

12. Képtömörítés

Kató Zoltán

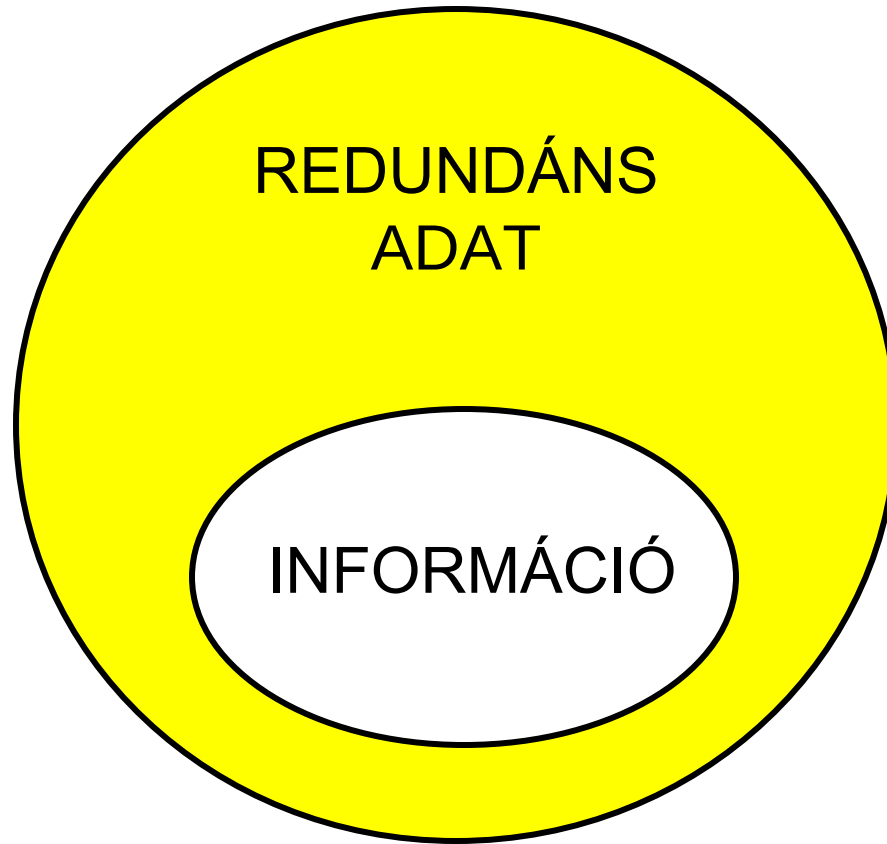
**Képfeldolgozás és Számítógépes Grafika tanszék
SZTE**

(<http://www.inf.u-szeged.hu/~kato/teaching/>)

Miért van szükség tömörítésre?

- A rendelkezésre álló adattárolási és továbbítási kapacitás véges
 - Pl. a tömörítetlen videó bitrátája 1Gbps!
- Mi teszi lehetővé a tömörítést?
 - A természetes képeknek struktúrája van → szomszédos pixelek hasonlóak
 - Egy videó szomszédos frame-einek tartalma nagyjából azonos
- Kép / videó tömörítés: a reprezentációhoz szükséges bitek számát csökkentjük
 - Veszteségmentes: az eredeti kép pontosan visszaállítható
 - Veszteséges: az eredeti kép nem állítható vissza, de vizuálisan nem érzékelünk különbséget.
 - Képelemzéshez általában nem használható!

Információ és adat



Adat = Információ + Redundáns adat

Bitek számának triviális csökkentése

$$\text{Compression Ratio} = 10\text{LOG} \frac{\text{Channel Bit Rate}}{N Q F}$$

- **N = pixelek száma**
 - Csökkenti a felbontást → mintavételezési hibák
- **Q = bitek száma pixelenként (kvantálástól függ)**
 - Artifaktumok (hamis élek)
- **F = frame ráta (videó esetén)**
 - Elmosódás
- **A képminőség megtartása így nem lehetséges**

Információelméleti megközelítés

szimbólumok, forrás-abc:

$$a_1, a_2, \dots, a_J$$

előfordulási valószínűségek:

$$S = (P(a_1), P(a_2), \dots, P(a_J))$$

entrópia (bit/szimbólum):

$$H(S) = - \sum_{j=1}^J P(a_j) \cdot \log_2 P(a_j)$$

Shannon-féle kódolási tétel

Legyen S egy $H(S)$ entrópiájú információforrás.
Ekkor

$$H(S) \leq m(S),$$

ahol $m(S)$ a minimális átlagos kódhossz S veszteségmentes kódolására (vagyis $H(S)$ a veszteségmentes kódoláshoz szükséges átlagos bitszám alsó korlátja).

Huffman kódolás

- A Huffman kódolás egy optimális változó hosszúságú kódot szolgáltat. (TIFF és JPEG is használja.)
- A kódolás egy élcímkézett bináris kódfát épít fel mohó (*greedy*) stratégiával egy minimumos prioritási sor segítségével.
- J szimbólum esetén az eljárás $(J-1)$ -lépéses: egy lépésben a 2 legkisebb valószínűségű elemet (szimbólumot vagy részfat) kiveszünk a sorból és a belőlük alkotott új részfat visszahelyezzük a sorba (a 2 elem valószínűségösszegével).

Huffman kódolás

előfordulás
szimbólum gyakorisága (%)

A	12
B	2
C	7
D	52
E	14
F	13

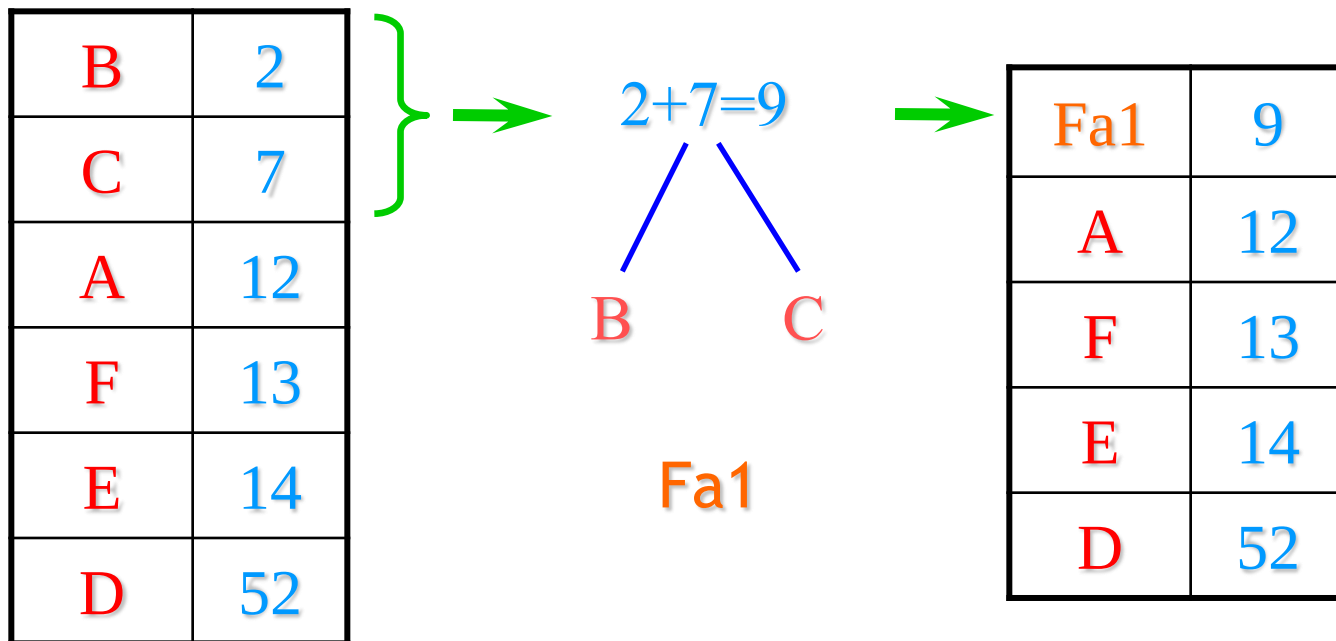
$$\Sigma=100\%$$



minimumos
prioritási sor

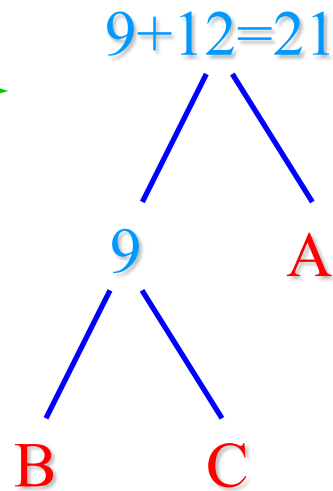
B	2
C	7
A	12
F	13
E	14
D	52

Huffman kódolás



Huffman kódolás

Fa1	9
A	12
F	13
E	14
D	52

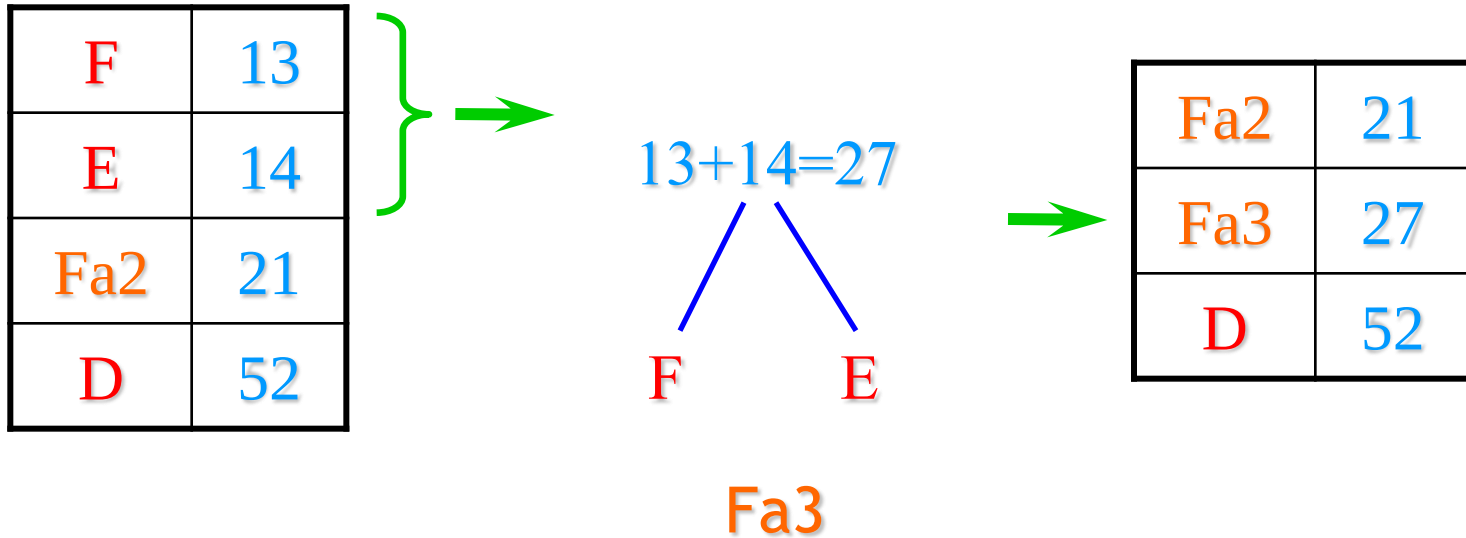


Fa2



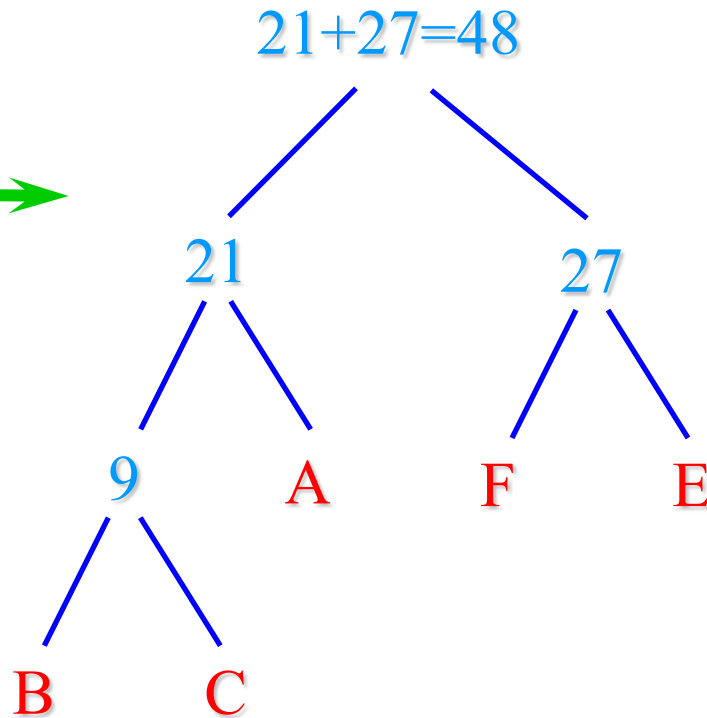
F	13
E	14
Fa2	21
D	52

Huffman kódolás



Huffman kódolás

Fa2	21
Fa3	27
D	52

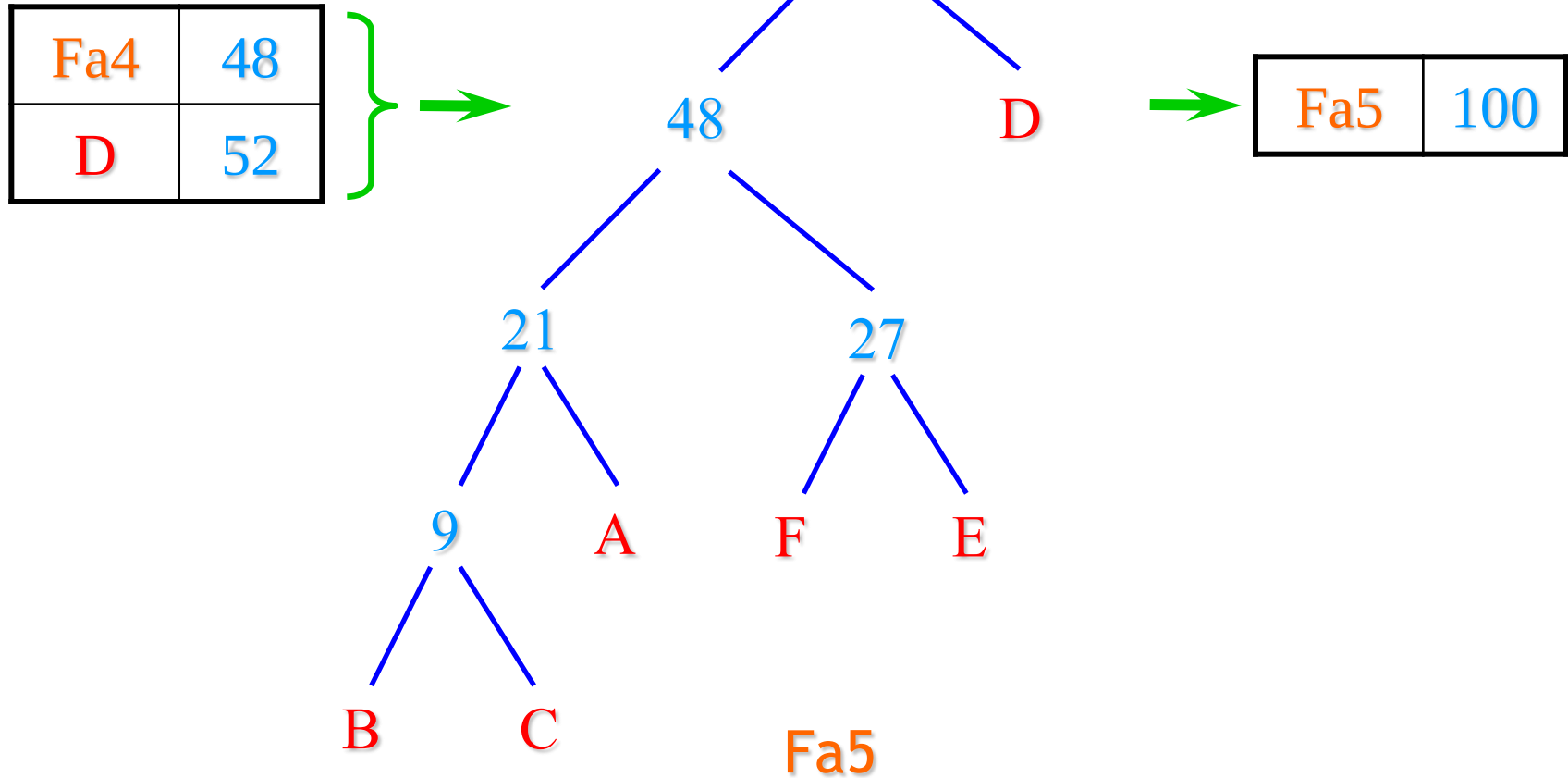


Fa4

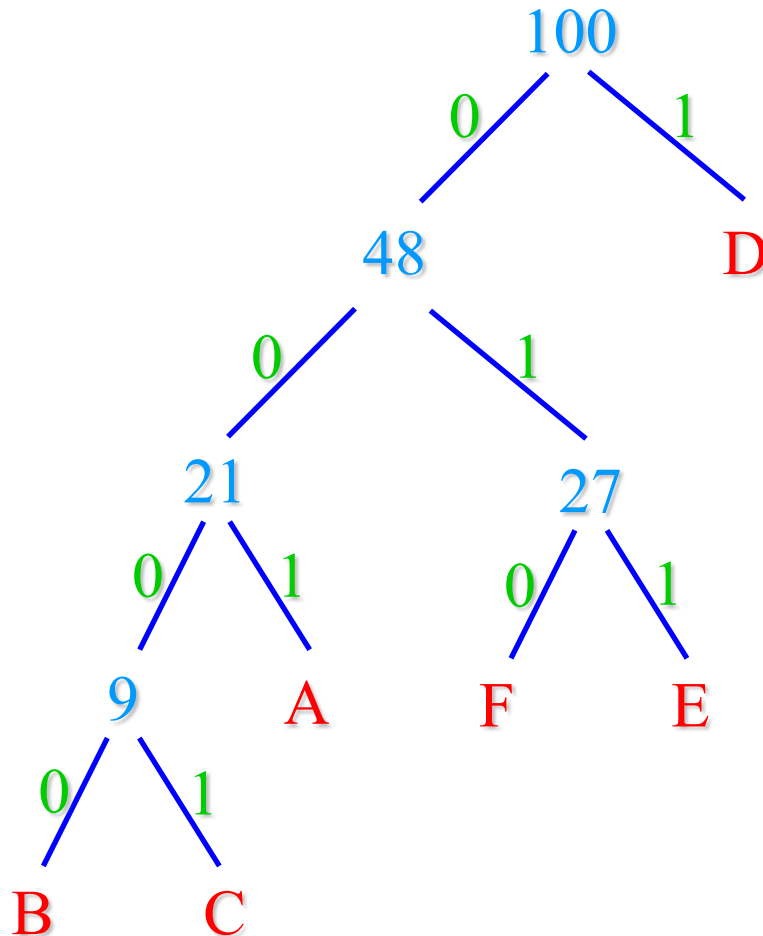


Fa4	48
D	52

Huffman kódolás



Huffman kódolás

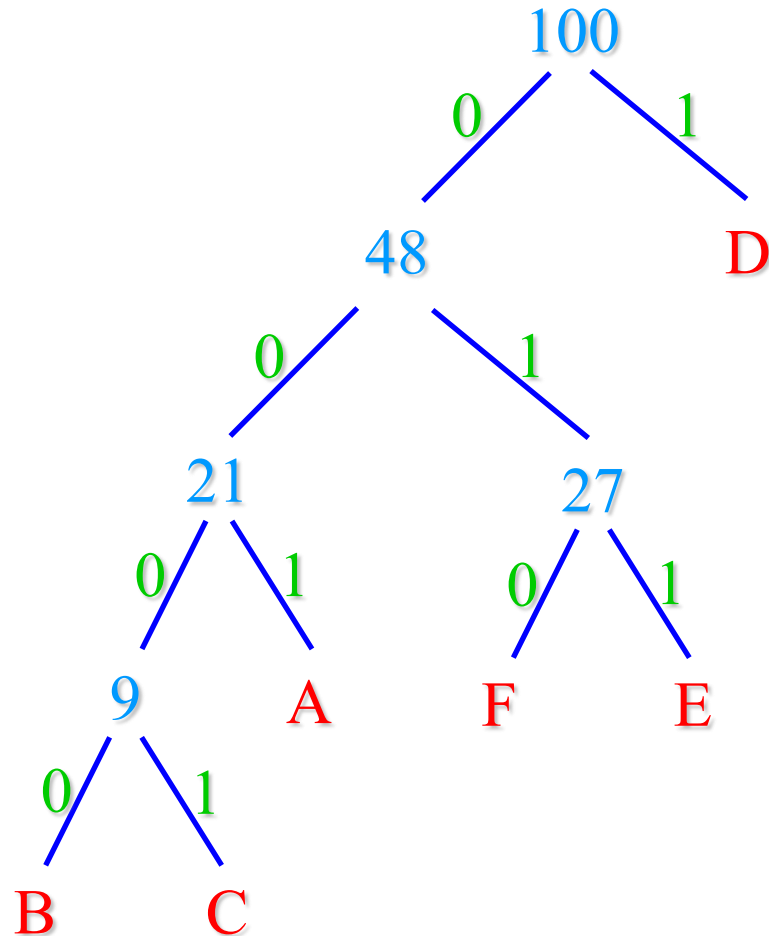


Élcímkézés:
0: balra,
1: jobbra

Huffman kódolás

szimbólum kód

A	001
B	0000
C	0001
D	1
E	011
F	010



A változó hosszúságú prefixmentes kódot a gyökértől a levélig vezető út élcímkéinek konkatenáltja adja.

Huffman kódolás

szimbólum kód

A	001
B	0000
C	0001
D	1
E	011
F	010

kódolt üzenet:

10100000100110110001

dekódolás:

10100000100110110001

D F B D A D E C

az eredeti üzenet:

DFBDADEC

A változó hosszúságú kód egyértelmű dekódolhatósága a prefix-mentességnek köszönhető.

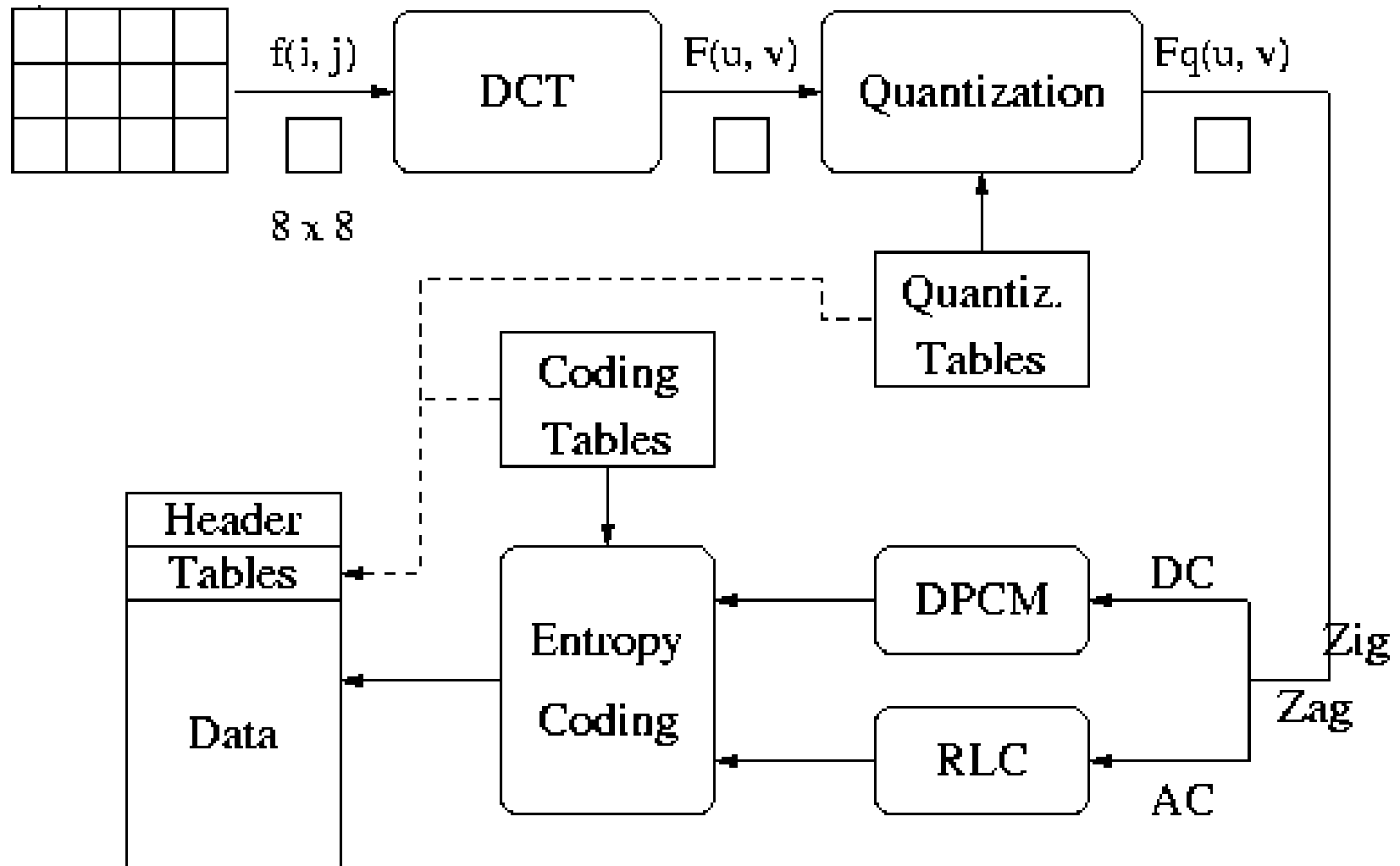
JPEG formátum/szabvány

- A JPEG (*Joint Photographers Experts Group*) az ISO és az CCITT szabványügyi szervezetek által létrehozott munkacsoport.
- A JPEG egy
 - veszteséges transzformáció kódolás
 - DCT-vel
 - kvantálótáblákkal és
 - az együtthatók (és a 0 lefutások) Huffman
 - kódolásával.

JPEG kódolás

- **a kódolandó kép felbontása 8x8-as részképekre, és a blokkok kódolása (egymástól függetlenül):**
 1. **eltolás: a képpont-értékekből levonunk 2^{m-1} -et (ha az intenzitástartomány $[0, 2^m - 1]$);**
 2. **2D DCT végrehajtása;**
 3. **a kapott együtthatók kvantálása (a megkívánt tömörítési arányhoz konstruált tábla szerint);**
 4. **a cikk-cakk minta szerint kiterített kvantált együtthatók speciális futamhossz és Huffman kódolása.**

JPEG kódolás



DPCM: Differential/Delta Pulse-Code Modulation
 RLC: Run-Length enCoding

Példa JPEG kódolásra (1.)

52	55	61	66	70	61	64	73		-76	-73	-67	-62	-58	-67	-64	-55
63	59	55	90	109	85	69	72		-65	-69	-73	-38	-19	-43	-59	-56
62	59	68	113	144	104	66	73		-66	-69	-60	-15	16	-24	-62	-55
63	58	71	122	154	106	70	69		-65	-70	-57	-6	26	-22	-58	-59
67	61	68	104	126	88	68	70	→	-61	-67	-60	-24	-2	-40	-60	-58
79	65	60	70	77	68	58	75		-49	-63	-68	-58	-51	-60	-70	-53
85	71	64	59	55	61	65	83		-43	-57	-64	-69	-73	-67	-63	-45
87	79	69	68	65	76	78	94		-41	-49	-59	-60	-63	-52	-50	-34

8x8-as blokk

eltolás
(-128)

eltolt blokk

Diszkrét koszinusz transzformáció (discrete-cosine transform, DCT)

(forward) transzformáció:

$$C(X, Y) = \sum_{x=0}^{n-1} \sum_{y=0}^{n-1} f(x, y) \cdot \alpha(x) \alpha(y) \cos\left(\frac{(2x+1)\pi X}{2n}\right) \cos\left(\frac{(2y+1)\pi Y}{2n}\right),$$

megegyeznek a
bázisfüggvények

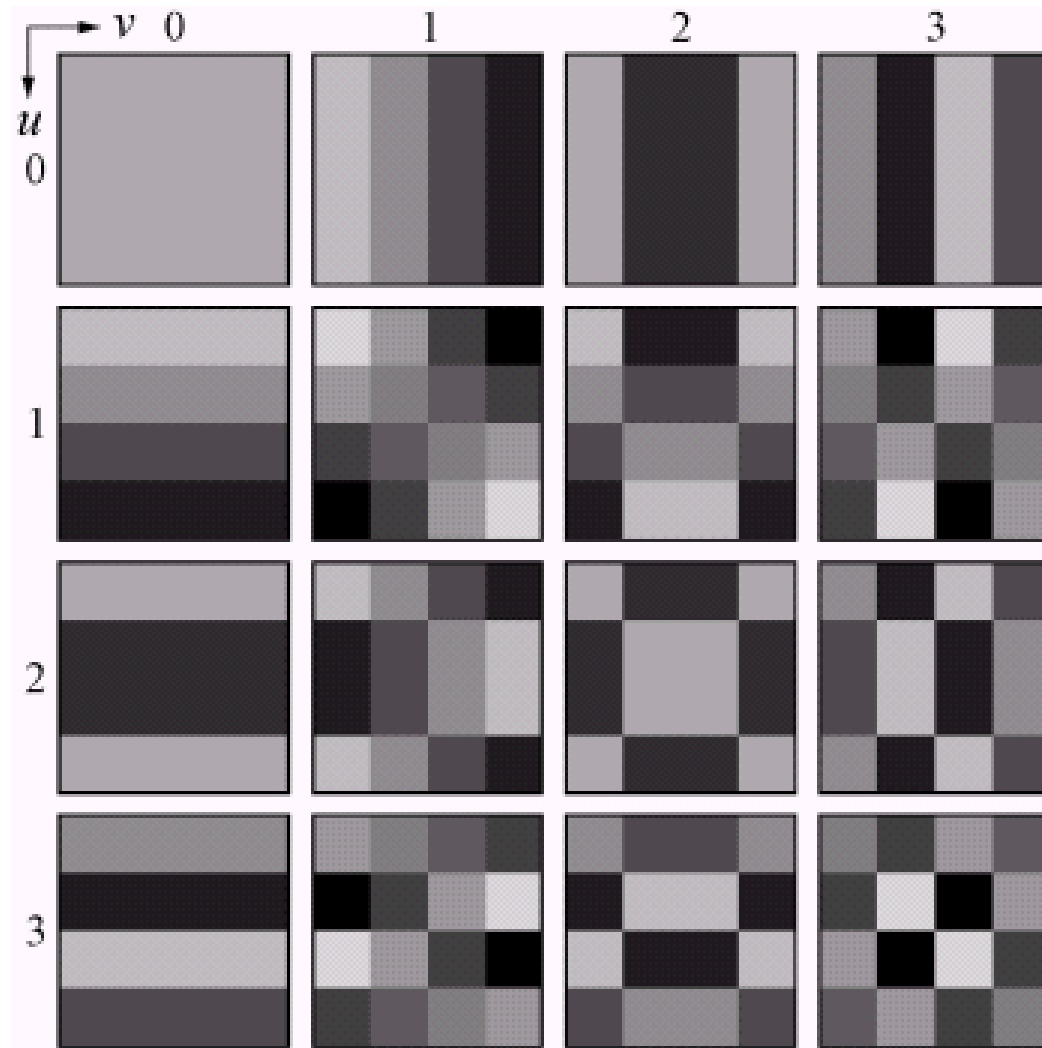
inverz transzformáció:

$$f(x, y) = \sum_{X=0}^{n-1} \sum_{Y=0}^{n-1} C(X, Y) \cdot \alpha(X) \alpha(Y) \cos\left(\frac{(2x+1)\pi X}{2n}\right) \cos\left(\frac{(2y+1)\pi Y}{2n}\right),$$

ahol:

$$\alpha(u) = \begin{cases} \sqrt{1/n} & , \text{ ha } u = 0 \\ \sqrt{2/n} & , \text{ ha } u = 1, 2, \dots, n-1 \end{cases}.$$

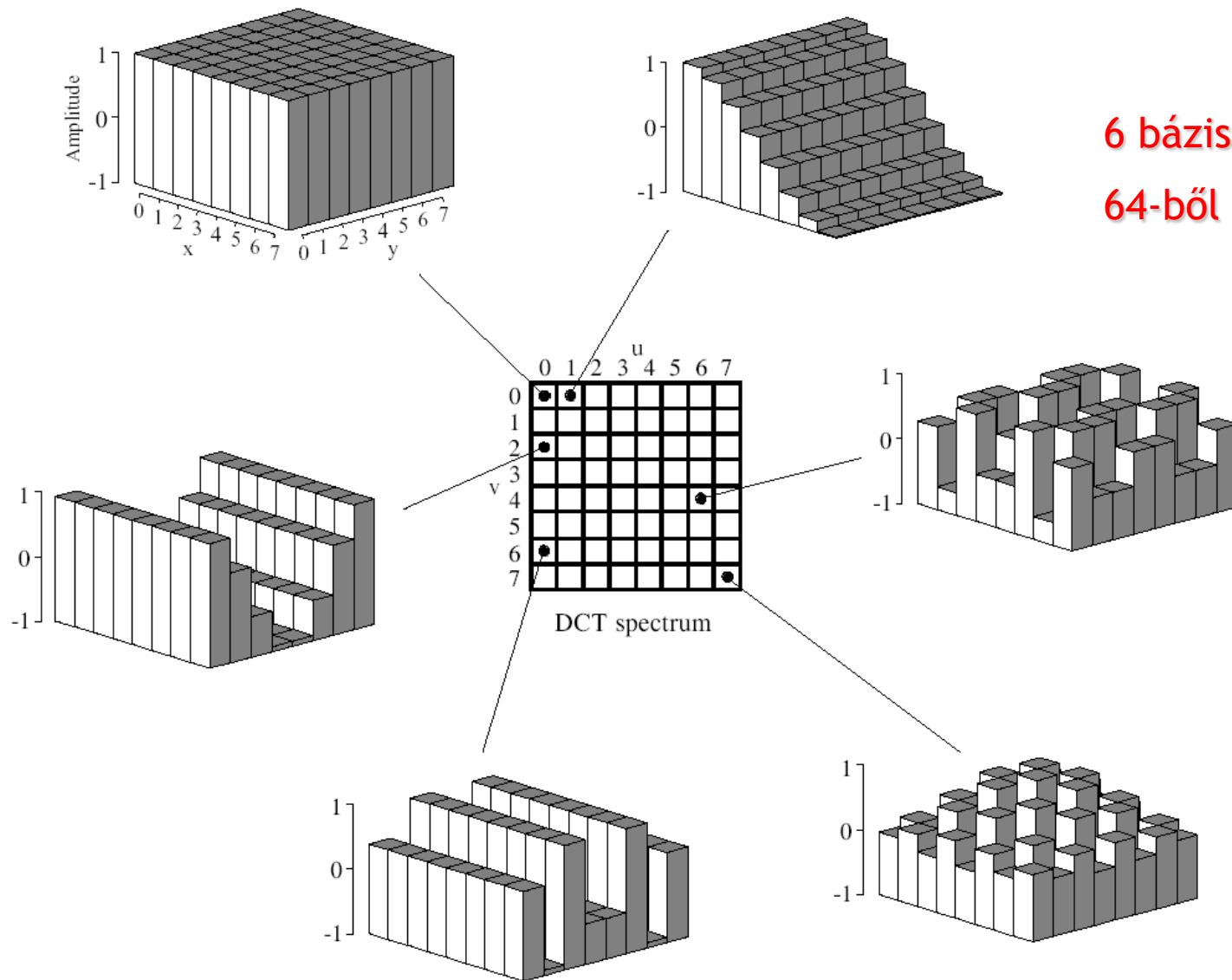
DCT bázisképek



$n=4$

DCT bázisképek

6 báziskép a
64-ből ($n=8$)



Blokkméret és a DCT kódolás

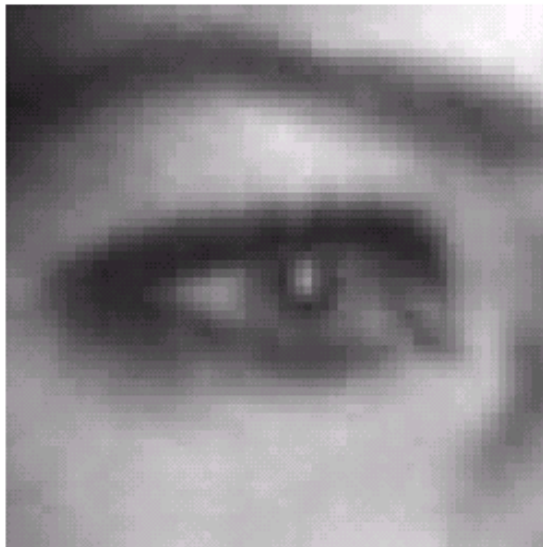
eredeti kép
részlete



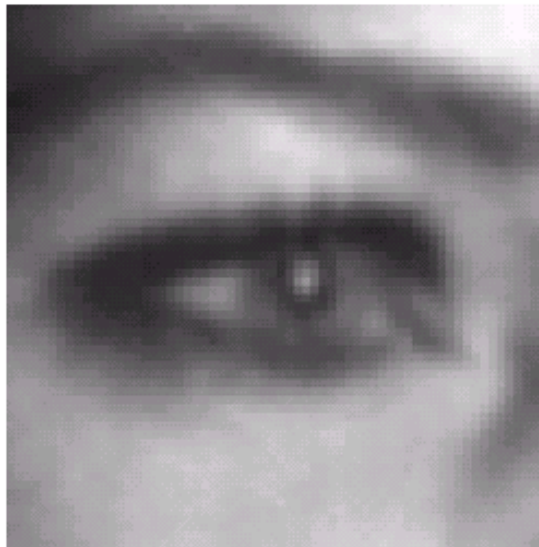
dekódolt
képrészlet
(n=2)



dekódolt
képrészlet
(n=4)



dekódolt
képrészlet
(n=8)



(Mindhárom kódolás az együttthatók negyedét használta fel.)

Példa JPEG kódolásra (2.)

$$\begin{bmatrix} -76 & -73 & -67 & -62 & -58 & -67 & -64 & -55 \\ -65 & -69 & -73 & -38 & -19 & -43 & -59 & -56 \\ -66 & -69 & -60 & -15 & 16 & -24 & -62 & -55 \\ -65 & -70 & -57 & -6 & 26 & -22 & -58 & -59 \\ -61 & -67 & -60 & -24 & -2 & -40 & -60 & -58 \\ -49 & -63 & -68 & -58 & -51 & -60 & -70 & -53 \\ -43 & -57 & -64 & -69 & -73 & -67 & -63 & -45 \\ -41 & -49 & -59 & -60 & -63 & -52 & -50 & -34 \end{bmatrix} \xrightarrow{\text{DCT}} \begin{bmatrix} -415 & -30 & -61 & 27 & 56 & -20 & -2 & 0 \\ 4 & -22 & -61 & 10 & 13 & -7 & -9 & 5 \\ -47 & 7 & 77 & -25 & -29 & 10 & 5 & -6 \\ -49 & 12 & 34 & -15 & -10 & 6 & 2 & 2 \\ 12 & -7 & -13 & -4 & -2 & 2 & -3 & 3 \\ -8 & 3 & 2 & -6 & -2 & 1 & 4 & 2 \\ -1 & 0 & 0 & -2 & -1 & -3 & 4 & -1 \\ 0 & 0 & -1 & -4 & -1 & 0 & 1 & 2 \end{bmatrix}$$

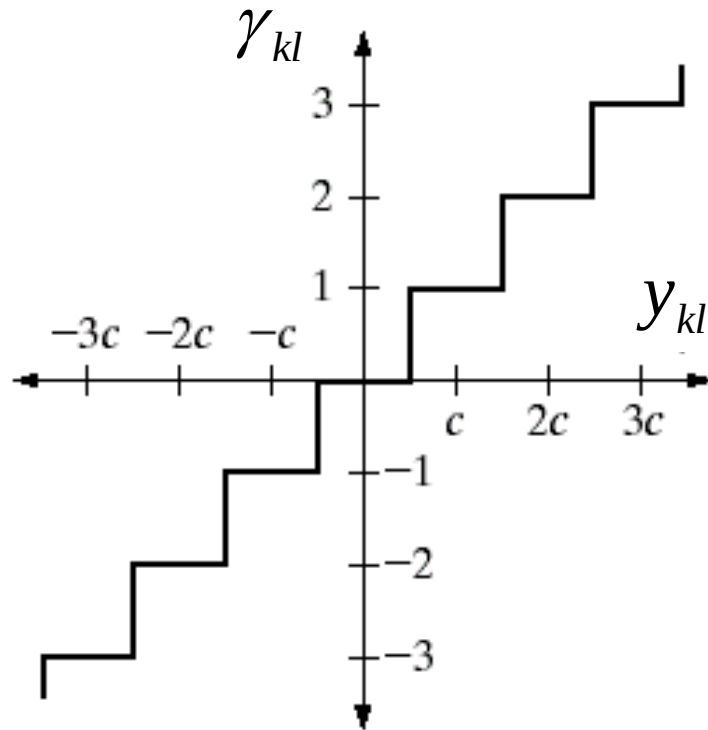
eltolt blokk

DCT

együtthatók

Kvantálás

$$X = \sum_{k=0}^{n-1} \sum_{l=0}^{n-1} y_{kl} \cdot B_{kl} \approx \sum_{k=0}^{n-1} \sum_{l=0}^{n-1} \gamma_{kl} \cdot B_{kl} = \hat{X}$$



$$\gamma_{kl} = \text{round}\left(\frac{y_{kl}}{T_{kl}}\right),$$

ahol:

T_{kl} a kvantálótábla megfelelő pozícióján lévő nem-negatív egész (esetünkben c).

Példa JPEG kódolásra (3.)

$$\begin{bmatrix} -415 & -30 & -61 & 27 & 56 & -20 & -2 & 0 \\ 4 & -22 & -61 & 10 & 13 & -7 & -9 & 5 \\ -47 & 7 & 77 & -25 & -29 & 10 & 5 & -6 \\ -49 & 12 & 34 & -15 & -10 & 6 & 2 & 2 \\ 12 & -7 & -13 & -4 & -2 & 2 & -3 & 3 \\ -8 & 3 & 2 & -6 & -2 & 1 & 4 & 2 \\ -1 & 0 & 0 & -2 & -1 & -3 & 4 & -1 \\ 0 & 0 & -1 & -4 & -1 & 0 & 1 & 2 \end{bmatrix} \xrightarrow{\text{kvantálás}} \begin{bmatrix} -26 & -3 & -6 & 2 & 2 & -1 & 0 & 0 \\ 0 & -2 & -4 & 1 & 1 & 0 & 0 & 0 \\ -3 & 1 & 5 & -1 & -1 & 0 & 0 & 0 \\ -4 & 1 & 2 & -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

együtthatók

kvantált együtthatók

kvantáló-
tábla

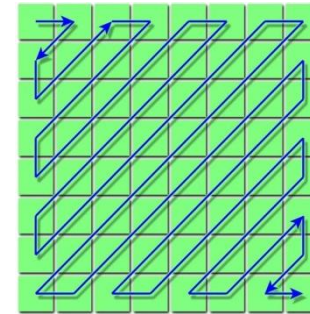
16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

$\text{round}(-415/16)$
= -26

1D futamhossz kódolás (RLC)

- Az 1950-es évektől javasolt technika a FAX-üzenetek kódolására.
- A képet soronként kódolja: egy sort a 0-k és 1-ek folytonos csoportjainak hossza jellemez. Feltételezi, hogy minden sor 0-k csoportjával indul (melynek hossza 0 is lehet).
- A kódolás önmagában is hatékony, de a kapott kód további tömörítése is általános

Példa JPEG kódolásra (4.)

$$\begin{bmatrix} -26 & -3 & -6 & 2 & 2 & -1 & 0 & 0 \\ 0 & -2 & -4 & 1 & 1 & 0 & 0 & 0 \\ -3 & 1 & 5 & -1 & -1 & 0 & 0 & 0 \\ -4 & 1 & 2 & -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$


kvantált együtthatók

„0”-lefutás

-26, -3, 0, -3, -2, -6, 2, -4, 1, -4, 1, 1, 5, 1, 2, -1, 1, -1, 2, 0, 0, 0, 0, 0, 0, -1, -1, 0, ..., 0

kvantált együtthatók a cikk-cakk bejárás szerint

veszteségmentes entrópia kódolás
(futamhossz + Huffman)

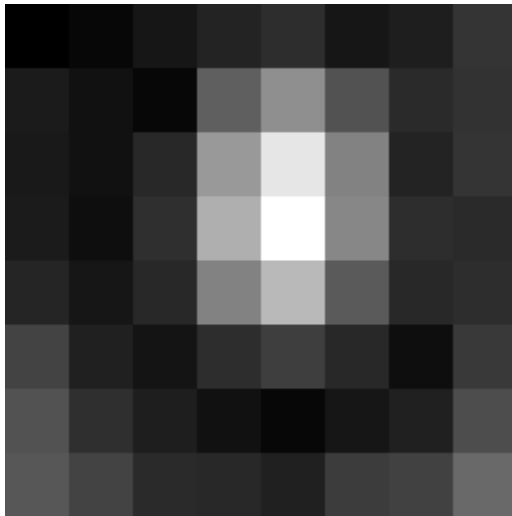


JPEG dekódolás

- **a blokkok dekódolása (egymástól függetlenül):**
 1. **a kvantált együtthatók dekódolása és 8x8-as elrendezésük a cikk-cakk minta szerint;**
 2. **a kvantálás „invertálása”;**
 3. **2D inverz DCT végrehajtása;**
 4. **„visszatolás”: a képpont-értékekhez hozzáadunk 2^{m-1} -et;**
- **a dekódolt kép „összerakása” a 8x8-as blokkokból.**

A JPEG hibája

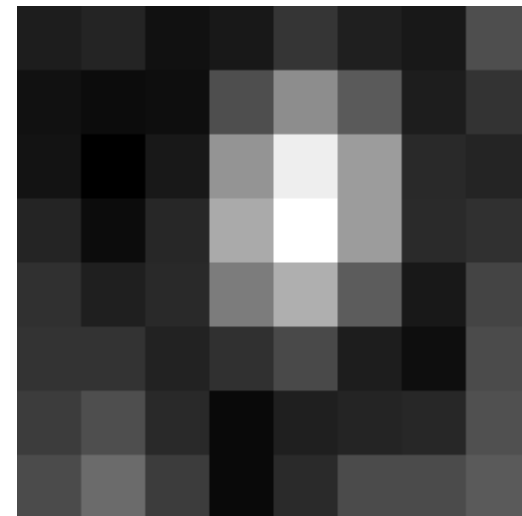
$$\begin{bmatrix} 52 & 55 & 61 & 66 & 70 & 61 & 64 & 73 \\ 63 & 59 & 55 & 90 & 109 & 85 & 69 & 72 \\ 62 & 59 & 68 & 113 & 144 & 104 & 66 & 73 \\ 63 & 58 & 71 & 122 & 154 & 106 & 70 & 69 \\ 67 & 61 & 68 & 104 & 126 & 88 & 68 & 70 \\ 79 & 65 & 60 & 70 & 77 & 68 & 58 & 75 \\ 85 & 71 & 64 & 59 & 55 & 61 & 65 & 83 \\ 87 & 79 & 69 & 68 & 65 & 76 & 78 & 94 \end{bmatrix}$$

$$\begin{bmatrix} 60 & 63 & 55 & 58 & 70 & 61 & 58 & 80 \\ 58 & 56 & 56 & 83 & 108 & 88 & 63 & 71 \\ 60 & 52 & 62 & 113 & 150 & 116 & 70 & 67 \\ 66 & 56 & 68 & 122 & 156 & 116 & 69 & 72 \\ 69 & 62 & 65 & 100 & 120 & 86 & 59 & 76 \\ 68 & 68 & 61 & 68 & 78 & 60 & 53 & 78 \\ 74 & 82 & 67 & 54 & 63 & 64 & 65 & 83 \\ 83 & 96 & 77 & 56 & 70 & 83 & 83 & 89 \end{bmatrix}$$


eredeti blokk

$$\begin{bmatrix} -8 & -8 & 6 & 8 & 0 & 0 & 6 & -7 \\ 5 & 3 & -1 & 7 & 1 & -3 & 6 & 1 \\ 2 & 7 & 6 & 0 & -6 & -12 & -4 & 6 \\ -3 & 2 & 3 & 0 & -2 & -10 & 1 & -3 \\ -2 & -1 & 3 & 4 & 6 & 2 & 9 & -6 \\ 11 & -3 & -1 & 2 & -1 & 8 & 5 & -3 \\ 11 & -11 & -3 & 5 & -8 & -3 & 0 & 0 \\ 4 & -17 & -8 & 12 & -5 & -7 & -5 & 5 \end{bmatrix}$$

különbségkép



dekódolt blokk

Példa JPEG tömörítésre



eredeti, 262 KB

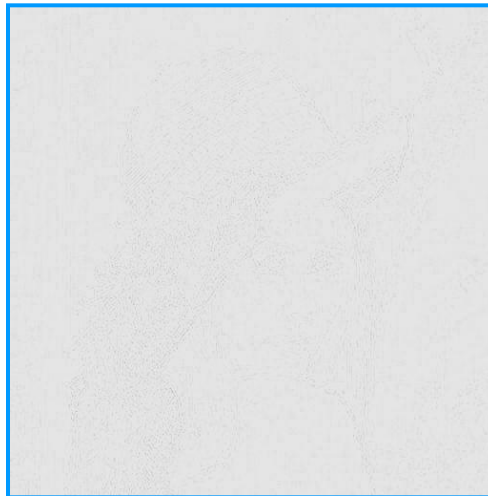


12:1, 22 KB



43:1, 6 KB

eltérés az
eredetitől →



Színes képek JPEG tömörítése

- YUV,YIQ színtérben az intenzitás (Y) és kromatikus (UV illetve IQ) információ szeparált
- Az emberi látás kromatikus felbontása kisebb → jobban tömöríthető

Original



Y - Blur



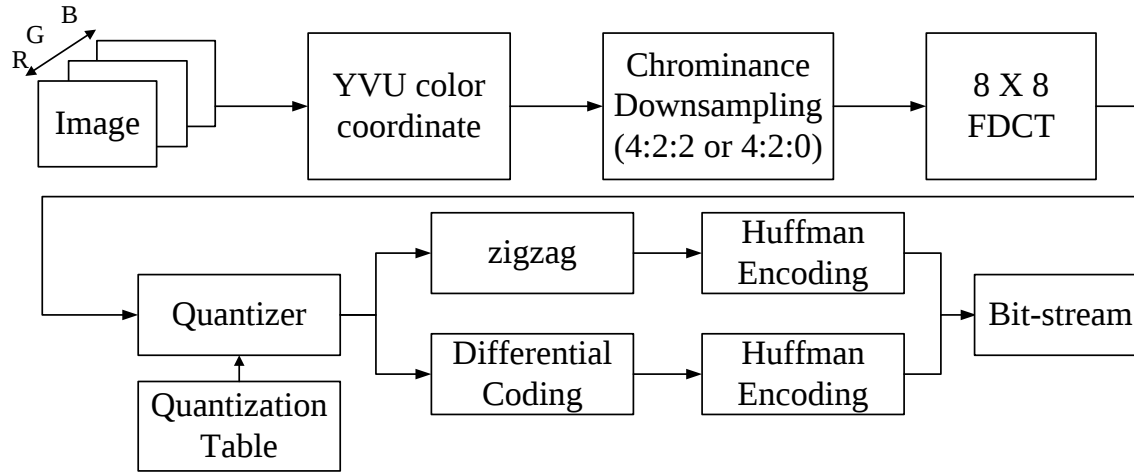
I - Blur



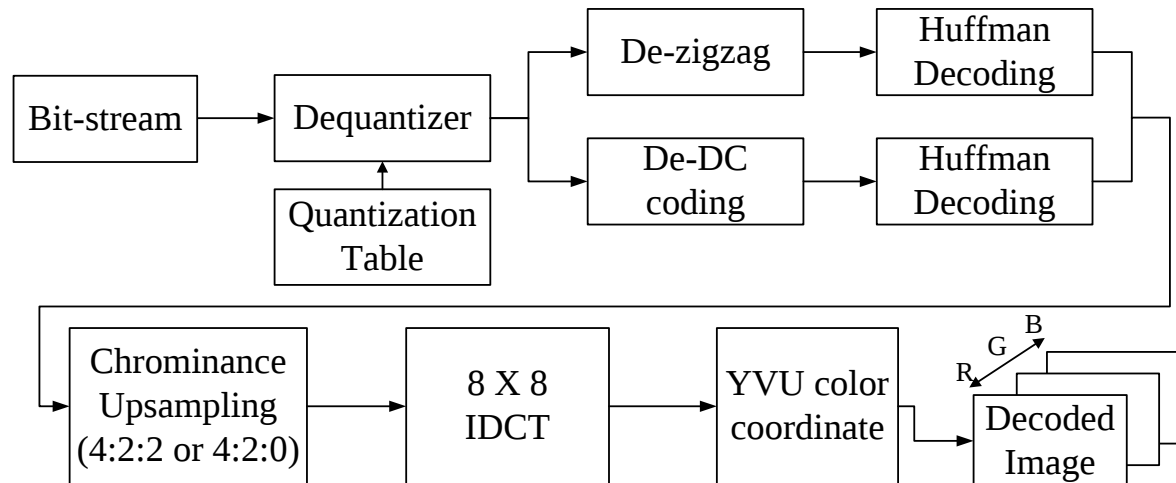
Q - Blur

JPEG folyamat színes képekre

Kódoló



Dekódoló



„Blokkosodás” JPEG képeken



eredeti kép



tömörített kép
79:1

JPEG 2000

- **Az újabb szabvány (*ISO group Joint Photographic Experts Group, ISO/IEC 29 WG01*) a DCT helyett wavelet transzformációt használ.**
- **A JPEG 2000 képeken általában nem érzékelhető a torzulás még 50:1 arányú tömörítés mellett sem.**

Original



JPEG
27:1



JPEG2000

27:1



JPEG – JPEG2000

tömörítési arány: 150:1



eredeti kép



JPEG2000

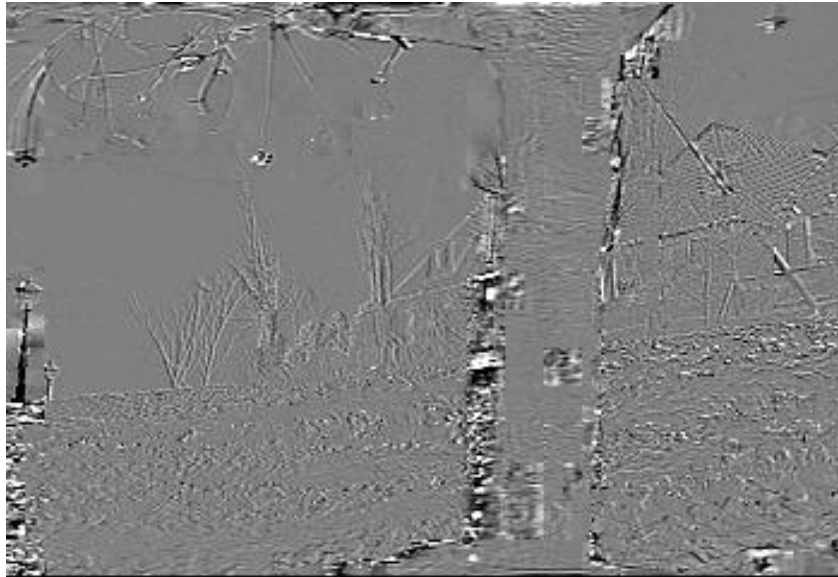


JPEG

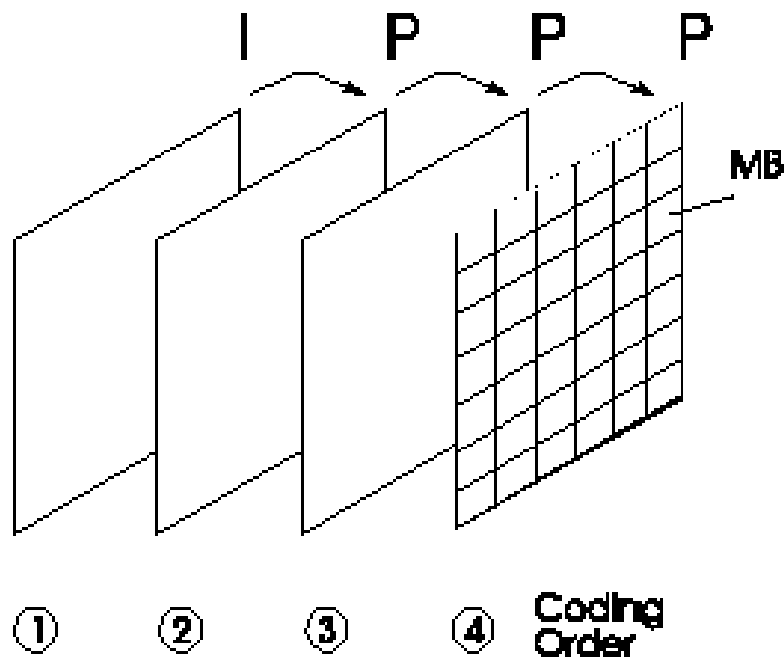
Mi az MPEG ?

- "Motion Picture Expert Group", 1990-ben létrehozott szervezet az audio és video átvitel szabványosítására.
 - Többféle szabvány. Pl. MPEG-1: VHS minőség CD-n (320 x 240 + CD audio @ 1.5 Mbits/sec) .
- Video tömörítés a framek közötti temporális redundanciát használja ki:
 - Temporális DPCM a redundancia kiszűrésére
 - A mozgáskompenzált hiba kódolása: DCT+Huffman.
- Minőséget alapvetően befolyásolja a jó mozgásbecslés
 - Predikció a mozgás trajektóriák mentén
 - A jó minőségű kódolás számításigényes (dekódolás nem)

Mozgás kompenzált predikció



MPEG-1



A.) Frame Prediction

Y1	Y2
Y3	Y4

U

V

B.) Macroblock

Felhasznált anyagok

- **Palágyi Kálmán: Digitális Képfeldolgozás**
/pub/Digitalis_kepfeldolgozas
- **További források az egyes diákon megjelölve**