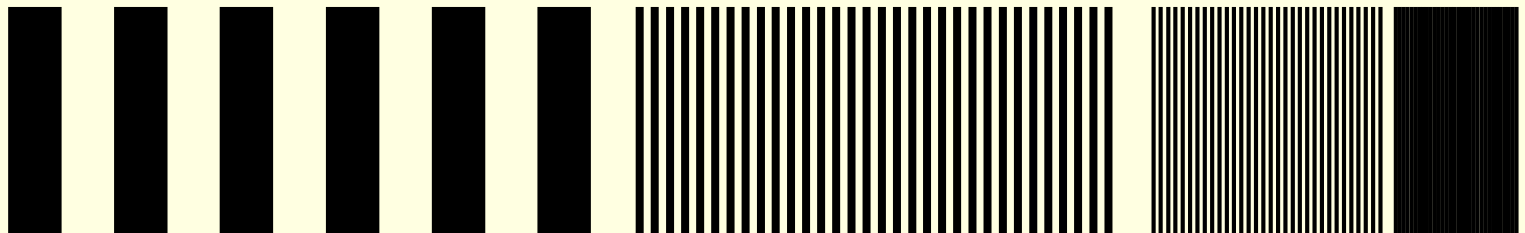
Ljudsko oko ima cca 10-20 puta više štapića nego čunjića

ZAKLJUČAK #1:

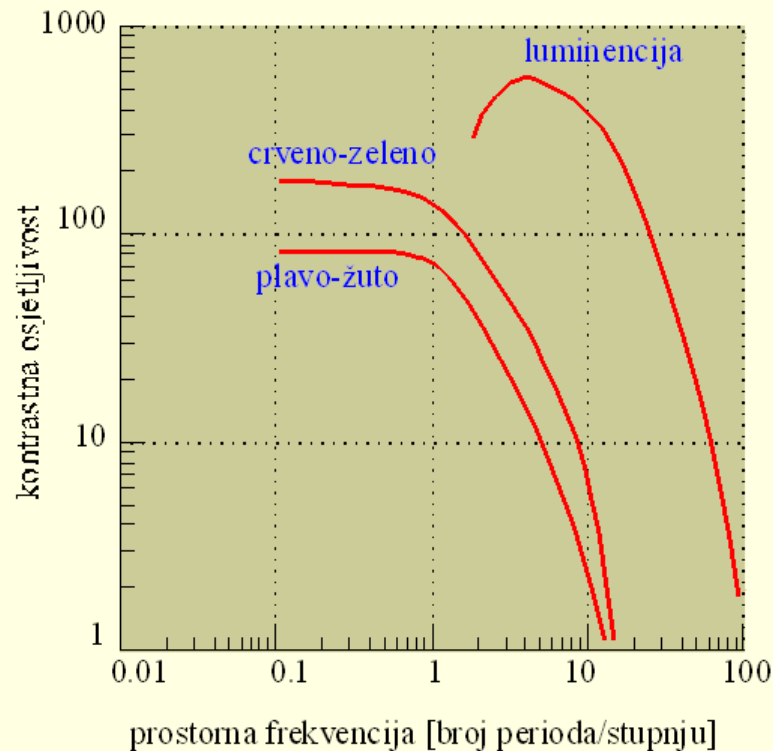
Znatno smo osjetljiviji na promjene u luminanciji nego u boji

HVS – prostorna osjetljivost

- Zbog konačne fizičke gustoće receptora, oko može razaznati detalje ako su zrake svjetlosti pod kutem upada većim od cca 1 minute
- Prostorna frekvencija



HVS – prostorna osjetljivost



ZAKLJUČAK #2:

Oko je neosjetljivo na visoke prostorne frekvencije , a posebno to vrijedi za komponente boje.

HVS Model za slike

- Kako iskoristiti prethodna saznanja o ljudskom vizualnom sustavu da bi omogućili veće omjere kompresije
- Cilj bi nam bio:
 - Zadržati što više informacija o luminantnoj komponenti
 - Izbaciti prostorne frekvencije koje oko ne vidi

Promjena prostora boja

- Ulazni podaci: uglavnom RGB
- Prikladan prostor boja: YUV
- RGB-YUV konverzija vrlo se lako može obaviti jednostavnom matričnom operacijom:

$$Y = (0,257 * R) + (0,504 * G) + (0,098 * B) + 16$$

$$U = -(0,148 * R) - (0,291 * G) + (0,439 * B) + 128$$

$$V = (0,439 * R) - (0,368 * G) - (0,071 * B) + 128$$

- Kompleksnost (za kasnije analize): za izračun svakog piksela treba 9 množenja i 9 zbrajanja (+dohvat, spremanje)

Promjena prostora boja

Postoji i nešto drugačija formula za konverziju koja se koristi u JPEG-u a definirana je u JFIF dokumentu:

$$Y = 0.299 R + 0.587 G + 0.114 B$$

$$Cb = - 0.1687 R - 0.3313 G + 0.5 B + 128$$

$$Cr = 0.5 R - 0.4187 G - 0.0813 B + 128$$

Što smo postigli konverzijom?

- Dobili smo dva skupa komponenata
 - Y: luminancija, oko je znatno osjetljivije na ovu komponentu
 - U,V: boja, oko manje osjetljivo
- Ideja je da prethodna dva skupa obrađujemo RAZLIČITO
- No ovime još uvijek nismo postigli apsolutno nikakvu promjenu u količini podataka

Poduzorkovanje

- “subsampling”
- Jedna od najjednostavnijih metoda kako nakon konverzije u YUV prostor boja možemo smanjiti količinu podataka je poduzorkovanje:
 - Oko nije toliko osjetljivo na prostornu frekvenciju komponenata boje te se one ne prenose za svaki piksel
 - 4:4:4, 4:2:2,...

Transformacija

- Prebacivanjem podataka u frekvencijsku domenu želimo dobiti informacije o frekvencijskim karakteristikama svake komponente
- Koju transformaciju izabrati?
 - Želja nam je da se nakon transformacije većina informacija zadrži u što manjem broju što nižih frekvencijskih elemenata
 - Prema teoriji: Karhunen-Loève (KLT) je idealna (ali je potpuno nepraktična za primjenu)

Transformacija: DCT

- Iz teorije se može vidjeti da je diskretna kosinusna transformacija (DCT) vrlo bliska idealnoj
- Prednost DCT:
 - Može se jednostavnije izračunati (brzi algoritmi)

DCT

$$F(u,v) = \frac{2}{\sqrt{M \cdot N}} \cdot C(u) \cdot C(v) \cdot \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} f(i,j) \cdot \cos\left[\frac{(2 \cdot i + 1) \cdot u \cdot \pi}{2 \cdot M}\right] \cdot \cos\left[\frac{(2 \cdot j + 1) \cdot v \cdot \pi}{2 \cdot N}\right]$$

$F(u,v)$ - transformacijski koeficijent

$f(i,j)$ - amplituda elemenata slike u bloku

u, v - koordinate u području transformacije (prostorne frekvencije)

i, j - koordinate u području elemenata slike

$C(u)=C(v) = (1/2)^{1/2}$, za $u,v = 0$

$C(u)=C(v) = 1$, za $u = 1,2,...M-1, v = 1,2,...N-1$

DCT za slike

- DCT kod obrade slika uglavnom se obavlja nad blokovima podataka veličine 8x8:

$$F(u,v) = \frac{1}{4} \cdot C(u)C(v) \cdot \sum_{i=0}^7 \sum_{j=0}^7 f(i,j) \cdot \cos\left[\frac{(2 \cdot i + 1) \cdot u \cdot \pi}{16}\right] \cdot \cos\left[\frac{(2 \cdot j + 1) \cdot v \cdot \pi}{16}\right]$$

- Poduzorkovanje je vrlo “primitivna” i nekvalitetna metoda
- DC koeficijent:

$$F(0,0) = \frac{1}{8} \cdot \sum_{i=0}^7 \sum_{j=0}^7 f(i,j)$$

(DC=8 x srednja vrijednost elemenata bloka)

IDCT

- Na sličan način definirana je i inverzna DCT:

$$f(i,j) = \frac{1}{4} \cdot \sum_{v=0}^7 \sum_{u=0}^7 C(u)C(v)F(v,u) \cdot \cos\left[\frac{(2 \cdot i + 1) \cdot u \cdot \pi}{16}\right] \cdot \cos\left[\frac{(2 \cdot j + 1) \cdot v \cdot \pi}{16}\right]$$

- Vidimo da DCT koeficijenti $F(v,u)$ u stvari predstavljaju faktore kojima množimo bazne valne oblike pri restauraciji signala

Opseg podataka

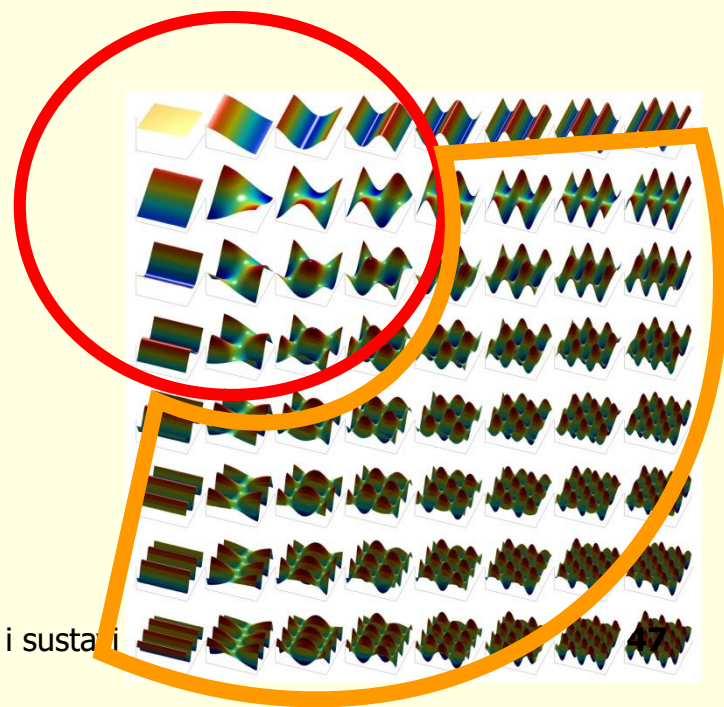
- DCT dovodi do proširenja opsega podataka (dinamički opseg za 8x8 2D DCT je 2^3 puta veći u odnosu na ulaz)
 - Ako na ulazu imamo 8 bitovne podatke nakon 2D DCT imati ćemo 11 bitovne koeficijente!!
- Još uvijek su SVI podaci sačuvani i moguće je obaviti perfektnu rekonstrukciju (uz uvažavanje nepreciznosti matematičkih izračuna)

Pomak kod DCT

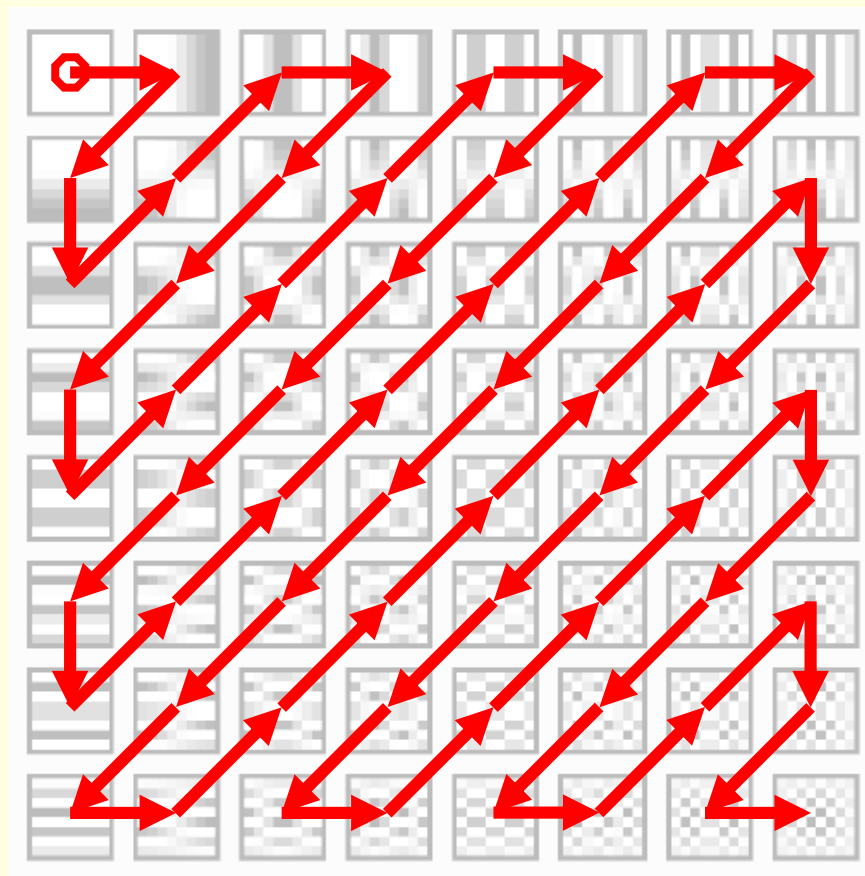
- Slikovni podaci na ulazu su pozitivni cijeli brojevi (npr. 0-255)
- S obzirom da je DCT definirana i za pozitivno i negativno područje na ovaj način bi se polovica ulaznog prostora izgubila (a time i znatno smanjila efikasnost)
- Zato se prije DCT sve vrijednosti na ulazu translaticiraju za polovicu opsega tj. -128
- Prema tome umjesto da ulazni elementi budu u opsegu [0-255] biti će u opsegu [-128 – 127]

Što smo dobili sa DCT?

- Energija signala koncentrirana u nižim frekvencijama
- Značajniji koeficijenti
- Manje značajni koef.



Cik-cak (Zig-zag) reorganizacija



Kako iskoristiti DCT?

- Nakon DCT, uz poznavanje prosječnih osjetljivosti oka možemo smanjivati količinu podataka koja opisuje frekvencije na koje naše oko nije osjetljivo
- To ne možemo učiniti potpunim odbacivanjem visokofrekventnih komponenata već smanjenjem njihove preciznosti (problemi naglih prijelaza !)

Što je kvantizacija

- Postupak kojim se smanjuje dinamički opseg ulaznih vrijednosti (a time i potreban broj bita)
- Ulazni podatak dijeli se sa zadanim brojem (kvantizacijski korak)
- Kod inverznog postupka podatak se množi sa kvantizacijskim korakom
- Primjer (Q=4):
 - Ulazni niz 5,11,4,17,1 (potrebno 5 bitova za prikaz)
 - Izlazni niz 1,3,1,4,0 (potrebno 3 bita za prikaz)
 - Restaurirani niz 4,12,4,16,0 (**GUBITAK PODATAKA**)
- **KVANTIZACIJOM DOLAZI DO NEPOVRATNOG GUBITKA INFORMACIJA !!!**

Kvantizacija nakon DCT

- kvantizacijska tablica sa 64 vrijednosti $q(u,v)$ određenih na temelju HVS (koraci kvantizacije)
- Svaki DCT koeficijent $F(u,v)$ dijeli se sa pripadnim (skaliranim) faktorom $q(u,v)$, a rezultat se zaokružuje na najbližu cjelobrojnu vrijednost
- Skaliranjem sa faktorom S se pojednostavljeno određuje stupanj kompresije i “kvaliteta” slike
- vrijednosti nastale kvantizacijom $S(u,v)$ su:

$$S(u,v) = \text{round}\left(\frac{F(u,v)}{Q(u,v)}\right) = \text{round}\left(\frac{F(u,v)}{q(u,v)S}\right)$$

Kvantizacija nakon DCT

- Utjecaj pojedinih prostornih frekvencija može se proizvoljno kontrolirati postupkom kvantizacije
- S obzirom da smo razdvojili Y i U, V komponente sada možemo Y DCT frekvencijske koeficijente kvantizirati sa različitim kvantizacijskim vrijednostima od U, V komponenti (kvant. koef. za U, V biti će veći zbog HVS)
- Također kvantizacijski koeficijenti za visoke frekvencije mogu biti veći

Tipične kvantizacijske tablice

Table K.1 – Luminance quantization table

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

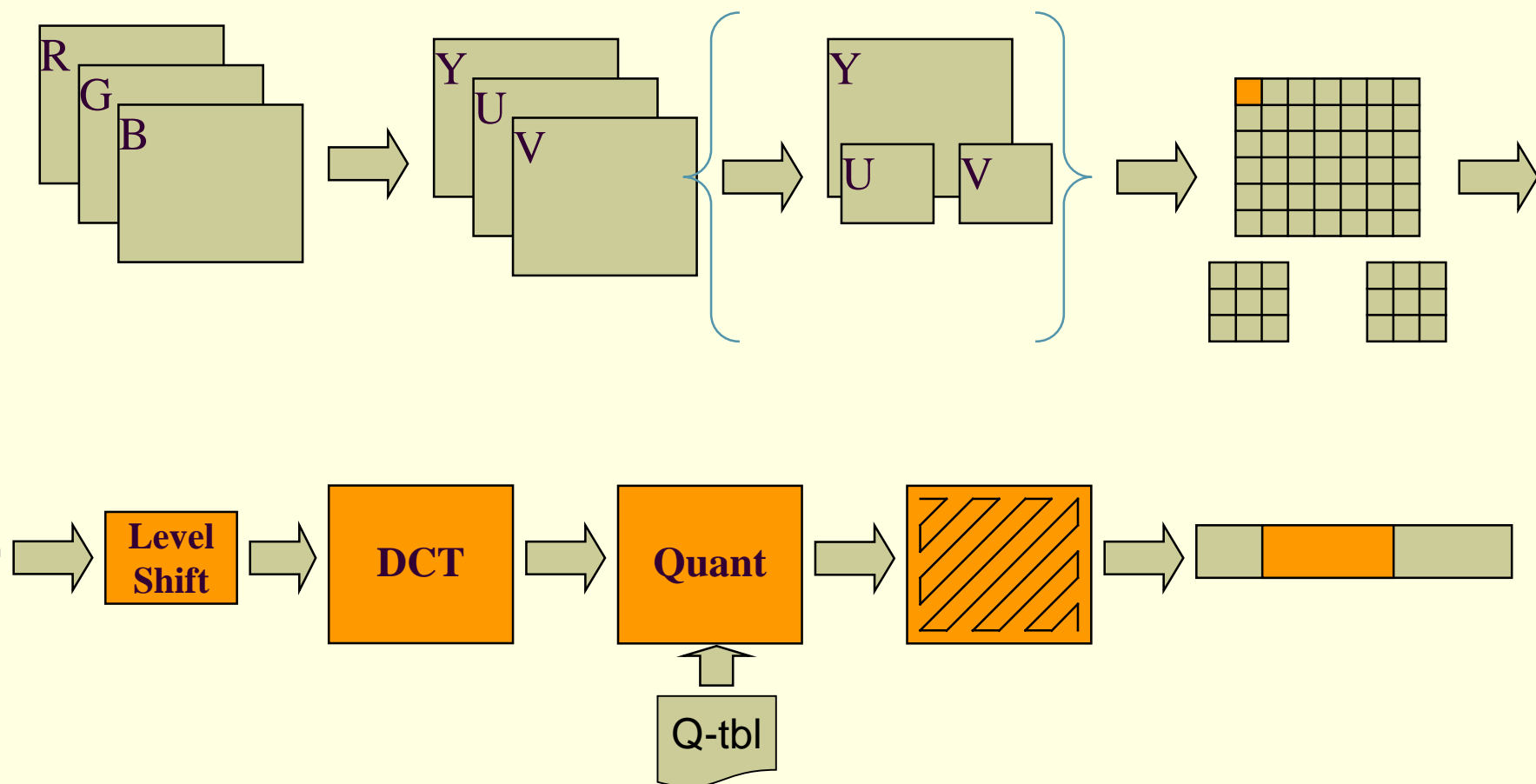
Table K.2 – Chrominance quantization table

17	18	24	47	99	99	99	99
18	21	26	66	99	99	99	99
24	26	56	99	99	99	99	99
47	66	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99

Što se postiže kvantizacijom

- Dijeljenje viših frekvencija sa većim kvantizacijskim faktorima rezultira time da veliki broj koeficijenata postaje 0
- Nakon zig-zag reorganizacije koeficijenti iz 2D se prebacuju u 1D na način da važni koeficijenti dolaze prvi u nizu a manje važni koeficijenti koji imaju veliku vjerojatnost da su jednaki nuli nakon njih
- To rezultira nizovima od 64 koeficijenta sa velikim brojem nula na kraju

JPEG koderski model



Što nakon HVS modela

- Statističkom analizom podataka nakon obrade u koderskom modelu može se vidjeti da je njihova raspodjela izvrsna za daljnju kompresiju algoritmima zasnovanim na statističkim modelima
- Drugi dio JPEG koda radi upravo to

Od čega krećemo

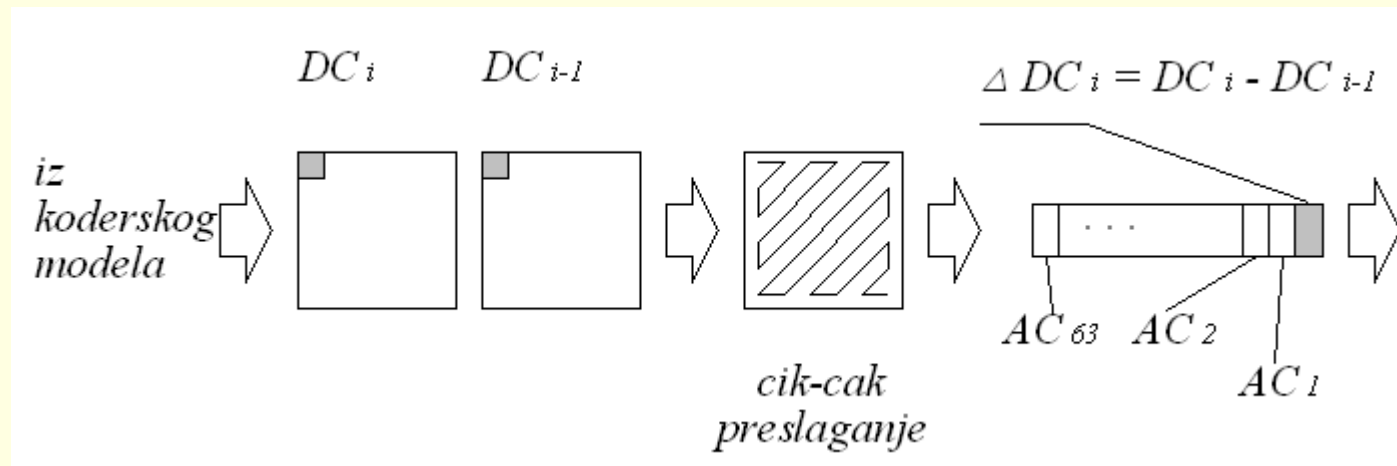
- Entropijski koder prima kvantizirane koeficijete presložene po cik-cak algoritmu u nizove od po 64 koeficijenta za svaku komponentu bloka
- Koje karakteristike možemo uočiti? (koristite DZ i proučite rezultantne vrijednosti za nekoliko slika)

DC koeficijenti

- Teorijska podloga: DC koeficijent predstavlja srednju vrijednost svih 64 elementa bloka
 - dva susjedna bloka obično imaju vrlo slične srednje vrijednosti
- Rezultat:
 - Distribucija DC vrijednosti je prilično jednolika i nepogodna za kodiranje
 - Distribucija RAZLIKE dva susjedna DC koeficijeta je vrlo gusta oko nule i idealna za kodiranje

DC razlika

- Iz prethodnoga odlučujemo da ne želimo kodirati vrijednost DC elementa već razliku između trenutnog i prethodnog
- Znatna ušteda u bitovima (malo po malo...)



Kodiranje duljine niza nula (ZRL)

- Slijedeći korak u našoj analizi proizlazi iz postupka kvantizacije nakon koje je mnogo elemenata viših frekvencija postalo nula.
- Za ovakve nizove izuzetno je pogodna metoda kodiranja duljine niza (ZRL coding)
- Metoda radi na način da umjesto da se kodira svaki koeficijent koji je nula u nizu, da se kodira broj uzastopnih nula koji se nalazi u nizu

DC/AC simboli

- Dodatno se u statističkom modelu određuje pripadnost pojedinog ulaznog elementa određenoj kategoriji prema apsolutnoj vrijednosti njegove amplitude
- Određivanje kategorije povezano je sa postupkom modificiranog Huffmanovog kodiranja o čemu će biti više riječi kasnije. Na izlazu iz statističkog modela svaka ulazna DC razlika i svaki ulazni AC koeficijent različit od nule biti će zamijenjeni sljedećim simbolima:
 - DC simbol: **[kategorija]** *[amplituda]*
 - AC simbol: **[[duljina niza nula],[kategorija]]** *[amplituda]*
- DC simbol nema komponentu koja određuje duljinu niza nula što je i logično jer JPEG zasebno obrađuje svaki slikovni blok a DC razlika uvijek je prvi element bloka.
- Za AC komponente duljina niza nula opisuje koliko koeficijenata ima vrijednost nula prije nekog koeficijenta koji je različit od nule.

ZRL/EOB

- Veličina polja koje opisuje duljinu niza normom je ograničena na 4 bita kojima se mogu opisati nizovi od 0 do 15 nula. U stvarnom nizu ulaznih elemenata može se pojaviti i niz koji sadrži i više od 15 nula te se u tom slučaju koristi specijalni simbol nazvan ZRL (Zero Run Length).
- ZRL simbol označava niz od 16 nula a nakon tog simbola nule se počinju brojati iz početka.
- Drugi, i zadnji, specijalni simbol označava da su od trenutnog elementa do kraja bloka svi elementi jednaki nuli. Ovaj simbol ima oznaku EOB (End Of Block).
- Slikovni blok sastoji se od 64 ulazna elementa te se nakon obrade u statističkom modelu na izlazu može naći najviše tri simbola ZRL. U normi je također definirano kako se niz ZRL simbola nakon kojih slijedi EOB simbol mora izbaciti iz izlaznog niza.
 - Razlog je prirodan. EOB simbol označava sve nule od trenutnog mjesta do kraja bloka pa su prema tome ZRL simboli koji eventualno prethode EOB simbolu nepotrebni.

Kategorija

Kategorija	DC razlika	AC koeficijent
0	0	
1	-1,1	-1,1
2	-3,-2,2,3	-3,-2,2,3
3	-7,...,4,4,...,7	-7,...,4,4,...,7
4	-15,...,-8,8,...,15	-15,...,-8,8,...,15
5	-31,...,-16,16,...,31	-31,...,-16,16,...,31
6	-63,...,-32,32,...,63	-63,...,-32,32,...,63
7	-127,...,-64,64,...,127	-127,...,-64,64,...,127
8	-255,...,-128,128,...,255	-255,...,-128,128,...,255
9	-511,...,-256,256,...,511	-511,...,-256,256,...,511
10	-1023,...,-512,512,...,1023	-1023,...,-512,512,...,1023
11	-2047,...,-1024,1024,...,2047	

Kategorija

- Neke vrijednosti se rijetko pojavljuju pa nije imalo smisla svakom broju pridjeljivati kod
- Kategorije su određene prema vjerojatnosti pojavljivanja ulaznih elemenata
- Kako se kategorijom ne može točno odrediti vrijednost elementa poslije kategorije mora se poslati još podatak [amplituda] koji unutar kategorije definira koji je to element.
- Amplituda ima različit broj bitova za svaku kategoriju (koji je jednak rednom broju kategorije)

Tablica simbola za AC komponente

Duljina niza	Kategorija				
	0	1	2	...	10
0	EOB	0/1	0/2	...	0/10
1	x	1/1	1/2	...	1/10
:	x	:	:		:
14	x	14/1	14/2	...	14/10
15	ZRL	15/1	15/2	...	15/10

Amplituda

- Za kodiranje amplitude koristi se sljedeće pravilo:
- Pretpostavimo da je koeficijent C zapisan u formatu dvojnog komplementa, a K je kategorija kojoj taj koeficijent pripada.
 - Ako je C pozitivan broj tada će se Huffmanovom kodu, kao proširenje, dodati K nižih bitova od C .
 - Ako je C negativan tada će se Huffmanovom kodu dodati K nižih bitova od vrijednosti koeficijenta C umanjenog za jedan, tj. $(C-1)$.

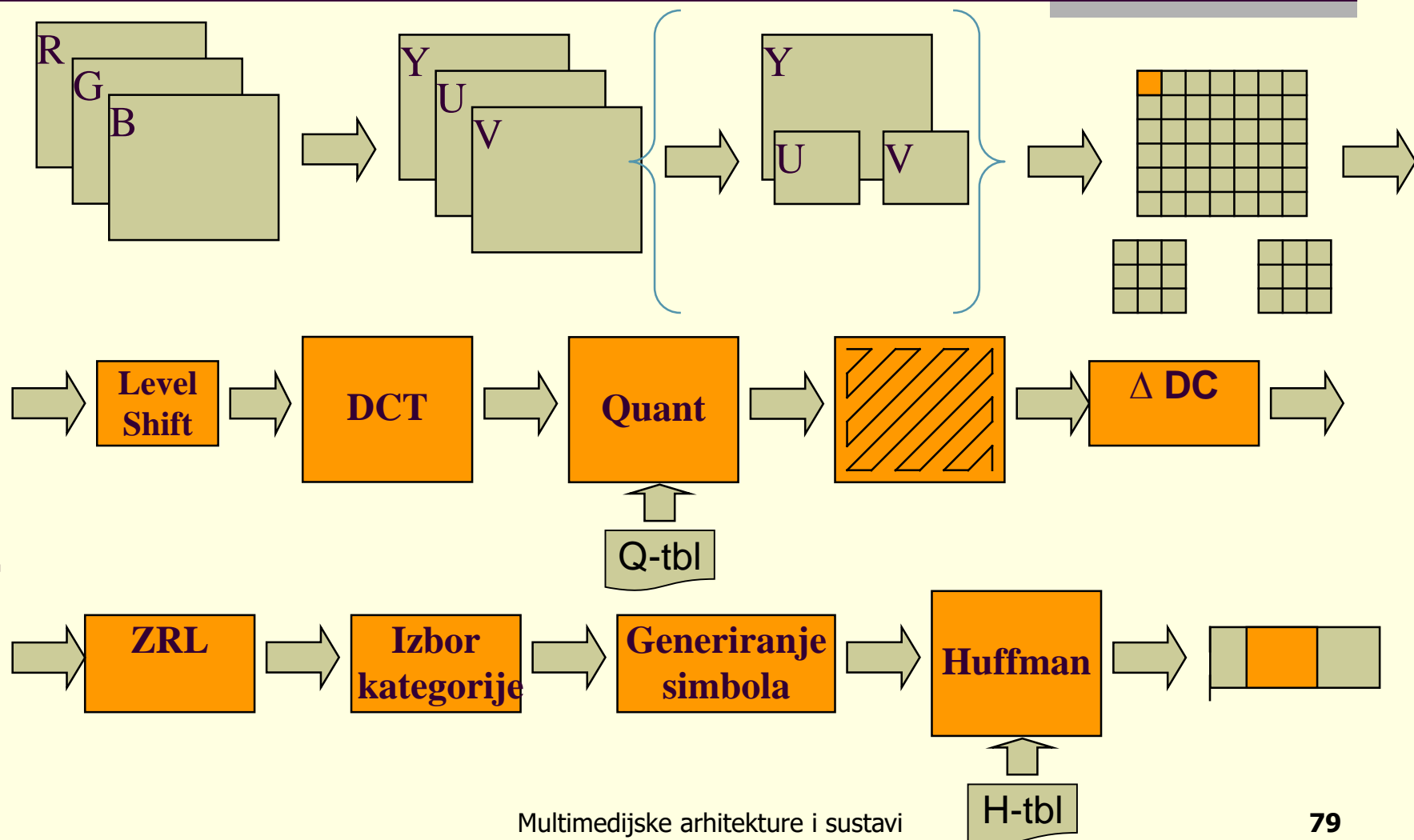
Huffman-ova tablica za DC simbole

Kategorija	Duljina koda	Kodna riječ
0	2	00
1	3	010
2	3	011
3	3	100
4	3	101
5	3	110
6	4	1110
7	5	11110
8	6	111110
9	7	1111110
10	8	11111110
11	9	111111110

Huffman-ova tablica za AC simbole

Duljina niza/Kategorija	Duljina koda	Kodna riječ
0/0 (EOB)	4	1010
0/1	2	00
0/2	2	01
0/3	3	100
0/4	4	1011
0/5	5	11010
0/6	7	1111000
0/7	8	11111000
0/8	10	1111110110
0/9	16	1111111110000010
0/10	16	1111111110000011
1/1	4	1100
1/2	5	11011
...

JPEG koder



Pomak (level shift)

$$\begin{bmatrix} 107 & 113 & 122 & 122 & 122 & 139 & 137 & 139 \\ 107 & 113 & 122 & 122 & 122 & 139 & 133 & 139 \\ 107 & 122 & 122 & 122 & 126 & 133 & 133 & 148 \\ 107 & 113 & 122 & 115 & 122 & 131 & 131 & 148 \\ 107 & 113 & 122 & 127 & 130 & 139 & 139 & 148 \\ 107 & 113 & 113 & 115 & 122 & 131 & 133 & 147 \\ 109 & 107 & 122 & 122 & 131 & 131 & 131 & 133 \\ 103 & 109 & 118 & 115 & 122 & 122 & 122 & 140 \end{bmatrix} \xrightarrow{\text{Pomak}} \begin{bmatrix} -21 & -15 & -6 & -6 & -6 & 11 & 9 & 11 \\ -21 & -15 & -6 & -6 & -6 & 11 & 5 & 11 \\ -21 & -6 & -6 & -6 & -2 & 5 & 5 & 20 \\ -21 & -15 & -6 & -13 & -6 & 3 & 3 & 20 \\ -21 & -15 & -6 & -1 & 2 & 11 & 11 & 20 \\ -21 & -15 & -15 & -13 & -6 & 3 & 5 & 19 \\ -19 & -21 & -6 & -6 & 3 & 3 & 3 & 5 \\ -25 & -19 & -10 & -13 & -6 & -6 & -6 & 12 \end{bmatrix} \xrightarrow{DCT}$$

2D DCT, kvantizacija

$$\begin{array}{ccc} \begin{array}{c} DCT \\ \rightarrow \end{array} & \begin{bmatrix} -31 & -84 & 2 & -16 & -5 & -10 & 13 & -3 \\ 12 & -2 & 1 & 1 & -9 & 2 & 1 & -7 \\ -11 & 7 & -6 & 3 & -1 & 4 & 1 & 1 \\ 5 & -6 & -2 & 7 & -2 & 6 & -1 & -2 \\ -1 & -2 & 1 & -4 & 0 & 0 & 2 & 1 \\ -1 & 3 & -0 & 1 & -0 & 1 & 0 & 4 \\ -2 & -2 & 9 & -4 & 1 & -6 & -4 & -3 \\ 12 & 0 & -8 & 1 & 1 & 2 & -3 & 0 \end{bmatrix} & \begin{array}{c} Kvantizacija. \\ \rightarrow \end{array} & \begin{bmatrix} -2 & -8 & 0 & -1 & 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} & \begin{array}{c} Cik-cak \\ \rightarrow \end{array} \end{array}$$

Cik-cak + entropijsko kodiranje

-2 -8 1 -1 0 0 -1 0 1 0 0 0 0 0 0 -1 00

[011:01][1011:0111][00:1][00:0][11100:0][1100:1][1111011:0][1010]

011011011011100100011100011001111101101010

Markeri

- Sve počinje definiranjem MARKERa
- Markeri su specijalni dvo-bajtni podaci koji počinju sa podatkom 0xFF kojeg slijedi podatak različit od 0 ili 0xFF a kojim se identificiraju različiti strukturalni dijelovi kompresiranog niza podataka

Minimalna struktura JPEG datoteke

SOI	FFD8	Start of Image
DQT	FFDB	Quantization table(s)
DHT	FFC4	Huffman table(s)
SOF	FFC0	Frame header
SOS	FFDA	Scan header
Kompresirani podaci		
EOI	FFD9	End of Image

Primjer: (test1_crno_gray.jpg)

