

# JPEG kódolás

Technikai megvalósítás, működés és implementáció  
Máthé Dávid György

# Áttekintés

- Digitális képek
  - Egyik legnépszerűbb tömörítési eljárás
  - Veszteséges (lossy compression)
- 
- JFIF, EXIF fájlformátum
  - JPEG → eljárás
  - JPEG Interchange Format (JIF)



ISO  
**120**

Exif Version  
**0220**

Date/Time Original  
**2016:09:20 10:01:21**

Create Date  
**2016:09:20 10:01:21**

Components Configuration  
**Y, Cb, Cr, -**

Exposure Compensation  
**0**

Metering Mode  
**Center-weighted average**

Light Source  
**Other**

Flash  
**No Flash**

Focal Length  
**3.5 mm**

# Szintér

- RGB-ből YCbCr-be színes esetén
- Chroma blue & Chroma red downsampling  
4 : 2 : 0
- Megvalósítás: szürkeárnyaltos (csak luma)
- Egy színcsatorna három helyett



# Előfeldolgozás

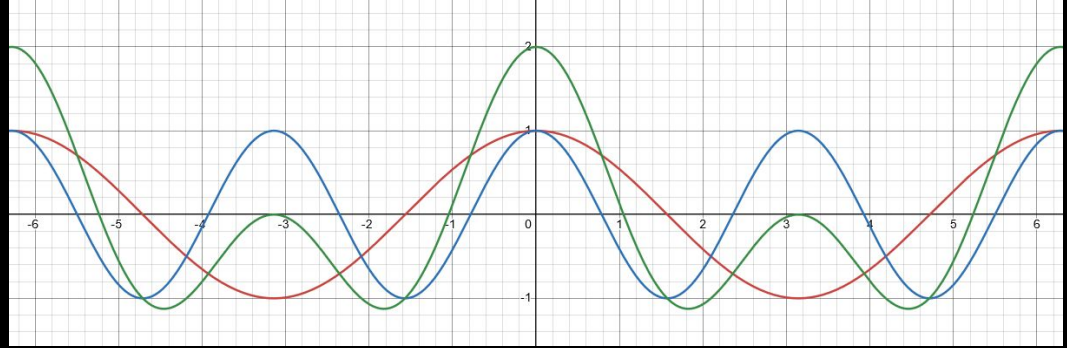
- Blokkokra vagy tile-okra bontás  
MCU (Minimum Coded Unit)
- Megvalósítás:  $8 \times 8$
- Függ a downsampling mértékétől
- Kép méret nem 8 többszöröse  $\rightarrow$  Padding  
Fix szín, szélső pixel ismétlése stb.
- Érték középpontosítás, középre igazítás  
 $(0 \sim 255) \rightarrow (-128 \sim 127)$





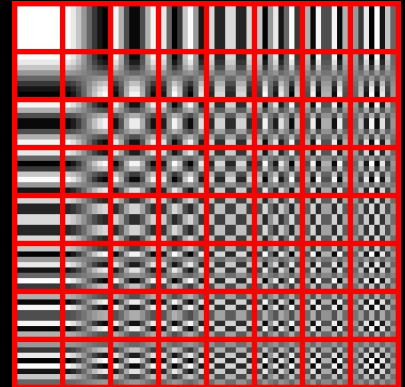
# Diszkrét koszinusz transzformáció

- Koszinusz függvények képi reprezentációja
- Egy 8x8 tile reprezentálható 64 különböző koszinusz fv. súlyozott (+ -) átlagával



$$G_{u,v} = \frac{1}{4} \alpha(u) \alpha(v) \sum_{x=0}^7 \sum_{y=0}^7 g_{x,y} \cos \left[ \frac{(2x+1)u\pi}{16} \right] \cos \left[ \frac{(2y+1)v\pi}{16} \right]$$

- u és v index koszinusz frekvencia (0..1)
- $\alpha$  Scalingfunction, Basisfunction...

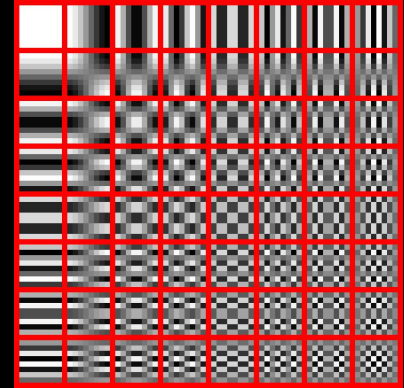


# Diszkrét koszinusz transzformáció

- Egy DCT együttható nem 1 pixelre számolódik, hanem a teljes tile-ra
- DCT együttható: mekkora súllyal kell alkalmazni az adott helyen lévő adott horizontális és vertikális frekvenciájú koszinusz fv.-t, hogy az összes koszinusz függvény értékeinek ezzel a súllyal vett átlaga az eredeti képet adja vissza (jó közelítéssel).

$$G_{u,v} = \frac{1}{4} \alpha(u) \alpha(v) \sum_{x=0}^7 \sum_{y=0}^7 g_{x,y} \cos \left[ \frac{(2x+1)u\pi}{16} \right] \cos \left[ \frac{(2y+1)v\pi}{16} \right]$$

- DC együttható, AC együttható és szerepük



# Kvantálás

- Nagy frekvenciájú koszinusz függvények sokkal kisebb súllyal
  - Adott helyen lévő DCT együttható / adott helyen lévő kvantálási mátrix érték
  - Cél → Kevesebb információ eltárolása a nagyfrekvenciás együtthatókról
- 
- Veszteséges
  - Kvantálási mátrix ~ Minőség (nagyobb értékek  $\rightarrow 0$ )
  - Legközelebbi egész számra kerekítés



# Futamhossz kódolás

- Cikk-Cakk bejárás  $\rightarrow$  rövidebb kód
- Futamhossz kódolás: Számpárok

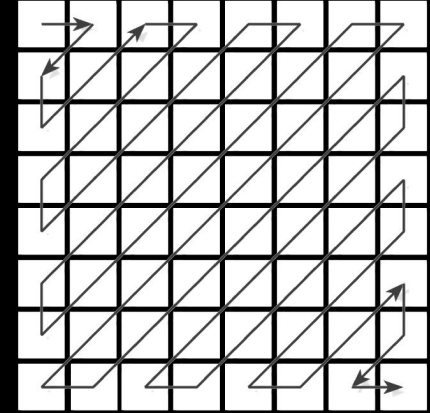
$\langle [0] \text{ darabszáma} , \text{ nem } [0] \text{ érték} \rangle$

- Speciális számpár:

$\langle 15 , 0 \rangle$

$\langle 0 , 0 \rangle$

- Csak AC



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



# Huffman kódolás

- Gyakoribb értékek → rövidebb kód
- Gyakorlatban Huffman-tábla a JFIF/EXIF fájlban



Ezzel hozható létre a fa

- Külön DC és AC
- Értéktartományokhoz rendelt kategóriák határoznak meg egy bitsorozatot
- Kategória kódszó + érték bitreprézentáció

# Huffman kódolás

- DC példa: -30

5. kategória: 110

Érték a tartományban: 00001

Végző kód: 11000001

-28 → 11000011

6 → 100110

- AC táblák jóval nagyobbak
- Párokhoz definiált kódok
- De ua. kategóriához tartozó kódszó + bitreprezent.

Range	Value Category	Bits for the value
0	0	-
-1,1	1	0,1
-3,-2,2,3	2	00,01,10,11
-7,-6,-5,-4,4,5,6,7	3	000,001,010,011,100,101,110,111
-15,...,-8,8,...,15	4	0000,...,0111,1000,...,1111
-31,...,-16,16,...,31	5	00000,...,01111,10000,...,11111

Category	Code length	Code word
0	2	00
1	3	010
2	3	011
3	3	100
4	3	101
5	3	110
6	4	1110
7	5	11110
8	6	111110
9	7	1111110
10	8	11111110
11	9	111111110

Köszönöm szépen a figyelmet!

Máthé Dávid György