

# **A MULTIMÉDIA TECHNOLÓGIÁK ALAPJAI**

## **Előadásvázlat**

*Mócsai Tamás*

BME Hálózati Rendszerek és Szolgáltatások  
Tanszék

2015.

# Videó bitsebesség csökkentés alapjai (ismétlés)

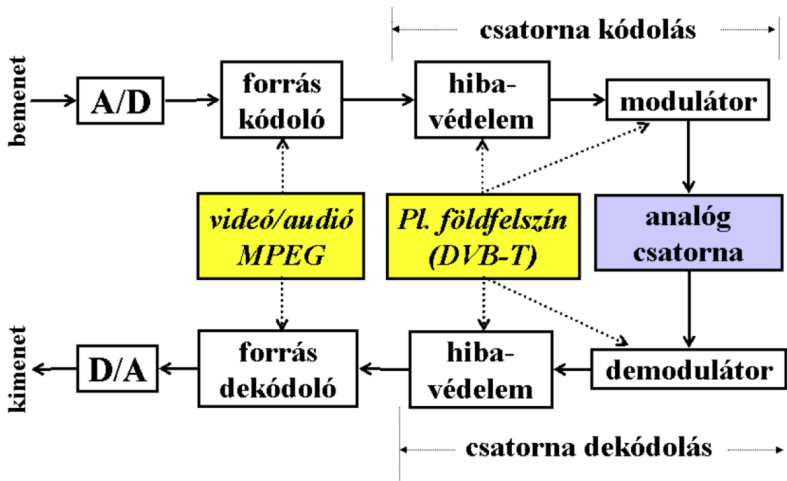
## Redundancia típusai

- A természetes mozgókép redundáns (statisztikus redundancia):
  - térben (intra-frame): egy képen belül a szomszédos pixelek,
  - időben (inter-frame): a szomszédos képkockák hasonlóak
- Észlelési redundancia: a videojel a HVS számára kevésbé, vagy nem észlelhető komponenseket is tartalmaz

## A HVS tömörítés szempontjából lényeges tulajdonságai

- A világosságjelre (fekete-fehér képtartalom) a látásunk térbeli felbontóképessége 3-5-ször nagyobb mint a színekre
- Álló és lassan változó képtartalomra a felbontásigényünk nagyobb
- Álló és lassan változó képtartalom esetében érzékenyebbek vagyunk a képminőség változásaira
- Hirtelen képváltás vagy gyors mozgás esetén kevésbé zavaró a gyengébb képminőség

# Digitális csatorna modell





## Csatornakódolás

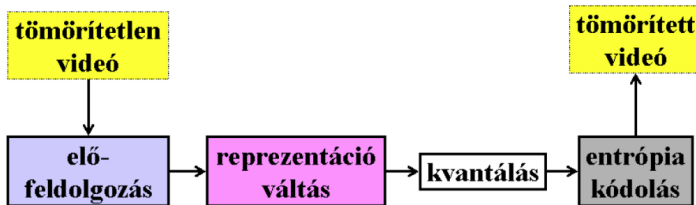
- A csatorna átviteli tulajdonságait véve választunk hibavédelmi algoritmust és modulációs eljárást

## Forráskódolás

- Figyelembe vesszük a forrás és a nyelő tulajdonságait: csökkentjük a forrás redundanciáját, a nyelő számára szükséges információt hagyjuk meg, úgy hogy a nyelő (HVS) "ne vegyen észre semmit"
  - Reprezentáció váltás: Az új reprezentációs térben kevesebb redundancia: pl. predikció, transzformációs kódolás, mozgás kompenzáció
  - Irreverzibilis kódolás: Az ábrázolás pontosságának csökkentése, a "lényegtelen" részek eltávolítása: pl: kvantálás, térbeli/időbeli alul-mintavételezés
  - Reverzibilis kódolás: Hatékonyabb kód-hozzárendelés, a statisztikai redundanciát csökkenti: pl: változó szóhosszúságú kódolás (VLC), futamhossz kódolás (RLC)



# Video forráskódolás modell



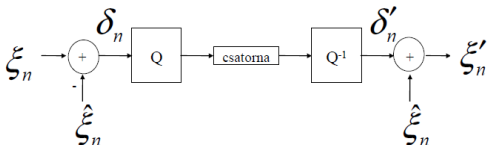
- előszűrés
- formátum konverzió

- predikció
- transzformáció
- részsávós kódolás

- futamhossz kódolás
- változó szóhosszúságú kódolás



# Differenciális kvantálás



- $\xi_n$ : kvantálandó forrás
- $\xi'_n$ : kvantált jel
- $\hat{\xi}_n$ : forrásminta becslése
- $\delta_n$ : differenciális jel
- $\delta'_n = \delta_n + \epsilon_n$ : kvantált differenciális jel
- $\xi'_n = \xi_n + \epsilon_n$ : kvantálás hatása additív zaj
- Eredő SNR :  $SNR_p = \frac{\mathbb{E}[\xi_n^2]}{\mathbb{E}[\epsilon_q^2]}$
- $SNR_p = \frac{\mathbb{E}[\xi_n^2]}{\mathbb{E}[\epsilon_q^2]} = \frac{\mathbb{E}[\xi_n^2]}{\mathbb{E}[\delta_n^2]} \frac{\mathbb{E}[\delta_n^2]}{\mathbb{E}[\epsilon_q^2]}$
- **Predikciós nyereség:**  
 $G_p = \frac{\mathbb{E}[\xi_n^2]}{\mathbb{E}[\delta_n^2]}$
- Optimális kvantáló SNR:  
 $SNR_q = \frac{\mathbb{E}[\delta_n^2]}{\mathbb{E}[\epsilon_q^2]}$
- $SNR_p = G_p SNR_q$



# Lineáris predikció (1D jelre)

- Cél a differenciális jel  $\delta_n$  teljesítményének minimalizálása
- $\min : \mathbb{E}[|\xi_n - \hat{\xi}_n|^2]$  (várható érték négyzet (LMS) értelemben)
- Legyen  $\hat{\xi}_n = \mathbf{a}^T \boldsymbol{\xi}_{n-1}$
- ahol  $\mathbf{a} = [a_1, \dots, a_N]$  és  $\boldsymbol{\xi}_{n-1} = [\xi_{n-1}, \dots, \xi_{N-1}]$
- $\mathbb{E}[|\xi_n - \hat{\xi}_n|^2] = \mathbb{E}[\xi_n^2 - 2\xi_n \mathbf{a}^T \boldsymbol{\xi}_{n-1} + \mathbf{a}^T \boldsymbol{\xi}_{n-1} \boldsymbol{\xi}_{n-1} \mathbf{a}]$
- Használjuk fel, hogy a forrás autokorrelációja:  
 $r_\xi(m) = \mathbb{E}[\xi_n \xi_{n-m}]$



$$\begin{aligned}
 \mathbb{E}[|\xi_n - \hat{\xi}_n|^2] &= \\
 &= \mathbb{E}[\xi_n^2 - 2\xi_n \mathbf{a}^T \boldsymbol{\xi}_{n-1} + \mathbf{a}^T \boldsymbol{\xi}_{n-1} \boldsymbol{\xi}_{n-1} \mathbf{a}] = \\
 &= r_\xi(0) - 2\mathbf{a}^T \begin{bmatrix} r_\xi(1) \\ r_\xi(2) \\ \vdots \\ r_\xi(N) \end{bmatrix} + \mathbf{a}^T \begin{bmatrix} r_\xi(0) & r_\xi(1) & \dots & r_\xi(N-1) \\ r_\xi(1) & r_\xi(0) & \dots & r_\xi(N-2) \\ \vdots & \vdots & \ddots & \vdots \\ r_\xi(N-1) & r_\xi(N-2) & \dots & r_\xi(0) \end{bmatrix} \mathbf{a}
 \end{aligned}$$





$$\mathbf{r} = \begin{bmatrix} r_{\xi}(1) \\ r_{\xi}(2) \\ \vdots \\ r_{\xi}(N) \end{bmatrix}$$

$$\mathbf{R} = \begin{bmatrix} r_{\xi}(0) & r_{\xi}(1) & \dots & r_{\xi}(N-1) \\ r_{\xi}(1) & r_{\xi}(0) & \dots & r_{\xi}(N-2) \\ \vdots & \vdots & \ddots & \vdots \\ r_{\xi}(N-1) & r_{\xi}(N-2) & \dots & r_{\xi}(0) \end{bmatrix}$$

$$\min : \mathbb{E}[|\xi_n - \hat{\xi}_n|^2] = r_{\xi}(0) - 2\mathbf{a}^T \mathbf{r} + \mathbf{a}^T \mathbf{R} \mathbf{a}$$

$$\frac{\partial \mathbb{E}[|\xi_n - \hat{\xi}_n|^2]}{\partial \mathbf{a}} = -2\mathbf{r} + 2\mathbf{R}\mathbf{a} = 0$$

$$\mathbf{a} = \mathbf{R}^{-1} \mathbf{r}$$



# Maximálisan elérhető predikciós nyereség

Adott  $N$  esetén

$$\max(G_p) = \frac{r_\xi(0)}{r_\xi(0) - \mathbf{r}^T \mathbf{R} \mathbf{r}}$$

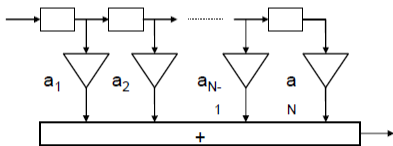
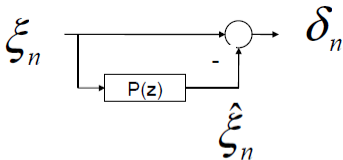
ha  $N \rightarrow \infty$ :

$$\lim_{N \rightarrow \infty} G_p = \frac{1}{\gamma_\xi}$$

Spektrális laposság:  $\gamma_\xi = \frac{\sqrt[N]{\prod_{k=0}^{N-1} S(k)}}{\frac{1}{N} \sum_{k=0}^{N-1} S(k)} \leq 1$ ,

ahol  $S(k)$  a  $\xi$  jel teljesítménysűrűség spektruma.

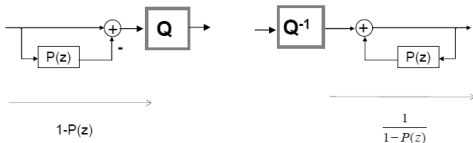
A lineáris prediktor egy FIR szűrő



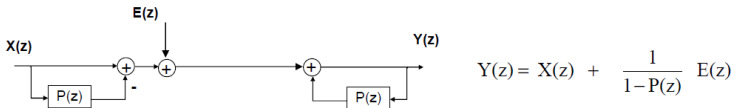
$$P(z) = \sum_{i=1}^N a_i z^{-i}$$



## Előrecsatolt differencia képzés



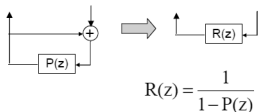
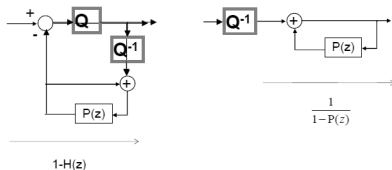
## Kvantálás hatása, mint additív zaj



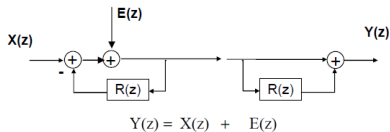


# Hátracsatolt differencia képzés

Kvantáló a hurkon belül



Kvantálás hatása, mint additív zaj

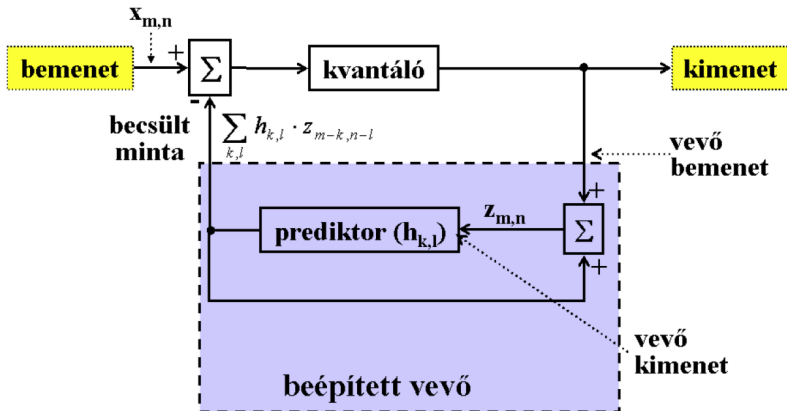




# Prediktív kódolás képtartalomra

- Alkalmazható egy adott kép pixeljei között (térbeli predikció), illetve az egymást követő képek között (időbeli predikció)
- Az  $x_{m,n}$  minta közvetlen kódolása helyett egy szomszédos (általában szintén becsült) mintához képesti különbséget kódoljuk
- A becsült mintát az  $x_{m,n}$  minta környezetében lévő minták lineáris kombinációjaként állítjuk elő
- A vevő oldalon csak a különbségi minták (és a lin. komb. együtthatók) állnak rendelkezésre
- A kvantálási hiba terjedése akkumulálódásának elkerülése érdekében a kódoló tartalmazza az idealizált vevő oldali dekódert is
- A becslést az inverz kvantált (dekódolt) minták alapján végezzük

# Prediktív kódoló blokkvázlata

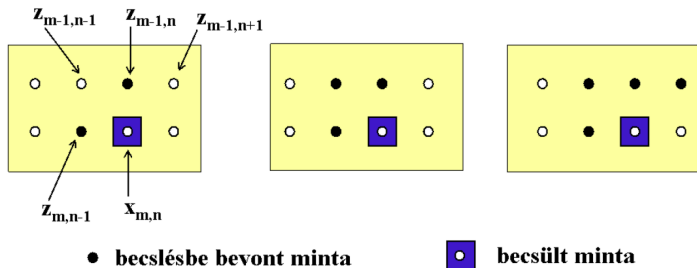




## Optimális együtthatók

- Az optimális együtthatók kiszámítása a hiba négyzetes várhatóérékének (Mean Square Error - MSE) minimalizálásával történik (elméletben)

A leggyakoribb fix együtthatós, térbeli (2D) predikciós típusok







# Transzformációs kódolás I.

---

- A természetes képek pixelei közötti korreláció jelentős
- A transzformációs kódolás általában blokkalapú: egy  $N \times N$ -es blokkon hajtjuk végre
- Célunk a jel reprezentálása egy alkalmasan megválasztott koordináta rendszerben, melyben a redundancia kisebb, így hatékonyabban ábrázolható
- A transzformáció lineáris algebrai művelet: A minták nem átlapolódó  $N \times N$ -es blokkjait a transzformáció mátrixával szorozzuk
- A transzformáció után kapott együtthatók közötti korreláció általában kisebb mint az eredeti minták között
- A kódolás további lépései a transzformált tartományban történnek



## 1D transzformáció

- Bemenő jelvektor:  $\mathbf{x} = [x(0) \ x(1) \ \dots \ x(N-1)]^T$
- Kimenő jelvektor (koefficiens vektor):  
 $\mathbf{y} = [y(0) \ y(1) \ \dots \ y(N-1)]^T$
- A jelvektor és koefficiens vektor kapcsolata:
  - $\mathbf{y} = \mathbf{A}\mathbf{x}$ , ahol  $\mathbf{A}$  a transzformáció mátrixa, és
  - $\mathbf{x} = \mathbf{B}\mathbf{y}$ , ahol  $\mathbf{B}$  az inverz transzformáció mátrixa
- Ha a transzformáció mátrixa unitér ( $\mathbf{A}^H \mathbf{A} = \mathbf{A} \mathbf{A}^H = \mathbf{E}$ ), akkor  $\mathbf{B} = \mathbf{A}^{-1} = \mathbf{A}^H$
- Ha a transzformáció mátrixa ortogonális (ekkor elemei valósak), akkor  $\mathbf{A}^T \mathbf{A} = \mathbf{A} \mathbf{A}^T = \mathbf{E}$ , akkor  $\mathbf{B} = \mathbf{A}^{-1} = \mathbf{A}^T$ : az  $\mathbf{A}$  mátrix sorvektorai ortonormált bázisrendszert feszítenek ki, ez a transzformáció bázisa
- A képtömörítésben alkalmazott transzformációs kódolások unitér, illetve ortogonális transzformációkat alkalmaznak



## 1D transzformáció

- Unitér transzformáció esetén az inverz transzformáció felírható a

$$\mathbf{x} = \sum_{k=0}^{N-1} y(k) \mathbf{a}_k^*$$

alakban, ahol  $\mathbf{a}_k^*$  az  $\mathbf{A}^H$  mátrix oszlopai, és ezek a transzformáció **bázisvektorai**.

- Ortogonális transzformáció esetében, hasonlóan,

$$\mathbf{x} = \sum_{k=0}^{N-1} y(k) \mathbf{a}_k$$

alakban, ahol  $\mathbf{a}_k$  az  $\mathbf{A}^T$  mátrix oszlopai.

- Az inverz transzformáció tehát nem más, mint  $\mathbf{x}$  jelvektor  $\mathbf{a}_k^*$  (vagy  $\mathbf{a}_k$ ) bázisvektorok szerinti sorfejtése,  $y(k)$  együtthatókkal.



## 2D tanszformáció

- A bemenő 2D mintasorozat  $N \times N$ -es blokkjait egy hipermátrix-al tanszformáljuk:

$$\mathbf{Y} = Y[k, l] = \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} x[m, n] A(k, m; l, n)$$

a tanszformáció kifejezése, és

$$\mathbf{X} = x[m, n] = \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} Y[k, l] A^T(k, m; l, n)$$

az inverz tanszformáció kifejezése ortogonális tanszformáció esetén



## 2D szeperábilis transzformáció

- Ha a transzformáció hipermátrixa  $A(k, m; l, n)$  felírható  $A(k, m; l, n) = A_1(k, m) A_2(l, n)$  alakban, akkor a 2D transzformáció szeperábilis - a kép oszlopain elvégzett 1D transzformáció, majd a transzformált kép sorain elvégzett 1D transzformáció egymásutánjaként elvégezhető
- Ha  $A(k, m; l, n)$  unitér/ortogonális, akkor  $A_1(k, m)$  és  $A_2(l, n)$  is unitér/ortogonális, és a gyakorlatban alkalmazott transzformációk esetében  $\mathbf{A}_2 = \mathbf{A}_1^T$ 
  - Ekkor  $\mathbf{Y} = \mathbf{A}\mathbf{X}\mathbf{A}^T$ , illetve
  - $\mathbf{X} = \mathbf{A}^T\mathbf{Y}\mathbf{A}$ , ha a transzformáció ortogonális.



## Tulajdonságok

- Unitér/ortogonális transzformációk esetén teljesül a Parseval (energiamegmaradás) tétele:
- $$\sum_{m=0}^{N-1} \sum_{n=0}^{N-1} |x[m, n]|^2 = \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} |y[k, l]|^2$$
  - Jelöljük  $\mathbf{E}_y = \mathbf{Y} - \hat{\mathbf{Y}}$ -nal a transzformációs tartományban elkövetett (pl. kvantálási) hibákat
  - Jelöljük  $\mathbf{E}_x = \mathbf{X} - \hat{\mathbf{X}}$ -nal a visszaállítás utáni hibákat az eredeti képhez képest
- A Parseval-tétel, és a transzformáció linearitása miatt ( $\mathbf{E}_y = \mathbf{A}\mathbf{E}_x\mathbf{A}^T$ ),  $\|\mathbf{E}_x\|^2 = \|\mathbf{E}_y\|^2$ , vagyis ha a transzformációs tartományban minimalizáljuk a reprezentációs hiba-energiát (MSE), akkor az eredeti jeltartományban is minimális hibaenergiát (MSE) kapunk



# Transzformációs típusok

---

- Ha az együtthatók közötti korreláció a transzformált tartományban nulla, akkor a transzformáció optimális
- Ez akkor teljesül, ha a jelet az autokorrelációs mátrixának sajátvektorai által kifeszített térbe transzformáljuk
- Ebben az esetben a transzformáció bázisa jelfüggető
- A gyakorlatban a fix bázisú transzformációkat alkalmazzuk
- Ezek szuboptimálisak, de a valós idejű jelfeldolgozáshoz alkalmasabbak
- Az optimális transzformáció a KLT (Karhunen-Love)



## Nem-harmonikus bázisfüggvényű

- DWHT
- DHT
- DST
- ...

## Harmonikus bázisfüggvényű

- DFT
- DCT
- DST
- Wavelet





## Nem-harmonikus bázisfüggvényű

- Valós, ortogonális transzformáció
- $2 \times 2$ -es transzformációs mátrixa:

$$\mathbf{H}_{2 \times 2} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

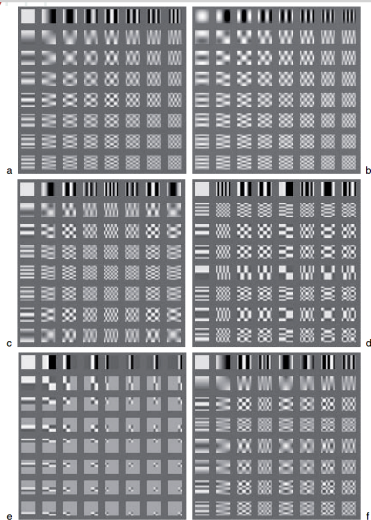
- Nagyobb blokkméretre  $\mathbf{H}$  rekurzívan számolható:

$$\mathbf{H}_{2N \times 2N} = \frac{1}{\sqrt{2}} \begin{bmatrix} \mathbf{H}_{N \times N} & \mathbf{H}_{N \times N} \\ \mathbf{H}_{N \times N} & -\mathbf{H}_{N \times N} \end{bmatrix}$$

- Az 2D transzformáció kifejezése
  - $\mathbf{y} = \mathbf{H}\mathbf{x}\mathbf{H}^T$  és
  - $\mathbf{x} = \mathbf{H}^T\mathbf{y}\mathbf{H}$

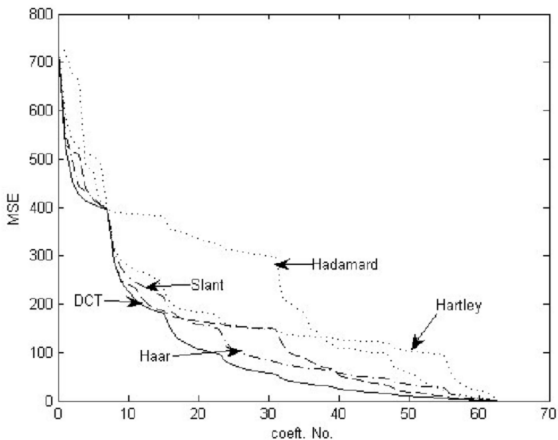


# 2D transzformációs bázisképek



ahol a: DCT, b: DST, c: Hartley, d: Hadamard, e: Haar, és f: Slant transzformációk bázisképei

# MSE a transzformáció típusának függvényében



Camerman ábra



## Optimális transzformáció

- Legyen az  $\mathbf{x}$  véletlen vektor, és  $\mathbf{x}$  transzformáltja  $\mathbf{y} = \mathbf{T}^H \mathbf{x}$
- $\mathbf{x}$  kovariancia mátrixa  
$$\mathbf{C}_x = E[(\mathbf{x} - E[\mathbf{x}])(\mathbf{x} - E[\mathbf{x}])^H] = E[\mathbf{x}\mathbf{x}^H] - E[\mathbf{x}]E[\mathbf{x}^H]$$
- $\mathbf{C}_x = \mathbf{C}_x^H$ , vagyis hermitikus mátrix (valós elemek esetén szimmetrikus)
- $\mathbf{C}_x$  diagonális, akkor és csak akkor, ha  $\mathbf{x}$  elemei korrelálatlanok
- Célunk egy olyan transzformáció keresése, mely  $\mathbf{C}_x$ -t diagonális mátrixba viszi át (Legyen  $\mathbf{C}_y = \mathbf{D}$ )



## Optimális transzformáció

- Vagyis  $\mathbf{C}_y = \mathbf{T}^H \mathbf{C}_x \mathbf{T}$ , ahol  $\mathbf{C}_y$  diagonális.
- Mivel  $\mathbf{C}_x$  hermitikus/szimmetrikus, létezik spektrálfelbontása, mely  $\mathbf{C}_x = \mathbf{U} \mathbf{D} \mathbf{V}^H$ , és  $\mathbf{U} = \mathbf{V}$ , tehát  $\mathbf{C}_x = \mathbf{U} \mathbf{D} \mathbf{U}^H$
- Ha a transzformáció  $\mathbf{T}$  mátrixának  $\mathbf{U}$ -t választjuk,  
$$\mathbf{C}_y = \mathbf{T}^H \mathbf{C}_x \mathbf{T} = \mathbf{U}^H \mathbf{U} \mathbf{D} \mathbf{U} \mathbf{U}^H \mathbf{U} = \mathbf{D}$$
- A sajátvektorok, illetve sajátértékek rendezését a sajátértékek csökkenő sorrendje szerint végezzük (legszignifikánsabb-legkevésbé szignifikáns):  
 $\sigma_1^2 > \sigma_2^2 > \dots > \sigma_N^2$ , ahol  $\sigma_i^2$  sajátérték az  $y_i$  varianciája.



## Rekonstrukció hibája

- Legyen  $\mathbf{x} = \hat{\mathbf{x}} + \mathbf{e} = \sum_{i=1}^M y_i \mathbf{u}_i + \sum_{j=M+1}^N y_j \mathbf{u}_j$
- Vagyis  $\mathbf{x}$   $\hat{\mathbf{x}}$ -el jelölt közelítésében csak az első  $M$  sajátvektort vesszük figyelembe.
- Ekkor  $E [\|\mathbf{e}\|^2] = E [\|\mathbf{x} - \hat{\mathbf{x}}\|^2] = E [\|\mathbf{y} - \hat{\mathbf{y}}\|^2] = \sum_{j=M+1}^N \sigma_j^2$
- Vagyis a rekonstrukció hibája (MSE értelemben) pont azon sajátértékek összege, melyekhez tartozó sajátvektorokat nem vettük figyelembe az inverz transzformációnál (sorfejtés)

## Elméleti 2D KLT:

- $\mathbf{X} = x[m, n]$  egy 2D véletlen mátrix.
- Célunk  $\mathbf{X}$  felírása unitér/ortogonális bázismátrixok segítségével  $\mathbf{X} = \sum_{k=0}^{M-1} \sum_{l=0}^{N-1} Y(k, l) \mathbf{U}_{k,l}$ , vagy
- másik jelöléssel  $x[m, n] = \sum_{k=0}^{M-1} \sum_{l=0}^{N-1} Y(k, l) u_{k,l}[m, n]$ .
- Az  $\mathbf{U}_{k,l}$  mátrixok, hasonlóan az 1D esethez, sajátmátrixai a  $C_x(m_1, n_1, m_2, n_2)$  4 dimenziós kovariancia hipermátrixnak.



# 2D KLT gyakorlati megvalósítása II.

## 1D-2D

- Ha  $\mathbf{X} = x[m, n]$  egy  $M \times N$ -es mátrix, akkor  $\mathbf{X}$  soraiból sor-folytonosan képzett (vagy  $\mathbf{X}$  oszlopaiból oszlop-folytonosan képzett)  $M \times N$ -es 1D vektoron az 1D KLT végrehajtható

## Szeeparábilis 2D

- Ha feltételezzük, hogy  $\mathbf{X}$  egyes sorainak kovariancia mátrixa azonos, valamint  $\mathbf{X}$  egyes oszlopainak kovariancia mátrixa azonos, akkor ezekre a  $\mathbf{K}_r$  illetve  $\mathbf{K}_c$  jelölést bevezetve
  - $\mathbf{K}_r = \mathbf{U}_r \mathbf{D}_r \mathbf{U}_r^H$ , illetve
  - $\mathbf{K}_c = \mathbf{U}_c \mathbf{D}_c \mathbf{U}_c^H$  felhasználásával
  - a 2D transzformáció szeeparálható  $\mathbf{Y} = \mathbf{U}_c^H \mathbf{X} \mathbf{U}_r$ , illetve
  - az inverz transzformáció  $\mathbf{X} = \mathbf{U}_c \mathbf{Y} \mathbf{U}_r^H$





# 2D KLT gyakorlati megvalósítása II.

## $\mathbf{K}_r$ és $\mathbf{K}_c$ közelítése

- $\hat{\mathbf{K}}_r = \frac{1}{M-1}(\mathbf{X} - \bar{\mathbf{X}})^T(\mathbf{X} - \bar{\mathbf{X}})$
- $\hat{\mathbf{K}}_c = \frac{1}{N-1}(\mathbf{X} - \bar{\mathbf{X}})(\mathbf{X} - \bar{\mathbf{X}})^T,$
- ahol  $\bar{\mathbf{X}} = \bar{x}[m, n] = \frac{1}{M} \frac{1}{N} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} x[m, n]$ , a  $\mathbf{X}$  mátrix átlagértéke
- Megjegyzés: belátható, hogy a fenti közelítések felhasználásával elvégzett szeparábilis 2D KLT (inverz)transzformáció ekvivalens a  $\mathbf{X} - \bar{\mathbf{X}}$  mátrix SVD felbontásával.



# Az unitér/ortogonális transzformációs kódolásokról általában

- A transzformáció önmagában véve veszteségmentes
- Az oda- és vissza-transzformáció elméletben az eredeti jelet adja vissza (a numerikus hibáktól eltekintve)
- A transzformált tartományban a jelenergia nagy részéhez csak néhány koeficiens járul hozzá
- A transzformáció hatékonyságát az jellemzi, hogy a koeficiensok közül hány jelentős
- Ha az inverz transzformációt csak a jelentős együtthatókkal végezzük el, a kapott blokk általában eltér az eredetitől, de az eltérés nem számottevő



## Előnyei

- A természetes képeket majdnem teljesen dekorrelálja
- Közel optimális
- A jelenergia nagy részét néhány kisfrekvenciás koefficiens hordozza
- Valós együtthatókat eredményez
- Számítástechnikailag hatékony algoritmusokkal számolható



## 1D DCT és IDCT

- DCT

$$X(k) = \sqrt{\frac{2}{N}} \alpha(k) \sum_{n=0}^{N-1} x(n) \cos \left[ \frac{\pi}{N} \left( n + \frac{1}{2} \right) k \right] \quad k = 0, \dots, N-1$$

- IDCT

$$x(n) = \sqrt{\frac{2}{N}} \sum_{k=0}^{N-1} \alpha(k) X(k) \cos \left[ \frac{\pi}{N} \left( n + \frac{1}{2} \right) k \right] \quad n = 0, \dots, N-1$$

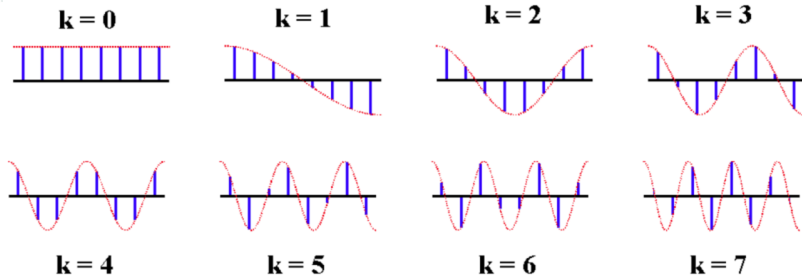
- ahol

$$\alpha(k) = \begin{cases} \frac{1}{\sqrt{2}} & \text{ha } k \equiv 0 \\ 1 & \text{ha } k \neq 0. \end{cases}$$

## Elnevezések

- DC koefficiens:  $X(0)$
- AC koefficiensek  $X(1), \dots, X(N - 1)$
- A  $k$ . bázisvektor:  $\sqrt{\frac{2}{N}} \alpha(k) \cos \left[ \frac{\pi}{N} \left( n + \frac{1}{2} \right) k \right] \quad n = 0, \dots, N - 1$

# $N = 8$ esetén a DCT bázisvektorok



- A bemenő jelvektort különböző harmonikus frekvenciájú elemi komponensek súlyozott összegeként (lin. komb.) állítjuk elő
- A súlytényezők a DCT együtthatók
- Ha a bemeneti vektoron belül a jel "lassan" változik, akkor a DC, illetve a kisfrekvenciás AC komponensek dominálnak

- 2D DCT

$$X(k, l) = \frac{2}{N} \alpha(k)\alpha(l) \sum_{n=0}^{N-1} \sum_{m=0}^{N-1} x(n, m) \cos \left[ \frac{\pi}{N} \left( m + \frac{1}{2} \right) k \right] \cos \left[ \frac{\pi}{N} \left( n + \frac{1}{2} \right) l \right] \quad k = 0, \dots, N - 1$$

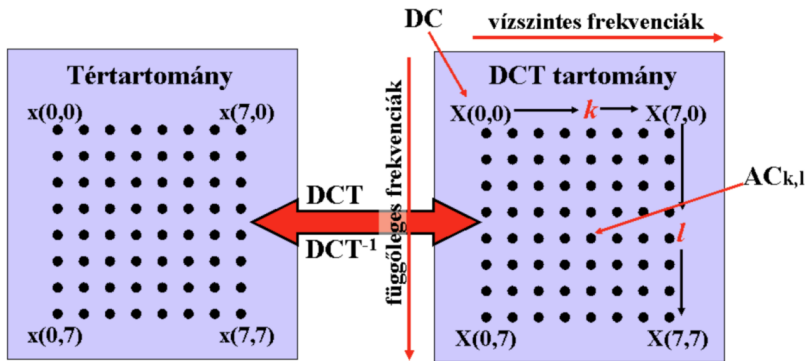
- 2D IDCT

$$x(m, n) = \frac{2}{N} \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} \alpha(k)\alpha(l) X(k, l) \cos \left[ \frac{\pi}{N} \left( m + \frac{1}{2} \right) k \right] \cos \left[ \frac{\pi}{N} \left( n + \frac{1}{2} \right) l \right] \quad k = 0, \dots, N - 1$$

- ahol

$$\alpha(k), \alpha(l) = \begin{cases} \frac{1}{\sqrt{2}} & \text{ha } k, l \equiv 0 \\ 1 & \text{ha } k, l \neq 0. \end{cases}$$

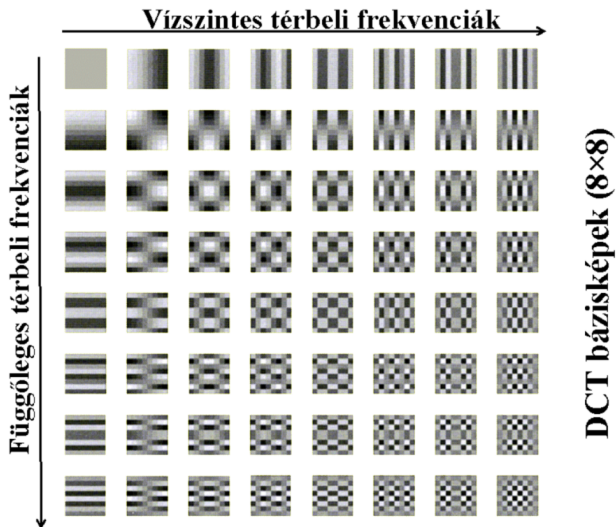
# 2D DCT illusztráció I.







# 2D DCT illusztráció II.





# 2D DCT illusztráció III.

**8 × 8-as pixel mátrix**

83	87	92	90	89	91	47	95
98	81	77	96	71	44	58	49
27	43	65	40	64	99	61	55
83	21	45	51	59	80	48	63
87	94	56	62	41	74	75	57
98	82	68	79	54	46	52	60
72	42	70	84	69	50	97	67
76	53	85	88	73	66	78	86

**DCT**



**8 × 8-as koefficiens mátrix**

248	18	-12	2	3	-7	0	0
10	9	8	1	-4	1	0	0
-6	2	1	0	2	4	2	-1
2	1	-1	1	-3	0	2	0
0	0	0	1	2	0	0	-1
1	-2	1	0	1	-2	3	0
0	0	0	-1	1	2	0	0
0	0	0	0	0	0	0	0



# 1D DCT számítása DFT-ből

## 1D DCT és DFT

- 1D DCT számítható az  $N$  hosszúságú  $\mathbf{x} = x[n] \quad n = 0 \dots N - 1$  mintavektor szimmetrikus  $2N$  hosszú kiterjesztésének DFT-jéből, ahol
- a szimmetrikus kiterjesztés  $\bar{x}[p] = x[p]$ , ha  $p < N$ , és  $\bar{x}[q] = x[2N - 1 - q]$ , ha  $q > N$
- Ha  $Y(k)$  jelöli az  $\mathbf{x} = x[n]$  DCT együtthatóit,
- akkor  $Y(k) = \Re \{ W F(k) \}$ , ahol  $F(k) \quad k = 0 \dots N - 1$  a  $2N$  hosszúságú kiterjesztett mintavektor első  $N$  DFT együtthatója, és
- $W$  egy jelfüggetlen skálatényező.

# 2D DCT számítása

- Ha  $\mathbf{A} = A[k, m] = \sqrt{\frac{2}{N}} \alpha(k) \cos \left[ \frac{\pi}{N} \left( m + \frac{1}{2} \right) k \right]$  az 1D transzformáció mátrixa, és
- $\mathbf{X} = x[m, n]$  a 2D kép mátrix, akkor  $\mathbf{Y}_c = \mathbf{A}\mathbf{X}$ , majd  $\mathbf{Y}_{c,r} = \mathbf{A}\mathbf{Y}_c^T$ , akkor
- $\mathbf{Y}_{c,r} = \mathbf{A}(\mathbf{X}\mathbf{A})^T = \mathbf{A}\mathbf{X}^T\mathbf{A}^T$ , amiből az eredeti kép DCT-je:
- $\mathbf{Y}_{k,l} = \mathbf{Y}_{c,r}^T$



## Kvantáló mátrix

- A kvantálás blokkonként ún. kvantáló mátrixszal történik
- $X^*(k, l) = \frac{X(k, l)}{W(k, l)}$ , ahol  $W(k, l)$  jelenti a  $(k, l)$  együtthatóhoz tartozó kvantálási lépcsőt
- A kvantáló mátrix elemeinek megválasztása alapvetően a HVS tulajdonságainak megfelelően történik
- Maradék elhagyása



## Kvantálás a HVS tulajdonságai alapján

- A HVS az apró részletekre (nagyfrekvenciás komponensek) kevésbé érzékeny
- A HVS szempontjából fontos együtthatókat (DC, néhány kisfrekvenciás AC) finomabban, a kevésbé fontos AC együtthatókat durvábban kvantáljuk
- Egy tipikus kvantáló mátrix pl. a következő

$$W(k,l) = \begin{bmatrix} 8 & 16 & 19 & 22 & 26 & 27 & 29 & 34 \\ 16 & 16 & 22 & 24 & 27 & 29 & 34 & 37 \\ 19 & 22 & 26 & 27 & 29 & 34 & 34 & 38 \\ 22 & 22 & 26 & 27 & 29 & 34 & 37 & 40 \\ 22 & 26 & 27 & 29 & 32 & 35 & 40 & 48 \\ 26 & 27 & 29 & 32 & 35 & 40 & 48 & 58 \\ 26 & 27 & 29 & 34 & 38 & 46 & 56 & 69 \\ 27 & 29 & 35 & 38 & 46 & 56 & 69 & 83 \end{bmatrix}$$

- A bitsebesség pl. a  $W(k,l)$  konstanssal való szorzásával állítható
- A kvantáláson túlmenően a DC komponensek általában a szomszédos blokkhoz képest differenciálisan kódoltak (JPEG és MPEG)

# DCT együtthatók kvantálása illusztráció I.

**DCT koeficienssek**

$$\begin{bmatrix} 284 & 18 & -12 & 2 & 3 & -7 & 0 & 0 \\ 10 & 9 & 8 & 1 & -4 & 1 & 0 & 0 \\ -6 & 2 & 1 & 0 & 2 & 4 & 2 & -1 \\ 2 & 1 & -1 & 1 & -3 & 0 & 2 & 0 \\ 0 & 0 & 0 & 1 & 2 & 0 & 0 & -1 \\ 1 & -2 & 1 & 0 & 1 & -2 & 3 & 0 \\ 0 & 0 & 0 & -1 & 1 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

**Kvantáló mátrix (W(k,l))**

$$\begin{bmatrix} 8 & 16 & 19 & 22 & 26 & 27 & 29 & 34 \\ 16 & 16 & 22 & 24 & 27 & 29 & 34 & 37 \\ 19 & 22 & 26 & 27 & 29 & 34 & 34 & 38 \\ 22 & 22 & 26 & 27 & 29 & 34 & 37 & 40 \\ 22 & 26 & 27 & 29 & 32 & 35 & 40 & 48 \\ 26 & 27 & 29 & 32 & 35 & 40 & 48 & 58 \\ 26 & 27 & 29 & 34 & 38 & 46 & 56 & 69 \\ 27 & 29 & 35 & 38 & 46 & 56 & 69 & 83 \end{bmatrix}$$

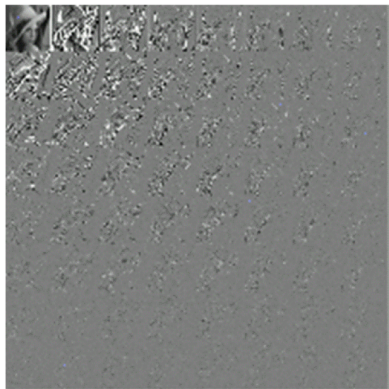
/

**Kvantálás után**

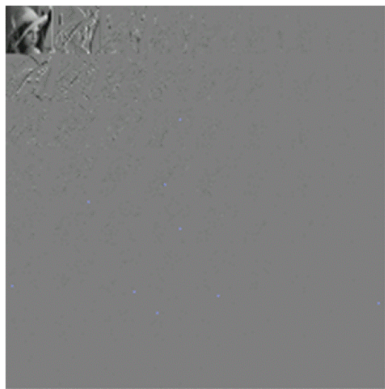
=

$$\begin{bmatrix} 35 & 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

# DCT együtthatók kvantálása illusztráció II.



**DCT felbontás**  
eredeti DCT együtthatók



**DCT felbontás**  
újrakvantált DCT együtthatók





- Kvantálás után az együtthatók nagy része zérussá válik
- Csak a nullától különböző együtthatókat kell tárolni/továbbítani, viszont ezek pozícióját ismerni kell
- Cikk-cakk kódolás: növekvő frekvenciák szerinti rendezés

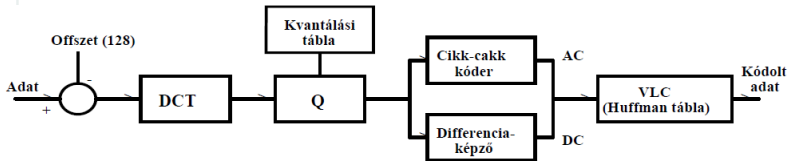
## Cikk-cakk kódolás

- Frekvencia szerint növekvő sorrendbe rendezzük az együtthatókat
- A DC együtthatót külön kezeljük
- A DC együtthatót általában veszteségmentesen, differenciálisan kódoljuk az előző blokk DC együtthatójával, mivel a szomszédos blokkok DC együtthatója hasonló
- Az AC együtthatók cikk-cakk rendezése miatt több nulla kerül egymás után
- A nullákra futamhossz kódot alkalmazunk
- (Zérusok száma, értékes amplitúdó) párokat (run,level) képzünk a cikk-cakk sorrend alapján
- A blokk végét EOB szimbólum zárja

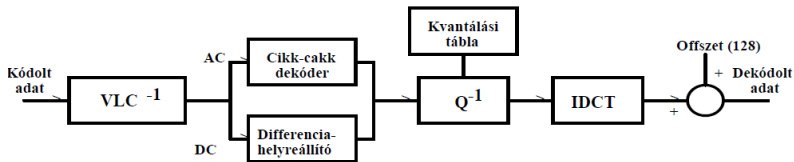




# JPEG kódoló/dekódoló vázlata



DCT alapú szekvenciális JPEG kódér



DCT alapú szekvenciális JPEG dekódér

- képméret maximum 65535 x 65535,
- komponensek száma maximum 255,
- a színmérő-rendszer nem specifikált.
- a kóder és dekóder szimmetrikus felépítése,
- képminőség - sebesség (bpp) kompromisszum
- 4-féle üzemmód:
  - **DCT alapú szekvenciális kódolás,**
  - **DCT alapú progresszív kódolás,**
  - veszteségmentes DPCM alapú kódolás,
  - hierarchikus kódolás (DCT, DPCM).



# DCT alapú JPEG kódolás lépései

- A kép 8x8-as, egymással át nem lapolódnó blokkjainak transzformációja DCT-vel.
- Az együtthatók kvantálása blokkonként egy felhasználó által definiált súlyozó mátrix és egy kvantálási tényező segítségével.
- Maximum 4 súlyozó mátrix használható egyszerre (pl. komponensenként más és más).
- A DC együttható: az előző blokkhoz képesti különbség kódolása. (1-D veszteségmentes DPCM hurokkal).



# DCT alapú JPEG kódolás lépései

- Az AC együtthatók cikk-cakk-ba rendezése, futamhossz kódolt párok 0-ák futási hossza, nem zérus amplitúdó képzése.
- A differenciális DC együtthatók és a futamhossz kódolt párok VLC kódolása.
- A blokk végét EOB jelzi.
- Maximum 2 DC és 2 AC Huffman tábla használható.
- A képkomponensek függetlenül kódoltak.
- A dekóder a bitfolyam fejlécéből nyeri ki azokat az információkat, amelyek a dekódolás vezérléséhez kellenek.
- A dekódolás számítás igénye kicsit kisebb mint a kódolásé.
- A rekonstruált kép minőségét a súlyozó mátrix, a kvantálási tényező és a DCT és IDCT pontossága határozza meg,
- A nem definiált kvantálási és Huffman táblákat (8 bites 8x8 db. koefficiens) a fejlécben kell továbbítani.
- Nincs tényleges sebesség vezérlés.



# DCT alapú JPEG kódolás lépései

- Az alapértelmezett (fix) kvantálási és Huffman táblákat használva a kóder és dekóder szimmetrikus a műveletigényt tekintve.
- A DC differenciák osztályozása amplitúdó szerint, osztály, amplitúdó párok képzése.
- Az osztály Huffman kódolása + az amplitúdó változó hosszúságú egészként való kódolása (VLI).
- Az osztály megadja a VLI bitszámát.
- A koefficiens osztályba sorolása és azon belüli érték meghatározása.
- VLI képzés: pozitív számoknál a szükséges bitszámon előjel nélküli bináris ábrázolása, negatív számoknál ennek az 1-es komplemente.



# JPEG VLI tábla, példa

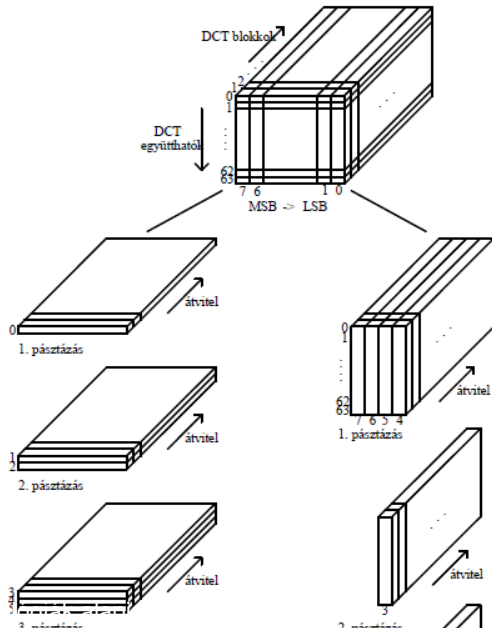
Osztály	Amplitúdó	VLI
1	-1, 1	0, 1
2	-3, -2, 2, 3	00, 01, 10, 11
3	-7...-4, 4...7	000...011, 100...111
4	-15...-8, 8...15	0000..0111, 1000..1111
5	-31...-16, 16...31	.....
6	-63...-32, 32...63	.....
7	-127...-64, 64...127	.....
8	-255...-128, 128...255	.....
9	-511...-256, 256...511	.....
10	-1023...-511, 511...1023	.....
11	-2047...-1024, 1024...2047	0000000000...0111111111 1000000000...1111111111





# Progresszív JPEG

- A kép átvitele több lépcsőben történik (durvától a legfinomabb részletekig)
- Különböző minőségi szintű hozzáférést biztosít a réteges kódolással (alapréteg + javítórétegek)
- Két fajta eljárás:
  - Spektrális kiválasztás: A cikk-cakkba rendezett DCT együtthatók sávokra osztása. Az első sáv mindig csak a DC komponenst tartalmazza. Az egyes sávokat a kezdő és záró cikk-cakk pozícióval azonosítják. A sávok a teljes képre vonatkozóan egymás után tárolódnak a bitfolyamban.
  - Szukcesszív approximáció: A DCT együtthatók bitsíkonkénti csoportosítása. Az első csoport az együtthatókat csökkentett precizitással tartalmazza (pl. az első 3 legértékesebb bit). A további csoportok a mintafelbontást javítják. Az egyes csoportokat a kezdő és záró bitpozíciójukkal azonosítják



# MPEG szabványok

- A szabványok nem specifikálja a kódert (kódolási algoritmusokat) sem a videó, sem az audio részben! Csak a kódolás eszközeinek tárháza, és az azok által létrehozott kimeneti adatfolyam bitszintaxisa specifikált!
- Szintén definiált a szabványos bitszintaxisú adatfolyam dekódolásának módja.
- MPEG-1 (ISO 11172 /1993/): Alacsony bitsebességű multimédiás alkalmazások (Video-CD, CD-I), kb. 1,5 Mb/s VHS képminőség, SIF felbontású kódolt videó.
- MPEG-2 (ISO 13818 /1994-95/): Műsorszórás (Digital Video Broadcasting, DVB), 2-8 Mb/s terjesztési minőség (distribution quality); Stúdiótechnika, 18-50 Mb/s újrafeldolgozási minőség (contribution quality); Digital Versatile Disc (DVD), 3-7 Mb/s (VBR!) jobb mint a PAL minőség; általában ITU-R BT-601 a felbontás mindegyikben. Az MPEG-2 kibővített MPEG-1, a kódolási elv mindkét eljárásban azonos.
- MPEG-4 (ISO/IEC 14496): Alacsony bitsebességű kódolások (kiindulás x 64 kbit/s) szabványának indult (1994), de jelenleg az interaktív multimédia szolgáltatások objektum orientált szabványává vált. *Ezen belül a két legfontosabb rész:*
  - MPEG-4 Part 2: Egyik szabványos változat a H.263, illetve implementációi a DivX, és XVID
  - MPEG-4 Part 10: H.264/AVC



# MPEG-1

---

- DCT alapú mozgáskompenzációt használó hibrid kódolás.
- A kódolás és dekódolás számításigénye különböző, a rendszer tehát aszimmetrikus.
- MPEG videókódolás egyik alapjellemzője a réteges szerkezete.
- A rétegszerkezet 6 egymásba ágyazott egységet tartalmaz, melyekben az alsóbb rétegek általában nem dekódolhatók a felsőbb szintek nélkül.

# Mozgásbecslés, mozgáskompenzáció alapú predikció

- A mozgókép soron egymás utáni képkockái erősen korreláltak
- A képváltásokat és a gyors mozgásokat (pl. sport) leszámítva a képtartalom képről-képre viszonylag kevésbé változik
- Mozgásbecslés alapú predikcióval a predikció hatékonysága javítható
- A redundancia csökkentés hatékonysága a mozgás meghatározás pontosságától függ
- A mozgásvektorok pontos meghatározása nehéz és időigényes
- Megfelelően definiált költségfüggvényekkel, kompromisszumokkal keressük a mozgásvektorokat (pl. átlagos abszolút hiba, vagy átlagos négyzetes hiba)
- Offline tömörítés esetén pontosabb mozgásbecslő algoritmusok használhatók



# Képek közötti differencia (lassú mozgás)



**N. kép**



**(N+1). kép**



**N. kép - (N+1). kép**



# Képek közötti differencia (képvtáltás)



**N. kép**



**(N+1). kép**

**N. kép - (N+1). kép**





# Blokk alapú mozgásbecslés I.

- Az  $N \times N$ -es blokk alapú transzformációs kódoláshoz illeszkedően blokk alapú mozgásbecslést és predikciót célszerű megvalósítani
- A cél az aktuális kép minden  $N \times N$ -es blokkjához egy referenciakép egy-egy  $N \times N$ -es területét rendelni, amely a kódolandó blokk prediktora lesz
- A blokkok pixel-koordinátáinak különbsége az adott blokk mozgásvektora
- Az optimális prediktor-blokk kiválasztása szélsőérték keresési feladat, melyben a kódolandó blokk körül kijelölt ablakon belül a referencia kép és a kódolandó blokk illeszkedést maximalizáljuk egy költségfüggvény szerint





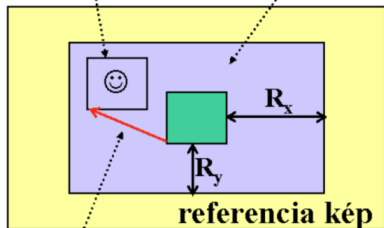
## Blokk alapú mozgásbecslés II.

- A mozgásbecslést az Y komponensen végezzük, a kapott mozgásvektorokat a Cr,Cb blokkok mozgásvektorai is lesznek
- Minden nem intra blokkhoz tartozik minimum egy mozgásvektor, a blokkon belüli mozgás azonban nem detektálható (predikciós hibaként jelenik meg)
- A mozgásvektorok is differenciálisan kódoltak, a blokk-raszter sorrendjében a szomszédos MB mozgásvektorához képesti különbséget kódolják (hatékony, mert a szomszédos makroblokkok valószínűleg hasonlóan mozognak)
- A mozgásvektorok meghatározása 1, 1/4 ill. 1/8 pixel pontossággal történhet, a kódoló típusától függően. Tört-pixeles pontosság esetén a kép pixeleihez tartozó minták interpolálásával számolják ki a pixelek közötti mintákat, és ez az interpolált kép a mozgáskeresés, illetve a predikció alapja.



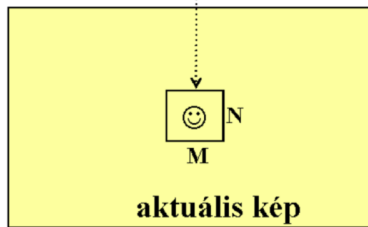
# Blokk alapú mozgásbecslés III.

legjobb illeszkedés      keresési ablak



mozgásvektor

kódolandó blokk





## Keresési eljárások

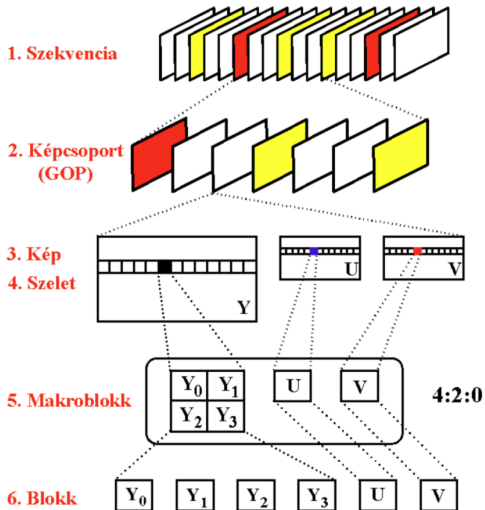
- Teljes/kimerítő keresés: pontos, de nagyon időigényes
- Pixel-rekurzív keresés: előzetes becslés pl. előző kép, szomszédos MB-k alapján, az előzetes becslés finomítása a mozgásvektor koordinátájának környezetében
- Logaritmikus keresés: Durvább, majd finomabb léptékű keresés a referencia illetve az adott kódolandó kép durvább majd finomabb felbontású változatain. Minden finomítás után pixel rekurzív keresés a korábbi durvább felbontású mozgásvektor eredmény környezetében



# MPEG rétegszerkezet

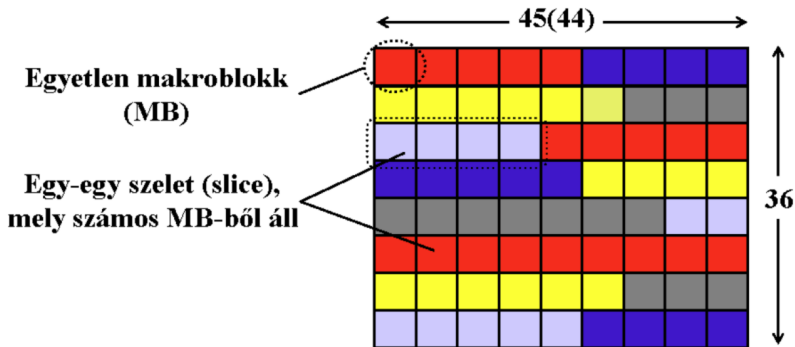
- Szekvencia réteg** A kódolt szekvenciát azonosítja, a fejléc tartalmazza a rendszeradatokat (képméret, bitsebesség, stb.)
- Képcsoport réteg (GOP)** Legalább egy önmagában kódolt (I) képet tartalmazó, meghatározott számú kép együttese, a véletlen hozzáférés egysége
- Képréteg** Egy kép kódolt adatait tartalmazza
- Szelet (slice) réteg** MB-k sor-folytonos csoportja, az újraszinkronizáció egysége (a legalsó szint, amelyen a dekóder még képes feléledni bithiba esetén)
- Makroblokk réteg** Az Y  $16 \times 16$ -os (4 db  $8 \times 8$ -as blokk), és a Cr,Cb  $8 \times 8$ -as blokkjaiból áll (4:2:0 esetén 1 Cr 1 Cb, tehát 1 MB összesen 6 blokk), a mozgáskompenzáció egysége
- Blokk réteg** A MB  $8 \times 8$ -as blokkjai, a DCT kódolás egysége

# MPEG rétegszerkezet





# MPEG képszerkezet



**Egy MPEG-2 EU SD kép 45(44) × 36 MB tartalmaz**



# MPEG képtípusok

- I (intra)** Önmagában kódolt kép, a dekódoláshoz szükséges minden adatot tartalmaz
- P (predictive)** egy múltbéli referencia képhez képest prediktíven kódolt kép, referenciája egy előző I vagy P kép, dekódolásához a referencia kép szükséges
- B (bidirectional)** két irányból kódolt kép, referenciája az előző I vagy P kép és a következő I vagy P kép. B kép nem lehet referencia



# MPEG képsorrend, MB típusok

- A kijelzési és átviteli képsorrend általában különböző
- A műsorszórás leggyakoribb kijelzési sorrendje: IBBPBBPBBPBB (Long GOP - EU)
- A stúdiótechnika kijelzési sorrendje általában: csak I képek
- MB típusok
  - I képekben: minden MB önmagában kódolt, intra típusú
  - P képekben: az MB lehet intra, vagy prediktíven kódolt
  - B képekben: az MB lehet intra, illetve a múltból, a jövőből, vagy múltból és jövőből prediktíven kódolt (múltból és jövőből történő predikció esetén a mozgás-kompenzált múltbéli illetve jövőbéli referenciaképek átlaga a predikció alapja)
- Frame és field alapú predikció (többfajta...)





- A globálisan elérhető tömörítés 10 és 50 között van.
- B képek alkalmazásával növekszik a dekóder oldalon a számítás- és memóriaigény, és a kóder oldali a késleltetés.
- Az IPB struktúra nem specifikált, B képek használata nem kötelező: a szekvencia állhat csak I, vagy csak I és P képekből.
- Az I, P és B képek aránya kompromisszum kérdése:
- Gyors véletlen hozzáférés (szerkeszthetőség, dekódolás megkezdhetősége): sok I kép, kevés P és B kép.
- Jó tömöríthetőség: sok B, kevés I kép.
- Alacsony késleltetés, alacsony hardverköltés: nincsenek B képek.
- A B képek miatt kétféle képsorrend (kijelzési, átviteli) definiált, hiszen a jövőbeli referenciaképeket előbb kell a bitfolyamba tenni mint a belőlük jósolt B képeket.

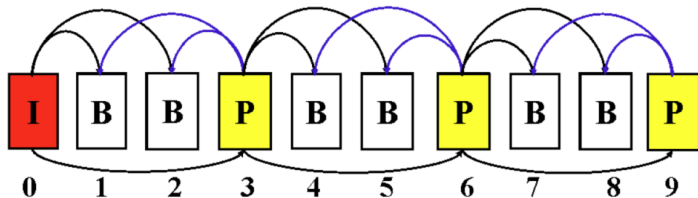


- Zárt képcsoport: a benne lévő első B képeknek nem referenciája az előző képcsoport utolsó I vagy P képe.
- Nyílt képcsoport: a benne lévő első B képeknek múltbeli referenciája az előző képcsoport utolsó I, vagy P képe.

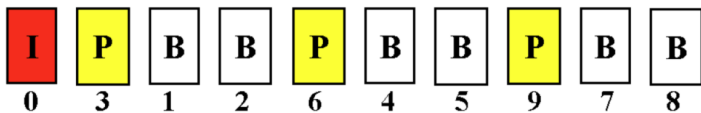


# MPEG képsorrend

## Kijelzési képsorrend



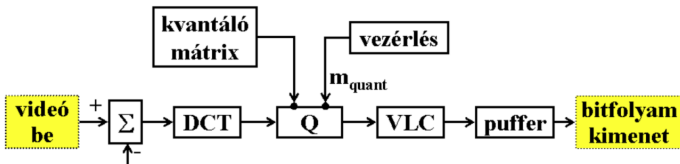
## Átviteli képsorrend





# MPEG kódolás lépései

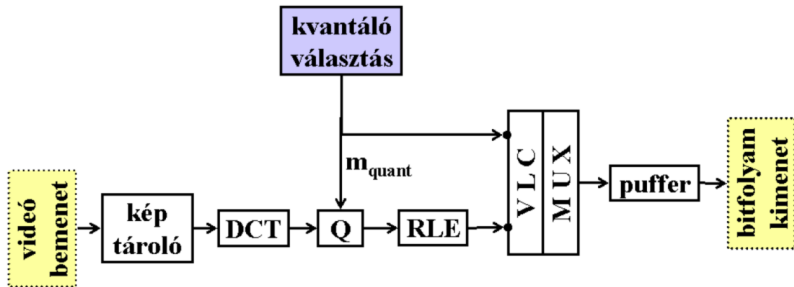
- A MB predikciós hibájának, vagy az intra MB kódolása DCT transzformációval
- A DCT együtthatók újrakvantálása: a kvantálási lépcsőt a kvantálási mátrix megfelelő elemének és a kvantálási skálafaktornak a szorzata adja
- Kvantálási mátrix perceptuális alapú az I képekre, és általában konstans a P és B képekre
- A skálafaktor szeletenként, vagy MB-nként újradefiniálható (bitrate vezérlés)







# MPEG I-only (stúdió) kódoló





## Konstans bitsebesség (CBR)

- A kódoló kimenetén egységnyi idő alatt keletkező bitek száma változó, mert
  - A különböző típusú képek (I,P,B) különböző bitszámmal kódolhatók azonos minőség mellett
  - Egy képen belül is változhat a képtartalom, az összetettebb blokkok leírásához több bitre van szükség
  - A VLC kódolás miatt is ingadozik a kimeneti bitmennyiség
- A legtöbb átviteli csatorna konstans bitsebességű forrást igényel
- A probléma a kódoló kimenetére helyezett pufferrel oldható meg, melyből a csatorna fix ütemben olvassa a biteket



# Bitsebesség vezérlés lépései

---

- GOP-ra elosztható bitek száma adott, a cél bitszám meghatározása I, P, B képekre (globális minőség biztosítása).
- A kvantálás meghatározása: MB komplexitástól és textúrától függő referencia kvantáló felvétele (lokális minőség biztosítása).
- és azok módosítása a puffer telítettségnek megfelelően.





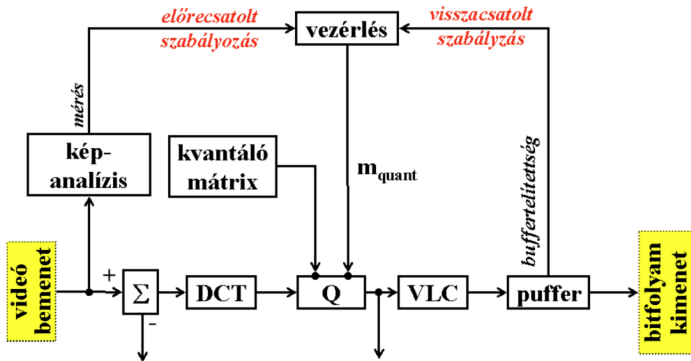
# MPEG bitsebesség vezérlés II.

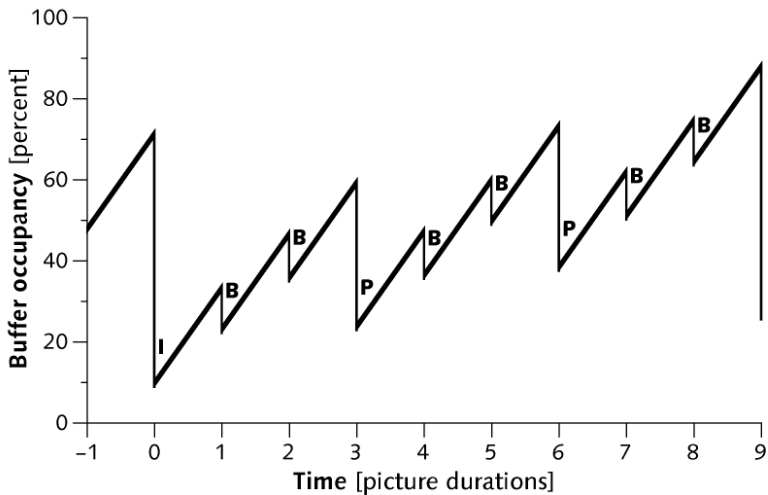
## Változó (VBR)

- Két alapvető típusa:
  - Nyílt hurkú: nincs puffer szabályozás
  - Visszacsatolt: van puffer szabályozás
- Nyílthurkú VBR használható, ha a bitfolyam olvasás vezérelhető eszközről történik (pl. DVD, memória, stb), ekkor a puffer gyakorlatilag végtelen nagy
- Nyílthurkú VBR-ek két alaptípusa:
  - Konstans skálafaktorú kódolás: a kvantálási zajt egyenletesen teríti a képen (nem feltétlenül a legjobb képminőség)
  - Konstans képminőségű kódolás: a skálafaktort a képminőséghez igazítja
- A visszacsatolt VBR lényege, hogy a változó bitsebességet puffereléssel simítjuk mielőtt a bitfolyam a VBR csatornába kerül, így a sebesség egy előre definiált csúcssebesség (max. bitrate) alatt marad
- A VBR kódolás kb. 20-30 %-al kisebb átlagos bitsebességet eredményez, mint az azonos minőségű CBR kódolás

# MPEG bitsebesség vezérlés vázlata

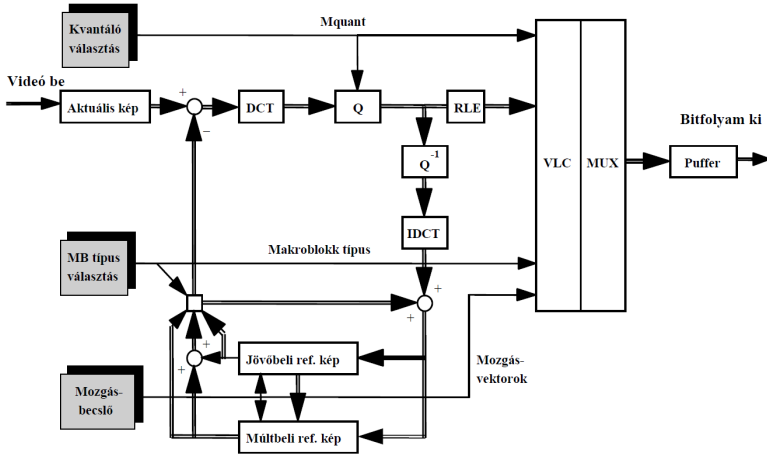
Lehetőségek: Előrecsatolt bitsebesség szabályzás, globális (kép) szintű, és lokális (MB) szintű komplexitás elemzéssel vagy visszacsatolt szabályzás a buffer telítettségének figyelésével

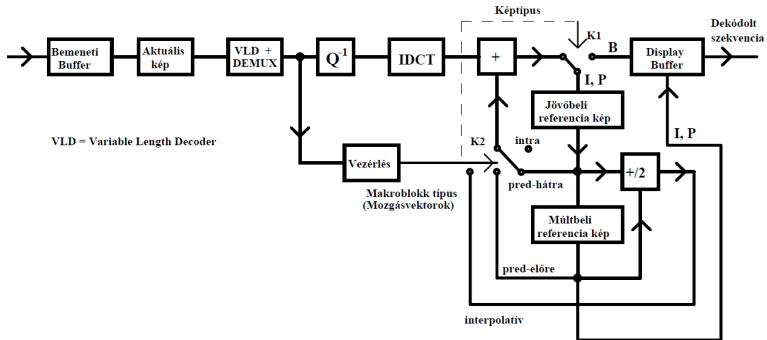






# MPEG-1 kódoló







# MPEG-1 kódolás alapparaméterei

---

- 4:2:0-ás színkülönbségi jel mintastruktúra
- Képfrekvenciák: 24, 25, 29.97, 30 kép/sec
- Max. képméret: 4095x4095
- Csak progresszív képek kódolása
- Letölthető kvantáló mátrixok
- I,P,B képek
- Elérhető tömörtés: 1:25 - 1:30



# MPEG-2 kódolás alapparamétere

---

- MPEG-1 -re épül, vele felülről kompatibilis
- Váltott-soros videó kezelés, félkép alapú feldolgozás (DCT és mozgásbecslés) is lehetséges
- Réteges kódolási módok (elavult)
- Profile-Level szerkezet
- Egy videón belül a kép és félkép alapú feldolgozás keverhető



# MPEG-2 videó kódolás

- Az MPEG-2 videó szintaktikailag az MPEG-1-re épül, s azzal felülről kompatibilis, tehát egy MPEG-2 dekódernek tudnia kell dekódolni egy MPEG-1 bitfolyamot.
- Legfontosabb eltérések:
  - Váltott-soros szekvenciák kezelése is, frame (kép) és field (félkép) alapú feldolgozás.
  - Új kódolási módok.
  - Léptékelhetőség.
- A frame kódolásakor a bennük lévő két félképet együtt, míg field esetén a félképeket egymástól függetlenül kezelik.
- Egy videó szekvencián belül a frame és field keverhető.
- A műsorszóró alkalmazások mind kettőt, a kis késleltetésű interaktív rendszerek általában a field-et használják.





# MPEG-1 és MPEG-2 videó közötti különbségek

---

- Progresszív mellett váltottsoros képek kezelése is:
- kép-félkép alapú adaptív DCT
- új félkép alapú mozgásbecslési algoritmusok:
- félkép alapon félképre
- félkép alapon képre
- Dual Prime predikció (félképre, ha nincs B kép)
- MPEG-1-nél magasabb bitsebességek, nagyobb képméret
- Csak félpixeles mozgásbecslés (egész pixeles már nincs), a cikk-cakk letapogatás helyett alternatív letapogatás is
- 4:2:2 és 4:4:4 mintavétel is (MPEG-1-ben csak 4:2:0)
- Profil és szint szerkezet
- Skálázható kódolás is: jel-zaj viszony és térbeli skálázás



# MPEG-2 Profile-Level szerkezet

- Különböző osztályokat definiál, melyek meghatározzák egy adott Profile/Level-nek megfelelő bitfolyam dekódolásához szükséges dekóder képességeket és kapacitást
- A **profile** a mintavételi formátumokat, a képtípusokat, és az egyéb kódolási eszközöket definiálja, illetve korlátozza
- A **level** a képméretet, a képfrekvenciát, és a bitsebességet definiálja, illetve korlátozza
- Célzott alkalmazási területek Level-enként:
  - Low: videokonferencia
  - Main: SDTV
  - 422-ML: stúdiótechnika
  - High: HDTV

# MPEG-2 Profile-Level tulajdonságok

Minden, adott Profile/Level szintnek megfelelő dekóder a tőle balra álló Profile/Level szinteket is dekódolni tudja. Professzionális és konzumer elterjedés szempontjából a sárga területek jelentősek.

## Szintek (Levels)

	Low	Main	High-1440	High	422P
<b>Horizontális méret</b>	352	720	1440	1920	720
<b>Vertikális méret</b>	288	576	1152	1152	608
<b>Képfrekvencia (Hz)</b>	30	30	60	60	30
<b>Bitsebesség (Mbit/s)</b>	4	15	60	100	50

## Profilok (Profiles)

	Simple	Main	SNR	Spatial	High	422P
<b>YUV</b>	4:2:0	4:2:0	4:2:0	4:2:0	4:2:0 4:2:2	4:2:0 4:2:2
<b>Képtípus</b>	I, P	I, P, B	I, P, B	I, P, B	I, P, B	I, P, B
<b>Skálázhatóság</b>	nem	nem	igen	igen	igen	igen

# MPEG-2 Profile-Level tulajdonságok

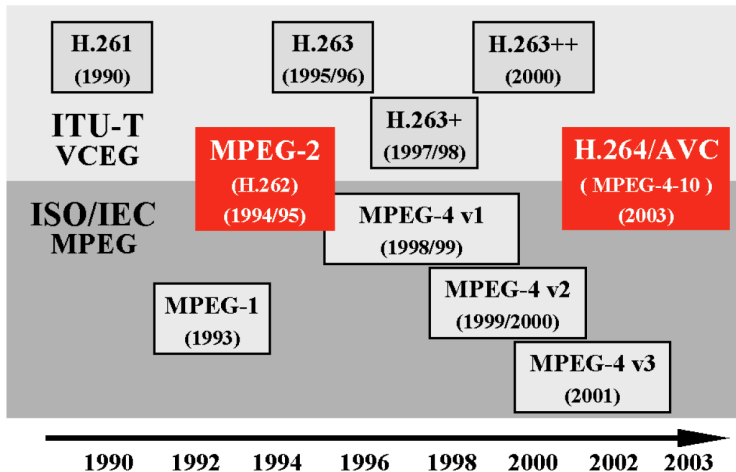
<i>Profile</i> @Level	MPEG-1 CPB	Simple (no B pictures)	Main (MP)	4:2:2 (422P)
High (HL)			1920×1152 60 Hz 80 Mb/s	1920×1088 60 Hz 300 Mb/s
High-1440 (H14)			1440×1152 60 Hz 47 Mb/s	
Main (ML)		720×576 30 Hz 15 Mb/s	720×576 30 Hz 15 Mb/s	720×608 30 Hz 50 Mb/s
Low (LL)			352×288 30 Hz 4 Mb/s	
MPEG-1 CPB † <i>max 99 Kpx</i>	768×576 <sup>†</sup> 30 Hz 1.856 Mb/s			



# MPEG-2 Profile-Level tulajdonságok

<i>Profile@Level</i>	<i>Image columns (<math>N_C</math>)</i>	<i>Image rows (<math>N_R</math>)</i>	<i>Frame rate, Hz</i>	<i>Luma rate [samples/s]</i>	<i>Bit rate [Mb/s]</i>	<i>VBV size [KBytes]</i>
422P@HL	1920	1088	60	62,668,800	300	5,760
MP@HL	1920	1088	60	62,668,800	80	1,194
MP@H-14	1440	1088	60	47,001,600	60	896
422P@ML	720	608	60	11,059,200	50	1,152
MP@ML	720	576	30	10,368,000	15	224
MP@LL	352	288	30	3,041,280	4	58

# Video tömörítés történelem 1990-2003 között





# MPEG-4

- Az MPEG-4 kiterjeszti az MPEG-2 korlátozott adat modelljét a multimédia alkalmazások számára is
- Az MPEG-4 olyan reprezentációs szabvány, mely lehetővé teszi az interaktivitást az audiovizuális objektumokkal, és
- közösen alkalmazható technikát biztosít számos konvergáló szolgáltatási formának
- A stúdiótechnika szempontjából kevésbé fontos



# MPEG-4 (Part 2) és H.263 I.

---

- Számos profile és Level, melyek célul tűzték ki a teljes audiovizuális alkalmazási spektrum lefedését: alacsony bitsebességű alkalmazások (videokonferencia, zárt láncú biztonsági kamera hálózatok), közepes és magas bitsebességű alkalmazások (konzumer video, DVD, HDTV műsorszórás), és stúdió alkalmazások
- Az Advanced Simple Profile (ASP) implementált változatai a DivX (kivéve a Divx Plus HD, ami már H.264/AVC), és az XVID (open source)
- Az ITU által szabványosított kódolási változat a H.263, mely MPEG-4 video dekóder által dekódolható, és internetes videotelefon és internetes média alkalmazások (Flash video, stb.) legelterjedtebb kódolási módja volt a H.264 megjelenéséig
- Sokat vártak tőle, de csak részben váltotta be a hozzáfűzött reményeket





# MPEG-4 (Part 2) és H.263 II.

## Kódolási elemek, eszközök

- Alapvető kódolás elemek, módszerek megegyeznek az MPEG-1/MPEG-2 - vel
- 1/4-pixel pontosságú mozgásbecslés
- Globális (frame) alapú mozgáskompensáció (pan, rotate, zoom, és warp definiálása az egész képre), mely a teljes képtartalom mozgása esetén (pl. kameramozgás) esetén hatékonyabban leírja a mozgást, mint a blokk alapú mozgásbecslés - a gyakorlati implementációkban nem terjedt el, ezért a H.264-be már nem is került bele
- Átlapolt blokk alapú mozgáskompensáció (blokkosodás csökkentése érdekében), az implementációja nem terjedt el
- Simple Studio Profile támogatja a 10/12 bit és 4:2:2, 4:4:4 mintaformátumokat (HDCAM SR elterjedt professzionális formátuma)



# MPEG-4 (Part 10) H.264/AVC

- Közös fejlesztése az ITU-T Video Coding Experts Group-nak és az ISO Joint Video Team-nek, ezért ITU-T H.264 szabvány, és ISO/IEC MPEG-4 AVC szabvány is !
- Jobb video minőség, mint az MPEG-2 -vel
- Jelentős kompresszió növekedés (kétszeres, vagy akár háromszoros tömörítési hatékonyság azonos képminőség mellett, az MPEG-2-höz képest)
- Javított predikció
- Javított mozgáskompensáció pontosság
- Javított entrópia-kódolási hatékonyság
- Profile-Level szerkezet
- Széles alkalmazási spektrum (videotelefon, internet-média, digitális műsorszórás), kezdetben elsősorban "entertainment" (tehát SD és kisebb felbontású) alkalmazásokra
- A Fidelity Range Extensions (FREXT) kiegészítéssel professzionális (stúdió, digitális mozi, stb) alkalmazásokra is



# MPEG-4 H.264/AVC Profilok

	<i>Profile</i>	<i>Baseline (BP)</i>	<i>Extended (XP)</i>	<i>Main (MP)</i>	<i>High (HiP)</i>
<i>Kódolási eszköz</i>					
Több referencia kép használata		•	•	•	•
Változó blokkméretű mozgásbecslés		•	•	•	•
I- és P-szeletek		•	•	•	•
1/4 pixel pontosságú mozgásbecslés		•	•	•	•
16 bites aritmetikájú integer transzformáció		•	•	•	•
UVLC		•	•	•	•
CAVLC		•	•	•	•
Deblocking szűrő		•	•	•	•
Flexibilis Makroblokk sorrend (FMO)		•	•		
Tetszőleges szelet sorrend (ASO)		•	•		
Redundáns szeletek (RS)		•	•		
Adat partícionálás			•		
SI és SP szeletek			•		
B-szeletek			•	•	•
Interlace kódolás (PICAFF, MBAFF)			•	•	•
Súlyozott (és ofszet) inter-predikció			•	•	•
CABAC entrópiakódolás				•	•
8x8 luma intra predikció					•
Nagyobb bitmélységek (8bitnél)					•
4:4:4 és 4:2:2 alumintavételezés					•
Veszteségmentes inter-predikció					•
Adaptív 8x8/4x4 transzformáció váltás					•
Kódoló által specifikálható kvantálási mátrixok					•
Cb és Cr független kvantálása (független Qstep)					•
Monkróm (alfa)csatorna kódolás					•



# MPEG-4 H.264/AVC Szintek

<i>Level</i>	<i>Typ. image format</i>	<i>Typ. frame rate [Hz]</i>	<i>Max. bit rate [b/s]</i>
L1	176×144	15	64 k
L1b	176×144	15	128 k
L1.1	352×288 or 176×144	7.5 or 30	192 k
L1.2	352×288	15	384 k
L1.3	352×288	30	768 k
L2	352×288	30	2 M
L2.1	352×480 or 352×576	30 or 25	4 M
L2.2	SD	15	4 M
L3.0	SD	30 or 50	10 M
L3.1	1280×720	30	14 M
L3.2	1280×720	60	20 M
L4.0	1920×1080	30	20 M
L4.1	1920×1080	30	50 M
L4.2	1920×1080	60	50 M
L5	2048×1024	72 or 30	135 M
L5.1	4096×2048	30	240 M

- Új profilkövetelmények:
  - 8 bit/mintánál nagyobb bitmélységek támogatása
  - 4:2:2 és 4:4:4 mintastrukturák támogatása
  - Video-kulcsolás, és transzparencia (alpha csatorna) támogatása a stúdióalkalmazások céljára
  - Nagyobb adatsebességek támogatása
  - Veszteségmentes tömörítés támogatása (lináris PCM, vagy transzformációs kódolás nélküli entrópiakódolás)
  - A színtér transzformációk numerikus hibáinak elkerülése (YCgCo színtér)
  - RGB (nem csak YCbCr) komponens reprezentáció támogatása
- $8 \times 8$ -as transzformáció
- Mára a FRExt High Profile-jai fontosabbá váltak (mind konzumer, mind profi felhasználásra), mint a Main profile konzumer profiljai, mert számottevően jobb kódolási hatékonyságot biztosítanak, miközben az implementálás komplexitása (pl. kódoló, dekódoló hardver) nem nőtt jelentősen



# MPEG-4 H.264/AVC FRExt profiljai

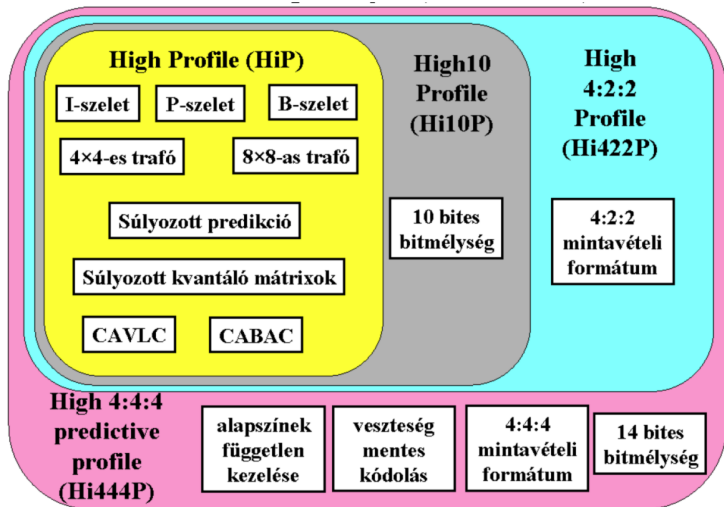
---

- High Profile (HP): 8 bit/minta, 4:2:0 mintastruktúra, csúcsminőségű fogyasztói célokra, ha nem szükséges a kiterjesztett szín mintastruktúra
- High 10 Profile (Hi10P): 10 bit/minta, 4:2:0 mintastruktúra
- High 4:2:2 Profile (H422P): 10 bit/minta, 4:2:2 mintastruktúra
- High 4:4:4 Profile (H444P): 12 bit/minta, 4:4:4 mintastruktúra, járulékosan támogatja a veszteségmentes kódolást

A H.264/AVC HD Intra kiterjesztésének profiljai (pl. High 10 Intra Profile) megegyezik a fenti profilokkal, de csak intra kódolási eszközöket tartalmaz, professzionális kamera és editáló alkalmazások (stúdió) céljára



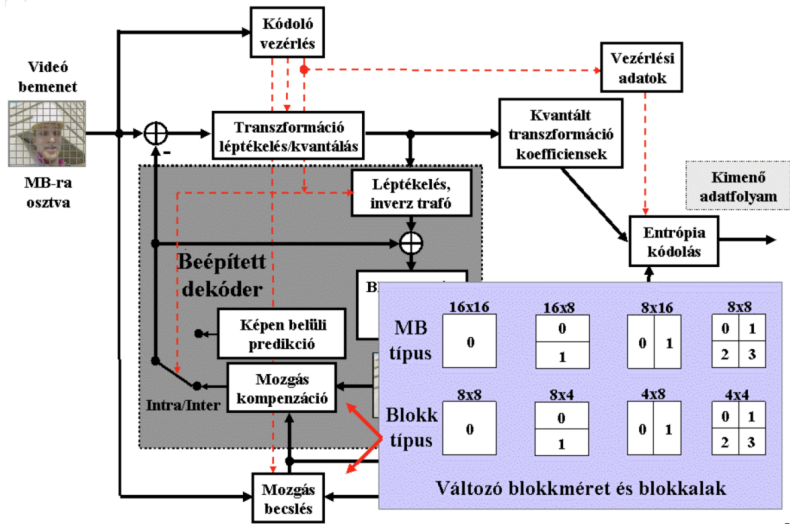
# MPEG-4 H.264/AVC FRExt profiljai





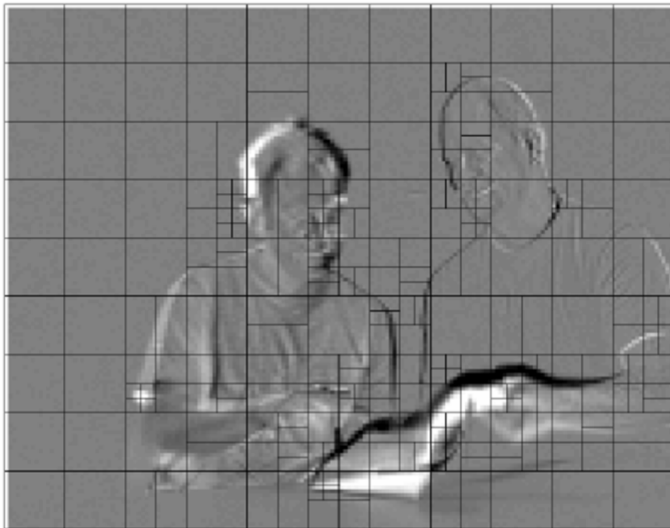


# MPEG-4 H.264/AVC MB méretek





# MPEG-4 H.264/AVC MB méretek





# MPEG-4 H.264/AVC részletek

- A MB-ok  $16 \times 16$ -os méretűek, azon belül a mozgáskompenzáció szub-MB-en történik, melyek  $16 \times 16$ ,  $16 \times 8$ ,  $8 \times 16$ ,  $8 \times 8$ ,  $4 \times 8$ ,  $4 \times 8$  és  $4 \times 4$  méretűek lehetnek
- A predikciós hiba kódolása  $4 \times 4$ -es vagy  $8 \times 8$ -as (FRExt) blokkokon történik
- Integer transzformáció a DCT helyett
- Képfelépítés: Szekvencia - Kép - Szelet - MB - szub-MB - blokk - minta
- Blokkosodást csökkentő szűrés: jelentősen csökkenti a predikciós hiba maradványait, a blokk határokon megjelenő élek simításával. Javul a dekódolt kép minősége, ezáltal az azonos minőséghez szükséges adatsebesség csökkenthető 5-10%-al
- Alapvetően kétfajta képtípus: referencia és nem-referencia kép (predikció szempontjából)
- Képeken belül I,P,B típusú képszeletek
- I,P,B típusú MB-ok (pl. lehet intra MB egy P szeleten belül)



# MPEG-4 H.264/AVC részletek

- Intra- predikciós újdonságok
  - Intra predikció teljes MB-on, vagy blokkokon (4x4, vagy 8x8 (FRExt) blokkon): a MB-ok, illetve blokkok prediktív kódolása
- Inter- predikciós újdonságok:
  - Több referenciaképes mozgáskompenzáció
  - B képszeletek használata referenciaként is
  - Súlyozott predikció
  - A mozgásvektorok meghatározása 1/4 ill. 1/8 pixel pontossággal történhet.
  - Különböző méretű szub-MB alapú mozgáskompenzáció
  - Field/frame alapú predikció eldöntése képszinten (Picture-Adaptive Frame/Field coding), vagy MB szinten (Macroblok Adaptive Frame-Field Coding)
- Új entrópia kódolási módok (CABAC, CAVLC) - részletekkel nem foglalkozunk
- A vizsgáltak szerint az SD/HDTV alkalmazásokban a H.264/AVC a 4/9 - 2/5 -részére csökkenti az MPEG-2-höz képesti adatsebességet



## Integer transzformáció I.

- Az MPEG-1/MPEG-2 DCT transzformációja idealizált trigonometrikus függvényeket használ. A dekódoló beépítése a kódoló hurokba megszünteti a DCT együtthatók újra-kvantálásából származó predikciós hiba-akkumulálódást, de nem garantálható, hogy a tényleges dekóder pontosan állítja elő a DCT bázisfüggvényeit (sin,cos). Az MPEG-1/MPEG-2 a DCT bázisfüggvények számítására vonatkozóan hibatoleranciákat definiál
- A H.264/AVC-nél definiált 16 bites fixpontos aritmetikával implementálható transzformáció nem okoz eltérést a kódoló és dekódoló bázisfüggvényei között
- A H.264/AVC integer transzformációja közelítő DCT-ként is ismert (az integer transzformáció mátrixa a 4x4-es 1D DCT transzformációs mátrix megfelelően skálázott, legközelebbi egész értékekre kerekített, majd ortonormált elemeiből áll)
- Számítása emellett egyszerűbb, mint a DCT-é (a 4x4-es transzformáció csak összeadást, kivonást, bit-shiftet igényel)
- A 4x4-es transzformáció kisebb bitsebesség esetén fellépő hibája (nem folytonos blokkhatárok: pl. blokkosodás és ringing) kevésbé látható (granuláris eloszlású), mint az MPEG 8x8-as blokkjai esetében



## Integer transzformáció II.

- Bizonyos predikciós módokban a  $16 \times 16$ -os integer transzformált blokkon egy további Hadamard transzformációt is végrehajtanak (ld. később)
- Szintén újdonság, hogy a kvantálási skálafaktor (bitrate vezérlés) nem lineárisan, hanem logaritmikusan növeli a kvantálási lépcsőket (8 bites esetben 52 kvantálási skálafaktor, ahol 6 érték növelés pontosan egy bittel növeli a kvantálási szóhosszt)
- A FRExt profil MB szinten adaptívan válthat a  $4 \times 4$ -es, illetve  $8 \times 8$ -as transzformáció között



## 4x4-es integer transzformáció mátrixa

A 4x4-es 1D DCT közelítő transzformációs mátrixa

$$\mathbf{T}_{4 \times 4} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix} \quad \mathbf{D}_{st} = \begin{bmatrix} 1/2 & & & \\ & 1/\sqrt{10} & & \\ & & 1/\sqrt{2} & \\ & & & 1/\sqrt{10} \end{bmatrix}$$

ahol a diagonális  $\mathbf{D}_{st}$  mátrix-al való szorzás  $\mathbf{T}_{4 \times 4}$  elemeit ortonormálja

$$(D_{st}[i, i] = \frac{1}{\sqrt{\sum_j (T[i, j])^2}})$$

Ezek alapján a transzformáció felírható:

$$\mathbf{A}_1 = \mathbf{D}_{st} \mathbf{T}_{4 \times 4} \quad \mathbf{Y} = \mathbf{A}_1 \mathbf{X} \mathbf{A}_1^T$$



## 4x4-es integer transzformáció mátrixa (folytatás)

A gyakorlatban a  $\mathbf{D}_{st}$  diagonális mátrix-al való balról-jobbról szorzást a főátló elemeiből alkotott vektor diadikus szorzatából képzett mátrix-al való elemenkénti szorzással ( $\star$ ) valósítják meg:

$$\mathbf{Y} = \mathbf{D}_{st} \mathbf{T}_{4 \times 4} \mathbf{X} (\mathbf{D}_{st} \mathbf{T}_{4 \times 4})^T = \mathbf{D}_{st} \mathbf{T}_{4 \times 4} \mathbf{X} \mathbf{T}_{4 \times 4}^T \mathbf{D}_{st}^T = (\mathbf{T}_{4 \times 4} \mathbf{X} \mathbf{T}_{4 \times 4}^T) \star (\mathbf{d}_{st} \mathbf{d}_{st}^T)$$

vagyis

$$\mathbf{Y} = (\mathbf{T}_{4 \times 4} \mathbf{X} \mathbf{T}_{4 \times 4}^T) \star \mathbf{S}_t$$

ahol

$$\mathbf{S}_t = \mathbf{d}_{st} \mathbf{d}_{st}^T$$

A transzformált együtthatók újrakvantálása a következőképpen valósul meg:

$$\mathbf{Y}_q = \frac{1}{2^{15}} (\mathbf{T}_{4 \times 4} \mathbf{X} \mathbf{T}_{4 \times 4}^T) \star \mathbf{W}_t$$

$$\mathbf{W}_t = \frac{2^{15}}{Q_{step}} \mathbf{S}_t$$



## 4x4-es integer inverz transzformáció mátrixa

$$\mathbf{T}_{4 \times 4} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1/2 & -1/2 & -1 \\ 1 & -1 & -1 & 1 \\ 1/2 & -1 & 1 & -1/2 \end{bmatrix} \quad \mathbf{D}_{si} = \begin{bmatrix} 1/2 & & & \\ & \sqrt{2/5} & & \\ & & 1/2 & \\ & & & \sqrt{2/5} \end{bmatrix}$$

$$\mathbf{A}_2 = \mathbf{D}_{si} \mathbf{T}_{4 \times 4} \quad \mathbf{X} = \mathbf{A}_2^T \mathbf{Y} \mathbf{A}_2$$



4x4-es integer inverz transzformáció mátrixa (folyt.)

Hasonlóan az előzőekhez:

$$\mathbf{A}_2 = \mathbf{D}_{si} \mathbf{T}_{4 \times 4} \quad \mathbf{X} = \mathbf{A}_2^T \mathbf{Y} \mathbf{A}_2$$

$$\mathbf{X} = (\mathbf{D}_{si} \mathbf{T}_{4 \times 4})^T \mathbf{Y} (\mathbf{D}_{si} \mathbf{T}_{4 \times 4}) = \mathbf{T}_{4 \times 4}^T \mathbf{D}_{si}^T \mathbf{Y} \mathbf{D}_{si} \mathbf{T}_{4 \times 4} = \mathbf{T}_{4 \times 4}^T (\mathbf{Y} \star \mathbf{S}_i) \mathbf{T}_{4 \times 4}$$

ahol

$$\mathbf{S}_i = \mathbf{d}_{si} \mathbf{d}_{si}^T$$

Az inverz kvantálás a következőképpen valósul meg:

$$\mathbf{X} = \frac{1}{2^6} \mathbf{T}_{4 \times 4}^T (\mathbf{Y}_q \star \mathbf{W}_i) \mathbf{T}_{4 \times 4}$$

ahol

$$\mathbf{W}_i = 2^6 Q_{step} \mathbf{S}_i$$



## Intra predikció

- I képszeleteken belül a teljes MB (16x16 pixel), vagy egy 4x4-es, vagy egy 8x8-as (FRExt) blokk prediktíven kódolt a szomszédos, már korábban kódolt/dekódolt blokkok pixeleiből. A predikció az adott blokk szomszédos pixeleiből történik többféle módon: a predikció iránya lehet vízszintes, függőleges, DC alapú (átlag) vagy síkbeli (lin. függvénykapcsolat)
- A kódoló által választható 4x4-es intra predikció esetén 9 predikciós irány közül választhat a kódoló, melyek irányok a szomszédos pixelek különböző együtthatókkal képzett lineáris kombinációjaként állíthatók elő
- A H.264/AVC intra predikciója jelentősen megnöveli a tömörítés hatékonyságát szemben az MPEG-1/2-vel, ahol csak a DC komponensek differenciális kódolását alkalmazzák az intra MB-k esetében
- 16x16-os intra predikció esetén a 4x4-es integer transzformációból eredő blokkok DC együtthatóit (16 DC eü.) egy 4x4 blokk alapú Hadamard transzformációval kódolják



Intra\_4x4\_Vertical prediction mode

$\text{pred4x4}[x, y] = p[x, -1]$ , with  $x, y = 0..3$

Intra\_4x4\_Horizontal prediction mode

$\text{pred4x4}[x, y] = p[-1, y]$ , with  $x, y = 0..3$

Intra\_4x4\_DC prediction mode

$\text{pred4x4}[x, y] = ( p[0, -1] + p[1, -1] + p[2, -1] + p[3, -1] + p[-1, 0] + p[-1, 1] + p[-1, 2] + p[-1, 3] )$

Intra\_4x4\_Diagonal\_Down\_Left prediction mode

$\text{pred4x4}[x, y] = ( p[x + y, -1] + 2 * p[x + y + 1, -1] + p[x + y + 2, -1] )$

Intra\_4x4\_Diagonal\_Down\_Right prediction mode

If  $x$  is greater than  $y$ ,

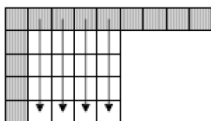
$\text{pred4x4}[x, y] = ( p[x - y - 2, -1] + 2 * p[x - y - 1, -1] + p[x - y, -1] )$

Otherwise if  $x$  is less than  $y$ ,

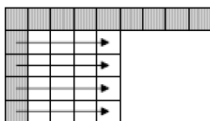
$\text{pred4x4}[x, y] = ( p[-1, y - x - 2] + 2 * p[-1, y - x - 1] + p[-1, y - x] )$

## Intra predikció példák

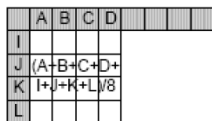
Mode 0: Vertical



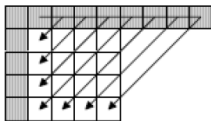
Mode 1: Horizontal



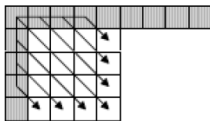
Mode 2: DC



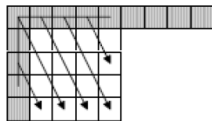
Mode 3: Diagonal-Down-Left



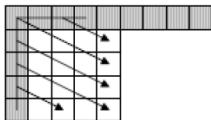
Mode 4: Diagonal-Down-Right



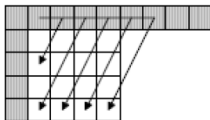
Mode 5: Vertical-Right



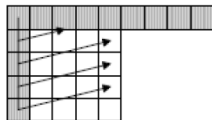
Mode 6: Horizontal-Down





Mode 7: Vertical-Left



Mode 8: Horizontal-Up



 Already coded and reconstructed pixels of neighboring blocks in the same frame

 Pixels of the current 4x4 block



## Veszteségmentes MB mód

- Nagy bitsebességek mellett (kis kvantálási lépcsőknél) a transzformációs kódolás, predikció, mozgáskompensáció adat overheadje miatt a kódoló tetszőleges MB-ra dönthet úgy, hatékonyabb a MB PCM kódolása (csak entrópiakódolás). A PCM makroblokk mód adatsebessége definiálja egyúttal a dekóder számára az adott képméret, képfrekvencia melletti legnagyobb kezelendő bitsebességet is.
- A FRExt definiál továbbá egy transzformációs kódolás nélküli prediktív kódolást+entrópiakódolást is.



## Súlyozott predikció és mozgáskompenzáció I.

- Több fajta sub-MB méret alapú mozgásbecslés (minden szub-MB-hoz egyedi mozgásvektorok).
- A H.264/AVC lehetőséget biztosít több referenciakép használatára a P és B típusú predikciókhoz
- Ez azt jelenti, hogy míg az MPEG-2-nél csak az adott prediktívan kódolt képet közvetlenül megelőző I, vagy P, kép (illetve kétirányú predikciónál az adott prediktívan kódolt képet közvetlenül megelőző, illetve közvetlenül követő I, illetve P képek) lehet a predikció alapja, a H.264 kódoló több, az adott képet megelőző, illetve követő kép közül választhat (és B kép is lehet referencia !), hogy melyiket használja referencia képként az adott kép mozgáskompenzált prediktív kódolásához. A felhasználható referenciaképek számát (a referenciaképtároló méretét) az adott Level szabályozza.
- A referenciaképek kiválasztása viszont MB alapú.



## Súlyozott predikció és mozgáskompenzáció II.

- Jelentős újdonság, hogy a referencia kép(ek) tetszőleges skála és offset tényezőkkel súlyozhatók, majd ezen skálázott referenciaképeken hajtódik végre a mozgáskompenzált predikció.
- Egyirányú predikció (P képek) esetében ez hatékony kódolási lehetőséget kínál pl. a Fade-in, Fade-out típusú effektek kódolásához, míg kétirányú predikció esetén a crossfade jellegű effektusok hatékony kódolásához
- A P és B típusú kép(szelet)ek esetén a mozgásvektorok is differenciálisan kódoltak
- A mozgásvektorok a képen belüli korábbi 3 szomszédos MB mozgásvektorainak mediánjához képest differenciálisan kódoltak
- A kódoló el is hagyhatja a mozgásvektor átvitelét, és utasíthatja a dekódert, hogy az adott blokk környezetének mozgásvektorai alapján becsülje a hiányzó mozgásvektort





## De-blocking filter

- A prediktív kódoló hurokba épített deblocking szűrő a blokkosodást csökkenti a szomszédos blokkok pixeleinek térbeli aluláteresztő szűrésével.
- Természetesen a szűrt képek a további predikció alapja (predikciós hibaterjedés ellen)
- A szűrő erősségét a MB típusa (I,P,B), a kvantálási súlyozótényező, és egyéb tényezők szabályozzák
- Alapvetően minél nagyobb kvantálási lépcsővel történik a MB kódolása, annál erősebb a szűrő hatása



# MPEG-4 H.264/AVC részletek

## Új szintér: YCgCo (FRExt)

- Az YCbCr komponensek kiszámítása pl. az RGB komponensekből lineáris transzformációt igényel, melyben az együtthatókkal való szorzás lebegőpontos műveleteket igényel, ez nem mindig hajtható végre ideális pontossággal
- Az integer transzformáció alapú kódolás esetén ezen lebegőpontos YCbCr komponensek kódolása elkerülhetetlenül kerekítési hibák forrása
- Az új szintér egy új Y komponens mellett a Cg (green chroma) és Co (orange chrome) komponensekből áll
- Számításuk :  $Y = \frac{1}{2} \left( G + \frac{(R+B)}{2} \right)$ ,  $Cg = \frac{1}{2} \left( G - \frac{(R+B)}{2} \right)$ ,  
 $Co = \frac{(R-B)}{2}$
- 4:4:4-es komponens (RGB) formátum esetén arra is lehetőség van, hogy csak a predikációs hiba komponenseken hajtja végre a kódoló az YCgCo transzformációt, a kóder ill. dekóder képtárolóiban (referencia képtárolók) a képek RGB ábrázolásban vannak (residual color transform)

	<i>Profile</i>	<i>Baseline (BP)</i>	<i>Extended (XP)</i>	<i>Main (MP)</i>	<i>High (HiP)</i>
<i>Kódolási eszköz</i>					
Több referencia kép használata		•	•	•	•
Változó blokkméretű mozgásbecslés		•	•	•	•
I- és P-szeletek		•	•	•	•
1/4 pixel pontosságú mozgásbecslés		•	•	•	•
16 bites aritmetikájú integer transzformáció		•	•	•	•
UVLC		•	•	•	•
CAVLC		•	•	•	•
Deblocking szűrő		•	•	•	•
Flexibilis Makroblokk sorrend (FMO)		•	•		
Tetszőleges szelet sorrend (ASO)		•	•		
Redundáns szeletek (RS)		•	•		
Adat partícionálás			•		
SI és SP szeletek			•		
B-szeletek			•	•	•
Interlace kódolás (PICAFF, MBAFF)			•	•	•
Súlyozott (és ofszet) inter-predikció			•	•	•
CABAC entrópiakódolás				•	•
8x8 luma intra predikció					•
Nagyobb bitmélységek (8bitnél)					•
4:4:4 és 4:2:2 alulmintavételezés					•
Veszteségmentes inter-predikció					•
Adaptív 8x8/4x4 transzformáció váltás					•
Kódoló által specifikálható kvantálási mátrixok					•
Cb és Cr független kvantálása (független Qstep)					•
Monkróm (alfa)csatorna kódolás					•

# MPEG-4 H.264/AVC Szintek, ismétlés

<i>Level</i>	<i>Typ. image format</i>	<i>Typ. frame rate [Hz]</i>	<i>Max. bit rate [b/s]</i>
L1	176×144	15	64 k
L1b	176×144	15	128 k
L1.1	352×288 or 176×144	7.5 or 30	192 k
L1.2	352×288	15	384 k
L1.3	352×288	30	768 k
L2	352×288	30	2 M
L2.1	352×480 or 352×576	30 or 25	4 M
L2.2	SD	15	4 M
L3.0	SD	30 or 50	10 M
L3.1	1280×720	30	14 M
L3.2	1280×720	60	20 M
L4.0	1920×1080	30	20 M
L4.1	1920×1080	30	50 M
L4.2	1920×1080	60	50 M
L5	2048×1024	72 or 30	135 M
L5.1	4096×2048	30	240 M

# HEVC (Highly Efficient Video Coding)

Új kódolási eljárás, a szabvány 2013. július óta elérhető

- A H.264/AVC szabvány végleges elfogadásakor máris elkezdtek vizsgálni a továbbfejlesztési lehetőségeket
- Az alapvető alkalmazási terület megegyezik a H.264/AVC-vel, de annak kódolási hatékonyságát javítani kellett elsősorban a HD és UHD (2K, 4K, akár 8K) alkalmazásokra
- Fő fejlesztési irányok:
  - Nagyobb felbontású (képméretű) formátumok hatékonyabb kódolása
  - Párhuzamos kódolási (és dekódolási) architektúrák támogatása (parallel processing)

## Főbb változások a H.264/AVC-hoz képest I.

- Fa struktúrájú makroblokk-blokk felosztás:
  - CTU (Coding Tree Unit): Y CTB + Cr és Cb CTB (Coding Tree Blocks), lehetséges méretük: 64x64, 32x32, 16x16
  - CU (Coding Units): Y CB + Cr és Cb CB (Coding Block) - vagy megegyezik a CTB méretekkkel, vagy fel van osztva további blokkokra. Legkisebb CB méret: 8x8
  - PB (Prediction Blocks): A CU-k a predikciós módtól függően feloszthatók PB blokkokra, az intra-inter predikció döntés a CU szinten meghatározott, tehát egy CU minden PB-je azonosan intra, vagy inter típusú. A PB blokkméretek 64x64-től 4x4-ig terjedhetnek, intra módban négyzetes (NxN-es), inter módban aszimmetrikus felosztások is lehetségesek
  - TB (Transform Block) az intra/inter predikció hibablokkjainak kódolási egysége: TB méretek: 32x32, 16x16, 8x8, 4x4 - Egy TB több PB-t is lefedhet (tehát a TB mérete lehet nagyobb, mint a PB- k mérete egy CU-n belül, de a TB csak négyzetes lehet)

# HEVC (Highly Efficient Video Coding)

## Főbb változások a H.264/AVC-hoz képest II.

- Transzformáció típusa: Integer DCT (mint a H.264-nél), de csak 32x32-es mátrixra definiálták, kisebb TB méret esetén 32x32-es mátrixot decimálják
- Alternatív 4x4 DST transzformáció: intra predikcióra optimálisabb
- Intra predikció: 33 lehetséges predikciós irány
- Sample Adaptive Offset (SAO): nemlineáris szűrés a kvantálás okozta sávosságok elkerülésére, valamint az élek-kontrasztátmenetek pontosabb rekonstruálása érdekében
- Mozgásvektorok hatékonyabb predikciója a szomszédos PB-k, valamint referencia kép mozgásvektorai alapján

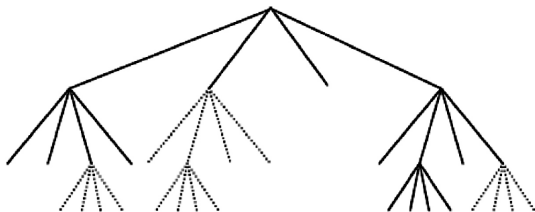
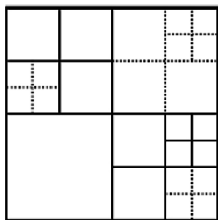
## Főbb változások a H.264/AVC-hoz képest III.

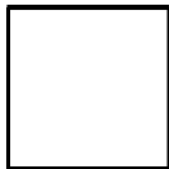
- Parallel processing támogatás:
  - Tile: a szeletek (sliceok) mellett egy képben belül egy önmagában kódolt egység, melynek CTB-i önmagukban dekódolható (egy Tile-on belüli predikciók nem függenek más Tile-ok CTB-itől). Tipikusan négyzet alakú képterületek, melyek párhuzamosan kódolhatók, dekódolhatók.
  - Wavefront Processing: Egy szeleten belül egy újabb CTU sor párhuzamos kódolása/dekódolása megkezdhető, ha az a korábbi CTU sorban legalább kettővel több CTU kódolása/dekódolása már megtörtént
  - Vagy Tile, vagy WPP alapú párhuzamosítás, egyszerre a kettő nem lehetséges



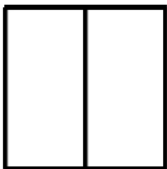


# HEVC CTB/CB/TB felosztás





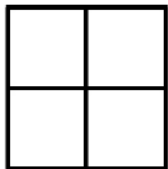
$M \times M$



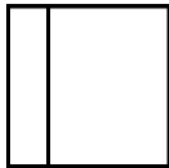
$M/2 \times M$



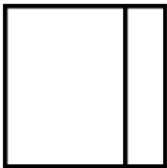
$M \times M/2$



$M/2 \times M/2$



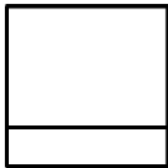
$M/4 \times M$  (L)



$M/4 \times M$  (R)

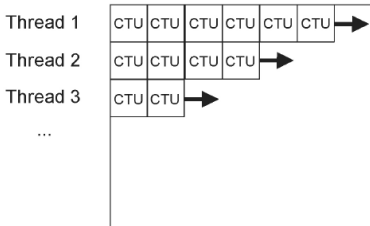
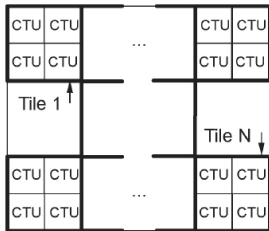
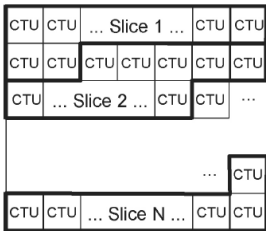


$M \times M/4$  (U)

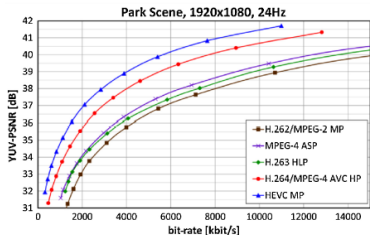
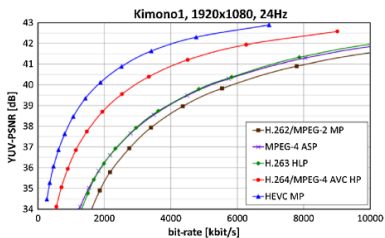


$M \times M/4$  (D)

# HEVC parallel processing lehetőségek



# HEVC kódolási hatékonyság összehasonlítás



AVERAGE BIT-RATE SAVINGS FOR EQUAL PSNR FOR ENTERTAINMENT APPLICATIONS

Encoding	Bit-Rate Savings Relative to			
	H.264/MPEG-4 AVC HP	MPEG-4 ASP	H.263 HLP	MPEG-2/ H.262 MP
HEVC MP	35.4%	63.7%	65.1%	70.8%
H.264/MPEG-4 AVC HP	–	44.5%	46.6%	55.4%
MPEG-4 ASP	–	–	3.9%	19.7%
H.263 HLP	–	–	–	16.2%