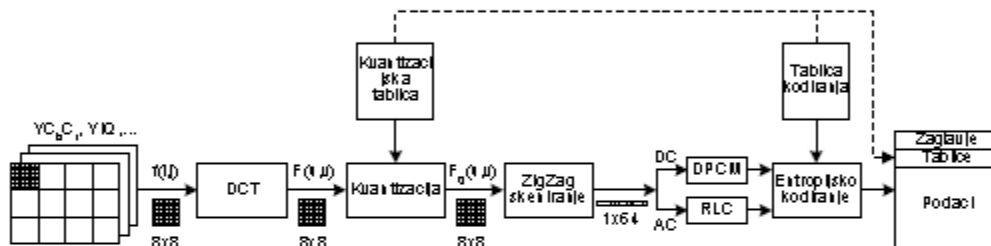


11.2.1. Sekvencijalni (osnovni) JPEG mod

Sekvencijalni mod je (kao i progresivni i hijerarhijski) mod s gubitkom dijela informacije. Temelji se na sljedećim tehnikama:

- Diskretna kosinusna transformacija (DCT),
- Kvantizacija,
- Zig Zag skeniranje,
- DPCM na DC komponenti,
- RLC na AC komponentama,
- Entropijsko kodiranje.

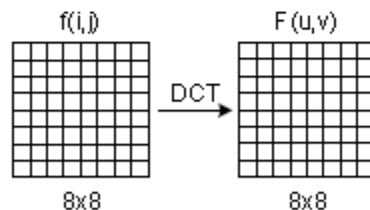
Na slici 11.3 dana je shema JPEG kodera. JPEG dekoder ima iste elementa samo u obrnutom redoslijedu. Napomenimo da većina JPEG datoteka ima kroma komponente koje su poduzorkovane i to najčešće 4:2:0.



Slika 11.3 Shema JPEG kodera

DCT – diskretna kosinusna transformacija

DCT je transformacija koja iz prostornog područja prebacuje signal u frekvencijsko područje što je za slikovni signal 8x8 piksela prikazano na slici 11.4.



Slika 11.4 DCT transformacija slikovnog signala od 8x8 piksela

DCT se za dvodimenzionalni signal 8x8, gdje su vrijednosti piksela $f(i,j)$ u intervalu $(-127,128)$, računa prema sljedećem izrazu:

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

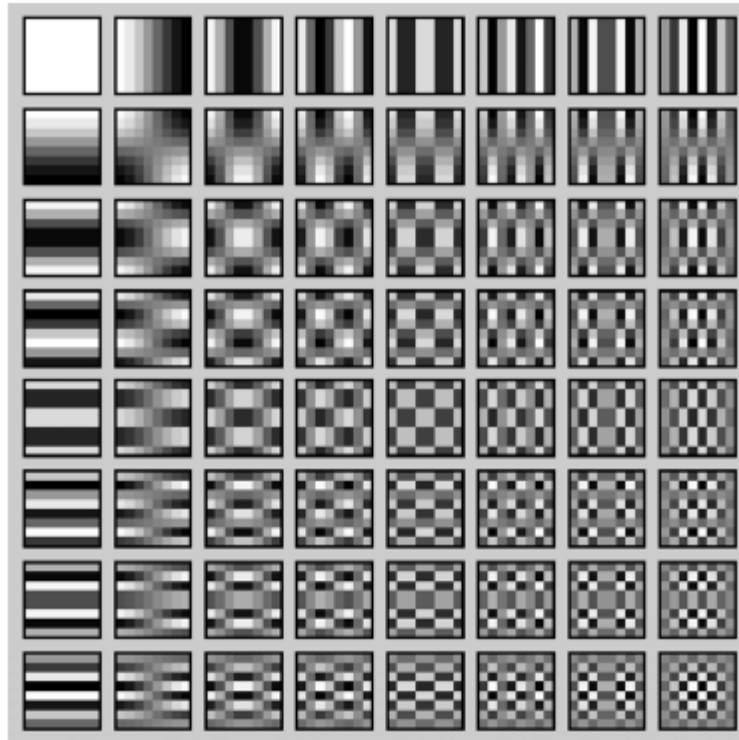
$$\Delta(\mu) = \begin{cases} 1/\sqrt{2} & \text{za } \mu = 0 \\ 1 & \text{za ostale} \end{cases}$$

Inverzna DCT (IDCT) računa se kao

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

$$\Delta(\mu) = \begin{cases} 1/\sqrt{2} & \text{za } \mu = 0 \\ 1 & \text{za ostale} \end{cases}$$

DCT osnovne matrice za dimenziju 8x8 prikazane su na slici 11.5. Osnovne matrice pokazuju kakvo je značenje pojedinih DCT koeficijenata. Tako npr. ako je koeficijent (0,1) bitno veći od drugih koeficijenata, to znači da se vrijednost piksela slike mijenjaju od bijelog prema crnom ako se pomičemo od lijevo prema desno.



Slika 11.5 DCT osnovne matrice

Dvodimenzionalna DCT (2D DCT) može se lakše i brže izračunati kao 2 puta jednodimenzionalna DCT (1D DCT):

$$F(u, v) = \frac{1}{2} \cdot \sum_{i=0}^7 \Delta(u) \cdot \cos \frac{(2i+1) \cdot u \cdot \pi}{16} \cdot G(i, v)$$

$$G(i, v) = \frac{1}{2} \cdot \sum_{j=0}^7 \Delta(v) \cdot \cos \frac{(2j+1) \cdot v \cdot \pi}{16} \cdot f(i, j)$$

$$\Delta(\mu) = \begin{cases} 1/\sqrt{2} & \text{za } \mu=0 \\ 1 & \text{za ostale} \end{cases}$$

Kvantiziranje

DCT je reverzibilna transformacija pa se, ako koeficijente izračunamo s dovoljnom preciznošću, nakon primjene IDCT dobije slika koja je potpuno jednaka početnoj (originalnoj). Teoretski gledano, DCT je transformacija bez gubitka dijela informacije (*engl. lossless*). Kvantizacija (odnosno greška kvantizacije) je razlog zbog kojeg JPEG kompresija jeste kompresija s gubicima (*engl. lossy*). Kvantizacijom se naime smanjuje broj bita potreban za prijenos koeficijenta.

JPEG standard za kvantiziranje koristi tablicu/tablice s 64 vrijednosti. Svaka vrijednost odgovara jednom od 64 DCT koeficijenta. To omogućava da se svaki koeficijent kvantizira posebno, čime se može uzeti u obzir relativna važnost svakog koeficijenta. Naime, ljudski vizualni sustav je najosjetljivije na niske frekvencije (gornji lijevi kut u matrici 8x8 DCT koeficijenata), a manje osjetljiv na visoke frekvencije (donji desni kut). Stoga i vrijednosti u kvantizacijskoj tablici (tablica 11.1) generalno rastu od gore lijevo prema dolje desno. JPEG standard dozvoljava definiranje bilo kakve kvantizacijske tablice (bitno je samo da su tablice u koderu i dekoderu iste). Ipak JPEG standard preporuča korištenje nekih konkretnih vrijednosti jer u prosjeku rezultiraju subjektivno najkvalitetnijim komprimiranim slikama. U tablici 11.1 dane su vrijednosti koje preporuča JPEG standard [Pennebaker 1993] i koje moraju biti sadržane u svakom JPEG koderu i dekoderu. Ako se koriste nestandardne kvantizacijske tablice potrebno ih je prenijeti zajedno s podacima.

Iako JPEG standard dozvoljava korištenje bilo kojeg modela boja, u pravilu se koriste modeli boja koji imaju komponentu osvjetljenja i dvije komponente krome. To su modeli boja YUV, YCbCr, YPbPr i sl. Tako se može iskoristi činjenica da je čovjek manje osjetljiv na promjene u komponentama krome nego u komponenti osvjetljenja. Stoga se komponente krome u pravilu kvantiziraju grublje (veći korak kvantizacije) nego komponenta osvjetljenja što se vidi u tablici 11.1.

Napomenimo da većina JPEG datoteka ima komponente krome koje su poduzorkovane. To znači da se za svaki piksel prenosi komponenta osvjetljenja, a da se za komponentu krome prenosi najčešće jedna vrijednost za blok od 2x2 piksela i to kao srednja vrijednost ta četiri piksela. Iz toga proizlazi da se obradom dva bloka krome (za dvije komponente) od 8x8 vrijednosti zapravo obradi blok od 16x16 piksela. Za to vrijeme je potrebno obraditi 4 bloka osvjetljenja od 8x8 vrijednosti jer to također odgovara bloku od 16x16 piksela.

16	11	10	16	24	40	51	61	17	18	24	47	99	99	99	99
12	12	14	19	26	58	60	55	18	21	26	66	99	99	99	99
14	13	16	24	40	57	69	56	24	26	56	99	99	99	99	99
14	17	22	29	51	87	80	62	47	66	99	99	99	99	99	99
18	22	37	56	68	109	103	77	99	99	99	99	99	99	99	99
24	35	55	64	81	104	113	92	99	99	99	99	99	99	99	99
49	64	78	87	103	121	120	101	99	99	99	99	99	99	99	99
72	92	95	98	112	100	103	99	99	99	99	99	99	99	99	99
Kvantizacijska tablica osvjetljenja q(u,v)								Kvantizacijska tablica krome q(u,v)							

Tablica 11.1 Kvantizacijske tablice koje preporuča JPEG standard

Kvantizirana vrijednost DCT koeficijenta $F(u,v)$ dobije se koristeći kvantizacijske vrijednosti $q(u,v)$ kao:

$$F(u,v) = \text{round}(F(u,v)/q(u,v))$$

To je dakle neravnomjerna kvantizacija, za razliku od ravnomjerne kod koje bi svaki DCT koeficijent $F(u,v)$ dijelili s istom konstantom.

Vrijednosti u kvantizacijskoj tablici mogu se skalirati čime se definira faktor kvalitete. U tom smislu vrijednosti u tablici 11.1 nisu važne samo u apsolutnom iznosu, nego i u relativnom. Skaliranjem vrijednosti u kvantizacijskoj tablici može se mijenjati (kontrolirati) stupanj kompresije, odnosno kvaliteta komprimirane slike.

Primjer: Ako imamo za koeficijent osvjetljenja $F(0,1) = 39.7$ slijedi da je kvantizirani DCT koeficijent jednak

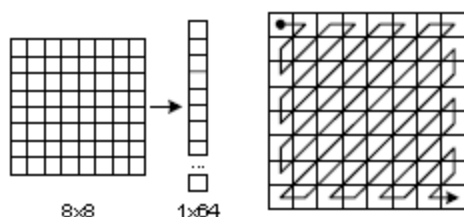
$$F(0,1) = \text{round}(39.7/11) = \text{round}(3.6091) = 4$$

Prilikom rekonstrukcije slike IDCT će koristiti koeficijent koji je jednak

$$F(0,1) \cdot q(0,1) = 4 \cdot 11 = 44$$

ZigZag skeniranje

Kvantizirani DCT koeficijenti imaju vrijednosti koje su vrlo često nula, posebno u donjem desnom dijelu matrice. Stoga se ZigZag skeniranjem mapira 8×8 vektor DCT koeficijenata u 1×64 vektor i to na način da najmanje važni koeficijenti (više frekvencije, uglavnom nule) budu na kraju vektora, a najvažniji koeficijenti (niže frekvencije) na početku vektora. Na tako dobiveni vektor može se uspješno primijeniti RLC kodiranje. ZigZag skeniranje prikazano je na slici 11.6.



Slika 11.6 ZigZag skeniranje

Diferencijalna pulsno kodirana modulacija istosmjernih komponente

Zbog bolje efikasnosti istosmjernih (DC) i izmjeničnih (AC) komponente se kodiraju različito. Iz slike 11.5 vidi se da je osnovna matrica koja odgovara DC koeficijentu konstanta. Stoga je DC koeficijent jednak srednjoj vrijednosti svih 8×8 piksela pomnoženoj s nekom konstantom.

Srednje vrijednosti susjednih blokova 8×8 piksela u pravilu se ne razlikuju puno, pa se DC komponenta mijenja sporo od bloka do bloka. Osim toga DC komponenta je uglavnom relativno velika. Stoga ima smisla kod DC komponente kodirati samo razliku u odnosu na DC komponentu

prethodnog 8x8 bloka pa se koristi diferencijalna pulsno kodirana modulacija (DPCM). To znači da se kvantizira razlika susjednih vrijednosti prema

$$\text{Razlika} = DC_i - DC_{i-1}$$

Run Length Code (RLC) izmjeničnih komponenti

Vektor koeficijenata 1x64 ima u puno nula, i to posebno za visoke frekvencije, dakle na kraju vektora. Stoga ima smisla za izmjenične (AC) komponente koristiti RLC kodiranje, pa su AC komponente kodirane u obliku

(PRESKOČI/IZNOS KOMPONENTE).

gdje se preskoči odnosi na broj nula koji prethode komponenti različitoj od nule.

Entropijsko kodiranje

JPEG osnovni mod koristi Huffmanovo kodiranje kao metodu entropijskog kodiranja koeficijenata. Vrijednost DCT koeficijenata može biti relativno velika pa je Huffmanovo kodiranje za tako veliki alfabet nepraktično. JPEG standard taj problem rješava podjelom mogućih vrijednosti (odnosno razlika za DC koeficijent) na kategorije. Podjela mogućih vrijednosti na kategorije dana je u tablici 11.2.

Kategorija	Moguće vrijednosti		Mogući kodovi
0	0		-
1	-1	1	0, 1
2	-3 -2	2 3	00, 01, 10, 11
3	-7 ... -4	4 ... 7	000, ..., 011, 100, ..., 111
4	-15 ... -8	8 ... 15	0000, ..., 0001, 1000, ..., 1111
5	-31 ... -16	16 ... 31	...
6	-63 ... -32	32 ... 63	...
7	-127 ... -64	64 ... 127	...
8	-255 ... -128	128 ... 255	...
9	-511 ... -256	256 ... 511	...
10	-1023 ... -512	512 ... 1023	...
11	-2047 ... -1024	1024 ... 2047	...
12	-4095 ... -2048	2048 ... 4095	...
13	-8191 ... -4096	4096 ... 8191	...
14	-16383 ... -8192	8192 ... 16383	...
15	-32767 ... -16384	16384 ... 32767	...
16	32768		-

Tablica 11.2 Podjela mogućih vrijednosti na kategorije

KATEGORIJA iz tablice 11.2 se kodira Huffmanovim kodom koji je dan u tablici 11.3. Primjerice, Kada se kodira DC koeficijent (razlika) potrebno je kodirati iznos komponente. To znači da treba poslati Huffmanov kod odgovarajuće kategorije iza kojeg mora doći broj koji kaže o kojoj se

vrijednosti unutar kategorije radi. Dakle kodirana razlika DC koeficijenta ima oblik (KATEGORIJA/VRIJEDNOST).

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

Tablica 11.3 Huffmanova kodne tablice za kodiranje KATEGORIJE kod DC koeficijenta

Kod AC koeficijenata stvar je malo drugačija jer prenosimo informaciju oblika (PRESKOČI/ZNOS KOMPONENTE) gdje se preskoči odnosi na broj nula koje prethode danoj komponenti. Iznos komponente se kodira na isti način kao i razlika kod DC komponente, pa kodirane AC komponente imaju oblik (PRESKOČI/KATEGORIJA/VRIJEDNOST).

PRESKOČI/KATEGORIJA kodira se Huffmanovim kodom. U tablici 11.4 dan je dio Huffmanove kodne tablice za kodiranje PRESKOČI/KATEGORIJA. Huffmanova tablica može biti bilo kakva, ali ako nije standardna onda se mora prenijeti u zaglavlju JPEG datoteke.

klasa	kodna riječ	klasa	kodna riječ	klasa	kodna riječ
0/0 (EOB)	1010			...	16/0 (ZRL) 11111111001
0/1	00	1/1	1100	...	16/1 111111111110101
0/2	01	1/2	11011	...	16/2 111111111110110
0/3	100	1/3	1111001	...	16/3 111111111110111
0/4	1011	1/4	111110110	...	16/4 111111111111000
0/5	11010	1/5	11111110110	...	16/5 111111111111001
:	:	:	:	:	:

Tablica 11.4 Dio Huffmanove kodne tablice za kodiranje PRESKOČI/KATEGORIJA

EOB (end-of-block) kod se šalje odmah iza koda zadnjeg koeficijenta različitog od nule u ZigZag matrici. ZRL kod se upotrebljava kada je broj uzastopnih nula u ZigZag scanu veći od 15 tako da ZRL predstavlja 16 uzastopnih nula. Primjerice 20 uzastopnih nula iz kojih je koeficijent iznosa -5 će se kodirati kao

(ZRL) (PRESKOČI/KATEGORIJA 4/3) (VRIJEDNOST -5)

Primjer:

Neka su nakon DCT transformacije i kvantizacije dobiveni sljedeći DCT koeficijenti:

48	12	0	0	0	0	0	0	0
-10	8	0	0	0	0	0	0	0
2	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	0	0	0	0	0	0	0

Ako pretpostavimo da je vrijednost DC koeficijenta u prethodnom bloku piksela bila 40, slijedi da je razlika 8 (48-40). Ako to uzmemo u obzir, te napravimo ZigZag skeniranje dobivamo matricu s vrijednostima

8, 12, -10, 2, 8, 0, 0, ..., 0.

Prva vrijednost (8) odnosi se na razliku DC koeficijenta. Kodiramo je tako da u tablici 11.2 pogledamo kojoj kategoriji taj broj pripada (a to je kategorija 4) te potom u tablici 11.3 pogledamo koji je Huffmanov kod za tu kategoriju (a to je 101). Nakon toga iz tablice 11.2 pogledamo koji je kod broja 8 (a to je 1000, koji kaže da je to peta vrijednost u danoj kategoriji). Slijedi da je DC koeficijent (razlika iznosa 8) kodiran s 101 1000.

Prvi AC koeficijent (broj 12) pripada kategoriji 4 i nema niti jednu nulu ispred sebe pa ga kodiramo s Huffmanovim kodom za klasu (0/4) što je prema tablici 11.4 jednako 1011. Nakon toga, prema tablici 11.2, slijede bitovi 1100 koji kažu da se radi o baš o broju 12 (trinaesti element) unutar dane kategorije. Stoga se prvi AC koeficijent kodira kao 1011 1100.

Drugi AC koeficijent (broj -10) također kodiramo Huffmanovim kodom za (0/4) ali ovaj put s drugim kodom za vrijednost. Ukupni kod za drugi AC koeficijent je 1011 0101. Slijedeći istu logiku kodiramo 2 kao 01 10 (klasa 0/2 + vrijednost 2), te 8 kao 1011 1000 (klasa 0/4 + vrijednost 8).

Kako je koeficijent s vrijednosti 8 zadnji koeficijent različit od nule, nakon njega šaljem Huffmanov kod za EOB, tj. 1010, čime je kodirano svih preostalih 59 nula. Slijedi da je ukupan kod za svih 64 vrijednosti:

1011000 10111100 10110101 0110 10111000 1010.

11.2.2. Progresivni mod

Progresivni mod prikazuje nisku kvalitetnu sliku i onda je sukcesivno poboljšava. Dva su načina da se to postigne:

1. Spektralna selekcija – Pošalji najprije DC komponentu i nekoliko prvih AC komponenti. Potom šalji ostale AC komponente.
2. Sukcesivna aproksimacija – Pošalji za sve koeficijente samo najznačajnije bite (MSB) a onda postupno šalji i ostale bite sve do najmanje značajnog bita (LSB) – u praksi se šalju kvantizirani koeficijenti i onda razlika tako kvantiziranih koeficijenata i koeficijenata s finijim korakom kvantizacije.

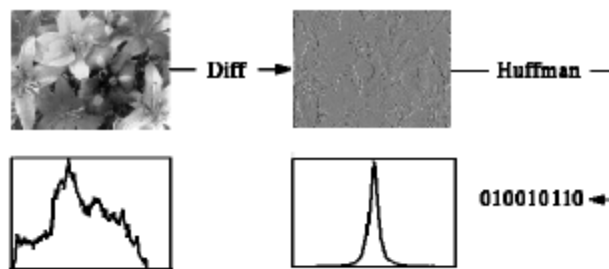
11.2.3. Hijerarhijski mod

Hijerarhijski mod je dobar za gledanje slika na zaslonima s malom rezolucijom. Sastoji se od sljedećih koraka:

1. Poduzorkuj sliku s faktorom 2 i to svaku dimenziju (npr. od 640x480 dobij 320x240).
2. Kodiraj manju sliku koristeći jedan od tri ostala moda (progresivni, sekvencijalni, bez gubitaka).
3. Dekodiraj i povećaj rezoluciju (naduzorkovanje) kodirane slike.
4. Kodiraj razliku između naduzorkovane slike i originalne koristeći progresivni, sekvencijalni ili mod bez gubitaka.

11.2.4. Mod bez gubitaka

Mod bez gubitaka (engl. lossless) je poseban slučaj JPEG gdje nema gubitka u kvaliteti. Mod bez gubitaka ne koristi DCT nego prediktivno kodiranje. Kombiniraju se vrijednosti do 3 susjedna piksela kako bi se predvidjela vrijednost trenutnog piksela. Kodira se (prenosi) samo razlika u odnosu na stvarnu vrijednost, a samo se prvi piksel prenosi kao stvarna vrijednost. Razlika ima puno manju dinamiku od originala što znači da za kodiranje razlike treba manje bita. Na slici 11.7 ilustrirana je dinamika originalne slike i slike razlike pri čemu je slika razlike dobivena najjednostavnijim prediktorom po kojem je trenutni piksel jednak prethodnom ($X=A$).



Slika 11.7 Dinamika originalne slike i razlike ($X=A$)

JPEG standardom za mod bez gubitaka definirano je sedam različitih mogućih prediktora (tablica 11.5). Relativni odnos piksela A, B, C i X prikazan je na slici 11.8. Prvi redak i stupac slike mogu koristiti samo 1D predviđanje.

		C	B	
		A	X	

Slika 11.8 Relativni odnos piksela A, B, C i X

Prediktor	Metoda
1	A
2	B
3	C
4	$A + B - C$
5	$A + (B - C) / 2$
6	$B + (A - C) / 2$
7	$(A + B) / 2$

Tablica 11.5 Mogući prediktori moda bez gubitaka

Relativno loša kvaliteta moda bez gubitaka jedna je od najslabijih dijelova JPEG standarda. Stoga je razvijen JPEG-LS standard (ITU-T Rec. T.87) koji postiže nešto bolju kompresiju u modu bez gubitaka.

12.1. VIDEO KODER/DEKODER

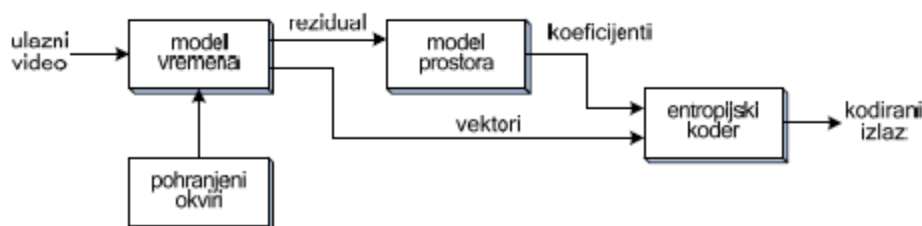
Kompresija signala podrazumijeva uklanjanje redundancije. Kako smo već vidjeli, uklanjanje samo statističke redundancije (kompresija bez gubitaka) nije dovoljno dobro za video signale. Stoga se za kompresiju video signala koristi uklanjanje subjektivne redundancije (kompresija s gubicima).



Slika 12.1 Vremenska i prostorna redundancija video signala

Većina video koda koristi uklanjanje vremenske (*engl. temporal*) i prostorne (*engl. spatial*) redundancije (slika 12.1). Uklanjanje vremenske redundancije temelji se na činjenici da su susjedni okviri visoko korelirani, posebno ako je broj okvira u sekundi velik. U prostornoj domeni, redundancija je sadržana u visokoj korelaciji susjednih piksela jer susjedni pikseli obično imaju slične vrijednosti. Nakon uklanjanja vremenske i prostorne redundancije, ostatak se entropijski kodira. Stoga se video koder sastoji od tri temeljna funkcionalna dijela (slika 12.2):

- model vremena,
- model prostora,
- entropijski koder.



Slika 12.2 Blok dijagram video koda

Ulaz u model vremena (*engl. temporal model*) je nekomprimirani video. Model vremena smanjuje vremensku redundanciju koristeći sličnost između susjednih video okvira i to najčešće tako da predvidi (procjeni) trenutni okvir. Predviđanje se može napraviti na temelju jednog ili više prošlih i/ili budućih okvira. Najjednostavniji način predviđanja (procjene) je pretpostaviti da je trenutni okvir jednak prethodnom. Predviđanje se može poboljšati koristeći složeniji algoritam, odnosno koristeći neku kombinaciju okvira (primjerice da piksele predvidimo kao srednju vrijednost trenutnog i budućeg okvira). Osim toga predviđanje se može poboljšati i kompenzirajući razliku između okvira (kompenzacija pokreta). Stoga su izlazi iz vremenskog modela rezidual i set parametara. Parametri su najčešće upravo vektori pokreta. Rezidual je okvir čije su vrijednosti jednake razlici predviđenog i stvarnog okvira.

Rezidual je ulaz u model prostora koji ima za cilj eliminirati prostornu redundanciju sadržanu u činjenici da su susjedni pikseli slični. To se najčešće radi tako da se rezidual transformira te se potom tako dobiveni koeficijenti kvantiziraju. Dakle, model vremena sažima slike na istom načelu (ali obično malo pojednostavljeno) kao i bilo koji drugi koder slika (npr. JPEG). Izlaz iz modela vremena su kvantizirani koeficijenti. Kako se ne bi propagirala greška procjene, povremeno se okvir (ili dio okvira) prenese bez korištenja kompenzacije pokreta, dakle kao statična slika.

Koeficijenti i vektori pokreta entropijski se kodiraju kako bi se eliminirala eventualna preostala statistička redundancija pa se tu radi o kodiranju bez gubitaka.

Video dekodek radi suprotne operacije. Najprije se niz bitova entropijski dekodira te se na temelju koeficijenata rekonstruira rezidual. Potom se iz vektora pokreta te jednog ili više prethodno dekodiranih okvira, rekonstruira procjena trenutnog okvira. Sami trenutni okvir dobije se dodavanjem reziduala na tako procijenjeni okvir.

12.2. MODEL VREMENA

12.2.1. Korištenje sličnosti susjednih okvira

Kao što smo već rekli, najjednostavnija metoda predviđanja je korištenje prethodnog okvira kao prediktora za trenutni okvir. Na slici 12.3 prikazana su dva uzastopna okvira iz video sekvence. Okvir 1 se koristi kao prediktor za okvir 2. Na slici 12.4 je prikazan rezidual koji se dobije oduzimajući prediktor (okvir 1) od trenutnog okvira (okvir 2). Razlika je normalizirana tako da sredina sive boje predstavlja razliku nula, dok svjetlije odnosno tamnije nijanse sive boje odgovaraju pozitivnim odnosno negativnim razlikama. Iz slike 12.4 može se uočiti relativno puno tamnijih i svjetlijih područja što znači da je u rezidualu još uvijek preostalo dosta energije, a to znači da je preostalo dosta informacije koju treba sažimati u modelu prostora.

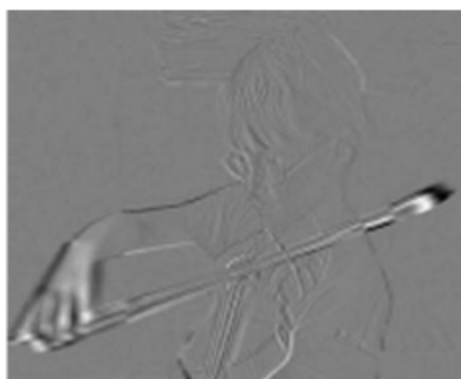
Energija u rezidualu je preostala zbog razlike između susjednih okvira. Većina preostale energije posljedica je pokreta, bilo kamere bilo samog objekta. Očito je da se bolje predviđanje može postići ako kompenziramo pokrete (*engl. motion compensation*). Općenito, razlike između susjednih okvira mogu se javiti uslijed:

- pokreta objekata (krutih objekata kao što je automobil ili objekata s promjenjivim oblikom kao što su prsti na ruci),
- pokreta kamere (pomicanje, zumiranje i sl.),
- otkrivanja objekata (primjerice dio pozadine koji se pomicanjem objekta otkrije),
- promjena osvjetljenja.

S izuzetkom otkrivanja objekta i promjene osvjetljenja, te razlike odgovaraju stvarnom pomicanju piksela. Stoga je moguće procijeniti/izračunati trajektoriju svakog piksela. Trajektorije svih piksela tvore optički tok (*engl. optical flow*). Na slici 12.5 prikazan je optički tok za okvire prikazane na slici 12.3. Radi preglednosti prikazan je optički tok samo za svaki drugi piksel. Kad bi prenijeli cijeli optički tok, rezidual bi bio vrlo mali pa bi i broj bita za njegovo prenošenje bio mali ili nula.



Slika 12.3 Okvir 1 (lijevo) i okvir 2 (desno)



Slika 12.4 Razlika okvira 2 i okvira 1

12.2.2. Procjena pokreta i kompenzacija pokreta temeljena na blokovima

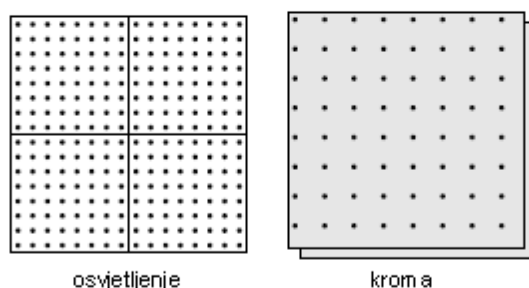
Kompenziranje pokreta na temelju optičkog toka nije praktično jer je računski vrlo zahtjevno. Osim toga, potrebno je za svaki piksel poslati njegov vektor optičkog toka što rezultira velikom količinom podataka koju je potrebno prenijeti. Time se gubi prednost malog reziduala. Općenito je kod kodiranja videa potrebno naći kompromis između broja bita potrebnih za kompenzaciju pokreta i broja bita potrebnih za rezidual kako bi njihova suma bila što je moguće manja.

U praksi se kompenzacija pokreta radi na blokovima piksela. Najčešće je osnovna jedinica za kompenzaciju pokreta blok veličine 16×16 piksela koji se naziva makroblok (*engl. macroblock*). Za video poduzorkovan 4:2:0 s modelom boja YCbCr, makroblok se sastoji od 256 uzoraka sive boje

(organizirane kao 4 bloka od 8×8 piksela), 64 uzorka za komponentu Cb (jedan blok od 8×8 piksela) i 64 uzorka za komponentu Cr (jedan blok od 8×8 piksela). Tako jedan makroblok ima ukupno 6 blokova od 8×8 piksela što je prikazano na slici 12.6.



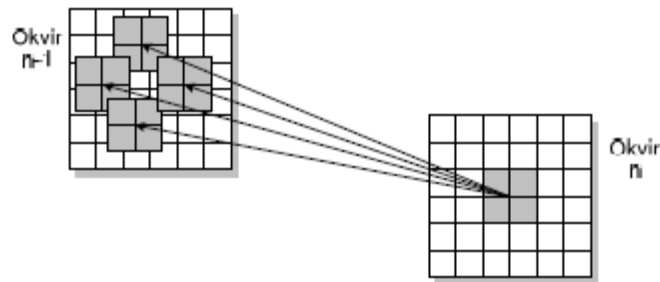
Slika 12.5 Optički tok



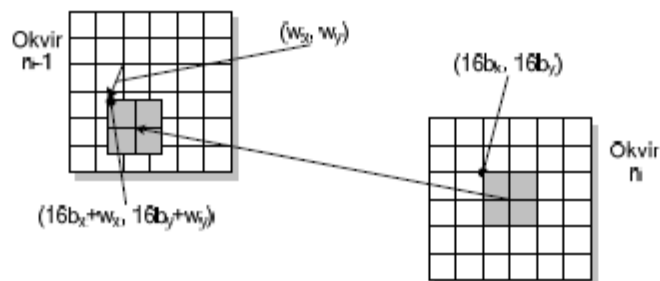
Slika 12.6 Makroblok (za poduzorkovanje 4:2:0)

Da bi mogli kompenzirati pokret najprije je potrebno napraviti procjenu pokreta (*engl. motion estimation*). Procijeniti pokret makrobloka znači pronaći blok od 16×16 piksela u referentnom okviru koji je najbliži trenutnom makrobloku. Referentni okvir je prethodno kodirani okvir koji može biti ispred ili iza trenutnog okvira. Da bi se dobilo na vremenu obično se ne pretražuje cijeli referentni okvir nego samo okolina makrobloka što je ilustrirano na slici 12.7. Postoje razni načini za definiranje sličnosti makroblokova. Najjednostavnija mjera sličnosti je srednja apsolutna distorzija (Mean Absolute Distortion - MAD) (slika 12.8) pri čemu se traži makroblok koji minimizira MAD.

Kada smo pronašli najbliži makroblok, kompenzacija pokreta se vrši tako da se taj najbliži makroblok oduzme od trenutnog makrobloka čime se dobije rezidualni makroblok (osvjetljenje i kroma). Rezidualni makroblok se kodira i zajedno s vektorom pokreta (*engl. motion vector*) šalje dekođeru.



Slika 12.7 Procjena pokreta (za prethodni okvir kao referentni)



$$\frac{1}{256} \sum_{j=0}^{15} \sum_{i=0}^{15} | \text{frame}_n[16b_x + i, 16b_y + j] - \text{frame}_{n-1}[(16b_x + w_x) + i, (16b_y + w_y) + j] |$$

Slika 12.8 Mjera sličnosti makroblokova

Referentni okvir može biti neki od prethodnih okvira, neki od budućih okvira ili kombinacija prošlih i budućih okvira. Ako se kao referentni okvir koristi neki od budućih okvira, on mora biti kodiran prije trenutnog okvira, pa se u tom slučaju kodiranje ne radi po stvarnom vremenskom slijedu okvira.

Kada je između trenutnog i referentnog okvira značajna razlika (primjerice promjena scene) tada je obično efikasnije makroblok kodirati bez kompenzacije pokreta. Tako koder može birati između intra moda (kodiranje bez vektora pokreta) i inter moda (kodiranje s kompenzacijom pokreta) i to za svaki makroblok posebno.

12.2.3. Promjena veličine bloka za procjenu i kompenzaciju pokreta

Realni objekti rijetko odgovaraju dimenzijama 16x16. Stoga može biti efikasnije koristiti promjenjivu veličinu bloka za procjenu pokreta i kompenzaciju pokreta. Na slici 12.9 prikazana su dva uzastopna okvira video sekvence. Rezidual, bez kompenzacije pokreta, dobiven oduzimanjem okvira 1 od okvira

2 prikazan je na slici 12.10. Na slici 12.11 prikazan je rezidual kada se koristi vektor pokreta i to za blokove veličine 16x16, 8x8 i 4x4.

Očito je da se smanjivanjem veličine bloka, smanjuje i energija u rezidualu. Međutim, smanjivanjem veličine bloka povećava se složenost (treba više traženja) ali se i povećava broj bita potreban za prijenos vektora pokreta. Time se poništi prednost dobivena zbog manje energije u rezidualu (a to znači i manje bita za prijenos). Rješenje je da se koristi promjenjiva veličina bloka, u ovisnosti o karakteristikama okvira. Na ravnim, jednoličnim dijelovima, blok može biti velik, dok na dijelovima s puno detalja i složenim pokretima, blok može biti mali. Napomenimo da blok ne mora biti kvadrat već može biti i pravokutnik, primjerice 8x4.

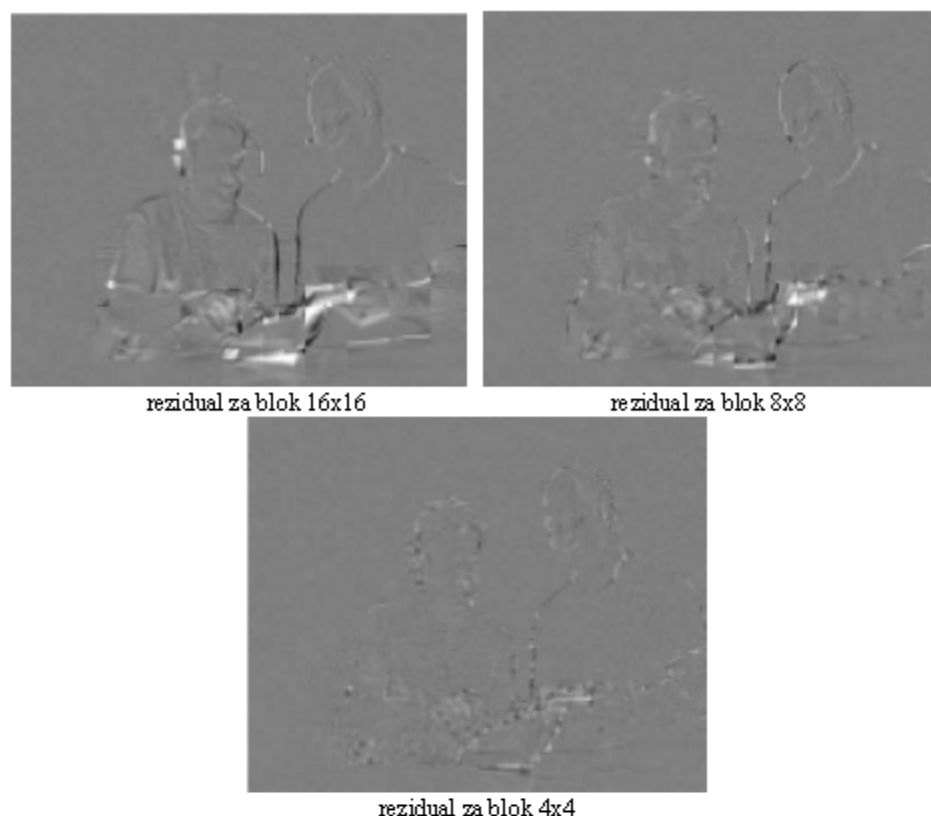
Idealno bi bilo da blok odgovara stvarnom objektu, dakle da ne mora nužno biti ni pravokutnik. To bi bila kompenzacija pokreta temeljena na područjima (*engl. region based motion compensation*), a ne na blokovima. Jedini standard koji predviđa kompenzaciju pokreta i općenito kodiranje temeljeno na područjima (objektima) je MPEG-4.



Slika 12.9 Okvir 1 (lijevo) i okvir 2 (desno)



Slika 12.10 Rezidual (bez kompenzacije pokreta)



Slika 12.11 Rezidual za blok 16x16 (lijevo), za blok 8x8 (desno) i blok 4x4 (dolje)

12.2.4. Kompenzacija pokreta za manje od jednog piksela

U nekim slučajevima predviđanje pokreta je bolje ako koristimo interpolirane uzorke, a ne same uzorke referentnog okvira pa u tom slučaju govorimo o procjeni i kompenzaciji pokreta za manje od jednog piksela (*engl. sub-pixel motion estimation and compensation*). Koristi se procjena i kompenzacija pokreta od pola piksela²⁷ (*engl. half-pixel*) i četvrtine piksela (*engl. quarter-pixel*). Uočite da, iako se koristi izraz piksel, najčešće se procjena vrši samo na komponentama osvjetljenja.

U pravilu je postupak sljedeći. Najprije koder pronađe najbolji vektor pokreta za cijeli piksel (kružići na slici 12.12), dakle vektor pokreta koji minimizira energiju u rezidualu. Potom se provjere sve pozicije pomaka od pola piksela koje su u okolini tog izračunatog vektora pokreta (kružići na slici 12.12) i odabere se onaj koji minimizira energiju u rezidualu. Ako treba, postupak se ponovi i za pomaka od četvrtine piksela, opet u okolini prethodno izračunatog najboljeg pomaka od pola piksela.

Finija interpolacija rezultira manjom energijom u rezidualu. Na slici 12.13 prikazan je rezidual za blok od 4x4 piksela i to za slučaj kompenzacije pokreta od pola piksela i četvrtine piksela. U praksi kompenzacija od pola piksela rezultira značajnim dobikom u odnosu na kompenzaciju cijelog piksela.

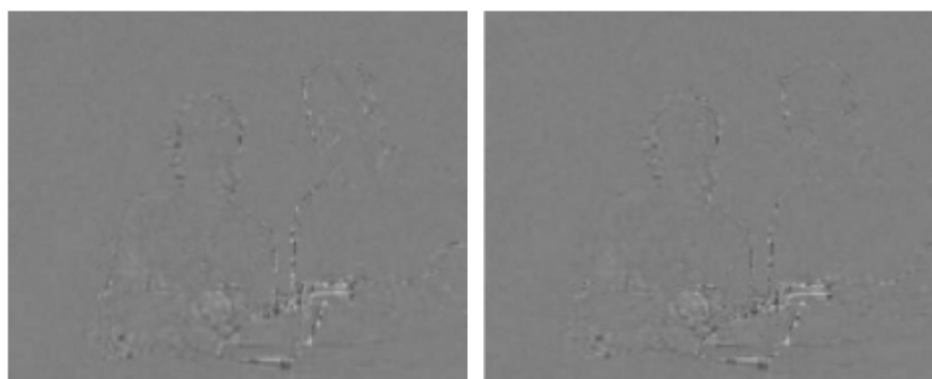
²⁷ Uočite da „pola piksela“ i „četvrtina piksela“ zapravo ne postoje na slici. Radi se o tome da se vrijednosti „pola piksela“ i „četvrtina piksela“, u slučaju potrebe, interpoliraju na temelju susjednih, stvarnih piksela.

Kompenzacija od četvrtine piksela više ne donosi tako značajan dobitak, kod kompenzacije od osmine piksela dobitak je još manji, itd.



Slika 12.12 Kompenzacija pokreta za manje od jednog piksela

Korištenje složenije sheme za procjenu i kompenzaciju pokreta (primjerice blok od 4x4 piksela i procjena pokreta za četvrtinu piksela) rezultira s manje energije u reziduala pa time i manje bita za njegovo prenošenje. Međutim korištenje složenije sheme za procjenu i kompenzaciju pokreta znači i više bita za prenošenje informacije o vektoru pokreta (više bita treba za prenijeti vektor pomaka ako se radi o procjeni pokreta za četvrtinu piksela nego za cijeli piksel). Stoga je zadatak kodera da nađe kompromis kojim se minimizira ukupna suma bita.

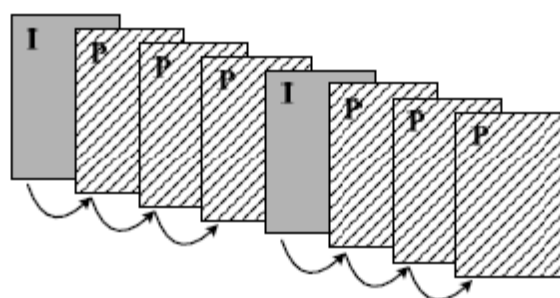


Slika 12.13 Rezidual za blok 4x4 i kompenzaciju od pola piksela (lijevo) i četvrtinu piksela (desno)

Tehnički detalji standarda H.261

H.261 standard je predvidio dva tipa kodiranja: Intra-okvir (*engl. Intra frame, I frame*) i Inter-okvir (*engl. Inter frame, P frame*). Razlikuju se po načinu kodiranja: kod Intra-okvira (I-okvir) makroblokovi se kodiraju koristeći metodu sličnu JPEG standardu, dok se kod Inter-okvira (P-okvir) kodiraju koristeći pseudo razliku u odnosu na prethodni okvir (predviđanje), pa makroblokovi P-okvira ovise jedan o drugom. Primjer sekvence I i P okvira prikazan je na slici 13.3.

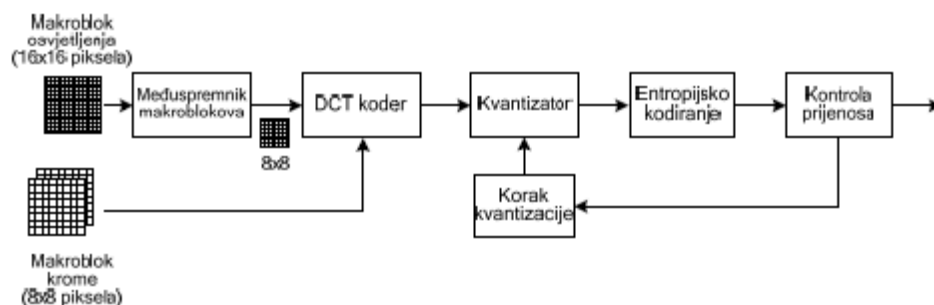
Kako bi postigli bolje kodiranje na vrlo niskim brzinama većina H.261 koda ima vrlo malo makroblokova kodiranih kao I-okvir. Ipak, ograničenje standarda je takvo da se svaki makroblok mora kodirati u intra-okvirnom modu barem jednom svako 132 okvira. Time se osigurava da se kod uključivanja u videokonferencijsku vezu ili kod problema u prijenosu svi makroblokovi ispravno osvrježe u roku od nekoliko sekundi.



Slika 13.3 Primjer sekvence I i P okvira kod H.261 standarda

Intra-okvorno kodiranje temelji se na diskretnoj kosinusnoj transformaciji (DCT) što znači da se I-okvir tretira kao mima slika. Za razliku od JPEG standarda, koji također koristi DCT, za kodiranje I-okvira koristi se linearna kvantizacija; svi koeficijenti se kvantiziraju istom, konstantnom vrijednošću.

Kvantizacija je adaptivna i korak kvantizacije ovisi o tome koliko je memorije preostalo u prijenosnom međuspremniku (*engl. buffer*). Radi se o kontroli prijenosa odnosno kontroli bitske brzine. U slučaju potrebe, kada je prijenosni međuspremnik pun, mogu se preskočiti cijeli okviri (zamrznuta slika). Blok dijagram Intra-okvornog kodiranja prikazan je na slici 13.4.



Slika 13.4 Blok dijagram Intra-okvornog kodiranja kod H.261 standarda

Kod Inter-okvornog kodiranja (P-okviri) kodira se razlika u slikama/blokovima, a ne sama slika/blok. Kodiranje razlike vrši se procjenom pokreta (*engl. motion estimation*) te kompenzacijom tog pokreta (*engl. motion compensation*). To se kod H.261 standarda radi na razini makrobloka (4 bloka od po 8x8 piksela) i to tako da se:

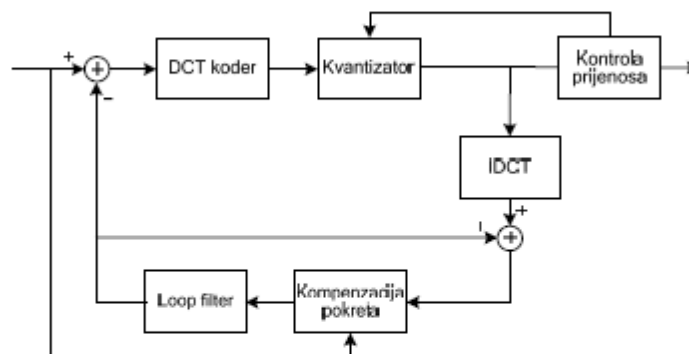
Usporedi trenutni makroblok sa susjednim makroblokovima u prethodnom okviru kako bi pronašao makroblok u prethodnom okviru koji je najbliži trenutnom (da bi se uštedjelo na vremenu uspoređuju se samo makroblokovski osvjetljenja).

Ako je razlika ta dva makrobloka mala, ne komprimiraj makroblok, nego samo prenesi poziciju odgovarajućeg makrobloka u prethodnom okviru.

Ako je razlika velika, pošalji razliku između trenutnog makrobloka i prethodnog najbližeg makrobloka i to tako da se razlika DCT kvantizira na isti način kao i I-okviri.

Dekoder procjenu pokreta vrši na rekonstruiranoj slici, a ne na originalnoj, stoga se i u koderu procjena pokreta vrši na rekonstruiranoj slici čime se sprječava propagacija greške. Zato je u koder

uključena inverzna diskretna kosinusna transformacija (IDCT). U koder je uključen i prostorni filter koji gladi (*engl. smooth*) piksele prethodnog okvira čime se minimizira greška procjene pokreta. Naravno i kod Inter-okvirnog kodiranja potrebno je kontrolirati brzinu prijenosa. Shema inter-okvirnog kodiranja dana je na slici 13.5.



Slika 13.5 Inter-okvirno kodiranje kod H.261 standarda

Procjena pokreta i traženje vektora pokreta se vrši na makroblokovima. Vektor pokreta pokazuje na blok 16x16 piksela, u prethodnom okviru, koji se minimalno razlikuje od trenutnog. Standard ne kaže kako pronaći niti kako definirati najbližnji vektor. Kako smo već vidjeli, najjednostavniji način traženja vektora pokreta je traženje relativnog pomaka makrobloka koji minimizira srednju apsolutnu distorziju (Mean Absolute Distortion - MAD) (slika 12.8).

Tehnički detalji standarda MPEG-1

I, P i B okviri

Video dio standarda MPEG-1 temelji se, kao i standardi H.26x, na eliminaciji prostorne redundancije (za što služi DCT) i na eliminaciji vremenske redundancije (za što služi kompenzacija pokreta). MPEG-1 video sintaksa podržava tri tipa kodiranih okvira (*engl. frame*) odnosno slika (*engl. picture*) što je ilustrirano na slici 13.11. To su:

I-okvir (Intraframe)

- kodira se neovisno o drugim okvirima,
- koristi DCT kodiranje (slično JPEG),
- najniža kompresija unutar MPEG,
- pristupna točka za slučajni pristup (*engl. random access*);

P-okvir (Predicted frame – inter frame)

- kodira se u odnosu na prošle I ili P okvire,
- slično kao i H.261 samo što okvir u odnosu na koju se kodira nije uvijek najbliži prethodni,
- kompresija je bitno veća nego kod I-okvira;

B-okvir (Bidirectionally predictive-coded frame)

- komprimira se koristeći prošle i buduće okvire (I ili P),
- najveća kompresija,
- B-okvir je razlika u odnosu na H.261 i H.263 standarde.

Svi makroblokovi (16x16 piksela) unutar I-okvira moraju se kodirati u I-modu, dakle slično kao i kod JPEG sekvencijalnog moda. Makroblokovi unutar P-slike mogu se kodirati u I ili P modu. Makroblokovi u B slici mogu se kodirati u I, P ili B modu.

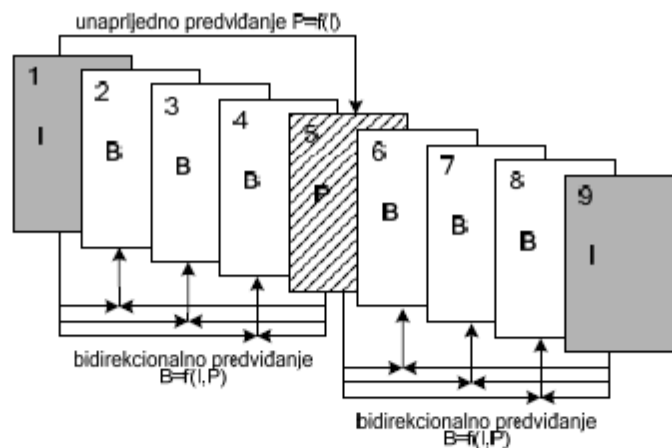
Kod MPEG-1 standarda video sekvenca se najprije podijeli na grupu slika odnosno okvira (GOP – Group of pictures). Veličina grupe slika određena je udaljenošću između dvaju najbližih I-okvira (za slučaj prikazan na slici 13.11 to je 8).

Uočite da, zbog predviđanja (kompenzacije pokreta), redoslijed prijenosa nije isti kao i stvarni redoslijed. Za slučaj prikazan na slici 13.11 redoslijed prijenosa bi bio:

1,5,2,3,4,9,6,7,8

Kod realizacije koda važno je da algoritam odrediti pravi omjer i razmještaj I, B i P okvira. Što je više I-okvira to je bolji slučajni pristup, ali je slabija kompresija. Razmještaj okvira, kako smo već rekli, nije ni na koji način definiran standardom već je prepušten dizajnerima koda. Tipična sekvenca MPEG-1 kodiranog videa koja je dobra za veliki broj aplikacija je:

(IBBPBBPBB)(IBBPBBPBB)...



Slika 13.11 I, P i B okviri kod MPEG-1 standarda

Tipično je kompresijski omjer jednak:

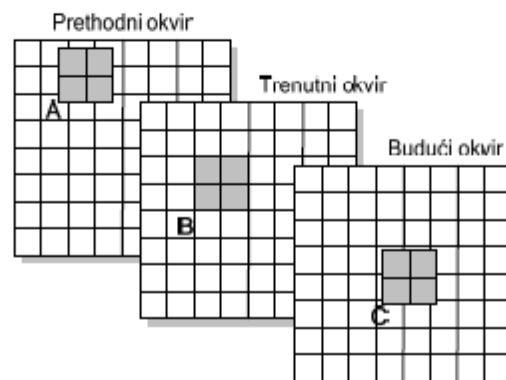
I-okvir	1:10
P- okvir	1:40
B- okvir	1:90

čime, za gore definirani niz, u prosjeku dobijemo kompresijski omjer od 1:70.

Interpolacija

Ako se kod P i B okvira koristi procjena pokreta, P-okvir makroblokove uvijek traži samo na prošlim okvirima, dok B-okvir makroblokove traži i u prethodnim i u budućim makroblokovima. Dakle B može najbliži makroblok tražiti u (slika 13.12):

- prethodnom okviru ($B=A$) – unaprijedno predviđanje,
- budućem okviru ($B=C$) – povratno predviđanje,
- prošlom i budućem okviru ($B=(A+C)/2$) – interpolacija,
- istom okviru.



Slika 13.12 Interpolacija i predviđanje kod kodiranja B-okvira za MPEG-1 standard

Makroblokovi se udružuju u isječke (*engl. slice*), pri čemu koder odlučuje koliko će makroblokova biti u jednom isječku. Unutar jednog isječka vrijednosti vektora pokreta i DC komponente svakog makrobloka kodiraju se koristeći DPCM.

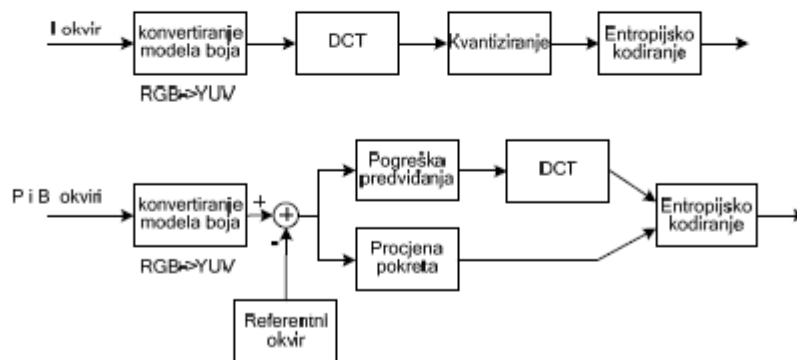
MPEG standard ne specificira način traženja vektora pokreta niti koliki dio slike se pretražuje. Ipak najčešće se koristi block-matching tehnika koja ima razne varijante:

- exhaustive search (brute force) algoritam,
- three-step-search algoritam,
- 2-D logarithmic search algoritam,
- conjugate direction search algoritam.

MPEG-1 video koder

Na slici 13.13 dana je shema MPEG-1 koder. I-okviri kodiraju se koristeći DCT, na sličan način kao i kod JPEG-a. P i B okviri kodiraju se pomoću procjene pokreta. U slučaju potrebe (ukoliko se procijenjena slika značajno razlikuje od originalne), osim vektora pokreta šalje se i razlika slike (pogreška predviđanja - residual). Razlika se kodira na isti način kao i I-okvir, ali uz različite kvantizacijske matrice.

Na slici 13.14 dan je primjer kvantizacijskih tablica iz MPEG Test Model 5 [ISO/IEC, Test Model 5 1993]. Vrijednosti u kvantizacijskoj matrici za ne-intra kodiranje imaju manju dinamiku (međusobnu razliku) u odnosu na kvantizacijske vrijednosti kod intra kodiranja. To je i razumljivo jer slika razlike (ne-intra kodiranje), a onda i odgovarajući DCT koeficijenti, ima manju dinamiku od originalne slike (intra kodiranje). Pri tome koder odlučuje da li će se pogreška uopće kodirati. Kao i kod H.263, koristi se preciznost izračuna vektora pokreta od pola piksela.



Slika 13.13 Pojednostavljena shema MPEG-1 koder

8	16	19	22	26	27	29	34
16	16	22	24	27	29	34	37
19	22	26	27	29	34	34	38
22	22	26	27	29	34	37	40
22	26	27	29	32	35	40	48
26	27	29	32	35	40	48	58
26	27	29	34	38	46	56	69
27	29	35	38	46	56	69	83

Intra kodiranje

16	17	18	19	20	21	22	23
17	18	19	20	21	22	23	24
18	19	20	21	22	23	24	25
19	20	21	22	23	24	26	27
20	21	22	23	25	26	27	28
21	22	23	24	26	27	28	30
22	23	24	26	27	28	30	31
23	24	25	27	28	30	31	33

Ne-intra kodiranje

Slika 13.14 Kvantizacijske matrice za intra i ne-intra kodiranje

MPEG tok bitova

MPEG video tok bitova (*engl. bitstream*) sastoji se od nekoliko slojeva (*engl. layer*) koji sadrže zaglavlje i podatke. Slojevi i njihov odnos prikazani su na slici 13.15.

Najviši sloj u hijerarhijskoj strukturi je sekvenca (*engl. sequence*) koja se sastoji od zaglavlja sekvence i niza grupa slika (*engl. Group of pictures - GOP*). Zaglavlje video sekvence definira video format, dimenzije slike, aspect ratio, frame rate, tip poduzorkovanja, itd.

Grupa slika (GOP) sadrži jednu ili više slika te predstavlja najmanju jedinicu koja se može neovisno kodirati unutar video sekvence. GOP zaglavlje sadrži vrijeme početka emitiranja i može se koristiti kao točka za slučajni pristup (*engl. random access*).

Slika (*engl. picture*) je osnovna jedinica u MPEG postupku kodiranja pri čemu se može kodirati u jednom od tri formata: I, P ili B.

Isječak (*engl. slice*) je dio slike, odnosno skupina od nekoliko makroblokova. Isječak je osnovna jedinica za uspostavljanje sinkronizacije postupaka kodiranja koji se provode na razini bloka i makrobloka. Ako dođe do pogreške u prijenosu, nije potrebno odbaciti cijelu sliku, nego samo odgovarajući isječak.

Makroblok (*engl. macroblok - MB*) je osnovna jedinica za kodiranje s kompenzacijom pokreta. Sastoji se od 16x16 piksela.

Blok (*engl. block*) je osnovna jedinica za primjenu diskretne kosinusne transformacije. Sadrži 8x8=64 elementa slike (piksela). S obzirom na poduzorkovanje od 4:2:0, jedan makroblok ima 4 bloka osvjetljenja i dva bloka krome.

Tehnički detalji standarda MPEG-2

Najznačajnije razlike MPEG-2 u odnosu na MPEG-1 standard su:

1. Skalabilno kodiranje (tako da se isti set signala može koristiti za HDTV i za standardni TV)
 - SNR skalabilnost (slično JPEG progresivnom modu, prilagođava korake kvantizacije DCT koeficijenata),
 - prostorna skalabilnost – slično hijerarhijskom JPEG,
 - vremenska skalabilnost – različite brzine okvira;
2. Modovi predviđanja za polje/okvir (*engl. field/frame prediction modes*) za podršku isprepletenog video ulaza;
3. DCT kodna sintaksa za polje/okvir (*engl. field/frame DCT coding syntax*);
4. Korištenje posebnih kvantizacijskih matrica (*engl. downloadable quantization matrix*) i alternativni redoslijed skeniranja (*engl. alternative scan order*);
5. Nelinearni kvantizacijski faktori za makroblobove.

Nabrojene razlike odnose se na poboljšanje efikasnosti kodiranja ili podršku isprepletenom videu. Osim toga, ima još i nekoliko poboljšanja koja se ne odnose na samu kompresiju, kao što je:

1. Veličine okvira može biti bitno veća (do 16383x16383);
2. Osim 4:2:0, poduzorkovanje može biti i 4:2:2 te 4:4:4,
3. Syntax to facilitate 3:2 pull-down in the decoder,
4. Pan and scan codes with 1/16 pixel resolution.

U nastavku je ukratko opisano nekoliko najznačajnijih promjena MPEG-2 standarda u odnosu na MPEG-1 standard.

Modovi predviđanja za polje/okvir

U MPEG-1 videu, svaka slika se kodira kao okvir, bez obzira da li je originalni video materijal isprepleten ili ne. Ako je originalni video isprepleten, svaki okvir se sastoji od dva polja, gornjeg i donjeg. MPEG-1 koristi predviđanje na temelj okvira, dakle ne vodeći računa o tome da su pikseli gornjeg i donjeg polja pomaknuti u vremenu.

MPEG-2 ima dva moda rada, jedan koji se temelji na okvirima (kao i kod MPEG-1) i drugi koji se temelji na poljima. Koder, na temelju odabranih kriterija, odluči koji će mod koristiti. Kod predviđanja na temelju polja, gornje polje u P-okviru može se predviđati ili na temelju gornjeg ili na temelju donjeg polja prethodnog I ili P okvira. Slično vrijedi i za donje polje P okvira. Kod B-okvira, svako polje se bidirekcionalno predviđa i to ne pomoću dva vektora pokreta (prošli i budući P ili I

okvir), nego na temelju 4 vektora pokreta (gornje i donje polje prošlog i budućeg P ili I okvira). Polja i okviri ilustrirani su na slici 13.16.

DCT kodna sintaksa za polje/okvir

Još jedno značajno poboljšanje koje se odnosi na podršku isprepletenom videu je mogućnost izbora DCT kodne sintakse prilagođene za polje ili za okvir.

Na slici 13.16, u sredini je prikazan makroblok osvjetljenja (16x16 piksela) gdje zatamnjeni pravokutnici predstavljaju piksele gornjeg polja, a svijetli pravokutnici piksele donjeg polja. Lijevo je ilustriran DCT prilagođen kodiranju polja jer svaki blok od 8x8 piksela sadrži piksele samo jednog polja, gornjeg ili donjeg.

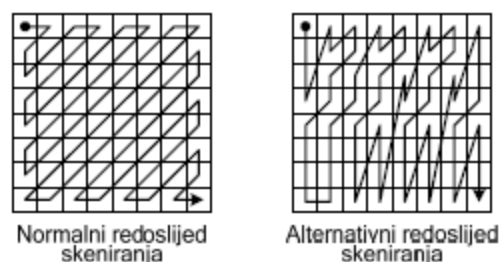
Izbor DCT kodiranja prilagođenog polju obično ima smisla za isprepleteni video kod kojeg scene imaju manje detalja ali veliku promjenu (pokret) između polja. U tom slučaju je efikasnije udružiti polja zajedno, a ne okvire. Time se dobiju blokovi od 8x8 piksela s većom korelacijom između piksela što u konačnici znači da će nakon provedene diskretne kosinusne transformacije trebati manje bita za kodiranje.



Slika 13.16 DCT kodiranje okvira i polja kod isprepletenog videa

Korištenje posebnih kvantizacijskih matrica i alternativni redoslijed skeniranja

MPEG-2 dozvoljava korištenje posebnih kvantizacijskih matrica (koje se onda moraju slati zajedno s ostalim podacima), i to na razini okvira. To može biti korisno u slučaju kada je ulazni video izrazito dinamičan. Ako drugačije nije definirano koriste se standardne kvantizacijske matrice i njih nije potrebno posebno slati jer se vrijednosti podrazumijevaju. U načelu, kvantizacijske matrice su različite za intra kodiranje i ne-intra kodiranje.



Slika 13.17 Normalni i alternativni redoslijed skeniranja

Dok MPEG-1 koristi samo normalno (zigzag) skeniranje, MPEG-2 dozvoljava korištenje i alternativnog skeniranja, i to na razini sloja slike. Normalno (zigzag) i alternativno skeniranje prikazano je na slici 13.17. Alternativno skeniranje može biti bolje od normalnog, posebno za DCT kodiranje okvira kod isprepletenog videa. Naime, spektralna distribucija je takva da više nula može završiti na kraju ako se koristi alternativno skeniranje.

Osnovne komponente MPEG-4 koda

- Procjena pokreta
 - reducira vremensku redundanciju
- Koder teksture
 - koristi se za kodiranje reziduala nakon procjene vektora pokreta
- Koder oblika
 - specificira područje objekta i konture unutar scene
- Sprite coding
- Wavelet kodiranje teksture
- Vremenska, prostorna i hibridna skalabilnost

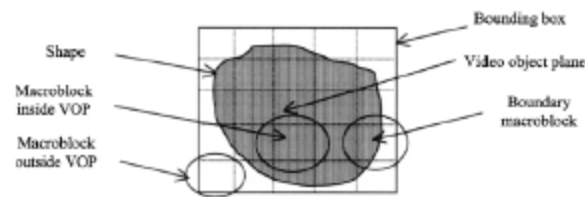
Procjena pokreta

- temelji se na block matching tehnici
- razlika najbližnjeg bloka u odnosu na trenutni naziva se vektor pokreta Motion vector (MV)
- razlika najbližnjeg i trenutnog bloka kodira se pomoću DCT
- Procjena pokreta se temelji na sličnim tehnikama kao i kod MPEG-1 i MPEG-2 (kriterij sličnosti, veličina prozora, točnost jedan ili pola piksela). Neka poboljšanja su
 - adaptivna selekcija veličine bloka (16x16 piksela ili četiri bloka od 8x8 piksela)

Kodiranje teksture

- kodiranje teksture koristi se kod Intra-VOP-ova i kod razlike dobivene nakon procjene pokreta.
- Temelji se na DCT a algoritam je sličan onome kod H.261, MPEG-1 i MPEG-2. Definicija I-VOP, P-VOP i B-VOP je slična definiciji I slike, P slike i B slike kod MPEG-1 MPEG-2.
- Razlike i poboljšanja uglavnom se javljaju zbog kodiranja objekata. Poboljšanja uključuju
 - Intra DC i AC predviđanje (predviđanje nije samo AC nego i DC koeficijenta)

kodiranje texture za VOP bilo kakvog oblika



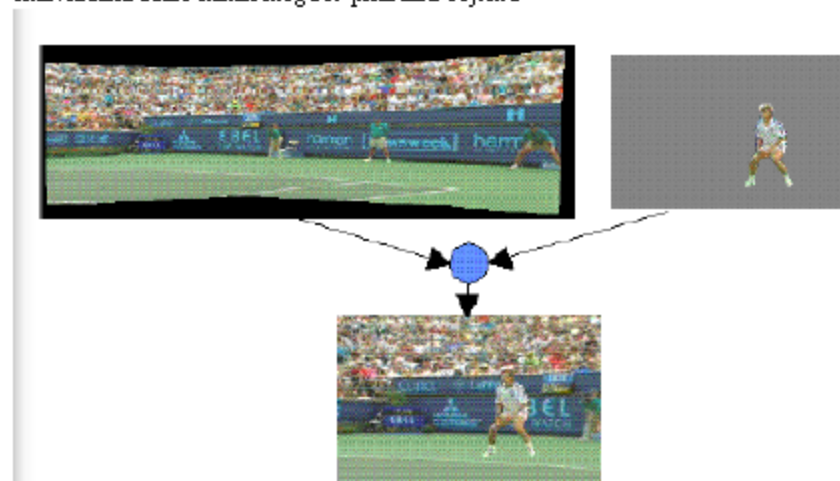
Video objekti i pozadine (sprites)

Mima pozadina (static sprite) – velika mima slika koju treba kodirati samo jednom za više slika

Dekoder kombinira VOPove i pozadine da bi kreirao originalni video

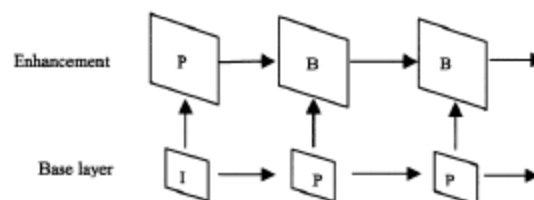
Moguća je skalabilnost na razini sadržaja i na razini objekta

Individualni audio kanali mogu se pridružiti objektu



Vremenska, prostorna i hibridna skalabilnost

Skalabilnost je moguća na razini video objekata (različita rezolucija za različite dijelove scene).



ilustracija prostorne skalabilnosti

Osnovne razlike H.264 u odnosu na MPEG-2

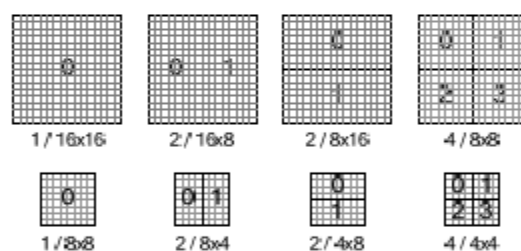
Inter-okvimo predviđanje

Inter-okvimo predviđanje koristi kompenzaciju pokreta slično kao i MPEG-2, ali uz nekoliko značajnih poboljšanja:

- Promjenljiva veličina bloka za kompenzaciju pokreta,
- Višestruka referenca za predviđanje,
- Generalizirano predviđanje B okvira,
- Mogućnost korištenja B okvira kao reference,
- Otežano predviđanje,
- Točnost vektora pokreta od jedne četvrtine piksela.

Kompenzacija pokreta kod H.264 koristi promjenjive veličine blokova, pa se makroblok može kodirati kao jedan blok od 16x16 piksela, dva od 16x8, dva od 8x16 ili 4 od 8x8. Nadalje, i svaki blok od 8x8 piksela može se kodirati kao jedan blok od 8x8 piksela, dva od 4x8, dva od 8x4 ili 4 od 4x4. Moguće podjele prikazane su na slici 13.21. I-okvir može koristiti do 16 referentnih okvira.

Biprediktivni makroblok (B-makroblok) se kod H.264 može predviđati na temelju bilo koja dva referentna makrobloka. Podsjetimo, kod MPEG-2, predviđanje je moralo biti na temelju jednog prošlog i jednog budućeg okvira (makrobloka). Nadalje, za razliku od MPEG-2, B-okvir se može koristiti kao referentni okvir (dakle kao okvir na temelju kojeg se može vršiti predviđanje).



Slika 13.21 Moguće podjele makroblokova i blokova kod inter-okvirnog predviđanja

Intra-okvimo predviđanje

H.264 bolje koristi prostornu redundanciju u odnosu na MPEG-2 i to tako da dozvoljava intra-okvimo predviđanje. Makroblok (MB) se kodira kao intra-MB kada vremensko predviđanje nije moguće (za prvi okvir u videu) ili kada je neefikasno (kod promjene scene).

Predviđanje kod intra-MB se temelji na susjednim pikselima u istom okviru jer je dosta vjerojatno da su slični pikselima MB kojeg kodiramo. MB se može kodirati kao jedan blok od 16x16 piksela, kao 4 bloka od 8x8 piksela ili kao 16 blokova od 4x4 piksela.

Na slici 13.22 prikazani su susjedni pikseli koji se koriste za predviđanje bloka 4x4 piksela. Trenutni pikseli (označeni rednim brojevima) predviđaju se koristeći otežane prosjeke susjednih piksela (označenih slovima). Kodira se razlika između predviđenih i stvarnih vrijednosti.

M	A	B	C	D	E	F	G	H
I	1	2	3	4				
J	5	6	7	8				
K	9	10	11	12				
L	13	14	15	16				

Slika 13.22 Intra-okvorno predviđanje kod H.264

Transformacijsko kodiranje

H.264, kao i MPEG-2 koristi transformacijsko kodiranje (engl. transform coding) kako bi kodirao rezidual (razliku predviđene i stvarne vrijednosti). Međutim, za razliku od tradicionalno korištene DCT, H.264 koristi cjelobrojnu transformaciju (engl. integer transform) i to na blokovima veličine 8x8 i 4x4 piksela. Cjelobrojna transformacija je napravljena tako da koristi samo zbrajanje i posmicanje (Shifting), te tako da ne nema nikakve razlike, u odnosu na original, nakon inverzne transformacije.

Ovakav pristup, u usporedbi s DCT, značajno smanjuje složenost. Veličina blokova za transformacijsko kodiranje neovisna je o veličini blokova za kompenzaciju pokreta odnosno intra-okvorno predviđanje.

Entropijsko kodiranje

Entropijski koder ima zadatak reducirati statističku redundanciju u komprimiranom videu. Entropijski koder u H.264 koristi univerzalni VLC (Variable-length coding) za sve elemente sintakse osim za kvantizirane koeficijente. Koeficijenti se mogu kodirati ili koristeći context-based adaptive VLC (CAVLC) ili koristeći context-based adaptive binary arithmetic coding (CABAC). CABAC daje najbolje rezultate, ali je računski najzahtjevniji.

S druge strane, MPEG-2 video koder koristi fiksnu VLC tablicu za sve elemente sintakse. Te VLC tablice izračunaju se na temelju velikog broja video uzoraka kako bi u općem slučaju dale dobre rezultate. Fiksne VLC tablice su jednostavne za implementaciju, ali ne mogu u potpunosti iskoristiti statističku redundanciju kao što mogu adaptivne.