

## 4 Videojelek JPEG- és MPEG-rendszerű forráskódolása

Jelentős sávszélességigénye miatt a közelmúltig a videojel átvitele digitálisan hagyományos csatornákon keresztül nem volt lehetséges. A probléma megoldására legjobb lehetőség az átviendő adatmennyiség tömörítése.

A 2.3-as bekezdésben láttuk, hogy a videojelek 4:2:2 formátumú digitalizálásakor a teljes adatsebesség 216 Mbit/s. A 4.1. táblázat a videojelek egyszerű adatcsökkentési módszerek alkalmazásával elérhető redukált adatsebességeit tartalmazza. Az un. irrelevanciaredukcióról van szó, amikor az alkalmazás szempontjából nem fontos információkat elhagyjuk. A képfrekvencia 25 Hz, a szóhosszúság (felbontás) 8 bit valamennyi formátum esetében. A videojel kioltási idejének az átvitel alatti átugrásával (kihagyásával) pl. a 216 Mbit/s 166 Mbit/s-ra csökkenthető, ill. a sorszekvenciás átvitel alkalmazásával (4:2:0 formátum) 124,5 Mbit/s-ra.

### 4.1. táblázat

Videojelek adatsebességének csökkentése irrelevanciaredukcióval

Jelek	Mintavételi fr. [MHz]	minta/sor	Sorszám	v [Mbit/s]	V <sub>össz.</sub> [Mbit/s]	Formátum
R	13,5	864	625	108		4:4:4
G	13,5	864	625	108		ITU 601
B	13,5	864	625	108	324	
Y	13,5	864	625	108		4:2:2
C <sub>B</sub>	6,75	432	625	54		ITU 601
C <sub>R</sub>	6,75	432	625	54	216	
Y	13,5	720	576	83		4:2:2
C <sub>B</sub>	6,75	360	576	41,5		csak az
C <sub>R</sub>	6,75	360	576	41,5	166	aktív kép
Y	13,5	720	576	83		4:2:0
C <sub>B</sub> /C <sub>R</sub>	6,75	360	576	41,5	124,5	csak aktív kép
Y	6,75	360	288	20,7		4:2:0, SIF
C <sub>B</sub> /C <sub>R</sub>	3,375	180	288	10,4	31,1	csak aktív kép

Végül pedig a vízszintes és függőleges irányú kétszeres alulmintavételezés az adatsebességet negyedére, azaz 31,1 Mbit/s-ra csökkenti („Source Input Format”, SIF). Ez a jelformátum az egyik félkép elhagyásával létrejövő folytonos (nem váltott soros) képbontásra vonatkozik, függetlenül a későbbi képviisszaadás módjától.

Természetesen műsorszórási célokra még ennél is kisebb adatsebességekre van szükség. A probléma megoldása az adattömörítő kódolási eljárások alkalmazása. Ezért ebben a fejezetben a digitális tv-technikában releváns képkódolási eljárásokkal foglalkozunk, részletesebben a műsorszórásban alkalmazottakkal. Tekintsük át a különböző célokra kifejlesztett eljárásokat:

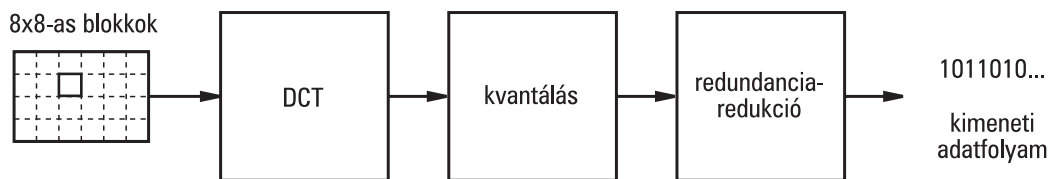
- JPEG (Joint Photographic Experts Group), amely főleg állóképek tömörítésére használható, bár adaptálták mozgóképekhez is (Motion JPEG).
- H.261, amely kis sávszélességigényű videokonferencia- és videotelefon-alkalmazásokra optimális.
- MPEG-1 (Moving Pictures Experts Group), amely közepes sávszélességű átvitelhez, pl. CD-n rögzített és PC-ken visszajátszható mozgóképek adattömörítésére kifejlesztett eljárás.
- MPEG-2, amely az MPEG-1-nél még nagyobb tömörítést tesz lehetővé. Kifejlesztése kifejezetten a nagy sávszélességigényű műholdas, földi, valamint kábeles jelátvitel sajátosságainak figyelembevételével történt. Az MPEG-2 dekóderek alkalmasak általában az MPEG-1-ben kódolt anyagok visszajátszására is.
- MPEG-3, amely a HDTV (High Definition Television) igényeire lett kialakítva, később beépítették az MPEG-2-be.
- MPEG-4, amely a nagyon alacsony bitsebességű videokonferencia, ill. mobil átvitel céljára alkalmas.

A tömörítési eljárások veszteségmentesek vagy veszteségesek. Az MPEG-rendszer az utóbbihoz tartozik. Az információvesztés természetesen látható hibát a képen rendszerint nem okoz. A tömörítési arányok különbözőek, pl. az MPEG-2-vel 33:1 arány is elérhető, de az ISDN-alkalmazásoknál (H.261) ez akár 470:1 is lehet.

## 4.1 JPEG - kódolás

### 4.1.1 Kóder és dekóder tömbvázlata

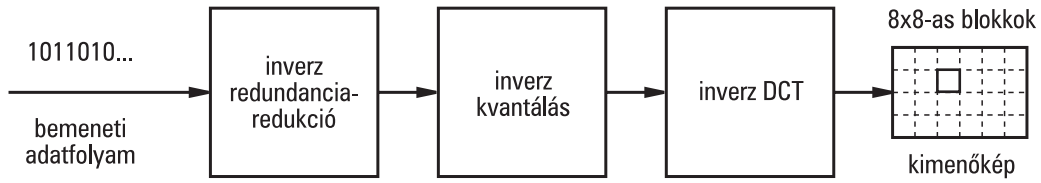
A 4.1. ábrán a következőkben vizsgálandó JPEG-kóder tömbvázlata látható. A kóder legelőször a bemenőképet 8x8 képpontból álló blokkokra bontja, majd kétdimenziós diszkrét koszinusztranszformációval (DCT) a képpontok mintáit a helyfrekvenciák tartományába transzformálja.



4.1. ábra  
JPEG-kóder tömbvázlata

A spektrális felbontás együtthatói a blokk különböző frekvenciájú komponenseinek nagyságát adják meg, amelyek kvantálásánál az emberi látás tulajdonságai jól figyelembe vehetők. A kvantálás az algoritmus tulajdonképpen veszteséges lépése (irrelevanciaredukció), minden további része (eltekintve a DCT kerekítési hibáitól) teljesen veszteségmentes, és megfelelő inverz műveletekkel a kiindulási állapot visszanyerhető. Az adatmennyiség további csökkentése érdekében a kvantálást redundanciaredukció követi, amely lényegében a futamhosszkódolás és a Huffman-kódolás kombinációja (4.1.4 fejezet).

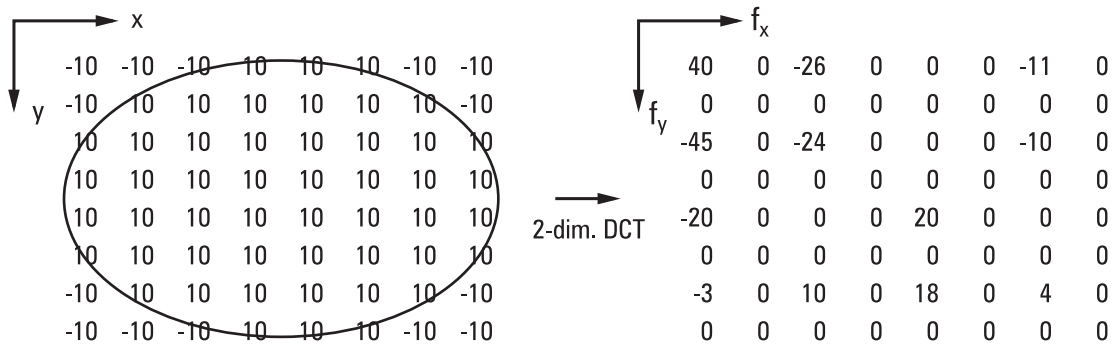
A megfelelő inverz lépések, ahogyan az a 4.2. ábrán látható, a dekóderben hajtódnak végre. A 8x8-as blokkok összerakásából létrejön a kimenőkép, amely megfelelően megválasztott kvantálási tábla esetén (4.1.3 fejezetet) vizuálisan alig különbözik a bemenőképtől.



4.2. ábra  
JPEG-dekóder tömbvázlata

#### 4.1.2 Diszkrét koszinusztranszformáció

A diszkrét koszinusztranszformációt úgy képzelhetjük el, mint 8x8 képpontnak a helytartományból egy ugyanakkora 64 tényezős frekvenciatartománybeli blokkba való leképezését. A 4.3. ábrán erre láthatunk példát.



4.3. ábra  
Példa diszkrét koszinusztranszformációra (DCT)

A bekarikázott képfelület a helyfrekvenciák tartományában nem ismerhető fel. Első pillanatra feltűnik, hogy a transzformált blokkok energiája (nagy értékű együtthatók) az alacsony frekvenciáknál koncentrálódik. Minél nagyobbak az  $f_x$ , ill.  $f_y$  helyfrekvenciák, az együtthatók tendenciózan annál kisebbek lesznek. Ezenkívül még sok a nulla értékű együttható, ami adatcsökkentési szempontból nagyon kedvező. A kétdimenziós DCT együtthatóinak értékét 8x8 képpontra az alábbi összefüggés adja meg:

$$G(f_x, f_y) = \frac{1}{4} C(f_x) C(f_y) \sum_{x=0}^7 \sum_{y=0}^7 g(x, y) [\cos(2x+1)f_x \frac{\pi}{16}] [\cos(2y+1)f_y \frac{\pi}{16}], \quad (4.1)$$

ahol

$$C(f) \text{ frekvenciafüggő konstans } \begin{cases} \frac{1}{\sqrt{2}}, & \text{ha } f = 0 \\ 1, & \text{ha } f > 0 \end{cases}$$

$f_x, f_y$  a helyfrekvenciák,  $\left( \frac{\text{makroblokkél képpontjainak száma}}{\text{egy periódusnak megfelelő képpontszám}} \right)$   
(makroblokk: 16x16 képpontból álló blokk)

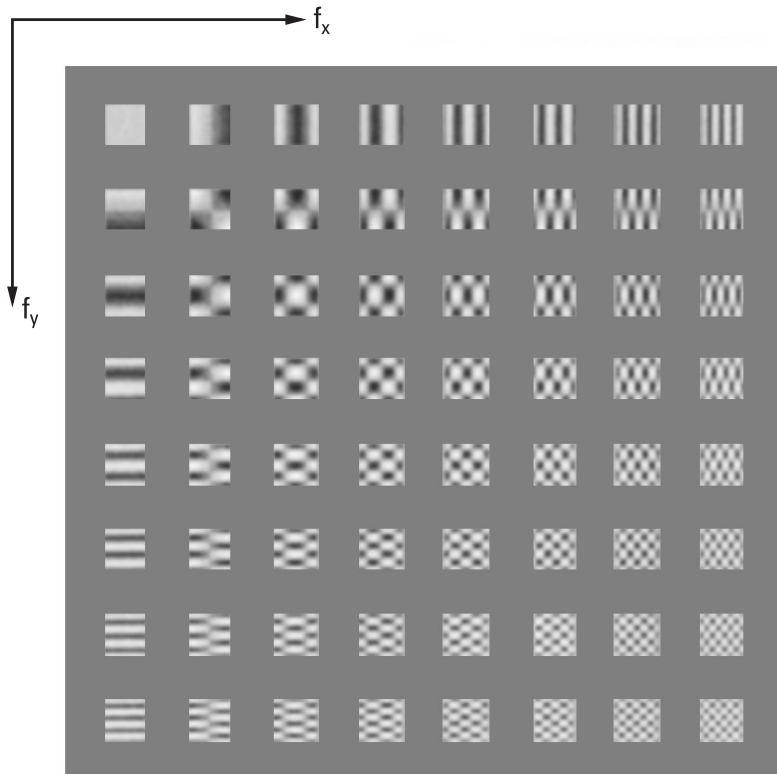
$G(f_x, f_y)$  a DCT-együtthatók,  
 $x, y$  a helykoordináták (képpontok sorszáma),  
 $g(x, y)$  a képpontok mintáinak értéke.

A blokk képmintáinak értékét a helyfrekvenciák együtthatóiból az inverz koszinusztranszformációval kapjuk vissza:

$$g(x, y) = \frac{1}{4} \sum_{f_x=0}^7 \sum_{f_y=0}^7 C(f_x) C(f_y) G(f_x, f_y) [\cos(2x+1)f_x \frac{\pi}{16}] [\cos(2y+1)f_y \frac{\pi}{16}] \quad (4.2)$$

A (4.1) összefüggés alapján megállapíthatjuk, hogy a 8 bites felbontással megadott  $g(x, y)$  bemenőjel egyenkomponense,  $G(0,0)$  11 bites felbontású lesz, hiszen 64 kvantált értéket kell összeadni ( $\cos 0 = 1$ ), majd szorozni  $\frac{1}{4} C(0)C(0) = \frac{1}{8}$ -dal. ( $64:8=8=2^3$ , ami plussz 3 bit.)

Az összes többi együttható is megadható 11 bittel egész számúan, mivel a koszinuszfüggvényértékek együtt az 1, ill.  $1/\sqrt{2}$  frekvenciafüggő állandókkal biztosítják a 11 bites értéktartományt. 64 8 bittel megadott egész szám átalakítása 64 11 bittel megadott egész számmá természetesen nem jelenti az adatmennyiség csökkenését, hanem éppen annak növekedését. De ha figyelembe vesszük, hogy az együtthatók nagy része nagyon kis értékű, akkor pl. egy durvább kvantálás az értéktartományt erősen korlátozza, csökkentve ezáltal az átvíendő adatmennyiséget, ami majd redundanciaredukcióval még tovább csökkenthető.



4.4. ábra  
DCT -bázisfüggvények

A bázisfüggvények alatt olyan blokkokat értünk, amelyek a frekvenciatartományban egyetlen együtthatóval megadhatók. A 4.4. ábrán ezek a blokkok a helyfrekvenciák mentén vannak elhelyezve, amelyek jobbra és lefelé növekvőek. Így a baloldali felső blokk egyetlen egységesen szürke érték, azaz  $G(0,0) \neq 0$ , és az összes többi  $(f_x, f_y)$ -ra érvényes,

A DCT nem az egyetlen eljárás, amely adatcsökkentésre alkalmas. Így pl. az un. Karhunen–Loeve-transzformációval valamivel még nagyobb redukciós tényező érhető el, de a hardverben és szoftverben szükséges ráfordítások aránytalanul nagyobbak. A DCT mint „csaknem ideális” transzformáció ezért kompromisszumnak tekinthető a ráfordítás és a teljesítmény között. Ugyanígy a 8x8-as blokkméret is kompromisszum. Ugyan a nagyobb méretű blokkok esetében valamivel nagyobb lenne az adatcsökkentés hatékonysága, azonban a számolási igény a blokkél hosszával négyzetes arányban nő, ami az elfogadható áru IC-k megjelenését késleltette volna.

A frekvenciatartományba történt transzformálás után a  $G(f_x f_y)$  együtthatók kvantálása következik a

$$G_Q(f_x, f_y) = kerekítés \left( \frac{G(f_x, f_y)}{Q(f_x, f_y)} \right) \quad (4.3)$$

összefüggés szerint, ahol	$f_x, f_y$	a helyfrekvenciák,
	$G_{\mathcal{Q}}(f_x, f_y)$	a kvantált DCT - együtthatók,
	$G(f_x, f_y)$	a kvantálendő DCT - együtthatók,
	$Q(f_x, f_y)$	a kvantálási lépcső értéke.

A (4.3) összefüggés szerint a kvantálás a  $Q(f_x, f_y)$  kvantálási lépcsővel való osztás és azt követő egész számra történő kerekítés révén valósul meg, ami lineáris kvantálást eredményez. Az emberi vizuális érzékelés tulajdonságaihoz illeszkedően az osztó függ a helyfrekvenciáktól. A 4.5. ábrán a bal oldali tábla a világosságjel, a jobb oldali pedig a krominanciajel DCT-együtthatóinak kvantálásához kísérleti úton meghatározott lépcsőértékeket tartalmaz.

	$f_x$							
$\downarrow f_y$	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

	$f_x$							
$\downarrow f_y$	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

$Q(f_x, f_y)$  a luminanciajel DCT-együtthatóira

$Q(f_x, f_y)$  a krominanciajel DCT-együtthatóira

4.5. ábra  
Példa kvantálási táblákra

A kísérlet során az egyes DCT-bázisfüggvények középszürke háttéren lettek megjelenítve, melyek amplitúdóit a kísérletben résztvevő személyek úgy állították be, hogy a bázisfüggvények mintázata már éppen ne legyen a háttértől megkülönböztethető. Az így meghatározott legkisebb eltérés „just noticeable difference” lett a kvantálási lépcső a luminanciajelek DCT-együtthatóira.

Ahogy látható, a helyfrekvenciák növekedésével egyre nagyobb lesz a kvantálási lépcső értéke. Ez azzal magyarázható, hogy az emberi szem a finom részleteket csak kisebb dinamikaátfogással tudja érzékelni. A luminanciajel esetében a DCT-algoritmussal meghatározott  $G(0,0)$  egyenkomponens 11 bitet igénylő szám lett, ami 16-tal elosztva 7 bitesre csökkent, és így  $G_Q(0,0)$  128 diszkrét értéket vehet fel, mialatt  $G_Q(7,7)$   $2^{11}/99$ -et, azaz csak 21-et.

A 4.5. ábra kvantálási táblái a JPEG-szabvány informatív függelékében megtalálhatók, azonban a komprimáláshoz előírva nincsenek. Ezek olyan empirikus táblák, amelyek 720x576 képpont felbontású képre és a képcsőméret négyszeresének megfelelő távolságból való szemlélés esetén optimálisak. A ténylegesen alkalmazott táblákat a komprimált képinformációval együtt kell átvinni, hogy a dekóder az inverz kvantálást el tudja végezni, ami nem más, mint a kvantálási tábla értékeivel való szorzás.

139	144	149	153	155	155	155	155	235,6	-1,0	-12,1	-,52	2,1	-1,7	-2,7	-,13	16	11	10	16	24	40	51	61
144	151	153	156	159	156	156	156	-22,6	-17,5	-6,2	3,2	-2,9	-0,1	0,4	-1,2	12	12	14	19	26	58	60	55
150	155	160	163	158	156	156	156	-10,9	-9,3	-1,6	1,5	0,2	-0,9	-0,6	-0,1	14	13	16	24	40	57	69	56
159	161	162	160	160	159	159	159	-7,1	-,19	0,2	1,5	0,9	-0,1	-0,0	0,3	14	17	22	29	51	87	80	62
159	160	162	162	160	155	155	155	-0,6	-0,8	1,5	1,6	0,1	-0,7	0,6	1,3	18	22	37	56	68	109	103	77
161	161	161	161	160	157	157	157	-1,8	-,02	1,6	-0,3	-0,8	-1,5	1,0	-1,0	24	35	55	64	81	104	113	92
162	162	161	163	162	157	157	157	-1,3	-0,4	-0,3	-,15	-0,5	1,7	1,1	-0,8	49	64	78	87	103	121	120	101
162	162	161	163	163	158	158	158	-2,6	-1,6	-3,8	-1,8	1,9	1,2	-0,6	-0,4	72	92	95	98	112	100	103	99
eredeti kép mintái								DCT- együtthatók								kvantálási tábla							
a)								b)								c)							
15	0	-1	0	0	0	0	0	240	0	-10	0	0	0	0	0	144	146	149	152	154	156	156	156
-2	-1	0	0	0	0	0	0	-24	-12	0	0	0	0	0	0	148	150	152	154	156	156	156	156
-1	-1	0	0	0	0	0	0	-14	-13	0	0	0	0	0	0	155	156	157	158	158	157	156	155
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	160	161	161	162	161	160	157	155
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	163	163	164	163	162	160	158	156
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	163	164	164	164	162	160	159	157
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	160	161	162	162	162	161	159	158
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	158	159	161	161	162	161	159	158
kvantált együtthatók								inverz kvantálás utáni együtthatók								rekonstruált minták							
d)								e)								f)							

4.6. ábra  
DCT és kvantálás együttes hatása a képre

A 4.6. ábra példája a DCT és a kvantálás együttes hatását szemlélteti. Az a) képrészlet az eredeti kép mintáinak értékét adja meg, b) az abból számított DCT-együtthatókat. A c) kvantálási táblával megvalósult kvantálás után kapjuk a d) részabrában látható kvantált együtthatókat, amelyekkel elvégezzük a következő szakaszban ismertetendő redundanciaredukciót. A példában ez a teljesen megfordítható folyamat az áttekinthetőség végett nem szerepel. Inverz kvantálás e) és inverz DCT után megkapjuk az f) rekonstruált képrészletet, amely ugyan az eredetitől különbözik, de éppen oly módon, hogy a szemlélő a különbséget nem érzékeli. Ezeknek a láthatatlan különbségeknek a megengedése nélkül, azaz un. veszteségmentes kódolással a tömörítési tényező csak kettő, vagy annál is kisebb lenne.

#### 4.1.4 Redundanciaredukció

Kódredundancia alatt a legjobb kódolással szembeni „bőbeszédűséget” értjük. Az abszolút kódredundancia kifejezése:

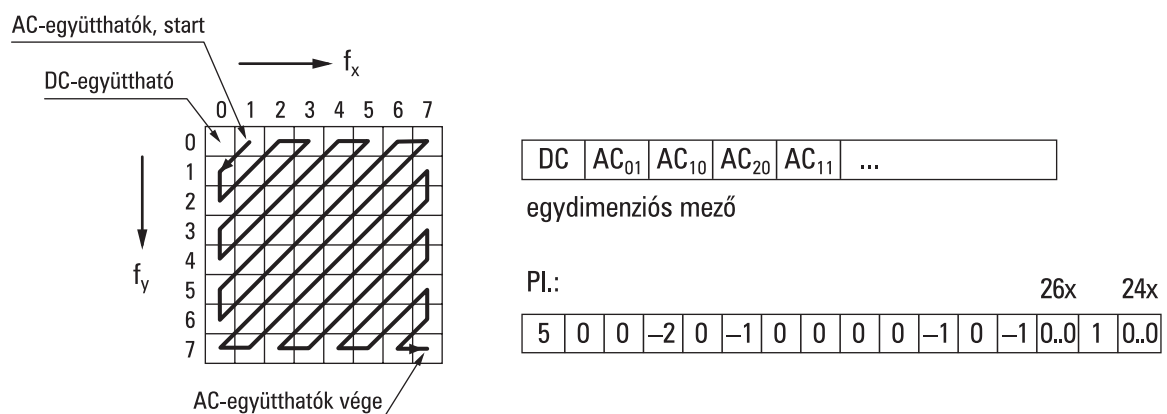
$$R = H_m - H, \quad (4.4)$$

ahol  $H_m$  az átlagos szóhosszúság,  $H$  pedig egy szimbólum átlagos információtartalma. A (2.4) összefüggés szerint egy valószínűségi változó átlagát, legyen az akár feszültség, információtartalom vagy szimbólumok bináris szóhosszúsága, úgy számítjuk ki, hogy a valószínűségi változót valószínűségével súlyozzuk (megszorozzuk), majd a súlyozott valószínűségi változókat összegezzük. Ennek értelmében:

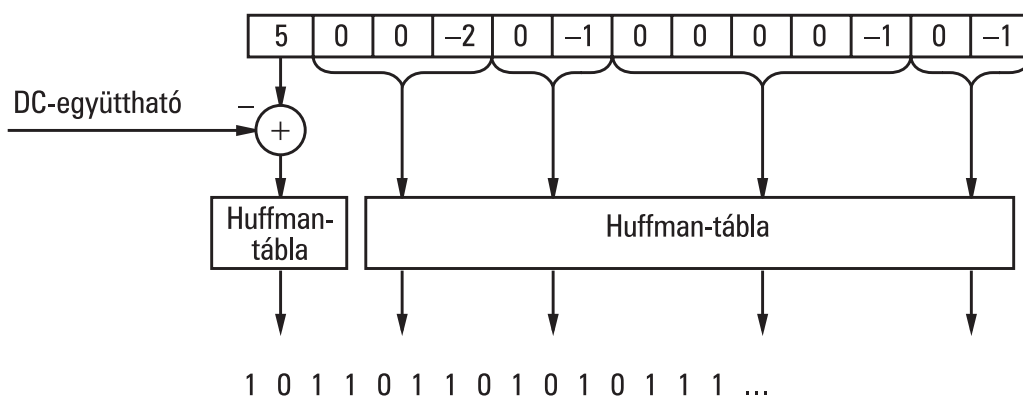
$$R = H_m - H = \sum_{i=1}^N P_i m_i - \sum_{i=1}^N P_i \lg \frac{1}{P_i} = \sum_{i=1}^N P_i \left( m_i - \lg \frac{1}{P_i} \right), \quad (4.5)$$

ahol  $m_i$  az  $i$ -edik szimbólum bináris szóhosszúsága,  $P_i$  a szimbólum „előfordulási” valószínűsége,  $N$  a szimbólumok száma,  $\lg(1/P_i)$  pedig az  $i$ -edik szimbólum információtartalma. A redundanciacsökkentő kódolásnak olyannak kell lennie, hogy  $R$  értéke minél kisebbre adódjon. A (4.5) kifejezés szerint ez akkor valósul meg, ha  $m_i$  minél jobban megközelíti az  $\lg(1/P_i)$  értéket, amiből következik, hogy gyakran előforduló szimbólumok esetében, amikor  $P_i$  nagy, viszont  $\lg(1/P_i)$  kis értékű, a szóhosszúságnak is kis értékűnek kell lennie.

Ez a követelmény a JPEG-rendszerű adattömörítés esetében a következők szerint valósul meg. Az első lépés a kvantált DCT-együtthatók átrendezése, ami a kétdimenziós mátrix elemeinek cikk-cakk sorrendben történő kiolvasását és utána egydimenziós sorozatban történő megjelenítését jelenti, ahogyan az a 4.7. ábrán is látható. A kétdimenziós mátrix így tehát egy egydimenziós együtthatósorozattá alakul, mégpedig növekvő frekvenciák szerinti elrendezésben. Ezáltal a nagy értékű együtthatók tipikusan a sorozat elején jelennek meg, míg a végén a kis értékűek és sok egymásra következő nulla.



4.7. ábra  
Együtthatók átrendezése egydimenziós sorozattá



4.8. ábra  
Redundanciaredukció

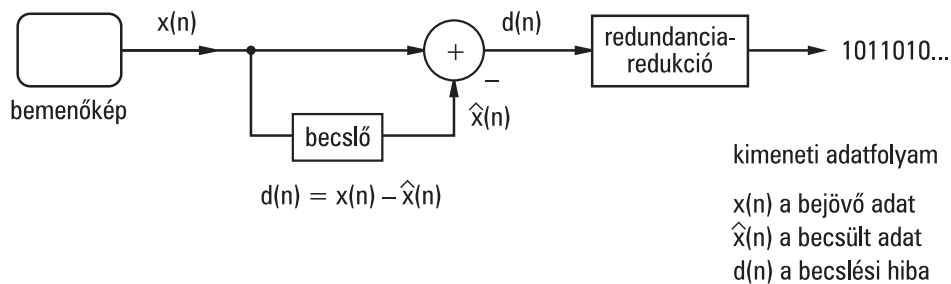
Ahogy a 4.8. ábra is szemlélteti, a szinuszos vagy AC-komponensek együtthatói úgy vannak kódolva, hogy az egymásra következő nullák száma és a következő nullától különböző együttható egy számpárt alkotnak. Így pl. a 0,0,-2 sorozatból keletkező számpár (2,-2), a 0,0,0,0,-1 sorozatból keletkező (4,-1). Az eljárás neve futamhosszkódolás, angolul Run Length Coding (RLC). A következő lépésben egy kódtábla, az ún. Huffman-tábla alapján a gyakran előforduló számpárokhoz rendelt kódszó kevés bitből áll, míg a ritkábban előforduló számpárok esetén a kódszó hosszabb. Az eljárás neve változó szóhosszra történő kódolás, angolul Variable Length Coding (VLC), amit szoktak még Huffman-kódolásnak is hívni. Mindezek alapján megállapíthatjuk, hogy a VLC olyan reverzibilis kódolási eljárás, amely a (4.5) összefüggéssel definiált kódredundanciát minimalizálja. Mivel az egyenkomponens együtthatója általában a legnagyobb értékű (és ennek megfelelően csak ritkán 0), ezért az AC-komponensektől eltérő módon kódoljuk. Ha képezzük az aktuális és az előző blokk DC-együtthatójának különbségét, akkor a különbségi jel általában kis értékű lesz, amit azután a jelstatisztikát figyelembe vevő Huffman-kódolásnak vetünk alá. Az eljárás szintén reverzibilis, és a dekóderben a különbségi értékek maradéktalanul visszanyerhetők.

Az így létrejött bitsorozat a járulékos információkkal, mint pl. a képfelbontás értéke, a kvantálási tábla stb. képezik a JPEG-adatfolyamot, amely azután tárolható vagy továbbítható. A leírt algoritmussal tipikusan 8-as értékű komprimálási tényező érhető el, amelynél a rekonstruált és az eredeti kép szubjektíve közel megkülönböztethetetlen.

#### 4.1.5 Veszteségmentes kódolás

Néhány alkalmazásnál az eredeti és a rekonstruált kép közötti mégoly kis különbségek – mint amilyenek pl. a 4.6. ábrán is fellépnek – sem engedhetők meg. A gyógyászatban pl. hamis diagnózishoz vezethet, ha a rekonstruált röntgenkép csak kis mértékben is eltér az eredetitől. Hogy ilyen esetekben is legyen lehetőség adattömörítés alkalmazására a JPEG-szabvány tartalmaz egy veszteségmentes kódolási eljárást is. Ahogy a 4.9. ábrán látható, itt nincs transzformáció, mert már a DCT-nél elkerülhetetlenül fellépő kis kerekítési hibák is zavaróan hatnának.





4.9. ábra  
Veszteségmentes JPEG-kódolás

Ehelyett az aktuális képpont értékének a környező képpontokból történő becslése valósul meg. A becsült és a tényleges érték közötti különbséget azután egy Huffman-tábla segítségével kódoljuk, hasonlóan, mint azt az alapkonfiguráció algoritmusában a DC-együtthatók átvitelénél láttuk. A Huffman-kódolás alapja itt az, hogy a becsült értékektől való kis eltérések sokkal gyakrabban fordulnak elő, mint nagyobbak. Ezek azután rövid kódszavakkal kódolhatók. A becsült értéket a 4.10. ábrán láthatóan hét különböző módon határozhatjuk meg.

	a becslési algoritmus hivatkozási száma	becslés
	0	nincs
	1	A
	2	B
	3	C
	4	$A + B - C$
	5	$A + \frac{B - C}{2}$
	6	$B + \frac{A - C}{2}$
	7	$\frac{A + B}{2}$

		C	B				
		A	X				

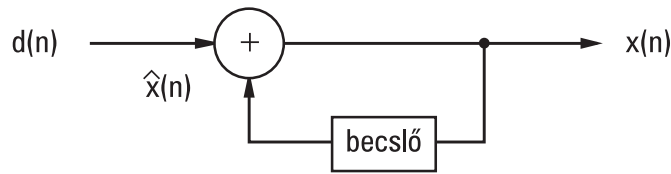
4.10. ábra  
Lehetséges becslési módok

A kóder kiválasztja a legkedvezőbb eredményt adó becslési algoritmust, és egyben jelzi a dekódernek is. A különbségi jelen alapuló kódolás neve differenciális pulzus-kódmoduláció, angolul Differential Pulse Code Modulation (DPCM). A veszteségmentes kódolással elérhető komprimálási arány egy átlagos képnél kettőnél kisebb, míg az alaprendszer szerinti kódolással szubjektíve közel azonos képminőség mellett nyolc. Alkalmazása ezért csak különleges esetekben indokolt, pl. röntgenképek kódolásánál, amelyeknél a nagy szürke felületek miatt a tömörítési arány még kettőnél is nagyobb lehet.

A dekóderben a teljesen reverzibilis inverz redundanciaredukció után visszanyerjük a  $d(n)$  becslési hibát. A  $d(n) = x(n) - \hat{x}(n)$  összefüggésből átrendezéssel kapjuk, hogy

$$x(n) = d(n) + \hat{x}(n). \quad (4.6)$$

Az egyenletnek megfelelő struktúravázlat, azaz a DPCM-dekóder tömbvázlata a 4.11. ábrán látható.

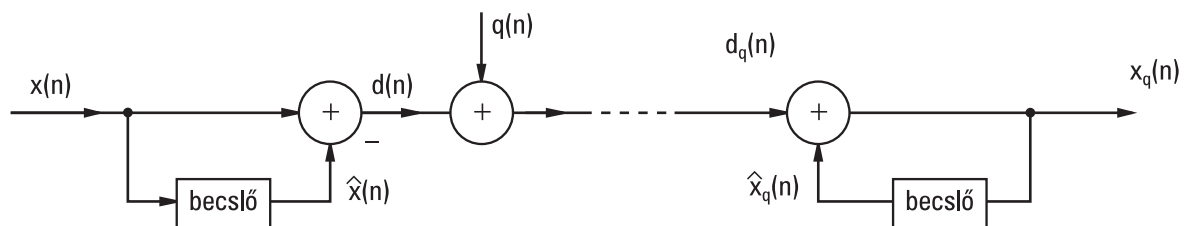


4.11. ábra  
DPCM-dekóder tömbvázlata

## 4.2 MPEG-szabványú kódolás

Természetesen több adattömörítő eljárás egymás utáni alkalmazása is megengedett, így pl. egy DPCM-mel kódolt különbségi képre alkalmazható a diszkrét koszinusz-transzformáción alapuló entrópiakódolás, ahol a kvantált együtthatók kódhosszúsága az előfordulási valószínűségük szerint optimalizált. Ezt hívjuk hibrid kódolásnak. Az MPEG-rendszerű kódolás pontosan ezen az elven alapul.

A DPCM-et követő DCT, ill. az együtthatók kvantálása azt okozza, hogy a  $d(n)$  becslési hiba (különbségi jel) az inverz DCT után  $q(n)$  kvantálási zajjal terhelődik, ami a DPCM-dekóderre jutva a pozitív visszacsatolás miatt a rekonstruált jelben több értékre kiterjedően is érezteti hatását. A kvantálási hiba figyelembe vehető a 4.12. ábra szerinti kódér-dekóder elrendezéssel, amely annyiban különbözik a 4.9. ill. a 4.11. ábrán megismerttől, hogy a kódér kimenetére most egy összegező is csatlakozik, ahol  $d(n)$ -hez hozzáadódik a keletkezett kvantálási zaj.



4.12. ábra  
DPCM kvantálással

A felírható összefüggések alapján vizsgáljuk meg  $q(n)$  hatását a kódolás-dekódolás folyamatára.

Az összefüggések:

$$d(n) = x(n) - \hat{x}(n), \quad (4.7)$$

$$d_q(n) = d(n) + q(n), \quad (4.8)$$

$$x_q(n) = d_q(n) + \hat{x}_q(n), \quad (4.9)$$

ahol

$q(n)$  a kvantálási hiba,  
 $x_q(n)$  a rekonstruált adat,  
 $d_q(n)$  kvantált különbségi jel.

(4.9)-be (4.7)-et és (4.8)-at behelyettesítve kapjuk, hogy

$$x_q(n) = x(n) - \hat{x}(n) + q(n) + \hat{x}_q(n), \quad (4.10)$$

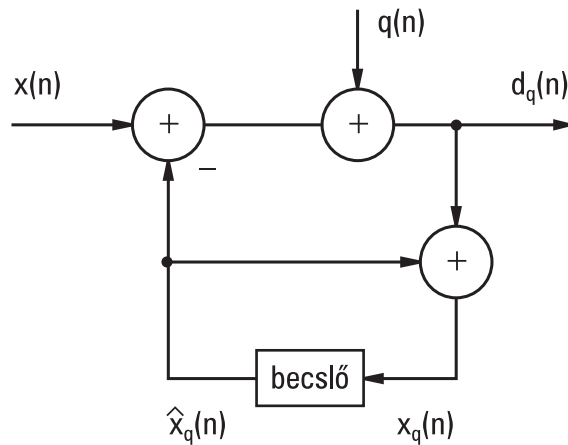
ill.

$$x_q(n) - x(n) = q(n) + \hat{x}_q(n) - \hat{x}(n), \quad (4.11)$$

ahol

$x_q(n) - x(n)$  az un. rekonstruálási hiba, és  
 $\hat{x}_q(n) - \hat{x}(n)$  a becsült értékek különbsége.

(4.11) szerint, ha a becsült értékek különbsége 0, akkor a rekonstruált adat,  $x_q(n)$  csak a kvantálási hibával különbözik az eredeti értéktől. Ehhez biztosítani kell az  $\hat{x}(n) = \hat{x}_q(n)$  feltételt, azaz a kóderben  $\hat{x}(n)$  helyett  $\hat{x}_q(n)$ -et kell képezni egy komplett dekóder-áramkörrel. Az így adódó elrendezés a 4.13. ábrán látható.



4.13. ábra

Kvantálási hibával megegyező rekonstruálási hibát  
adó DPCM-kóder

A bemenőjelre a kóder átviteli függvénye a benne lévő két visszacsatolt rendszerre felírható átviteli függvény alapján:

$$H_k(z) = \frac{1}{1 + \frac{H_b(z)}{1 - H_b(z)}} = \frac{1 - H_b(z)}{1 - H_b(z) + H_b(z)} = 1 - H_b(z), \quad (4.12)$$

ahol  $H_b(z)$  a becslő átviteli függvénye.

A dekóder átviteli függvénye:

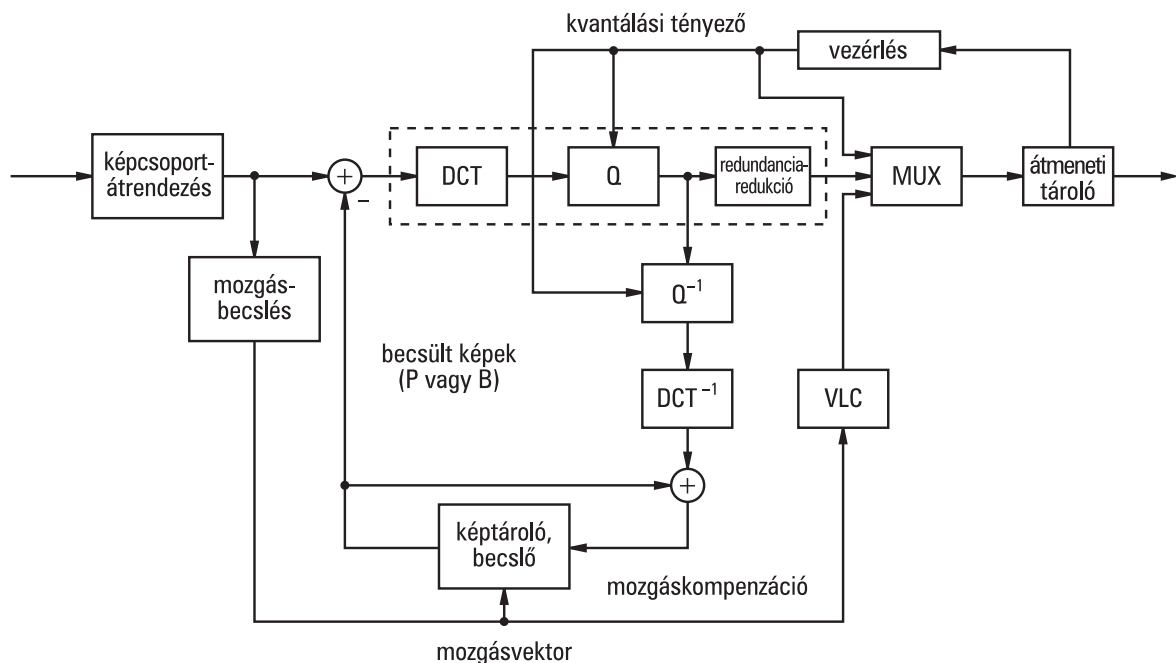
$$H_{dk}(z) = \frac{1}{1 - H_b(z)} \quad (4.13)$$

A két átviteli függvény szorzata, azaz a kódér-dekóder együttes eredő átviteli függvénye egységnyi, tehát a kimeneten, a kvantálási hibától eltekintve, visszakapjuk az eredeti értéket.

A kvantálási hibára felírható két átviteli függvény megegyezik a bemenőjelre felírhatóval, így a kvantálási hiba is változatlanul jelenik meg a kimeneten, azaz a rekonstruált érték, ahogy azt vártuk is, éppen a kvantálási hibával különbözik a bemenőértéktől.

#### 4.2.1 A kódolás-dekódolás folyamata

A 4.14. ábra MPEG-videokóderében ráismerünk a 4.13. ábrán felvázolt megoldásra, ezért a kvantálási hibára tett megállapítások itt is érvényesek. A következőkben a kódolási folyamatot a struktúravázlat alapján részletesebben is áttekintjük. Az ábrán szaggatott vonallal bekeretezett rész tulajdonképpen egy komplett JPEG-kóder, amely a különbségi kép blokkjait a helyfrekvenciák tartományába transzformálja.



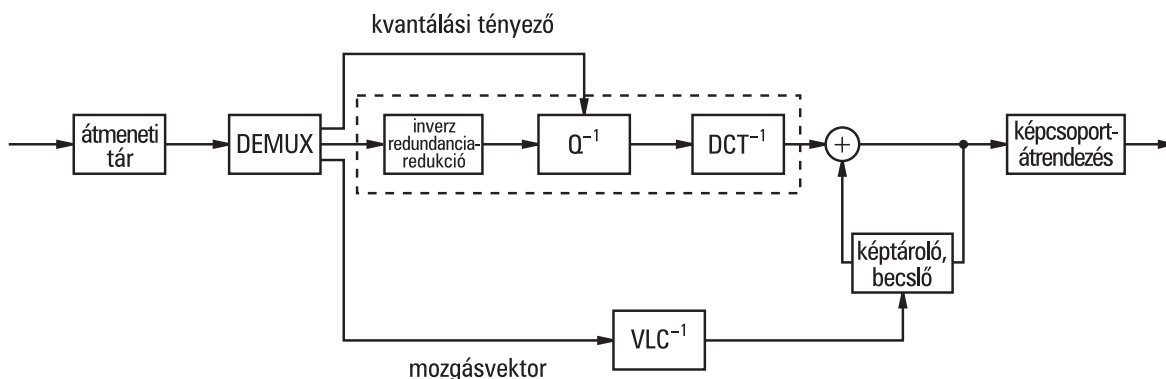
4.14. ábra  
MPEG-videokóder tömbvázlata

Mivel a bemeneti kvantált képminták 8 bites felbontásúak, és így 0-tól 255-ig vesznek fel értékeket, ezért a különbségi jel  $-255$  és  $+255$  között bármekkora lehet, ami 9 bites felbontásnak felel meg. Az MPEG-kódolásnál a DCT változatlanul 9 bites szóhosszal valósul meg, az adatsökkenés az újrakvantálás (Q) és a redundanciaredukció során jön

létre. A becslést a mozgásvektorokkal, ill. az ún. mozgáskompenzációval lehet hatékonyra tenni. A mozgásvektorokat a „mozgásbecslés”-tömb generálja. Mozgásbecslésnek hívjuk a mozgást leíró információk a képsorozatból való kinyerését, azaz az ún. makroblokkok (16x16 képpontméretű blokk) elmozdulását megadó mozgásvektorok kiszámítását. A mozgáskompenzáció pedig a mozgásvektoroknak a becslésnél történő alkalmazását jelenti. Hogy a dekóderben lévő becslő ugyanazt a becsült képet tudja előállítani, mint a kóderben lévő, ezért a mozgásvektorokat is továbbítani kell.

A becslés hatékonyságának növelése érdekében az MPEG-videokódolásnál a későbbi képeket is bevonják a mozgásvektorok meghatározásába, ami a kódolás előtt a képek átrendezését teszi szükségessé („képcsoport-átrendezés”).

A kvantálási lépcső nagyságának változtathatósága lehetővé teszi, hogy az kódolt adatfolyam állandó adatsebességű legyen. A kimeneten lévő átmeneti tároló (puffer) fogadja a változó adatsebességű adatfolyamot, majd állandó sebességgel továbbítja. Ha túlsordulás veszélye áll fenn, akkor az ún. kvantálási tényező révén a kvantálás durvábbá válik, a pufferbe kevesebb adat érkezik, beállítva ezzel az állandó adatsebességet. A dekóder tömbvázlata a 4.15. ábrán látható.



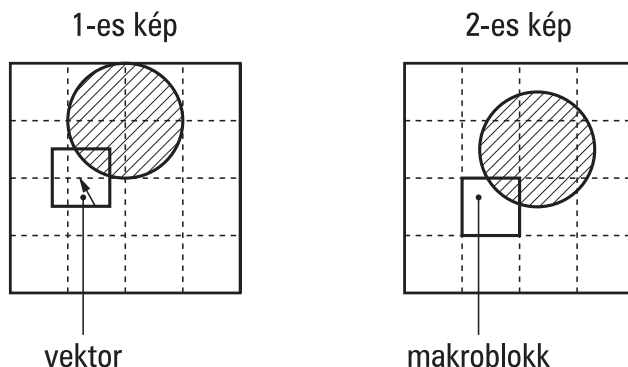
4.15. ábra  
MPEG-videodekóder tömbvázlata

Az átmeneti tárolóba érkező adatok a demultiplexerbe jutnak, ahol a kódolt képadatok a dekódoláshoz szükséges segédinformációktól (kvantálási tényező, mozgásvektor) szétválasztódnak. Az inverz redundanciaredukció után a kvantálási tényező ismeretében végbemegy az inverz kvantálás. Az inverz DCT után a dekódolt különbségi kép és a becsült kép mintáinak összegezésével visszkapjuk a kiindulási képet. A dekódolt képek eredeti sorrendjének visszaállításával fejeződik be a jelfeldolgozás.

A kóder és a dekóder tömbvázlatára pillantva azonnal megállapíthatjuk, hogy a dekóder sokkal egyszerűbb felépítésű. A dekóderben nincs szükség a mozgásvektorok kiszámítására, hiszen azok a kóderből rendelkezésre állnak. Az egyszerűbb dekóder kedvező a vevőkészülékek árának alakulása és megbízhatóságuk szempontjából is. Megjegyzendő, hogy az MPEG-videokóder a mozgásvektorokra is alkalmaz redundanciaredukciót biztosító változó szóhosszra történő kódolást („VLC”-tömb).

### 4.2.2 Mozgásbecslés

A mozgásbecsléssel megvalósítható mozgáskompenzációt (predikciót) szemlélteti a 4.16. ábra.



4.16. ábra  
Példa a mozgásbecslésre

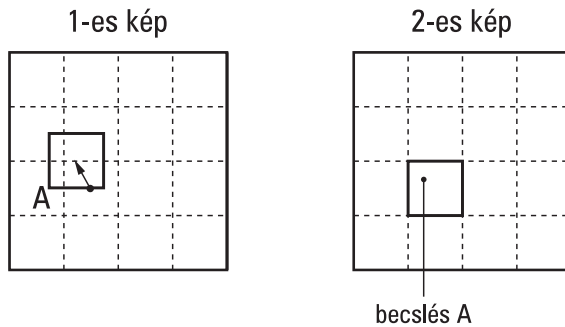
Az ábrán egy labda ferdén lefele irányuló mozgását követhetjük nyomon az 1-es és a 2-es képen. A mozgásbecsléshez a képet 16x16 képpontból álló makroblokkokra bontjuk. Ha a 2-es kép vastagon kihúzott makroblokkját az előző kép azonos helyzetű makroblokkjával hasonlítjuk össze, akkor megállapíthatjuk, hogy a labda mozgása miatt a két makroblokk között a felső sarokban jelentős az eltérés. Következésképpen a különbségi képnek ez a képrészlete jelentős adatmennyiséget fog tartalmazni. Ellenben ha az előző kép helyett egy a labda mozgását figyelembe vevő becsült képpel végezzük a különbségképzést, akkor a becsült kép és az aktuális kép közötti csekély eltérés miatt a különbségi kép kevés adatot fog tartalmazni. Tulajdonképpen a 2-es kép vastagított makroblokkjában lévő képrészletet kell megkeresni az 1-es képen, hogy tudni lehessen milyen irányú és nagyságú a labda elmozdulása. A 2-es kép valamely makroblokkjából az 1-es kép azonos tartalmú képrészletébe mutató vektor az ún. mozgásvektor. A mozgásvektor meghatározását mozgásbecslésnek hívjuk. A mozgásvektorok ismeretében az 1-es képből közelítően makroblokkokról makroblokkra megszerkeszthető a becsült 2-es kép. A valódi 2-es képpel összehasonlítva kapjuk a most már kevés adatot tartalmazó különbségi képet.

A mozgásvektorokat a dekóder a nagyobb pontosság végett a világosság-komponensből határozza meg, de ugyanaz a mozgásvektor alkalmazható a krominanciakomponensek mozgáskompenzációjához is.

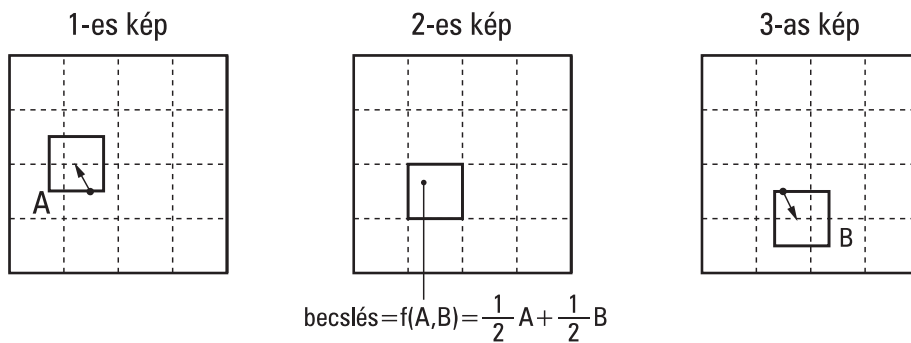
A mozgásbecslés bonyolult és nagymennyiségű művelet real-time elvégzését teszi szükségessé. A mozgásvektor kiszámításának módjára a szabvány nem tesz megkötést, ez a kóder fejlesztőjére van bízva.

A kódolási hatékonyság növelése érdekében az MPEG-rendszerrel a kódolandó kép utáni képet is felhasználjuk a becsléshez. Így kapjuk az ún. két irányból becsült (bidirectional predicted) vagy B-képet. Az előző, egy irányból becsült kép jele P (predicted frame). Az egy- és kétirányú becslést szemlélteti a 4.17. ábra.

### egyirányú becslés



### kétirányú becslés



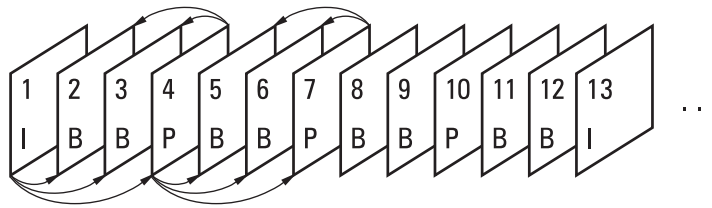
4.17. ábra  
Egyirányú és kétirányú becslés

Kétirányú becslésnél a 2-es képet követő 3-as képen is megkeressük az éppen kódolandó makroblokk megfelelőjét, a becslés azután lehet pl. a két makroblokk (A és B) átlaga. Ha a kódoló az 1-es vagy a 3-as képen nem talál becsléshez használható képrészletet (ami pl. képvágásnál fordul elő), akkor úgy is dönthet, hogy csak az egyik képből valósítja meg a becslést. Gyors mozgásoknál előfordulhat, hogy a mozgásbecslés felmondja a szolgálatot, ilyenkor a kódoló teljesen lemondhat a mozgáskompenzációról, és a mozgásvektor szimbólumainak átviteléről. A kódolási hatékonyság a kétirányú becslés következtében jelentősen növekszik. Kétirányból becsült makroblokkok átlagosan körülbelül fele annyi adatmennyiséggel kódolhatók, mint az egyirányból becsültek.

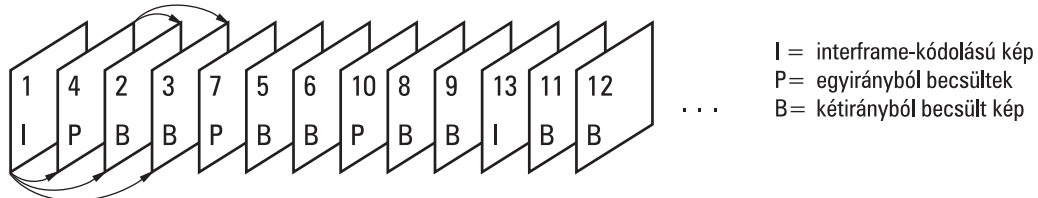
### 4.2.3 Képatrendezés

A kétirányú becslés nem csak a mozgásvektor meghatározásának számítási igényét kétszerezi meg, hanem a már említett képátcsoportosítást is szükségessé teszi. A 4.18. ábra felső részében egy kódolandó képsorozatot láthatunk.

megjelenítési (eredeti) sorrend



átviteli (kódolási) sorrend



4.18. ábra  
Képtrendezés

Ha minden képet differenciálisan kódolnánk, akkor sohasem találna belépési időpontot a dekódoló, mivel minden kép az öt megelőzőktől, ill. az öt követőktől függene, a stúdiókban pedig lehetetlenné válna a szerkesztés, ill. a vágás. Ha átviteli hiba lépne fel, az nagyszámú későbbi képen is jelentkezne a dekóder visszacsatolt struktúrája következtében. A jelenséget hibaterjedésnek hívjuk. Mindez elkerülhető, ha periódikusan differenciálisan nem kódolt, un. intrakódolású (képen belül kódolt) I-képeket iktatunk a differenciálisan kódolt, az un. interkódolású képek közé. Ilyenkor a kóderben szünetel a különbségképzés. Ha pl. minden 12. kép intrakódolású, akkor a készülék bekapcsolásakor vagy programváltásnál maximálisan  $\frac{1}{2}$  másodpercig tart, amíg a dekódolás elkezdődhet.

Az I-képek közé támaszként egyirányú becsléssel meghatározott P-képeket iktathatunk be, amelyek az őket megelőző I- vagy P-képből származnak. A többi kétirányból becsült képet az őket közrefogó P-, ill. I- és P-képből képezi a kóder. Ahogy a nyilakkal jelzett képek közötti kapcsolatból látható, a dekódernek mindig ismerni kell az egymásra következő P-, ill. I-képeket, hogy a közbenső B-képeket dekódolni tudja, de ugyanez érvényes a kóderre is, hiszen a visszacsatoló ágban egy teljes dekóder található. Ez az oka annak, hogy a képek sorrendjét felcseréljük. A kóder először mindig a B-képeket közrefogó I-és P-képeket kódolja, ill. továbbítja, ezért két egymást követő B-kép esetén összesen 4 képtárolóval kell rendelkeznie, és a következő lépéseket kell végrehajtania: kódolnia kell az I-képet, és az eredetit becslés céljára meg kell őriznie, míg a B-képek eredetieit egyelőre csak tárolnia kell, majd a P-képet kell kódolnia és az eredetit a B-képek becsléséhez meg kell őriznie, és csak ezek után kódolhatók a B-képek. Ezzel szemben a dekódernek az egymást követő B-képek számától függetlenül csak két képtárolóra van szüksége, mivel csak a közrefogó I-és P-képeket kell megőriznie, a B-képek a dekódolás után közvetlenül megjeleníthetők, és így a tárolóból törölhetők. A B-képek dekódolásához még szükség van bizonyos póttárhkapacitásra is, amelynek nagysága függ a megvalósítástól. Az I-, P- és B-képek sorrendjének kialakítása, az un. „group of picture”-struktúra kizárólag a kódolón múlik. A dekóder bármelyik sorozatot fel tudja dolgozni, mivel mindig csak valamivel több, mint két képtárolóra van szüksége. A megvalósítási ráfordítás a kóderben és ezzel együtt a képminőség azonban döntően a képcsoport felépítésétől függ. A minőség és a ráfordítás közötti jó kompromisszumnak bizonyult a 4.18. ábra szerinti struktúra, amitől azonban el is lehet térni. Az intra- és interkódolt képek közötti átkapcsolások a zajszint és a zajösszetétel megváltozását okozzák. Az I-képek csak az intrakódolás újrakvantálási zaját tartalmazzák, míg az P- és B-képek zajos különbségi adatoknak a zajos I-képhez történő hozzáadásával



jönnek létre, és ezért még zajosabbak. Nagy kompressziós arányoknál ez zajlúktetés formájában láthatóvá is válik.

#### 4.2.4 Adatsebesség-szabályozás

Ahogy már szó volt róla az állandó adatsebesség a kvantálás befolyásolása révén valósul meg. Ha a kóder kimenetén lévő átmeneti tárolót a túlsordulás fenyegetné, mert változatos a képtartalom, és csak sok nagyfrekvenciás komponenssel kódolható, vagy mert a képbecslés a korlátozott keresési tartomány miatt működésképtelenné vált, akkor az újrakvantálás durvábbá válik, és a képminőség esetleg észrevehetően rosszabbodik. Ha a körülmények kedvezőbbé válnak, újból finomodik a kvantálás. A kvantálástól függően több vagy kevesebb adat jut az átmeneti tárbá. Hogy a rekonstruált kép minősége egy minimális szintet mindig elérjen, az adatsebesség-szabályozás alól az intra-kódolt képek egyenkomponense ki van vonva. Az egyenkomponens DCT-együtthatójának kvantálásakor az osztó mindig 8. A P-és B-képeknél ezzel szemben az egyenkomponenshez és minden más frekvenciához tartozó együttható, hasonlóan, mint a JPEG-nél, egy táblázatból vett lépcsőmagassággal kvantálódik, amit azonban még egy további kvantálás is követ az adatsebesség szabályozása céljából.

Az I-képek DC-együtthatója:

$$G_Q(0,0) = \text{kerekítés} \left( \frac{G(0,0)}{8} \right), \quad (4.14)$$

Egyébként:

$$A(f_x, f_y) = \text{kerekítés} \left( \frac{G(f_x, f_y)}{Q(f_x, f_y)} \right), \quad (4.15)$$


$$G_Q(f_x, f_y) = \text{kerekítés} \left( \frac{A(f_x, f_y)}{Q_F / 8} \right), \quad (4.16)$$

ahol

$f_x, f_y$	a helyfrekvenciák,
$G_Q(f_x, f_y)$	a kvantált DCT-együtthatók,
$G(f_x, f_y)$	a kvantálatlan DCT-együtthatók,
$A(f_x, f_y)$	segédérték,
$Q(f_x, f_y)$	a kvantálási táblázat értékei,
$Q_F$	a kvantálási tényező.


$Q_F$  tehát az adatsebesség-szabályozást biztosító makroblokkokról makroblokkra változó kvantálási tényező, amelyet az átmeneti tárból érkező vezérlőjel 1 és 31 közötti értékekre tud beállítani. Ezáltal a kvantálási tábla kvantálási lépcsői egy közös tényezővel változtathatók, amely 1/8-os fokozatokban vehet fel értékeket. Ha  $Q_F = 31$  sem elegendő a tár túlsordulásának megakadályozásához, akkor az un. „skipped” (átugrott) makroblokkokkal, amelyek 0 bittel vannak kódolva, lehet csökkenteni az adatmennyiséget. Ilyenkor egyszerűen az előző kép megfelelő makroblokkja ismétlődik. Mivel ez természetesen nem pont ugyanolyan, mint az aktuális kép makroblokkja, ezért erősen feltűnő. Ellenben ha a képtartalom olyan egyszerű, hogy  $Q_F = 1$ -nél is kevés adat keletkezik, akkor un. kitöltőbitek kerülnek beiktatásra.

Az MPEG-1 és MPEG-2 szabványoknál a 4.19. ábrán látható kvantálási táblák rögzítettek, eltérően a JPEG-szabványtól. Így a táblák jelzőbitjei is megspórolhatók.



	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	

$Q(f_y, f_x)$  I-képekhez



16	16	16	16	16	16	16	16	
16	16	16	16	16	16	16	16	16
16	16	16	16	16	16	16	16	16
16	16	16	16	16	16	16	16	16
16	16	16	16	16	16	16	16	16
16	16	16	16	16	16	16	16	16
16	16	16	16	16	16	16	16	16
16	16	16	16	16	16	16	16	16

$Q(f_y, f_x)$  P-és B-képekhez

$Q(f_x, f_x)$  P- és B-képekhez

4.19. ábra  
MPEG-1 és MPEG-2 szabványosított kvantálási táblái

Ha a kóder mégis úgy találja, hogy eltérő kvantálási táblákkal jobb kódolási hatékonyság érhető el, akkor opcionálisan ezek a táblák is alkalmazhatók, ill. átvihetők.

Az MPEG-szabványú videokódolás hierarchikus képarcitektúráját szemlélteti a 4.20. ábra. A legkisebb egység a blokk. 4:2:0 mintavételezési struktúrájánál 4 világosságblokkhoz egy  $C_B$ - és egy  $C_R$ -blokk tartozik, melyek együtt alkotnak egy makroblokkot.

4.20. ábra  
MPEG-videokódolás hierarchikus képarchitektúrája

Szegmensnek hívjuk a vízszintes sorban elhelyezkedő meghatározott számú makroblokkot. A képcsoport (group of pictures) a legkisebb önállóan dekódolható egység. A képcsoportok sorozata a szekvencia. Az adatfolyam struktúrája a képarchitektúrát követi. A blokkrétegtől felfelé minden rétegnek saját fejléce (header) van, amely a dekódoláshoz szükséges információkat tartalmazza. A mozgásvektorok a makroblokkok adatai előtt kerülnek átvitelre.

#### 4.2.6 MPEG-1 sajátosságok

Az MPEG-1 komputeres és multimédiás alkalmazásokra szánt kódolási eljárás, amely jól alkalmazható videoanyagok hagyományos CD-ken való rögzítésére. Annak érdekében, hogy a behatárolt adatsebesség tartható legyen, a kódolandó kép felbontását az un. „Source Input Format”-ra (SIF) kellett korlátozni, ami vízszintes és függőleges irányban is 2-szeres alulmintavételezést jelent a luminancia- és a krominanciaértékekre, ahogy azt a 4. fejezet bevezetőjében láttuk. Az  $Y$ -,  $C_B$ - és  $C_R$ -minták helyzetét szemlélteti a 4.21. ábra (4:2:0 formátum).



4.21. ábra  
4:2:0 mintavételi struktúra MPEG-1 kódolásnál

Ennél a formátumnál a krominanciaminták pontosan a luminanciaminták között helyezkednek el, ugyanúgy, mint a JPEG-nél, és ahogy az a komputer-technikában is szokásos. Ha tv-jelet akarunk MPEG-1-gyel átvinni vagy rögzíteni, akkor a kódér előtt és a dekóder után formátumillesztésre van szükség. Kiindulva az 576 aktív sort tartalmazó tv-képből a függőleges alulmintavételezést az egyik félkép elhagyásával érhetjük el. Kevés mozgást tartalmazó képrészleteknél a félképek elhagyásából származó függőleges irányú alias-hatás adaptív szűréssel (az egymás alatti minták súlyozott átlagolásával) kiküszöbölhető.

A vízszintes irányú alulmintavételezés decimálósűrővel valósítható meg, amely a krominanciaértékeknél gondoskodik a minták helyzetének eltolásáról is. A decimálósűrővel a súlyozott átlagolást csak minden második képpontra végezzük el, miáltal megvalósul a vízszintes irányú 2-szeres alulmintavételezés. Az  $Y$ -,  $C_B$ -,  $C_R$ -értékek vízszintes, ill. a  $C_B$ -,  $C_R$ - értékek függőleges decimálósűrését megvalósító egységes algoritmus:

$$y(n) = \frac{-29u(n-3) + 0u(n-2) + 88u(n-1) + 138u(n) + 88u(n+1) + 0u(n+2) - 29u(n+3)}{256}, \quad (4.17)$$

ahol  $y(n)$  az  $n$ . képpont és még 6 szomszédos képpont értékéből számított új érték az  $n$ . képpontra. A makroblokkstruktúra miatt az egy sorban lévő luminanciaminták számának és a sorok számának 16-tal, míg a krominanciaminták számának 8-cal kell oszthatónak lennie. Így a bemenőkép 352x288 luminanciamintát és 176x144 krominanciamintát ( $C_B$  és  $C_R$ ) tartalmaz. A letapogatás progresszív, a képfrekvencia pedig 25 Hz.

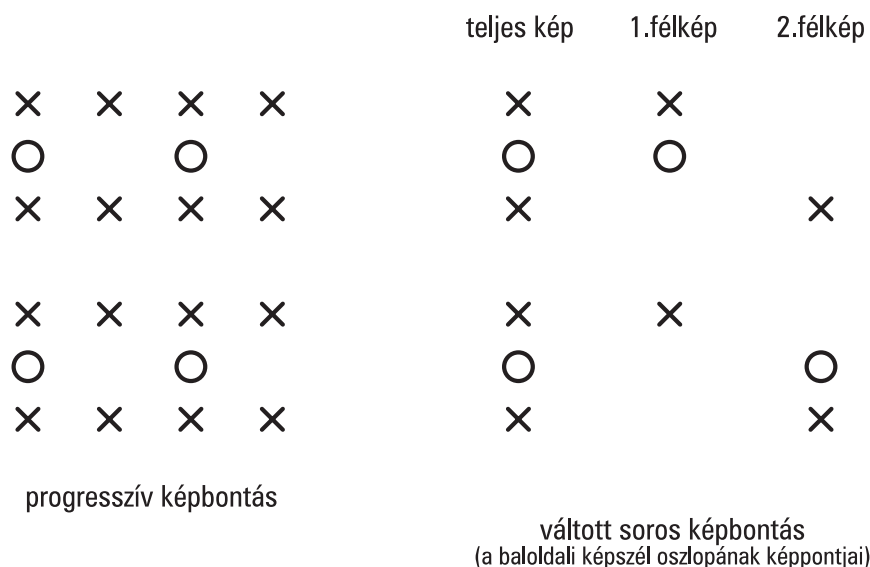
Dekódolás után az eredeti képformát inverz lépések segítségével vissza kell állítani. Fekete képszegélyek hozzáillesztése után következik a függőleges majd a vízszintes irányú túlmintavételezés. A hiányzó félkép ugyancsak túlmintavételezéssel pótolható.

Foglaljuk össze ezek után az MPEG-1 kódolás főbb jellemzőit:

- 1,5 Mbit/s adatsebesség SIF-jelek tömörítésekor; CD-hez és multimédiás felhasználásokhoz (SIF=Source Input Format: csak a páratlan számú félképeket kódoljuk, 625 soros képből kiindulva az aktív sorok száma 288 Y-sor és 144 krominanciasor. A képpontok száma egy Y-sorban 352 és egy krominanciasorban 176.)
- A kódolandó jelek  $Y'$ ,  $C'_B$  és  $C'_R$ .
- A letapogatási struktúra 4:2:0.
- A DCT DC-együtthatóinak kódolása DPCM révén valósul meg.
- A kvantálási táblák azonosak a luminancia- és a krominanciakomponensekre.
- A kvantálási táblák a szekvenciaretegben változtathatók meg.
- A VLC- táblák nem tölthetők le.
- A mozgásbecslés a P- és a B-képekre alkalmazható.
- A mozgásbecslés felbontása fél képpont.
- Csak progresszív képbontás lehetséges.

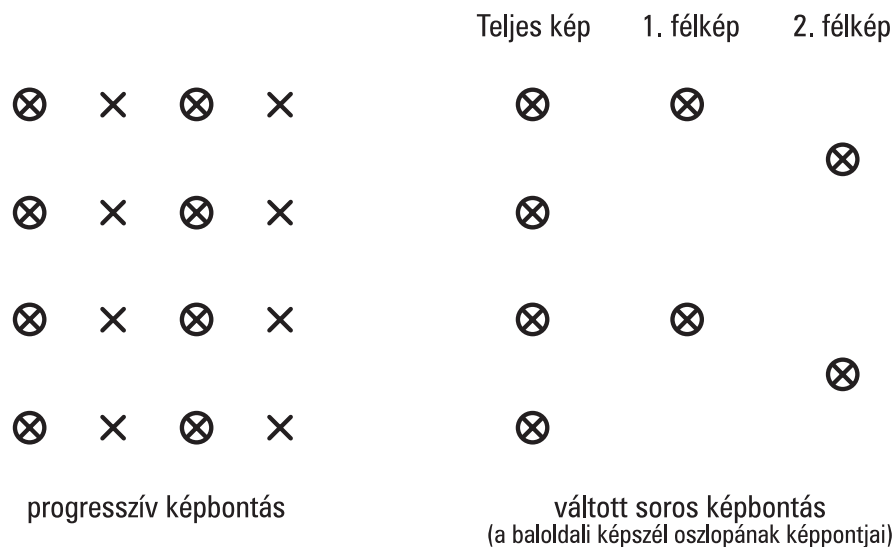
#### 4.2.7 MPEG-2 sajátosságok

Az MPEG-2 már mentes azoktól a lényeges korlátozásoktól, amelyek az MPEG-1-re érvényesek. A kódolás lehetséges teljes felbontású képre, és alkalmazható a 4:2:2, sőt a 4:4:4 krominanciaformátum is. Az európai DVB-projekt szerint a tv-műsorszórásban a 4:2:0 formátum kerül alkalmazásra (Main Profile), ami azonban a krominanciaszternek a luminanciaszterhez viszonyított helyzetében különbözik az MPEG-1-ben definiálttól. Ezenkívül a váltott soros képfelbontás is megengedett. A 4.22. ábra baloldali része a 4:2:0 formátumot szemlélteti progresszív letapogatású képbontásnál, ahol a luminanciaminták kereszttel, a krominanciaminták pedig körrel jelöltek.

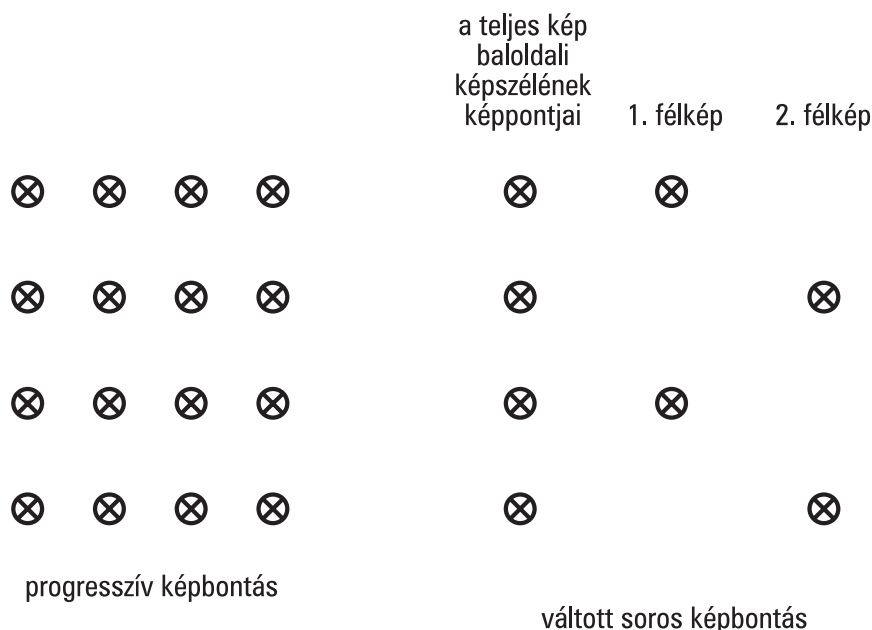


4.22. ábra  
4:2:0 képbontási struktúra MPEG-2-nél

Ahogy azt láttuk az MPEG-1-nél, szükség van egy horizontális szűrőre, ami az alulmintavételezést és egyben a krominanciaraszter vízszintes irányú eltolását is megvalósítja. Így olyan formátum jön létre, amely a számítógéptechnikában használatos. Az MPEG-2-t eleve tv-jelek kódolására fejlesztették, ahol a képbontási struktúra a 4.22. ábra szerinti. Váltott soros letapogatásnál a minták helyzete nem változik meg. A krominanciasoroknak a félképekhez való hozzárendelése a 4.22. ábra jobb oldalán látható a teljes kép első oszlopának képpontjaival szemlélítve. Az MPEG-2 szabvány „High Profile”-módjának mintavételi struktúrája 4:2:2, amit a 4.23. ábra érzékeltet. A 4.24. ábrán a szintén tervbe vett 4:4:4 formátum látható, amelyet végül mégsem vettek fel a „profile”-ok közé.



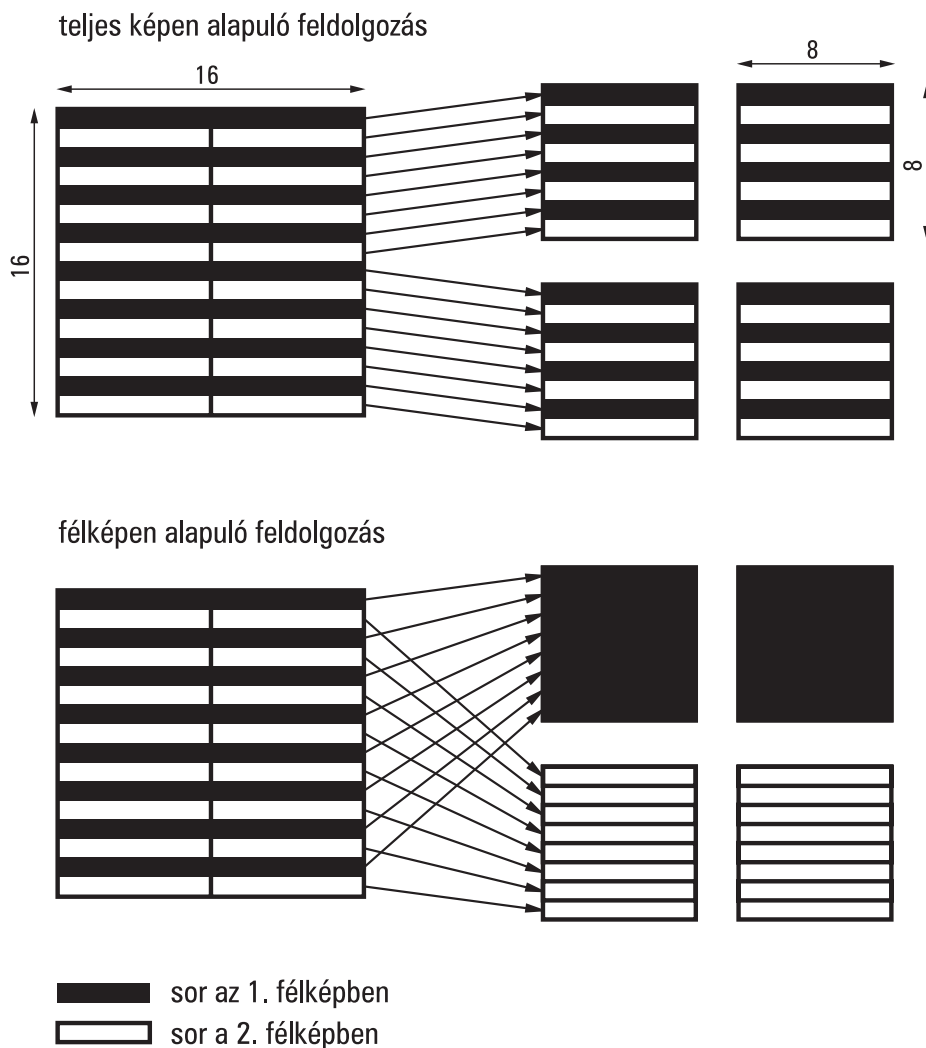
4.23. ábra  
MPEG-2 4:2:2 mintavételi struktúrája



4.24. ábra  
4:4:4-es mintavételi struktúra MPEG-2-höz

Jövőbeni alkalmazásokhoz a szabvány további „profile”-okkal egészíthető ki, amelyekhez a 4:4:4-es formátum is felhasználható.

A váltott soros képbontás a diszkrét koszinusztranszformáció megvalósítására is kihat. Az MPEG-2-nél választani lehet az un. teljeskép-mód és az un. félképmód között, melyeket a 4.25. ábra szemléltet. Az ábrán a teljes kép makrobloknagyságú képrészlete fekete, ill. fehér sorokból áll. Teljeskép-módban mindegyik makroblokkot négy 8x8-as DCT-blokkra bontjuk úgy, hogy mindegyik 8x8-as blokk mindkét mozgásfázist, azaz mindkét félkép sorait tartalmazza.

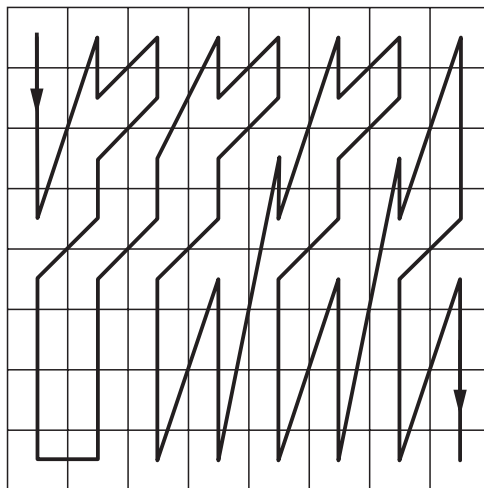


4.25. ábra  
Makroblokkoknak 8x8-as DCT-blokkokra történő bontása

A képen belüli gyors mozgások váltott soros képbontásnál a függőleges élek felszakadását okozzák, ami a függőleges frekvenciakomponensek megnövekedése miatt a kódolási hatékonyság csökkenéséhez vezet. Ebben az esetben előnyösebb a DCT-blokkokat az ábra alsó felében látható módon a félképekből képezni. Ilyenkor 4:2:0 formátumnál két 8x4-es nagyságú krominanciablokk jön létre az egy 8x8-as nagyságú helyett. Állóképeknél vagy csak kevés mozgást tartalmazó képeknél általában a teljes képen alapuló kódolás az előnyösebb, mivel a szomszédos sorok többnyire jobban hasonlítanak egymásra, ami azzal

jár, hogy a DCT-együtthatók értéke kisebb lesz. Hogy milyen módon történjen a blokkok létrehozása, azt a kóder dönti el.

MPEG-2-nél, eltérően a JPEG-től és az MPEG-1-től, opcionálisan a DCT-együtthatók nemlineáris kvantálása is lehetséges, ami szintén a kódolási hatékonyságot növeli. Alkalmazható egy, a félképmódusú blokkok futamhosszkódolásához jobban megfelelő második cikk-cakk görbe is (4.26. ábra), amely a hozzá tartozó módosított futamhosszkódolással és egy második Huffman-táblával szintén a kódolási hatékonyságot növeli.

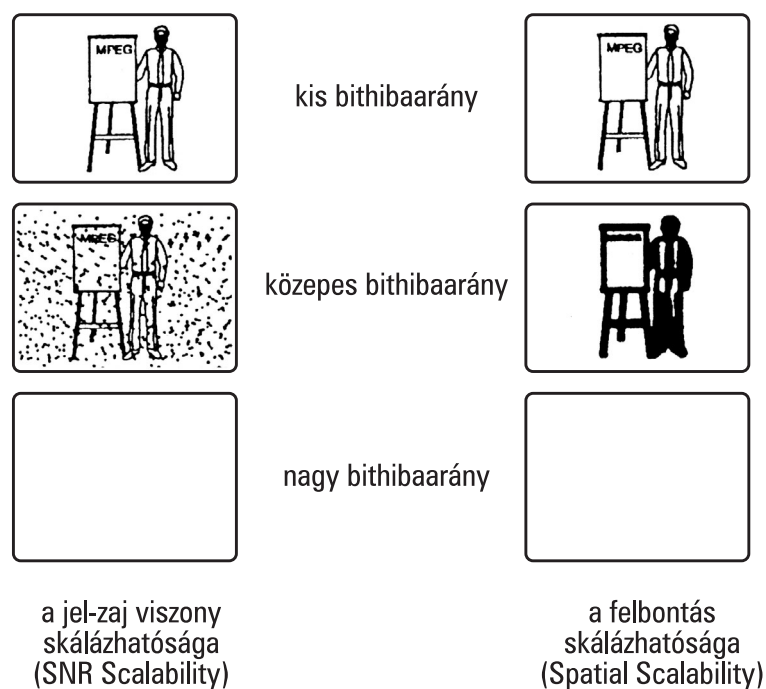


4.26. ábra  
Opcionális cikk-cakk görbe MPEG-2-höz

A képminőséget és a kódolás hatékonyságát javító megoldások összefoglalva a következők:

- 4:4:4, 4:2:2, 4:2:0 mintavételi struktúrák.
- DC-együtthatók javított pontossággal adhatók meg.
- Képenként változtatható kvantálási tábla.
- 4:2:2 formátumnál a kvantálási táblák különbözőek a krominancia- és a luminanciajel DCT-együtthatóira.
- Folytonos és váltott soros képbontású képek feldolgozhatósága.
- Félképre, ill. képre épülő mozgásbecslés.
- Félképre, ill. képre épülő DCT.
- A DCT-együtthatók nemlineáris kvantálása.
- Módosított cikk-cakk görbe és Huffman-tábla alkalmazhatósága.

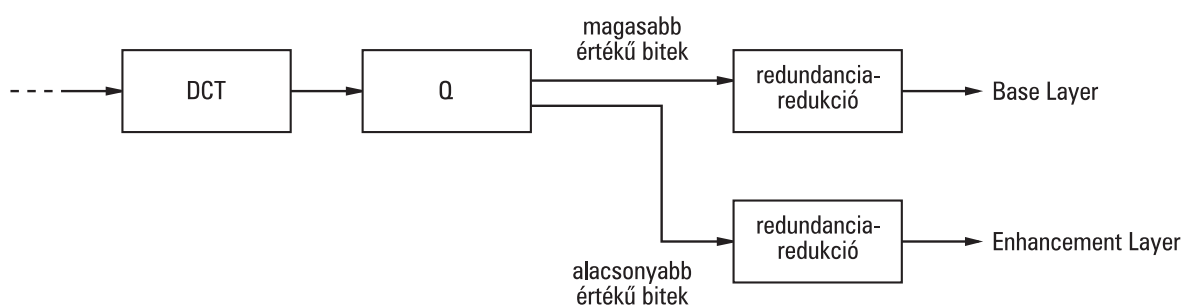
A digitális jelátvitelre általában jellemző, hogy az összeköttetés minősége hirtelen rosszabbodik, ha a jel-zaj viszony egy bizonyos szint alá csökken, ami a bithibaarány romlásának következményeként az addig hibátlan átvitel azonnali megszűnéséhez is vezethet. Digitális földi tv-műsorszórásnál megtörténhet, hogy az ellátottsági terület határán az egyik utcában a vétel kifogástalan, ugyanakkor egy másik utcában egyáltalán nincs vétel. Vagy kis átmérőjű parabolaantennával történő műholdas vételnél változékonnyá időjárási feltételek mellett a vétel periódikusan megszakadhat. Abból a célból, hogy ezeket a kellemetlen jelenségeket kiküszöböljék, az MPEG-2-nél kialakították az ún. skálázhatóságot, egy közbelső minőségi fokozatot a hibátlan vétel és a teljes vételkimaradás közötti átmenet megvalósítása érdekében. A skálázásnak a képre gyakorolt hatása függ a kialakítás módjától. Ezt szemlélteti a 4.27. ábra.



4.27. ábra  
Skálázhatóság hatása a képminőségre MPEG-2 forráskódolásnál

Kis bithibaaránynál továbbra is teljesen zavarmentes a vétel. Ha bithibaarány kissé növekedve közepes értékű lesz, akkor egy zajosabb vagy egy rosszabb felbontású, de még elfogadható képet kapunk. A vétel majd csak a nagyobb bithibaarányoknál szűnik meg teljesen. A skálázhatóság megvalósításának feltétele az ún. hierarchikus jelstruktúra, ill. moduláció.

A 4.28. ábrán a jel-zaj viszony skálázhatóságának kialakítását látjuk.

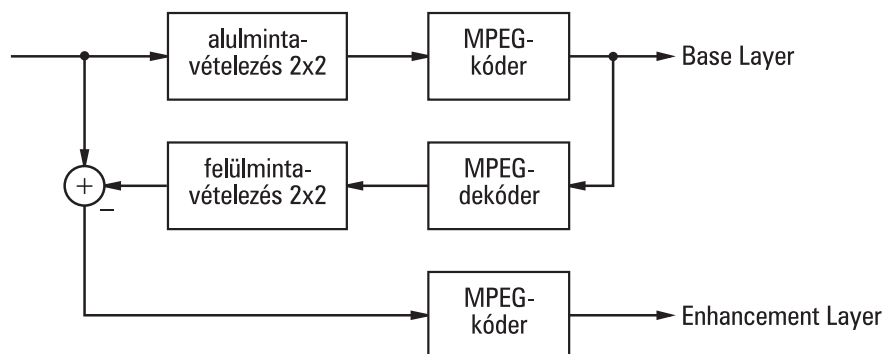


4.28. ábra  
Jel-zaj viszony skálázhatóságának megvalósítása MPEG-2-nél

A kvantálás után az egyes DCT-együtthatókat a biteik értéke szerint két csoportba soroljuk (alacsonyabb, ill. magasabb értékűek), majd utána a szétválasztott bitekre külön redundancia-redukciót alkalmazunk. Az ún. „Base Layer”-hez tartozó nagyobb értékű biteket a csatornakódoló az átvitel számára hatékonyabb hibavédelemmel látja el, mint az „Enhancement Layer” alacsonyabb értékű bitjeit. Ezáltal az átviteli csatorna tulajdonságainak romlásakor először a kvantált DCT-együtthatók kevésbé fontos bitjei esnek ki, ami a képben zajjellegű zavarként jelentkezik.



Egészen más a felbontás skálázását megvalósító kóder felépítése (4.29. ábra). A kódolandó képet először vízszintesen és függőlegesen kétszeresen alulminta-vételezzük, majd a csökkentett felbontású kép kódolására alkalmas MPEG-2-kóderrel kódoljuk.



4.29. ábra  
Felbontás skálázásának megvalósítása MPEG-2-nél

A létrejövő „Base-Layer” azután a csatornakódolóban ismét hatékony hibavédelmet kap. Ahogy látjuk a Base Layerhez tartozó jelet rögtön dekódoljuk, majd az így kapott képet az eredeti felbontására konvertáljuk, és mint becslült képet alkalmazzuk az „Enhancement Layer” számára. A különbségi kép MPEG-kódolás után a csatornakódolóban kevésbé hatékony hibavédelmet kap. A felbontás skálázhatóságának módszerével valósítható meg a HDTV-jel kompatibilis átvitele is. A két skálázhatóság kombinálásával pedig többfokozatúvá tehető a vételmegszűnés folyamata. Az európai DVB-projektben sem a jel-zaj viszony, sem a felbontás skálázhatósága nem képezi a rendszerspecifikáció részét. Megemlítenéd, hogy létezik még két további skálázhatóság is, az un. „Temporal Scalability”, amelynél az időbeni felbontást, azaz a képfrekvenciát csökkentik, és az un. „Data Partitioning” amely sajátágaiban a jel-zaj viszony skálázhatóságára hasonlít. E két módszerrel a továbbiakban nem foglalkozunk.

Végezetül tekintsük át az MPEG-2 kódolás Profile/Level táblázatát (4.30. ábra)

High Level		1920x1152 Pixel 80 Mbit/s			1920x1152 Pixel (960x576) Pixel 100(80,25) Mbit/s
High- 1440 Level		1440x1152 Pixel 60 Mbit/s		1440x1152 Pixel (720x576) Pixel 60(40,15) Mbit/s	1440x1152 Pixel (720x576) Pixel 80(60,20) Mbit/s
Main Level	720x576 Pixel 15 Mbit/s	720x576 Pixel 15 Mbit/s	720x576 Pixel 15(10) Mbit/s		720x576 Pixel (720x576) Pixel 20(15,4) Mbit/s
Low Level		352x288 Pixel 4 Mbit/s	352x288 Pixel 4(3)Mbit/s		
Levels Profiles	Simple Level	Main Profile	SNR Scalable Profile	Spatial Scalable Profile	High Profile
	mint a Main Prof. B-képek nélkül	4:2:0, nincs skálázhatóság	mint a Main Prof. + SNR-skálázhatóság	mint a SNR Scal. Prof. + Spat. Scal.	mint a Spat. Scal. Prof. + 4:2:2 - kódolás

4.30. ábra  
MPEG-2-kódolás Profile/Level táblázata

Balról jobbra az ún. „Profile”-ok vannak feltüntetve, amelyek alatt meghatározott szintaktikus elemeket és ezáltal meghatározott kódolási sajátosságokat értünk. (A szintaxis valamilyen jelrendszer elemeinek sajátos funkciót, ebből fakadó jelentést adó szerkesztési szabályok együttese.) Alulról felfelé helyezkednek el a szintek („Level”), amelyek a kódolási paraméterek fokozatait adják meg többek között az adatsebességre, ill. a képfelbontásra. A táblázat mezői a felbontás és az adatsebesség felső határát tartalmazzák. A zárójelben lévő számok a skálázható kódolás alacsonyabb szintjére, ill. szintjeire vonatkoznak. Az áthúzott mezők nincsenek definiálva. A „Low Level” tévéképek csökkentett felbontású kódolására létrehozott eljárás (LDTV, Low Definition Television), hasonló az MPEG-1-nél már megismert SIF-kép kódolásához, mindenesetre a 4 Mbit/s-ban meghatározott maximális adatsebesség miatt sokkal jobb minőségű képet kapunk, mint az MPEG-1-nél. A „Main Level” a mai normálfelbontású tv jeleinek kódolására alkalmas, és kb. PAL-képminőség érhető el vele. A HDTV (High Definition Television) kódolására „High-1440 Level” és a „High Level” alkalmas, az utóbbinál a soronkénti pontszám a 16:9-es képformátumhoz illesztett. A 16:9-es jelek átvitele azonban minden szinten (és minden profile-lal) lehetséges.

A „Simple Profile” és a „Main Profile” csak annyiban különböznek egymástól, hogy a „Simple Profile”-nál nem megengedettek a kétirányból becsült képek. Emiatt a kóder és a dekóder kevésbé összetett, mindazonáltal a kódolási hatékonyság oly mértékben csökken, hogy valószínűleg a „Simple Profile” nem kerül felhasználásra. Mindkét profile-nál valamennyi skálázhatóság kizárt, és rögzített a 4:2:0 krominanciaformátum. Az „SNR Scalable Profile” megengedi a jel-zaj viszony skálázhatóságát, míg a „Spatial Scalable Profile” esetében az előbbi a felbontás skálázhatóságával bővül, és csak a „High Profile” engedi meg a 4:2:2 krominanciaformátumot.

Az egész táblázat lefelé kompatibilis, ami azt jelenti, hogy ha egy dekóder a táblázat meghatározott mezőjéhez hozzárendelhető, akkor valamennyi tőle balra és lefele lévő profil, ill. szint adatfolyamát dekódolnia kell tudni. Minden MPEG-2-es dekódernek képesnek kell lennie továbbá bármilyen szintű és profilú MPEG-1 adatfolyam dekódolására is. A szabványban különleges kódolási eljárásokat is definiáltak, amelyek egyik Profile/Level kombinációban sem találhatók, mint pl. a 4:4:4 kódolás. Ezek a táblázat lehetséges jövőbeli bővítésére vannak szánva. Mindazonáltal a már definiált mezők nem változtathatók meg. A Profile/Level táblázat létrehozásának értelme abban van, hogy bizonyos alkalmazásokhoz egyszerű, lehetőleg könnyen megvalósítható eljárást kínáljon az MPEG-2 szabvány szerteágazó lehetőségei közül, nehogy minden dekóder szükségtelenül minden feladatra alkalmas legyen.

### 4.3 Összefoglalás

A 4. fejezetben a JPEG-, az MPEG-1 és az MPEG-2 videokódolással foglalkoztunk. A JPEG-szabványt, amely a DCT-t követően kvantálást és redundanciaredukciót alkalmaz, elsősorban állóképek adattömörítésére fejlesztették ki. A megfelelő teljesítményű és áru IC-k korai hozzáférhetőségével hamarosan mozgóképek kódolására is alkalmazni kezdték, azonban ez az ún. „Motion-JPEG” gyártóról gyártóra különböző megoldási módokat tartalmazott és így az egymás közötti kompatibilitás nem teljesült. A video-vágórendszerek területén a JPEG szilárd pozícióval rendelkezik.

Az MPEG-1 kezdettől fogva az alacsony minőségi szintű alkalmazásokra szánt kódolási eljárás (számítógép, multimédia), ahol széleskörűen alkalmazzák. Az adattömörítés révén elérhető költségcsökkenés vonzóvá tette ezt a szabványt az eredeti célkitűzéseket meghaladó területeken is, mint pl. a nagyobb adatsebességekkel megvalósítható magasabb

minőségi fokozatú tv-képtvitel. Ahogy a JPEG, úgy az MPEG-1 is transzformációs kódolásra épül, kiegészítve azt a képek között meglévő időbeni kapcsolatok kihasználásával és állandó adatsebességre való szabályozással.

Végül az MPEG-2 az összes szükséges eszközt tartalmazza a különböző minőségű tv-képek kódolására, kezdve a VHS-hez hasonló SIF-től egészen a HDTV-ig. Az Európai Digitális Tv-projekt (DVB) a digitális tv videokódolását az MPEG-2 szabvány „Main Profile at Main Level” (MP@ML) fokozatának felhasználásával valósítja meg. A készülékek közötti kompatibilitás végett a képfrekvencia 25 Hz-ben rögzített, és valamennyi DVB-vevőnek tudni kell a 4:3 és a 16:9 képméretarányú adásokat dekódolni és megjeleníteni, ami érvényes SIF-képekre is. Ezzel a DVB az első nagy alkalmazása a Moving Pictures Experts Group által a videokódolásra kifejlesztett szabvány második fokozatának.