

S Z A K D O L G O Z A T

Készítette: Szűcs Tibor

Debrecen

2008

Debreceni Egyetem

Informatikai Kar

A H.264 video szabvány bemutatása

Témavezető:

Dr. Tornai Róbert

egyetemi adjunktus

Készítette:

Szűcs Tibor

programozó matematikus

Debrecen

2008

Tartalomjegyzék

1	BEVEZETÉS	1
2	A SZEM ANATÓMIÁJA ÉS A LÁTÁS	2
2.1	Színterek (RGB, CYMK, YUV)	3
2.1.1	RGB	4
2.1.2	CYMK	4
2.1.3	YUV	4
2.1.4	Az YUV és RGB modell összefüggése	5
3	TÖRTÉNETI ÁTTEKINTÉS	6
3.1	Analóg fényképezés	6
3.2	PAL, NTSC, SECAM	7
3.3	Digitális fényképezés	8
3.4	A digitális képi információ nagysága	9
3.5	JPEG - (Joint Photographic Experts Group)	10
3.5.1	DCT	10
3.5.2	A kvantálás	12
3.5.3	Hamis kontúrok	15
3.5.4	A döntési folyamat modellezése	15
3.5.5	A DCT műveletigénye	18
4	A SZABVÁNY BEMUTATÁSA, ÖSSZEVETÉSE MÁS VIDEO SZABVÁNYOKKAL	19
4.1	Hogyan lesz az állóképből mozgókép	19
4.2	Az MPEG-1-2-4 rövid bemutatása	19
4.2.1	MPEG-1	19
4.2.2	MPEG-2	22
4.2.3	MPEG-4	24
4.3	MPEG-4 Part 10 AVC / H.264	27
4.3.1	Az AVC forráskódolási algoritmus	28
4.3.2	A H.264/MPEG-4 AVC profiljai	33
4.3.3	CAVLC példa	35
5	ÖSSZEGZÉS	39
6	IRODALOMJEGYZÉK:	40

1 Bevezetés

Témaválasztásomat két dolog is indokolta. Annál a cégnél, ahol felsőfokú tanulmányaim alatt szoftverfejlesztőként felvételt nyertem, adott volt egy projekt, aminek én is aktív tagjává váltam. Ez nem volt más, mint egy olyan videó codec (COder-DECoder) kifejlesztése, ami képes arra, hogy 1 Mb-es interneten Full HD minőségű (1920x1080 pixel) videót képes legyen valós időben szerveroldalon bekódolni, illetve kliens oldalon kikódolni. Ennek megvalósítására a H.264-es videó codec szabványát, és a C programozási nyelvet hívtuk segítségül. Akkori projektvezetőm – aki később ennek a szakdolgozatnak a külső konzulense lett – javaslatára tettük ezt, mivel ő már fejlesztett előzőleg, többek között H.263-as, H.261-es és MPEG2-es videó codeceket.

Az ő szakértelme, és az a tény, hogy ez volt életem első szoftverfejlesztési projektje megszeretette velem a codec írást. Ez az első indok, amiért ezt a témát választottam.

A másik pedig az, hogy a jelenlegi multimédia piacon én magam is ebben látom a jövőt. Lassan minden lakásban elérhetővé válik Magyarországon a szélessávú internet, és mindenki számára elérhetővé válik a legalább HD Ready (1280x720 pixel) felbontású tv készülék, akkor miért ne szolgáltatassunk olyan tartalmat, ami élvezhető minőséget nyújt a nézőnek, ha már minden technikai feltétel a kezünkben van.

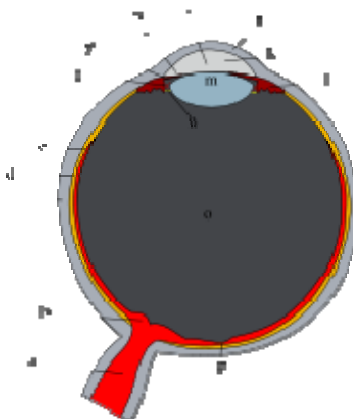
Ahhoz, hogy igazán megértsük a digitális videó tömörítés lényegét, mindenképpen szót kell ejtenem az emberi látásról, illetve a szem tulajdonságairól biológiai megközelítésből.

Majd az analóg és digitális videó rögzítés tulajdonságait számba véve mutatom be, hogy milyen nagy újításokat tartalmaz a H.264 videó codec szabványa.

2 A szem anatómiája és a látás

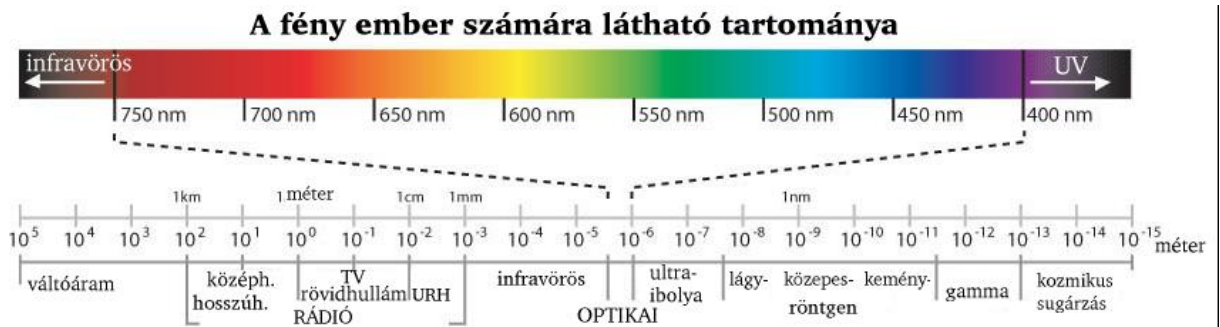
Számomra hihetetlen volt elfogadni azt a tényt, hogy mennyivel érzékenyebbek a gépek az emberi szemnél. Arról nem is beszélve, hogy valamennyi „csatornán” sugárzott adások ezt a tényt használják ki, és hogy van olyan része a felvételnek, amit mi sohasem láthattunk.

Az emberi szem egyszerű, két részből álló objektívvel rendelkezik. A külső a szaruhártya, a belső a lencse. A szivárványhártya, amely a szem színét is meghatározza, a szembe lépő fény mennyiségét csökkenti. Ez áthalad az üvegtesten és a retinán, ezután kognitív funkciók igénybevételével kialakul a kép.



1. ábra: Az emberi szem

A retinát a látás érzékelésére a foto receptorok, 130 millió pálcika és a színes látás érzékelésére 7 millió csap alkotja. A nyúlványok legnagyobb koncentrációja az úgynevezett sárgafoltban, tehát a legélesebb látás helyén van. Egy kis túlzással azt is mondhatnánk, hogy az emberi szem 137 megapixeles fényképezőgép. A szem és az agy képesek a környezetet akár három látószögben is látni. Az ember és vele együtt a többi főemlős is a világot a kéktől egészen a vörös színig érzékeli, tehát körülbelül a 400-tól 700 nm-ig terjedő hullámhosszú fényt.



2. ábra: A fény hullámhosszai

A fény elektromágneses sugárzás, és az energia megnyilvánulás egyik formája, melynek hullámhossza az emberi szem által látható tartományba, azaz az infravörös és az ultraibolya közé esik. A három alapvető mennyiség, amely meghatározza a fényt (és bármely elektromágneses hullámot):

- intenzitás (vagy amplitúdó, amelyet az ember fényerőként, fényességként érzékel)
- frekvencia (vagy hullámhossz, amelyet az ember színeként érzékel)
- polarizáció (azaz a rezgés iránya, ezt az átlagember normál körülmények között nem érzékeli, csak a rovarok egy része).

A hullám-részecske kettősség elmélete szerint a fénynek hullám és részecske tulajdonságai is vannak. A digitális képalkotás illetve a digitális tömörítési eljárások a fény hullámjainak tulajdonságait használja ki. Gyakorlatilag csak a szem számára látható tartományon belüli részeket tartják meg, a többit elhagyják.

2.1 Színterek (RGB, CYMK, YUV)

A színterek is fontos szerepet játszanak a képalkotásban. A különböző színterek, a színeket különböző módon tárolják, és különböző eljárással adják vissza elénk a letárolt képet. Nézzük rendre a három leggyakrabban használt színtér leírását.

2.1.1 RGB

A színábrázolás legelterjedtebb modellje, Az RGB színrendszerben a színek a három alapszín a vörös (R – red), zöld (G – green), kék (B – blue) egymásra vetítésével – összeadásával – állíthatók elő, ez tulajdonképpen az additív színkeverés, tehát a három alapszín egyforma keverése fehér, hiányuk fekete színt eredményez.

Az ábrázolás alapja, hogy a színeket a három színtonkomponens intenzitásával adják meg, tehát három számmal. Ezek a konkrét ábrázolástól függően különböző tartományból származhatnak, például 0-1 (valós számok), 0-100 (százalékok), 0-255 (egész számok) és egyéb kettő hatványáig terjedően. Ha egy képpontot egész számokkal ábrázolunk, akkor a piros, a kék és a zöld 256-256-256 féle árnyalatából áll össze, összesen 16 millió színárnyalattal. 24 illetve 32 biten tárolja az információt [10].

2.1.2 CYMK

A színeket három alapszín, a cián vagy zöldeskék (Cyan), a bíborvörös (Magenta), és a sárga (Yellow) szubsztraktív keverésével állítjuk elő; mivel így a fekete szín nem állítható pontosan elő – és egyéb gyakorlati okokból – a fekete színt (blacK) hozzávesszük az alapszínekhez. Ezt a színteret nyomdai és kiadványszerkesztésben használják.

2.1.3 YUV

Egy színt a fényerő vagy luminancia jellel (Y) és két ún. szín különbségi jellel (U és V) segítségével adunk meg.

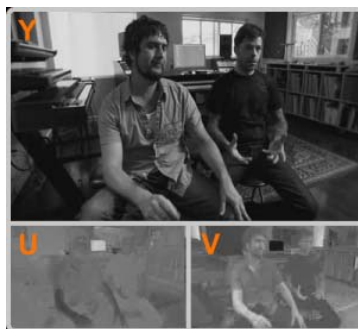
2.1.4 Az YUV és RGB modell összefüggése

Luminancia:
$$Y = 0.229 * R + 0.587 * G + 0.114 * B$$

(érdemes megfigyelni, hogy a fényerőhöz az egyes alapszínek nem egyforma mértékben járulnak hozzá (az arány kb. 3:6:1); az emberi szem legérzékenyebb a zöldre, kb. fele annyira érzékeny a vörösre, és hatodannyira a kékre)

Krominancia (színinformációk):
$$U = (B - Y) * 0.493$$

$$V = (R - Y) * 0.877$$



3. ábra: Egy YUV kép komponensekre bontva

3 Történeti áttekintés

Ebben a fejezetben az analóg és a digitális fényképezése mellett a műsorszórási szabványokat mutatom be.

3.1 Analóg fényképezés

Azokat az anyagokat, amelyek fény hatására kémiai vagy fizikai elváltozásokat szenvednek, fényérzékeny anyagoknak nevezzük. Valamennyi általánosan használt nyersanyag alapvető felépítése hasonlóságot mutat. Rendszerint egy vagy több fényérzékeny rétegből, hordozóból és segédrétegből áll. A fényképezésnél felhasznált fényérzékeny anyagok valamilyen hordozóra felvitt zselatinrétegben finoman elosztott ezüst-bromidból állnak. Az ezüstbromid-kristályban fény hatására fémezüst és bróm válik ki. A fény hatására közvetlenül bekövetkező változás szabad szemmel nem látható (látens kép) mert az így kivált ezüst mennyisége csekély (5-10 atom szemcsénként). Előhíváskor a kivált ezüst katalizátorként működik, az egész szemcse AgBr tartalma elbomlik. Ha szemcsében nem volt fény által kiválasztott ezüst, akkor előhíváskor nincs változás. A kép sötéttségét az előhívott szemcsék mennyisége szabja meg.

Az analóg fényérzékeny anyagoknál az anyag úgynevezett gradációjának meghatározásánál játszik szerepet. Ez leegyszerűsítve azt jelenti, hogy az expozíció során kapott, adott mértékű megvilágítás különbségre a kidolgozott anyag mekkora sötétedés különbséggel (denzitás érték változással) reagál. Ha adott megvilágítás különbségre az anyag kis denzitás különbséggel reagál, lágy gradációjú; ha nagyval, kemény gradációjú; a kettő közötti esetben pedig normál gradációjú.

A gradációt számszerűen a γ (gamma) érték fejezi ki. Ez az ún. jelleggörbe (más néven: gammagörbe) „egyenes” szakaszának meredekségével arányos érték (elvbem az „egyenes” szakasznak a gammagörbe vízszintes tengelyével bezárt szöge alapján határozzák meg). Minél meredekebb ez a szakasz, annál nagyobb a gamma érték, illetve annál keményebb az anyag. A jelleggörbe vízszintes tengelyén a megvilágítás értékek

logaritmusát, a függőleges tengelyén a kidolgozás után mérhető denzitást tüntetik fel. (A gamma érték kiszámítása a jelleggörbe „egyenes” szakaszának értelmezése függvényében különféle módszerekkel történik, ugyanis az „egyenes” szakasz gyakorlatilag sosem igazán egyenes.)

A gradáció fogalom a digitális szóhasználatban kontraszt elnevezéssel használatos.

3.2 PAL, NTSC, SECAM

Most nézzük az analóg televíziózásban használt leg elterjedtebb szabványokat, amiket a mai napig előszeretettel használnak [7].

A Phase Alternating Line (PAL) a német Telefunken cég laboratóriumában kifejlesztett analóg színes televíziós rendszer. Napjainkban még mintegy 60 országban sugározzák ebben a rendszerben a televízióadásokat. Elsősorban az európai országokban terjedt el, de sokfelé használják. A PAL szabvány szerint sugárzott kép 625 sorból áll (ebből azonban csak 575 sor alkotja magát a képet). A maximális horizontális felbontás (az egy sorra jutó képpontok száma) 768 pont. Az egy másodperc alatt megjelenített képkockák száma 25, ami tulajdonképpen 50 "fél-képkocka" sugárzását jelenti. (A TV először a páratlan sorszámú sorokat jeleníti meg, majd a páros sorszámúakat).

A National Television Standards Committee (NTSC), Nemzeti Televízió szabvány Bizottság, illetve az általa kidolgozott, elsősorban az Egyesült Államokban és Japánban használt televíziós szabvány. A PAL rendszerrel ellentétben az NTSC másodpercenként 30 darab, egyenként 525 pixelsorból felépített képkockát jelenít meg, szintén sorváltott üzemmódban.

A SECAM (Séquentiel a Memoire), Franciaországban használatos színes tévérendszer, amely különböző kelet-európai országokban is elterjedt. Másodpercenként 50 fél képpel dolgozik. Jellemzője, hogy csak minden második képsor színének információját tárolja el egy speciális tároló médiumon, ami aztán hivatkozásként szolgál

a köztük fekvő sorokra. A színekre vonatkozó információk „részlegesen vándorolnak a memóriába”.

3.3 Digitális fényképezés

A hagyományos fényképezőgépekhez hasonlóan a digitális kameráknak is fontos része a képrajzoló lencserendszer, az objektív. Ez vetíti az előtte lévő megvilágított motívum képét a mögötte elhelyezkedő fényérzékeny felületre, pont úgy, ahogy az analóg kameráknál is. Ez a felület a digitális fényképezőgépeknél nem film, hanem egy elektronikus alkatrész a CCD. A CCD sok apró, egymás mellett elhelyezkedő fényérzékelő cellából áll. Ezekben a cellákban a rájuk eső fény erősségével arányos elektromos feszültség, töltés keletkezik. Ezt a töltést egy elektronika kiolvassa és egy előre meghatározott kódolással számjegyek sorozatává alakítja. Így elérhető, hogy a látható képből egy számsor, más szóval digitális adat keletkezik.



4. ábra: A CCD az objektív mögött, a film helyén van.

Máris elérkeztünk az első problémánkhoz, amit meg kellett oldani. A digitális videó rendszerek által készített képi információ tárolása és kezelése mindig komoly kihívást jelentett a témával foglalkozó szakemberek számára. Azonban a keletkező adatmennyiség nagysága miatt a digitális rendszerek, vitathatatlan előnyeik mellett még mindig csak rövid ideig képesek elérni az analóg rendszerek valós idejű képrögzítését. Megoldást jelenthet az információ megfelelő kódolása, tömörítése, mellyel a tárolt adatok mennyisége jelentősen csökkenthető.

3.4 A digitális képi információ nagysága

Nézzük meg számokon keresztül, hogy mivel is állunk szemben. (Tételezzük fel, hogy egy digitális videó rendszer összes egysége és hardver eleme képes a képeket valós időben folyamatosan feldolgozni és tárolni.) Egy PAL valós idejű videojel rögzítéséhez szükséges képek száma másodpercenként 25, percenként $25 \cdot 60 = 1500$, óránként $25 \cdot 60 \cdot 60 = 90000$ képkocka. Ezt beszorozva egyetlen kép tárolásához szükséges memóriaigénnyel, a különböző felbontások szerint, az 1. táblázatban szereplő eredményeket kapjuk.

Grafikai felbontás	Inzenzítás (szín-) mélység	Egyetlen kép memóriaigénye	1 óra memóriaigénye	1 nap memóriaigénye
384(H) x 288(V)	8 bites szürkeskálás	108 KB	9 492 MB	222 GB
384(H) x 288(V)	24 bites RGB	324 KB	28 477 MB	667 GB
768(H) x 576(V)	8 bites szürkeskálás	432 KB	37 969 MB	890 GB
768(H) x 576(V)	24 bites RGB	1 296 KB	85 430 MB	2 670 GB

1. táblázat: Egy YUV kép komponensekre bontva

Láthatjuk, hogy a grafikai felbontás, a színmélység és a képformátum függvényében a tárolandó képek memóriaigénye között nagyságrendi különbségek lehetnek (példánkban 1:12). A másik jelentős kérdés az, hogy a keletkező óriási adatmennyiség hogyan kerül feldolgozásra. Hogyan történik az archiválás és hogyan

kell a kezelőszemélyzetnek biztosítani a folyamatos rögzítést? Az analóg technikával kapcsolatos gyakorlat, hogy a keletkező videokazettákat egy meghatározott ideig biztos helyen tárolják. Ez hogyan valósítható meg a digitális eszközökkel? Ezekre a kérdésekre a legtöbb rendszer megoldást kínál, viszont egyik sem nélkülözi a képtömörítési eljárásokat. Ezen algoritmusok nélkül a digitális videó rendszerek meg sem születtek volna.

3.5 JPEG - (Joint Photographic Experts Group)

A legáltalánosabb digitális képtárolási formátum a JPEG már tömörített képállomány. Kiterjesztéseként a .jpeg, .jpg, ritkábban a .jpe használt.

A képen lévő információt veszteségesen is tömörítheti ez a formátum (létezik veszteségmentes is). Bár a tömörítés információveszteséggel jár, akár 10-100× kisebb fájl méret mellett is élvezhető a tömörített kép. Elsősorban fényképek, rajzok tárolására való. A JPEG-ben nem képpontokat tárolnak le, hanem a képet transzformálják a frekvencia-tartományba a DCT-vel (diszkrét cosinus transzformáció).

3.5.1 DCT

A diszkrét koszinusz transzformáció egy Fourier-transzformáció, hasonló a diszkrét Fourier-transzformációhoz, de nem komplex, hanem valós számokon dolgozik.

A tömörítés során a lényegtelen információk elhagyására törekszünk. A kvantálási lépcsők számának csökkentésével (lásd később) jóval kisebb adathalmazt kapunk, miközben a képminőség csak az elfogadható mértékben romlott. Nem használtuk ki azonban az emberi látás néhány fontos tulajdonságát (lásd fent). Ha az adathalmazunkra alkalmazzuk a DCT-t, a kapott új adathalmaz nem lesz nagyobb, mint az eredeti, de ez most a kép spektrális tulajdonságaira lesz jellemző. Az emberi szem azonban a nagyfrekvenciás összetevőkre jóval kevésbé érzékeny, mint a DC-hez közeliekre. Elhagyva a nagyfrekvenciás összetevők kb. 50%-át, az inverz transzformáció után az eredeti kódolt információnak legfeljebb 5%-a veszik el.

A JPEG alapvetően az YUV színteret használja, 4:4:4, 4:2:2, 4:2:0 arányban. Ennek köszönhetően plusz 33-50%-os tömörítést is elérhet.

A JPEG kódolás során a képet 8x8 Pixelből álló blokkokra osztjuk, és ezt a 64 képpontot együtt transzformáljuk. Ezt eredetileg az indokolta, hogy az eljárás kifejlesztésének idején az integrált áramkörös technológia ennyit tett lehetővé. A 8x8-as technika jól bevált, ezért azóta sem módosították. Az MPEG technika a kódolás és a mozgásbecslés során szintén 8x8-as blokkokat használ, de ott a blokkméret növelése igen jelentős számítástechnikai többletigénnyel járna.

Ha egy N mintából álló sorozatot szimmetrikussá teszünk úgy, hogy a mintákat csökkenő index szerint megismételjük, akkor egy páros függvényt kapunk. A függvény Fourier transzformáltjának (spektrumának) N db harmonikus összetevője lesz (0 – N-1). Valamennyi összetevő koszinuszos és valós. Egyszerűsítve ez a DCT. Az N mintából álló sorozat diszkrét koszinusz-transzformáltja:

$$y(k) = \sqrt{\frac{2}{N}} \cdot \alpha_k \cdot \sum_{n=0}^{N-1} x(n) \cdot \cos\left(\frac{2n+1}{2N} \cdot k \cdot \pi\right)$$

$$\alpha_k = 1 \quad \text{ha} \quad k \neq 0 \quad \alpha_k = \frac{1}{\sqrt{2}} \quad \text{ha} \quad k = 0 \quad n, k = 0..N-1$$

Az N db mintát együtt transzformáltuk. Ha az N = 8, akkor 8 féle alapfüggvényt (harmonikus összetevőt) kapunk, amiből a k = 0-hoz tartozó összetevő a DC, és értéke:

$$y(0) = \frac{1}{\sqrt{N}}$$

Ne feledkezzünk meg azonban arról, hogy a képet három (Y, U, V) kétváltozós (x,y) függvény írja le, ezért kétdimenziós DCT-re van szükségünk. Ezt elvégezzhetjük úgy is, hogy két egydimenziós DCT-t egymás után kapcsolunk. A kétdimenziós DCT formula:

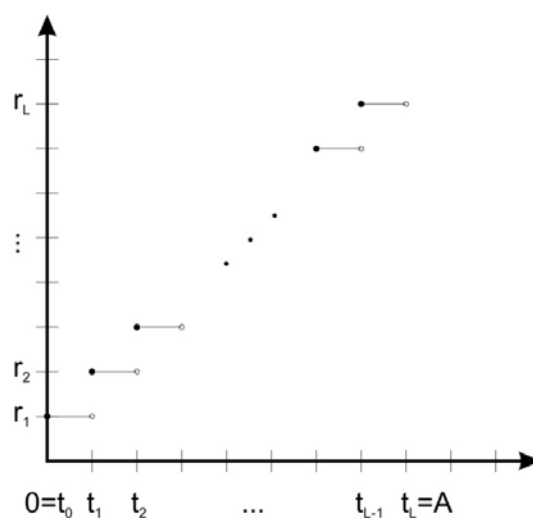
$$y(m, n) = \frac{2}{N} \cdot \alpha_m \cdot \alpha_n \cdot \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} x(k, l) \cdot \cos\left(\frac{2k+1}{2N} \cdot m \cdot \pi\right) \cdot \cos\left(\frac{2l+1}{2N} \cdot n \cdot \pi\right)$$

x(k,l) a bemeneti minta, am, an ugyanúgy alakul, mint az egydimenziós DCT-nél. 8x8 képpontból álló blokkok együttes transzformálásakor 64 féle alapfüggvényt

kapunk, vagyis megkaptuk a DC és a 63 féle koszinuszos összetevő együtthatóit. (A transzformáció szempontjából teljesen mindegy, hogy egy képpontot eredetileg hány biten kódoltunk.) A 64 minta helyett most van 64 másik számunk. Látszólag feleslegesen dolgoztunk. Rendezzük 8x8-as mátrixba az együtthatókat úgy, hogy a mátrix (0,0)-ás eleméhez tartozzon a DC, a (7,7) pozíciójú eleméhez pedig a legnagyobb frekvenciájú összetevő. A DCT tulajdonságából fakadóan az együtthatókat nem kell rendezni. Az együtthatókat kvantáljuk. A még elfogadható minőségromlás mértéke határozza meg, hogy hány kvantálási lépcsőt alkalmazunk. A kvantálás optimalizálható, ha a kvantálási mátrixot az adott képhez illesztjük (adaptáljuk), de akkor azt is át kell vinni.

3.5.2 A kvantálás

A minták értékei még mindig egy folytonos szürkességi érték tartományt fognak át. Ezért ahhoz, hogy egy digitális függvényt alkossunk, a szürkességi értékeket konvertálni (kvantálni) kell diszkrét mennyiségekre. A képfüggvény folytonos fényességértékei és a digitális megfelelőik közötti átalakítást nevezzük kvantálásnak. Eredménye egy képfüggvény, aminek véges sok, diszkrét értéke van, vagyis diszkrét értékű képpontokat kapunk, amelyeket általában egész számokkal reprezentálunk. A kvantálási szintek számának elegendőnek kell lennie ahhoz, hogy érzékelhetők legyenek a finom (an árnyalt) színátmenetek, részletek a képen.

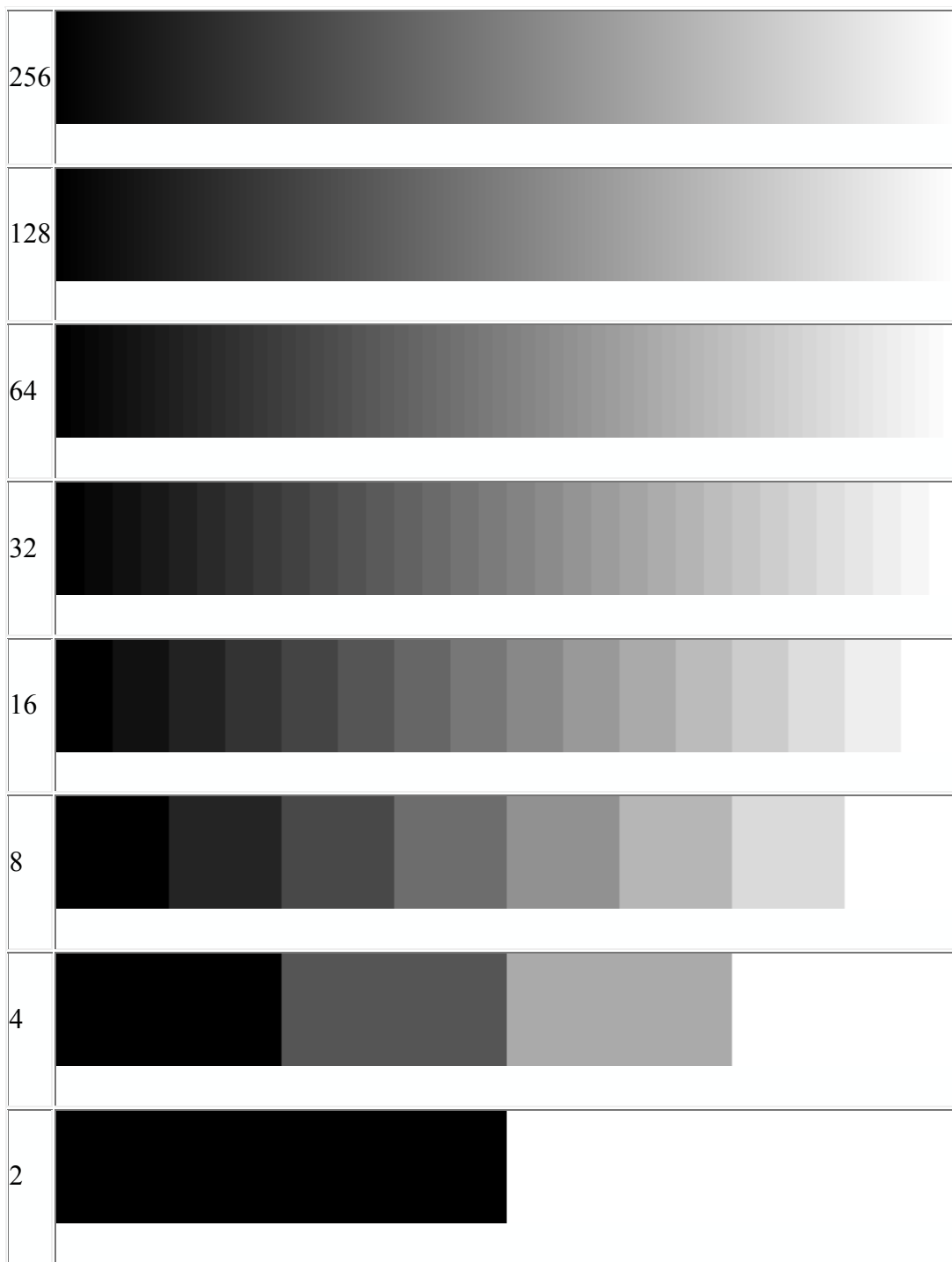


5. ábra: $f(x,y) = z \in \{r_1, r_2, \dots, r_L\}$ kvantálási szintek,
vagy reprezentálási szintek $z = r_k$, ha $z \in [t_k, t_{k+1})$ ($k=1, 2, \dots, L$)

Speciális esetben, ha $t_{k+1}-t_k=\Delta t$ minden k -ra, egyenletes kvantálóról beszélhetünk. Fontos megjegyeznünk, hogy a kvantálás nem invertálható, torzítást okoz.

A digitális képek előállításához használt minták számának megállapítására van megbízható módszerünk, de ez nem igaz a szürkeségi szintek számára. A szürkeségi szintek látható változásának mérése, felbecslése egy eléggé szubjektív folyamat. A tárolás és a végrehajtás miatt, valamint a mintavevéshez használt hardver megfontolásokból a szürkeségi szintek száma jellemzően egy egész kitevőjű kettő hatvány. Amikor egy kép 2^k -on számú szürkeségi értékkel rendelkezik, megszokott gyakorlat, hogy a képre, mint k bites képre hivatkozunk. Például egy kép 256 lehetséges szürkeségi szintű képet mint 8 bites képet említjük.

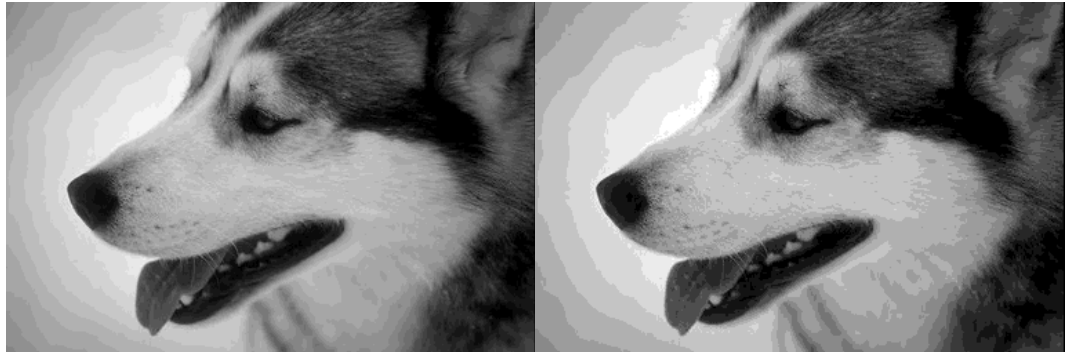
A legtöbb képfeldolgozó eszköz egyenletes kvantálást használ. Ha feltesszük, hogy k egyenlő részre, intervallumra osztjuk fel az értékeket, és b biten ábrázoljuk őket, akkor a szürkeségi szintek száma $k = 2^b$. A képpontok színének ábrázolására a legáltalánosabban, leggyakrabban használt a 8 bit (képpontonként), ez összesen 256 szürkeárnyalatot jelent, 0 a fekete, 255 a fehér ábrázolására. A tapasztalat szerint egyenletes (uniform) kvantálásnál ez általában elegendő. Néha ezeket a 0-1 zárt intervallumba eső számokkal ábrázolják, ebben az esetben szintén a 0 a fekete, és az 1 a fehér. Néhány speciális alkalmazásban, ahol a szürkeségi szint tartomány növelése fontos, a részletesebb ábrázolás is elképzelhető, például az orvosi képeknél 10-12, a csillagászatban 16 bites ábrázolás jellemző.



6. ábra: Szürkeségi szintek

3.5.3 Hamis kontúrok

Ha a kvantálási szintek száma nem elegendő, akkor hamis kontúrok keletkezhetnek (7. ábra).



7. ábra: Hamis kontúrok megjelenése

A hamis kontúrok hatását mérsékelheti egy kis mesterséges zaj:

$$z + \eta \xrightarrow{\text{kvantálás}}$$

ahol η egyenletes eloszlású valószínűségi változó, ezzel „széttörhetjük” a kontúrokat.

3.5.4 A döntési folyamat modellezése

A kvantálás definíció szerint két érték közötti döntést jelent, és így mindig információvesztéshez, és hibákhoz vezet/okoz. Hogy mekkora a hiba, és milyen pontos a "válasz"/választásunk/megoldás, ez a bitek/szintek (nem ua.) számától függ, amelyeken/amelyekkel ábrázolunk.

Ha a képfüggvényünk értéke: $0 \leq f(x,y) \leq A$ és L darab diszkrét értéket szeretnénk (kvantálási szintek).

Hogyan válasszuk meg

- L -et (a kvantálási szintek számát)
- $L+1$ darab döntési értéket: $0=t_0 < t_1 < t_2 < \dots < t_L = A$

- L darab reprezentálási szintet: $0 = t_0 < r_1 < t_1 < r_2 < t_2 < \dots < r_L < t_L = A$

Tehát, hogy hány részre osztjuk fel az intervallumot (L), ezeket pontosan hol vágjuk el (t_i), és a keletkezett új, kisebb intervallumok ábrázolására (reprezentálására) mely értéket (r_i) választjuk egy-egy ilyen intervallumon belül úgy, hogy a kvantálási hiba

$$\epsilon^2 = \sum_j \int_{t_{j-1}}^{t_j} (f - r_j)^2 \cdot p(f) df$$

minimális legyen, ahol

$$\int_0^A p(f) df = 1$$

Ha a t döntési értékek elég sűrűn vannak, akkor

$$\epsilon^2 \approx \sum_j p_j \cdot \int_{t_{j-1}}^{t_j} (f - r_j)^2 df$$

ahol $p_j = p(t_j) - p(t_{j-1})$.

Szélsőérték (minimum) ott lehet, ahol (a derivált 0):

$$\frac{\partial \epsilon^2}{\partial r_j} = 2p_j \cdot \int_{t_{j-1}}^{t_j} (f - r_j) df = 0$$

$r_j = 1/2 (t_{j-1} + t_j)$ felezőpont (általánosan nehéz megoldani).

Ha az eloszlás egyenletes, akkor az egyenletes kvantálás az optimális. Az ehhez tartozó lineáris kvantáló: $[f(x,y)]$, vagyis az egész-rész függvény. Ha az előbb elmondottaktól eltérően a színek eloszlása nem egyenletes, akkor először egyenletes

Az alkalmazott futamhossz-kódolás némiképp eltér a korábban megismerttől. Az értékes együtttható kódja után mindig a követő nullák száma (futási hossza) áll:

Eredeti számsorozat: 2,1,4,3,0,0,0,1,0,0,0,0,0,6,8,1,0,0,0, ...

RLE után: (2,0),(1,0),(4,0),(3,3),(1,5),(6,0),(8,0),(1,3), ...

A számpárokra még alkalmazhatunk változó szóhosszúságú (Huffman) kódolást is, akár bemenő blokkonként változó kódtáblával.

3.5.5 A DCT műveletigénye

Az alábbi táblázatban összefoglaltam, hogy $N \times N$ képpontból álló blokkok esetén a DCT végrehajtásához hány művelet szükséges:

	1 blokk 1 együttthatója		Teljes blokk	
	szorzás	összeadás	szorzás	összeadás
definíció szerint	N^2	N^2	N^4	N^4
$2x(1-D)$	$2N$	$2N$	$2N^3$	$2N^3$
gyors DCT	11 ($N=8$)	29 ($N=8$)	11 N^2	2

2. táblázat: A DTC műveletigénye

4 A szabvány bemutatása, összevetése más videó szabványokkal

4.1 Hogyan lesz az állóképből mozgókép

A mozgókép egymást követő álló képkockákból áll, és ezek gyakran csak kevésbé térnek el az őket megelőző, őket követő képkockáktól; ha csak a változást tároljuk le, akkor a szükséges adatmennyiség nagyságrendekkel csökkenthető.

Az eljárások abban különböznek, hogy milyen módon igyekeznek megtalálni ezen hasonlóságokat és eltéréseket; milyen módon határozzák meg azt, hogy az eredeti képet, vagy pedig az előzőektől való különbséget használják.

Nézzük rendre a manapság használt mozgókép tömörítési szabványokat.

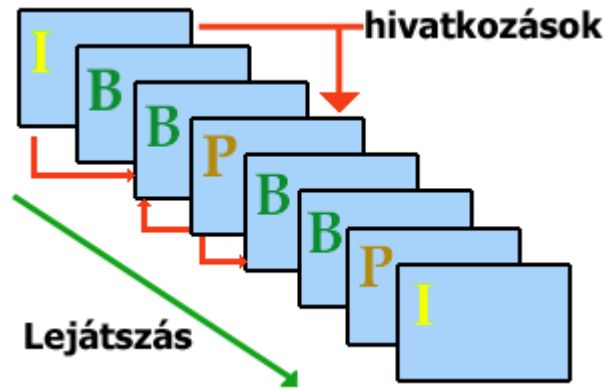
4.2 Az MPEG-1-2-4 rövid bemutatása

4.2.1 MPEG-1

Az MPEG-1 tömörítő eljárás az MJPEG tömörítő eljárásból alakult ki. Azért volt rá szükség, mert az MJPEG még mindig túl nagy adatmennyiséget jelentett. Márpedig a Philips a készülő CDI szabványához nem akart 2 CD-nél többet felhasználni egy filmhez. Ezért kellett keresni egy technológiát, amivel további tömörítést lehet elérni.

A filmeket megvizsgálva rájöttek, hogy rengeteg olyan jelenet van, amiben álló háttér előtt mozognak a szereplők. Kellott keresni tehát egy olyan technikát, amivel ezeket a mozdulatlan háttereket nem kell minden képkockán tárolni.

Megalkották tehát azt a módszert, amivel egy úgynevezett "mester-szolga" kapcsolat jött létre az egyes képkockák közt. Mit is jelent mindez?



9. ábra: A mester szolga viszony ábrázolása

Megalkottak 3 féle képkocka típust, és ezek egymáshoz viszonyított helyzete határozza meg a mester, vagy szolga mivoltukat. Ezek az I, B és P képkockák.

I képkocka:

Az I kockák teljes, MJPEG kódolású képkockák. Itt az egész képkocka van tömörítve és eltárolva.

Látni fogjuk, hogy a B és P kockák nem a teljes képkockát tartalmazzák.

P képkocka:

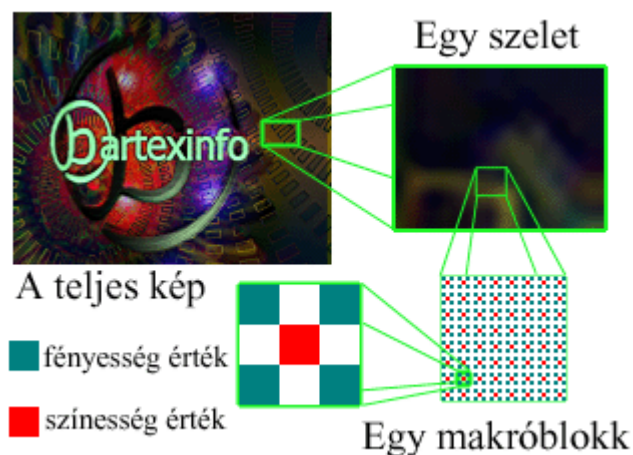
Ez az úgynevezett jószolt képkocka. A tömörítő algoritmus megvizsgálja, hogy az adott képszakaszban milyen mozgások történnek, és ezeket a mozgásokat tárolják el. Amennyiben nem tudja megállapítani az algoritmus a mozgás irányát, úgy a szakasz eleje, és vége közti különbséget tárolják el benne, természetesen tömörítve. Ez bizonyos túréssel történik, ezért a kapott kép soha nem egyezik az eredetivel, mindig csak hasonlít rá, persze olyan mértékben, hogy az ne legyen látható. Vagy legalábbis ne nagyon legyen látható.

B képkocka:

A Bidirectional /kétirányú/ szóból ered ennek a kockának az elnevezése. Ez a képkocka az előtte és utána következő I vagy P kockára hivatkozik, attól függően, hogy melyikre hasonlít jobban a kép adott része.

A B kockák kódolására 4 módszer létezik.

- Előrejöslással: az előző P vagy I kockából jósolják meg a mozgások irányát
- Visszajöslással: az utána következő P vagy I kockából történik a jóslás
- Kétirányú jóslással: az előző két módszert együttesen használják
- Mozgáskompensáció nélkül: ekkor csak az előző, és az utána következő I vagy P képkocka közti különbséget veszik.



10. ábra: Egy kép blokkbontása

Az B és P kockák egymás utáni sűrűségét a tömörítés folyamán meg lehet adni. Az MPEG 1 szabvány rendkívül rugalmas, mivel csak a fájl formátumot és a tömörítő eljárást adja meg. Ez azért jó, mert a számítási teljesítmény növekedésével jobb minőségű algoritmusok is megjelenhetnek, és ezek a kapott minőségre rendkívül jótékony hatással lehetnek.

Az MPEG 1-es eljárás a képkockát többszörösen szétosztja. Először úgynevezett szeletekre szedi szét. Ez azért jó, mert ha megsérül egy szelet, akkor azt egyszerűen figyelmen kívül hagyja a lejátszó program. Ebből is látszik, hogy minél kisebbek a szeletek, annál kevésbé észrevehetőek a hibák, viszont a fájl mérete is nagyobb lesz. A szeleteket is tovább osztották. Így keletkeztek a "makró blokkok". Ezek a makró blokkok 16*16 pixelnyi fényesség értéket, és 8*8 pixelnyi színesség értéket tárolnak.

Az MPEG 1 szabvány megalkotásakor a cél egy olyan tömörítő eljárás volt, ami a CD lemez adatátviteli sebességét vette alapul. Ezért a VideoCD szabvány 150kB/s-os sebességet ír elő, ami egy VHS minőségű videónak felel meg, és egy CD lemezre 74

percnyi anyag fér fel. Mindez köszönhető annak, hogy az MPEG 1 kódolás kb. az 1%-ra tömöríti össze az eredeti tömörítetlen videót. Természetesen az MPEG szabvány tartalmazza az audio sáv tömörítésére vonatkozó részt is. Persze ez a 150kB/s-os sebesség nem jelenti azt, hogy ennél nagyobb átviteli sebességű MPEG 1-es tömörítésű videót nem lehet készíteni. Az MPEG 1 említett rugalmasságának köszönhetően, csak a winchester mérete és a józanész szab határt az MPEG-1 kódolás átviteli sebességének.

4.2.2 MPEG-2

A két óra tartalmú video anyag tömörítés nélküli tárigénye kb. 144 Gigabájt hang nélkül. A CD-lemezekeken használt MPEG-1 tömörítés nem teszi lehetővé az elvárt minőségű DVD lemezek előállítását, ezért a DVD-lemezek alap tömörítési eljárásaként az MPEG-2 tömörítést választották. Ez a tömörítési eljárás azon alapszik, hogy az idő és térbeli ismétlések elhagyhatók, figyelembe véve az emberi szem tulajdonságait is. Az adatfolyamban I kép (képkockán belüli tömörítés), B kép (kétirányú), és P kép (előjelzett) adott sorrendű ismétlődése található meg. A képsorozatokat képcsoportnak (Group Of Pictures - GOP) nevezik. A tömörítési arány változtatható, a bitsebesség 1,5 és 40 Mbit/sec közötti érték lehet. A magasabb bitsebesség azt jelenti a képekhez több bit tartozik, és a tömörítendő információ bonyolultabb lehet. A képcsoportok felépítése is módosítható. MPEG-1 tömörítésnél egy képcsoportba általában 15 kép tartozik, MPEG-2-nél a képek száma 1 és 15 között változhat. Ha csak egy kép van a GOP-ban, az kötelezően I kép, melynek a legrosszabb a tömörítése, de legkevésbé veszteséges, azaz nagyobb bitsebesség kell az átviteléhez. Ha a 133 perces mozifilm DVD5-lemezre tömörítjük, az átlagos MPEG-2 bitsebességet olyan pontosan 3,5 Mbit/sec értéken kell tartanunk, amennyire csak lehetséges. A kódolás több fokozatból áll. Először a jelet komplexitásra értékeli ki, és minden képkockához meghatározzák az optimális bitsebességet (a bonyolultabb képekhez nagyobb bitsebességet, az egyszerűbb részekhez kisebb sebességet rendelnek). Ez az ún. adaptív, változó bitsebességű (VBR, Variable Bit Rate) eljárás. Az MPEG-1 tömörítés bitsebessége állandó (CBR, Constant Bit Rate). Az első fokozat eredménye egy lista (EDL, Encoding Decision List, kód döntési lista), melyet a második fokozatban, a tényleges tömörítéskor használnak fel. A második fokozat valós időben is folyhat, mert a felhasználó egyszerre kódolhatja és dekódolhatja (megtekintéshez) a video adatfolyamot.

A DVD formátum 4:2:0 komponens videó tömörítést használ, mellyel max. 10 Mbit/sec sebesség érhető el. Az MPEG-2-nek köszönhetően így egy lemezen 133 perc videó anyag, háromnyelvű 5.1 csatornás hanganyag és négynyelvű részképek férnek el. Mindez együtt 4,692 Mbit/sec átlagos bitsebességet igényel.

4.2.2.1 Állókép, részkép

Az állóképek MPEG-2 I képként tárolódnak, és általában menü célra használják őket. Az állóképeket hang is kísérheti. A részkép átlapolás a DVD lejátszásakor létrehozott kép, a mögötte lévő videóhoz kapcsolódik. A szöveg nem korlátozott. Bármilyen NTSC vagy PAL felbontású bittérkép szolgálhat részképként. Legfeljebb 32 részkép átlapolás lehetséges (32 nyelven feliratozható a film). A részképek elsődleges feladata a többnyelvű alcímek nyújtása, karaoke szöveg beszúrása, menük, animációk, utasítások és szinkronfeliratok megjelenítése. Minden részkép csoporthoz négy szín választható 16 színű palettáról, és négy árnyalat egy 16 színű készletből.

4.2.2.2 Képméretarány

Az eredeti 35 mm-es filmhez 4:3 képméretarányt (aspect ratio, a vízszintes és függőleges képméret aránya) választottak, melyet a televíziónál alkalmaznak. A filmtechnikában megnőtt a képméretarány. A szélesebb képeket kétféle módon lehet megjeleníteni a tv-n:

Pan/scan módszernél az eredeti képméretből kivágnak egy képernyőnyi ablakot, a levélszekrény (letterbox) módszernél pedig az eredeti képet lekicsinyítik, hogy beférjen a képernyő szélességébe (képernyő alján és tetején fekete csík található). A következő ábrán a bal oldalon 16:9-es és 4:3-as kép látható. A képen látható, hogy ezek a képek hogyan jelennek meg hagyományos tévén (4:3) és 16:9-es megjelenítéssel.



11. ábra: Képek különböző felbontásban

A 16:9 képarányt a széles képernyőjű tv számára fejlesztették ki, és ugyanezt az arányt használja a HDTV is. A DVD-Video képes alkalmazkodni minden típusú képforráshoz függetlenül a képméretaránytól. A lejátszó négy különböző úton állíthat elő kimenetet:

- teljes képkocka (4:3 video 4:3 kijelzőhöz)
- levélszekrény (16:9 video 4:3 kijelzőhöz)
- pan/scan (16:9 video 4:3 kijelzőhöz)
- széles kép (16:9 video 16:9 kijelzőhöz)

4.2.3 MPEG-4

Az MPEG-4 megjelölés egy az MPEG (Moving Picture Experts Group) 1998-ban bemutatott kölcsönös megállapodású audio/videó kódolási standard, amelynek elsődleges célja az alacsony bit gyorsaságú, 4800 bit/sec-tól kb. 4 Mbit/sec-ig, tartalmak kódolása.

Az Mpeg-4 szabványt alkalmazzák leginkább webes média folyamatok átvitelére és CD-k terjesztésére, videó-telefonokhoz és televíziós programok sugárzásához.

Az Mpeg-4 technológia magába foglalja az Mpeg-1 és az Mpeg-2 számos jellegzetességét, kiegészítve azokat olyan új sajátosságokkal, mint a kibővített VRML támogatás 3D megjelenítéshez; tárgy-irányítású összetett fájlokkal (beleértve az audio, videó és VRML tárgyakat), DRM (Digital Right Management, Digitális Jog Menedzsment) támogatással és más interaktív attrakciókkal.

Az Mpeg-4 legtöbb vonatkozásának fejlesztését és azok keresztülvitelét egyéni fejlesztőkre bízta, éppen ezért csak néhány teljesen komplett kivitelezése létezik.

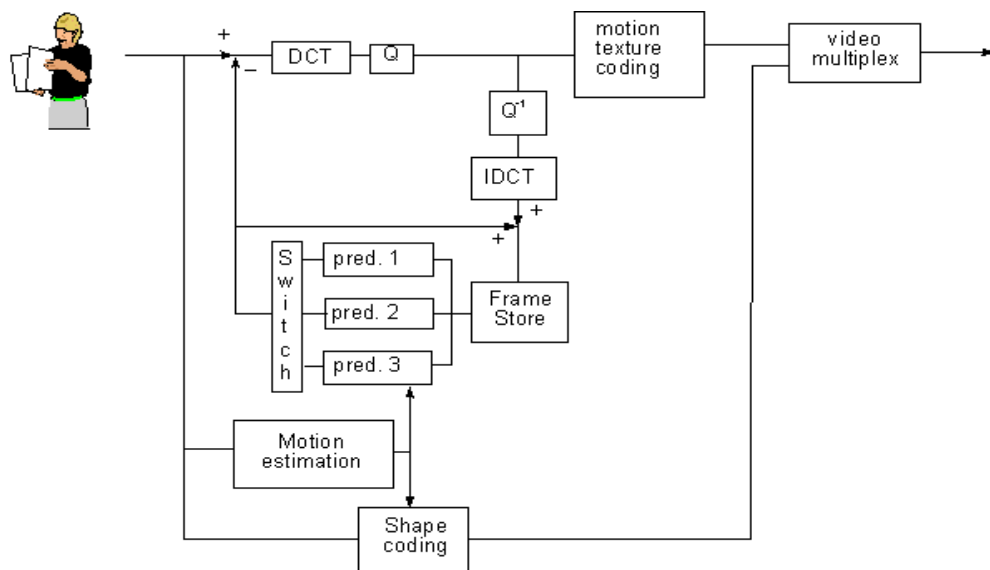
Erre felkészülve a fejlesztők hozzájárultak a "Profiles" (Profilok) elgondolással, ami különböző képességek csoportba foglalását segíti elő.

Az MPEG-4 több egységes elnevezésű "Layer"-ből (Réteg) áll:

1. réteg: a videó és audio szinkronizálását és multiplexelését írja le
2. réteg: tömörítő codec a videó jelekhez
3. réteg: tömörítő codec az érzékelhető audio jelek kódolására
4. réteg: vizsgálati egyeztetésre szolgáló eljárásokat ír le
5. réteg: szoftver szimulációs rendszereket ír le
6. réteg: Delivery Multimedia Integration Framework (DMIF)
7. réteg: H.264 fejlett videó codec

Az MPEG-4 elfogadott jelölése MP4.

A lenti ábra mutatja az MPEG-4 algoritmusának működését, hogy hogyan is készül el a bemeneti képsorozatból a kimeneti képsorozat.



12. ábra: Az alap MPEG-4 Video Coder blokkdiagramja

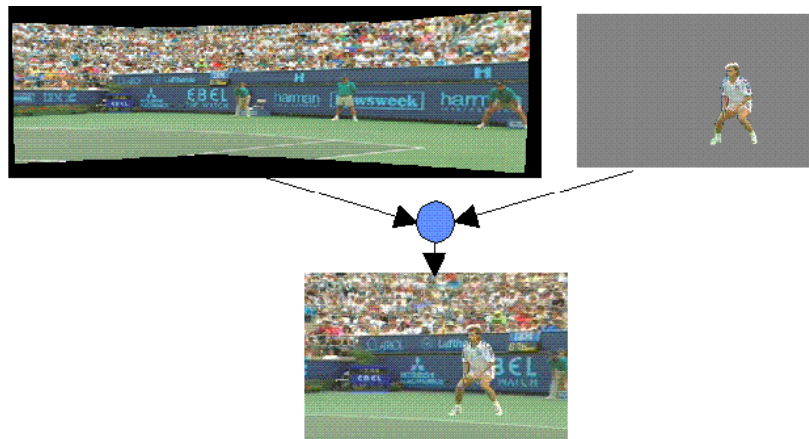
Az alapvető kódoló szerkezet formakódolással jár és mozgáskompenzációval mint ahogy a DCT központú textúra kódolás is

A content-based kódolásának egy fontos előnye az MPEG-4-ben, hogy a tömörítés hatékonyságát jelentősen javíthatják néhány videó sorozatnál azáltal, hogy

megfelelő mozgásvektort alkalmaznak az egyes képsorozatoknál. Ezek a következő lehetőségeket kínálják számunkra:

- Szabványos 8x8 vagy 16x16 pixeles blokk alapú mozgásbecslés és kompenzáció.
- Global Motion Compensation (GMC) az objektumoknak: Egy objektum globális mozgásának encódolásához kevés paramétert kell használnunk. GMC magába foglalja a globális mozgásbecslést, a kép megvetemedésének javítását, a mozgás irányának változását, és a textura kódolását a jóslathibák kiküszöbölésére.
- Quarter Pel Motion Compensation növeli a mozgáskompenzációs pontosságát és csak kis szintaktikus és számítási igénye van. Egy pontos mozgásleírás kisebb jóslathibához vezet és emiatt a vizuális minőség jelentősen javul.

A lenti ábrán láthatunk egy MPEG-4 stream-et, ami egy panoráma-képet használ. Láthatjuk, hogy egy objektumot, ami jelen esetben egy teniszező ember, hogyan helyezik rá a háttérképre. A háttérre helyezendő objektumot előzőleg nyerték ki a képsorozatból. A háttér egy statikus kép. Ez azért fontos, mert ezt a képet csak egyszer, a jelenet elején küldik át a dekódoló készüléknek és az ő dolga egyesíteni a képet az objektummal. A továbbiakban csak azokat információkat kell átküldeni, amik befolyásolják a háttérrel, illetve leírják magát az objektumot. Ezzel jelentős adatt mennyiséget tudunk megspórolni illetve sokkal nagyobb a hibatűrése. Ez azért fontos, mivel a média online sugárzása közben a jelkimaradást (egy adott kép nem érkezik meg egészében a dekóderhez) így kompenzálni tudjuk. A kép szétesése ellen alkalmazhatjuk azt a módszert, hogy a képet progresszív módon kisebb darabokban küldjük át a vevő oldalra.



13. ábra: Példa a videó stream kódolására

A szubjektív értékelési tesztjei az MPEG4-nek megmutatták, hogy ezeknek a technikáknak a kombinációja a többi verzióval összehasonlítva, akár plusz 50 százalékkal is megnövelheti a tömörítési arányt [8].

4.3 MPEG-4 Part 10 AVC / H.264

Az MPEG-4 Part 10 Advanced Video Coding (AVC) / H.264 videó kódolási szabvány a nevéből adódóan egyszerre ISO és ITU szabvány is, az MPEG (Moving Pictures Experts Group) és az ITU VCEG (Video Coding Experts Group) csoportok közös munkájának eredménye. A szabvány valóban rászolgál az „Advanced” jelzőre, hiszen az előtte ismert rendszerekhez képest jelentős hatékonyságot mutat.

Az új eszközök az alábbiak:

- Hatékonyabb intra kódolás
- Fejlettebb mozgásbecslési képesség:
- dinamikus blokkra bontás / fa struktúrájú makro blokk szegmentálás: 16x16-tól lefelé egészen 4x4-es blokkméretig
- $\frac{1}{4}$ képpont pontosságú mozgásbecslés

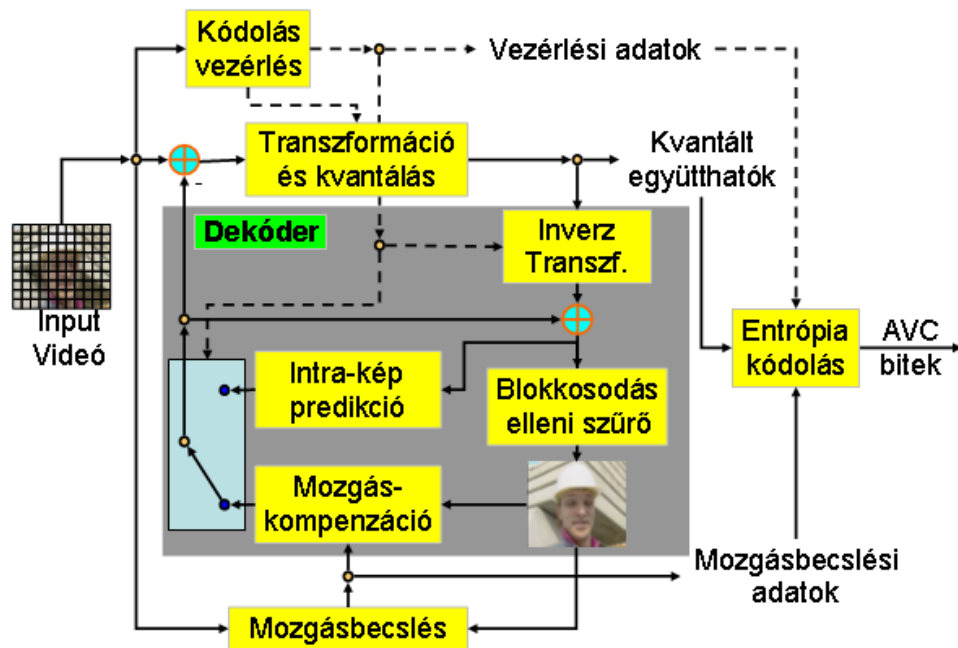
- Több referenciakép használata
- Súlyozott kétirányú predikció
- Kontextus adaptív VLC és aritmetikai kódolás
- Egész értékű transzformáció
- Beépített blokkosodás elleni adaptív szűrés
- Hálózati környezetre való adaptálhatóság

A fenti eszközöknek köszönhetően jelentős bitsebesség megtakarítás (akár 50%) tapasztalható a többi szabványhoz képest. Ugyanakkor ez a komplexitás növekedésével is jár a korábbi módszerekhez képest, a kódolás 3-szor, a dekódolás 2-szer nagyobb erőforrás igényű a többi szabványhoz képest.

4.3.1 Az AVC forráskódolási algoritmusa

A kódolás menetét az alábbi ábra mutatja be. Látható, hogy egy szokásos blokk alapú transzformációs kódolóval van dolgunk, azonban a különbség is szembetűnő. A transzformációs rész nem az eddig elterjedt DCT, hanem egy DCT bázisfüggvényeihez hasonló egész értékű transzformáció, amely eredetileg 4x4-es blokkokra működik, a FRext kiterjesztésben azonban már 8x8-as egész értékű transzformáció is megtalálható.

A transzformációs kódolás nem a képen, hanem az intra-, illetve inter (referenciaképpel való) predikció utáni becslési hibán történik. A blokk alapú predikció és kódolás miatt a dekódolt referenciaképek egy blokkosodás elleni szűréssel kaphatóak meg, ez a szűrés további minőségjavulást eredményez.



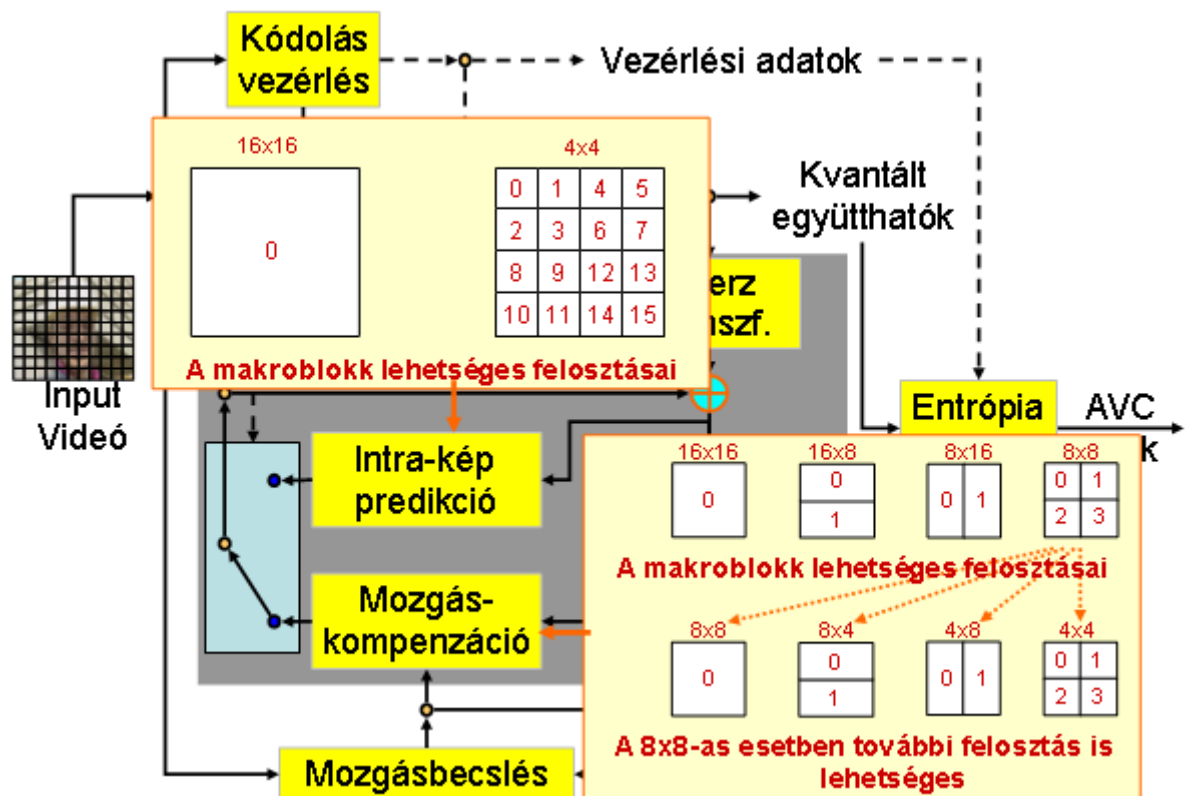
14. ábra: Az MPEG-4 AVC Video Coder blokkdiagramja

A mozgásbecslés hatékonyabb predickiót tesz lehetővé az eddig ismert rendszerekhez képest az alábbi eszközöknek köszönhetően:

- $\frac{1}{4}$ képpont pontosságú mozgásvektor
- Változó predickiós blokkméret (fa struktúrájú makro blokk szegmentálás):
 - 16x16-os terület tovább osztható részblokkokra
 - A 8x8-as részblokk tovább osztható
 - A részblokkok becslése egyedileg történik
- Több referenciaképes predickió:
 - Egy képhez több referenciakép is tartozhat, azonban egy blokknak legfeljebb két referenciája lehet, ezért az 1 vagy 2 referenciakép blokkonként kiválasztható.
 - B típusú kép is lehet referencia
 - B típusú blokk becslésekor a jövőbeli és múltbeli predickiók súlya az $\frac{1}{2}:\frac{1}{2}$ arány helyett $\alpha:(1-\alpha)$ is lehet

A mozgásbecslésben nem érintett makró blokk képpontjaira is lehetséges a predikció:

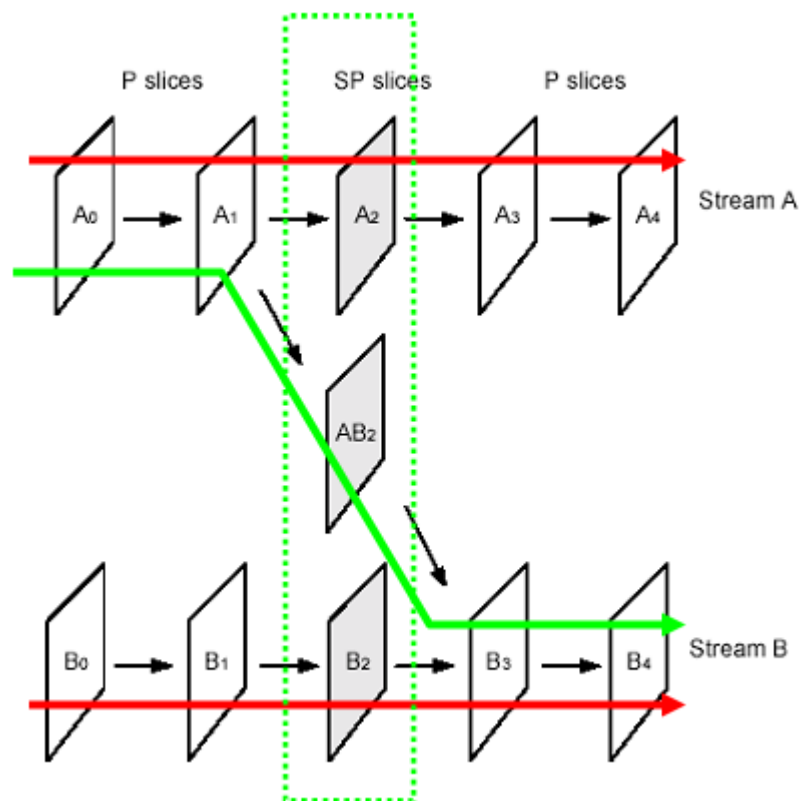
- 16x16-os intra predikció: a makró blokk becslhető a felső és balra eső szomszédos képpontokból.
- 4x4-es intra predikció: a makró blokkot 16 darab 4x4-es négyzetre osztjuk, és ezek a négyzetek becslhetőek a felső és balra eső szomszédos képpontokból.
- A becslést a blokkosodás elleni szűrés előtt kell elvégezni.
- Az intra típusú predikció és a mozgásbecslés (inter típusú predikció) helyét a kódolóban az alábbi ábra szemlélteti [9].



15. ábra: Az MPEG-4 AVC makró blokk felosztása

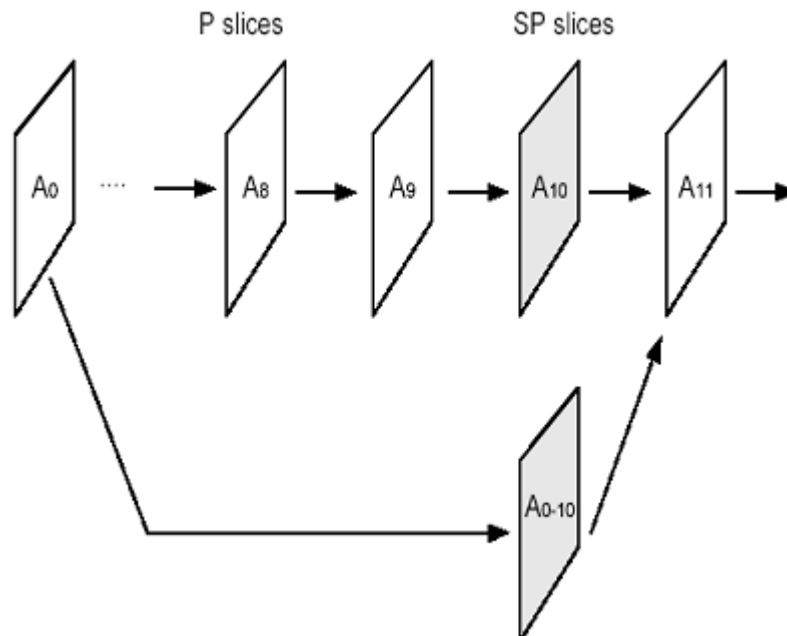
Az AVC érdekessége még a bitfolyamon belül, illetve kép különböző paraméterű bitfolyam közötti átkapcsolást lehetővé tevő képek (Switching P/I). Az SP képszelet olyan képszelet, amely I és P típusú makro blokkokat használ, de azért különböztetjük meg a P típusú képszelettől, mert redundáns módon egy olyan tartalmat ír le más referenciaképek használatával, amely képszelet tartalmát a bitfolyamban előzőleg egy másik képszelet már megadott. Hasonlóan az SI képszelet egy I típusú képszelet, amely redundánsan szerepel a bitfolyamban.

Az ilyen jellegű képek segítséget nyújtanak akkor, ha két bitfolyamunk van ugyanazon tartalommal, de más reprezentációban (pl. bitsebesség, képfrekvencia). A StreamA-ból StreamB-be átkapcsoláskor azoknál a képeknél, ahol az átkapcsolást engedélyezzük, beteszünk StreamA-ba egy harmadik képet, amely referenciája StreamA-ban van, de paramétereit tekintve StreamB képeihez jelent átmenetet. Ezt az esetet mutatja be az alábbi ábra.



16. ábra: Streamek közötti váltás

Azonos bitfolyam távoli pozíciójára ugráskor hasonlóan járunk el. Ebben az esetben egy képszelet többféle képszeletből is előállhat, például nemcsak egy P típusú képszelet, hanem egy SP típusú képszelet is leírhatja ugyanazt a képszelet tartalmát.



17. ábra: Hibajavítás

Ez a funkció azonos bitfolyamon belüli gyorssterkerést, keresést, hatékonyabb véletlen hozzáférést, vagy hibavédelmet is szolgálhat. A hibavédelmet például jól szemlélteti a fenti ábra. Ekkor az A11 kép nem lenne előállítható akkor, ha az A0 még megérkezik a dekóderbe, de az A1...A10 képek közül valamelyik referenciakép átviteli hiba miatt nem érkezik meg, és így a referenciaképek láncolata megszakad. Ekkor azonban az A0-10 SP típusú képpel lehetővé válik A10 rekonstrukciója (ha A0 megvan) és ezután A11 becslése is (16. ábra). Ugyanilyen módon SI típusú képeket beszúrva lehetőség van az adott kép predikciós láncolattól mentes dekódolására is, értelemszerűen ekkor nagyobb sáv szélesség felhasználása árán.

A többlet SP és SI kép beszúrásával a tároláskor redundancia lép fel, de a küldés során nem feltétlenül kell a képszeleteket redundánsan küldeni.

4.3.2 A H.264/MPEG-4 AVC profiljai

Az AVC eredetileg 3 profilt tartalmaz, három alapvető alkalmazási területet érintve.

4.3.2.1 Alap (Baseline) profil:

Videó konferencia, vezeték nélküli (Wireless) átvitel

Eszközök:

- I és P képtípusok (nincs B)
- Váltott soros és progresszív
- Kép-félkép kódolás képenként meghatározható
- In-loop deblocking filter
- 1/4-képpontos mozgáskompensáció
- Fa struktúrájú szegmentáció 16x16 blokkmérettől 4x4 blokkméretig
- VLC-alapú entrópiakódolás (CAVLC)
- Hibavédelem céljából
- FMO (Flexible Macroblock Ordering)
- Képszelet csoportok (slice group) száma: 1..8
- Redundáns képszeletelés

4.3.2.2 Fő (Main) profil

Főleg műsorszórás céljából

- Minden Baseline funkciót tartalmaz a hibavédelem célú eszközök kivételével
- A Baseline funkciókon túl az alábbi eszközök is:
- B típusú képek
- CABAC (context-adaptive binary arithmetic coding)

- Kép-félkép kódolás MB-onként meghatározható
- P és B típusú mozgásbecslésnél súlyozott átlagolás

4.3.2.3 X profil

Főleg streaming céljából

- Minden Baseline eszközt tartalmaz, ezen felül:
- B típusú képek
- Még több hibavédelmi funkció: adat particionálás
- Átváltás másik bitfolyamra/gyorstekerés: SP/SI típusú képek is
- Hálózati adaptációs képesség

Az AVC hálózati adaptációs rétege (NAL – Network Adaptation Layer) képes a folyamat ún. NAL egységekre szervezni. Egy egység egyúttal valamilyen kódolási egységet tartalmaz, célszerűen pl. egy képszeletet. A NAL réteg egy bájt-folyamot definiál, ahol HDLC-szerű keretezést használt bájt rendszerű start kóddal. Egy NAL egység start kóddal és egy RBSP típusú (Raw Byte Sequence Payload) tartalommal áll, amely szükség esetén tartalmaz olyan extra jelzőbájtokat (Emulation Prevention Byte, értéke 0x03), amely biztosítja, hogy a start kód (3 bájt, értéke 0x000001) ne forduljon elő a tartalmi részben. Ez a technika a csatornahibák detektálására, illetve a nem hibás NAL egységek megtalálására hatékony.

Amikor a jelenlegi munkahelyemen elkezdtem dolgozni, egy olyan csapatba kerültem ahol a H.264.-es szabvány implementálásán kezdtek dolgozni. Kezdetben annyit tudtam, hogy a fejlesztés C nyelven folyik, amit nagyon jól sikerült elsajátítanom az egyetemen. Számomra a gondot nem is maga a forrásprogram megírása okozta, hanem az a tény hogy maga a szabvány csak ajánlás. Az implementáció mikéntje a fejlesztőcsapaton múlik. Olyan tények természetesen meg voltak adva, hogy például a kibővített Huffman táblát kell használni, és hogy az hogyan néz ki, de minden más rajtunk múlott. Milyen profilt használunk, a választott profil milyen eszközöket tartalmaz. A profil választásánál, azt vettük figyelembe, hogy a mi feladatunk az

interneten történő műsorszóráshoz készíteni enkódert és dekódert. Erre az Alap (Baseline) profil a legalkalmasabb, mivel a hibatűrése még akkora hogy nem megy a képminőség rovására.

A csapatot két részre osztottuk. Az egyik fele a dekódert a másik fele pedig az enkódert fejlesztette. Én voltam az, aki mind a kettőbe fejlesztettem, mivel feladatul a CAVLC (Context-adaptive variable-length coding) implementálását kaptam. Nézzük meg egy kicsit részletesebben mit is jelent ez a teljesség igénye nélkül.

4.3.3 CAVLC példa

A CAVLC gyakorlatilag a 4X4-es vagy 2X2-es blokk zig-zag bejárása utáni tömböt kapja meg paraméterül [10]. A feladat gyakorlatilag ennek a tömb tartalmának a binarizálása. Nézzük meg, hogyan tudjuk ezt elvégezni lépésenként:

1. A perdikció, transzformáció és a kvantálás után a blokkok jellemzően ritkák (legnagyobb részt nulla értékeket tartalmaznak)

2. A zig-zag bejárás után a tömb elején találhatóak a legnagyobb értékű nem nulla együtthatók (non-zero coefficients vagy coeff), amit gyakran 1-esek sorozata ("Trailing 1s" vagy "T1s") követ.

3. A nem nulla együtthatók számát a szomszédos blokkokéhoz viszonyítjuk, még hozzá a T1s/coeff táblázat alapján, ami magában a szabványba található, és gyakorlatilag ez alapján kódoljuk ezeket az együtthatókat.

4. A nem nulla együtthatókhöz tartozó kódtáblázatban a nagyobb együtthatókhöz kisebb, a kisebb együtthatókhöz pedig nagyobb kódok tartoznak, ezzel segítve a hatékony tömörítést. Egy egyszerű példán keresztül nézzük meg az enkódolás és a dekódolás folyamatát.

Vegyünk egy 4x4-es blokkot:

0	3	-1	0
0	-1	1	0
1	0	0	0
0	0	0	0

Zig-zag bejárás után a következő sorozatot kapjuk:

0,3,0,1,-1,-1,0,1,0,0,0,0,0,0,0

TotalCoeffs (a nem nulla együtthatók száma) = 5 (az összeszámolást a legmagasabb indexű nem nulla együtthatótól kell kezdeni a legkisebbig, azaz a 0. indexig)

TotalZeros (a legmagasabb indexű nem nulla együttható előtti nullák darabszáma) = 3

T1s = 3 (a leghosszabban egymást követő egyesek „futáshossza”)

Ha megvannak ezek az értékek, elkezdődhet az enkódolás folyamata:

Elem	Érték	Kód
coeff_token	TotalCoeffs=5, T1s=3	0000100
T1 sign (4)	+	0
T1 sign (3)	-	1
T1 sign (2)	-	1
Level (1)	+1 (use Level_VLC0)	1
Level (0)	+3 (use Level_VLC1)	0010
TotalZeros	3	111

run_before(4)	ZerosLeft=3; run_before=1	10
run_before(3)	ZerosLeft=2; run_before=0	1
run_before(2)	ZerosLeft=2; run_before=0	1
run_before(1)	ZerosLeft=2; run_before=1	01
run_before(0)	ZerosLeft=1; run_before=1	Kódolását el kell hagyni.

Tehát ennek a blokknak a CAVLC által előállt bitsorozata:

000010001110010111101101

A bitsorozat dekódolása:

A kimeneti tömb lényegében ugyan azokból a táblázatokból áll vissza, amelyek alapján az enkódolást végeztük. Haladunk végig a bitsorozaton és keressük az értékeket a következő módon:

Kód	Elem	Érték	Output tömb
0000100	coeff_token	TotalCoeffs=5, T1s=3	Még üres
0	T1 sign	+	1
1	T1 sign	-	-1, 1
1	T1 sign	-	-1, -1, 1
1	Level	+1	1, -1, -1, 1
0010	Level	+3	3, 1, -1, -1, 1
111	TotalZeros	3	3, 1, -1, -1, 1
10	run_before	1	3, 1, -1, -1, 0, 1
1	run_before	0	3, 1, -1, -1, 0, 1
1	run_before	0	3, 1, -1, -1, 0, 1
01	run_before	1	3, 0, 1, -1, -1, 0, 1

Látszik, hogy a dekóder által visszaállított tömbben a nem nulla együtthatók előtti nullák száma 2, de a TotalZeros értéke pedig 3. Ebből az következik, hogy a tömb nulladik indexére be kell szúrunk a kimaradt nullát, és mivel a tömb 16 elemű ezért az utolsó nem nulla együttható utáni indexekre is nullákat kell illesztenünk. Így kapjuk meg a kiindulási tömböt:

0, 3, 0, 1, -1, -1, 0, 1,0,0,0,0,0,0,0,0

5 Összegzés

Láthatjuk, hogy a manapság oly népszerű digitális média milyen fejlődésen ment keresztül az elmúlt évek folyamán. Mindenki találkozik a népszerű DVD formátumú filmekkel, a mozikban élvezzük a Dolby Surround hangzást, de senki nem gondol bele abba, hogy tulajdonképpen milyen technikai háttérrel követel ez meg.

Dolgozatomban sikerült bemutatni, hogy a H.264-es szabvány milyen technikai újításokat hozott az előző szabványokhoz képest. A benne megalkotott profilok milyen széles körben alkalmazhatóak, és milyen hatékonyságú tömörítést sikerült elérni.

A H.264-es szabvány irányelveit már munkám során sikerült elsajátítanom, de ezen dolgozat megírása révén sikerült még alaposabb tudásra szert tennem. Jövőben munkám során, szeretnék továbbra is codec írással foglalkozni, mivel ez egy alkotói munka. Önálló gondolkodást, kreativitást és alapos tudást kíván meg.

6 Irodalomjegyzék:

- [1] <http://www.chiariglione.org/mpeg/standards/mpeg-4/mpeg-4.htm>
- [2] http://www.john-wiseman.com/technical/MPEG_tutorial.htm
- [3] <http://www.di.univr.it/documenti/OccorrenzaIns/matdid/matdid279434.pdf>
- [4] <http://www.chiariglione.org/mpeg/standards/mpeg-2/mpeg-2.htm>
- [5] <http://www.chiariglione.org/mpeg/standards/mpeg-1/mpeg-1.htm>
- [6] <http://dspr.com/www/technology/technology.htm>
- [7] Sikos László: Videoszerkesztés házilag, BBS_INFO Kft., 2006. szeptember
- [8] A MULTI-VIEW VIDEO CODEC BASED ON H.264 - *Cagdas Bilen, Anil Aksay, Gozde Bozdagi Akar* - Electrical and Electronics Engineering Department, Middle East Technical University Ankara, 06531, Turkey
- [9] FAST H.264 MOTION ESTIMATION WITH BLOCK-SIZE ADAPTIVE REFERENCING (BAR)
Changsung Kim, Siwei Ma and C.-C. Jay Kuo
Department of Electrical Engineering and Integrated Media Systems Center
University of Southern California, Los Angeles, CA 90089-2564
- [10] **H.264 and MPEG-4 Video Compression - Video Coding for Next-generation Multimedia**
Iain E. G. Richardson
The Robert Gordon University, Aberdeen, UK