JPEG Image Compression

Multimedia Databases SS23 (Exercises)

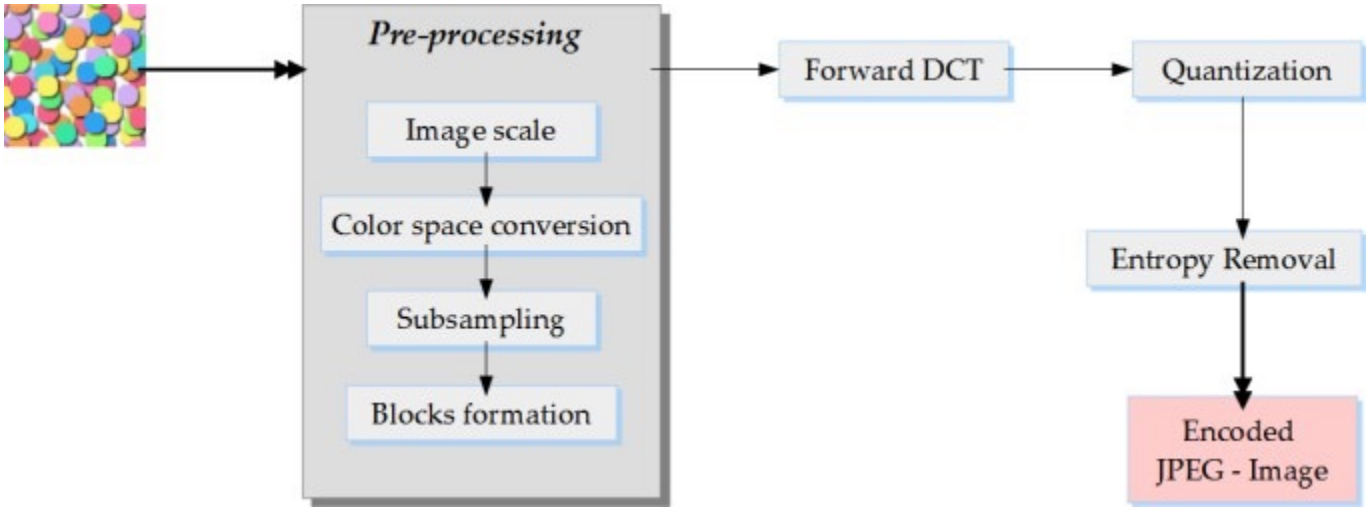
Prof. Dr. Harald Kosch / Prof. Dr. Mario Döller
Tutors: Kanishka Ghosh Dastidar, Alaa Alhamzeh

Outline

- JPEG Baseline Process
- Pre-processing
- Forward Discrete Cosinus Transformation (F-DCT)
- Quantization
- Entropy Coding

Task 1.1: JPEG compression algorithm

- Baseline Process
  - **Goal:** Reversible (but lossy) compression



Task 1.2

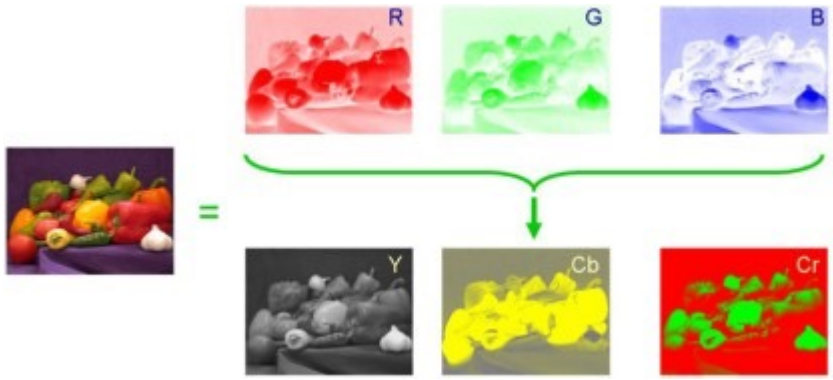
- Pre-processing:
  - Image scale + Blocks formation: divide the image into blocks of equal/specific size: Lossless.
  - Color space conversion: Lossless.
  - Subsampling: remove less significant details: Lossy
- DCT: Transform the image to frequency domain: Lossy
- Quantization: Remove less significant details: Lossy
- Entropy Removal: Remove redundancy: Lossless

Task 2: Pre-processing: Image Scale

- **DCT can only work with blocks of size 8x8**
  - Image dimensions must be divisible by 8
- **Missing pixels are completed as follows**
  - Extra columns on the right contain same value as furthermost column
  - Extra rows below contain same value as the last row of the image
  - Missing block on the bottom right contains same value as the last pixel of image

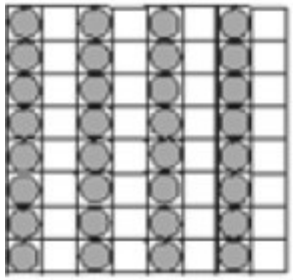
Task 2.1: Pre-processing: Color Space

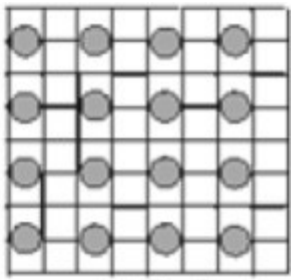
- **Conversion from RGB to YUV (YC_bC_r)**
- **Separates luma from chroma**
  - Luma (luminance): Y, isolated
  - Chroma (Chrominance): two color-difference components
    - C_b blue – luma
    - C_r red – luma
- **Enable data-reduction, through the fact:**
  - Human perception is less sensitive to changes in color than light intensity



Task 2.2: Pre-processing: Subsampling

- **Data reduction**
  - Keep all luma coefficients
  - Discard parts of chroma coefficients
- **Given by a ratio: X:a:b**
  - X: Reference
  - a: num of C_b or C_r per odd line
  - b: num of C_b or C_r per even line
- 4:4:4 → no changes

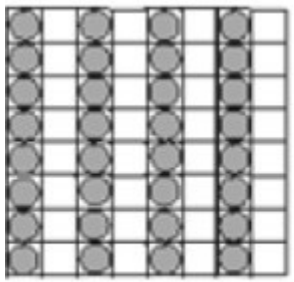

4:2:2


4:2:0

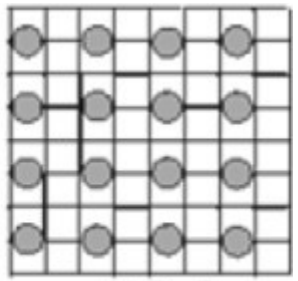
Task 2.3: Pre-processing: Blocks Formation

- **Image must be divided into blocks of MCU (Min Coded Units): 8x8**
- **But: Subsampling drops chroma coefficients**
  - So there might not be enough elements
  - 4:4:4 makes no changes → MCU = 8 x 8
  - 4:2:0 contains 50% in each dimension → MCU = 16 x 16

  →Image width/height must be divisible by 16 (or scaled)



4:2:2



4:2:0

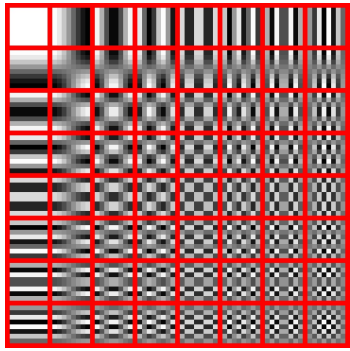
Task 3.1: DCT: Discrete Cosinus Transform.

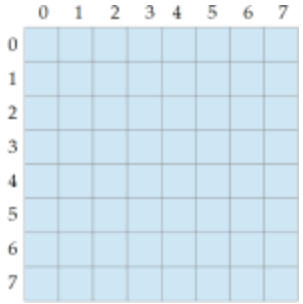
- **DCT translates image information of space domain (visible form of representation) into frequency domain**
  - Applied on 8x8 blocks
  - MUST be reversible
  - No data reduction during transformation, but basis for further processing
- **Frequency Domain:**
  - Frequency is the rate at which pixel values change over spatial distance
  - Color Variance → Frequency
    - A: Low, B: Moderate, C: High

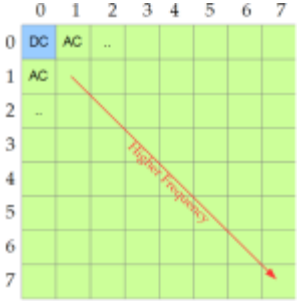
  - Transformation helps separating less important from more important information.
    - Human eye ist most sensitive to low frequencies than high frequencies.

Task 3.2 and 3.3: F-DCT: Coefficients

- **Image is divided into 8x8 blocks**
  - Each is considered as discrete signal with 64 coefficients
- **Two types of coefficients**
  - DC:DCT(0,0) Gives the basic hue of the image
  - AC: The other 63 coefficients
    - Represent color change across the block
- **Further parts (higher frequencies) are the more useful parts for compression.**







Task 4.1: Quantization

- **Quantization is performed by:**
  - Divide each element in the transformed matrix F(u,v) by the corresponding element in the quantization matrix
  - Round to nearest integer value
- **Quantization matrix is influential**
  - Not standartized, but recommendations are provided
  - Quantization tables for chroma and luma based on subjective experiments involving human visual system

162.3	40.6	20.0	72.3	30.3	12.5	−19.7	−11.5
30.5	108.4	10.5	32.3	27.7	−15.5	18.4	−2.0
−94.1	−60.1	12.3	−43.4	−31.3	6.1	−3.3	7.1
−38.6	−83.4	−5.4	−22.2	−13.5	15.5	−1.3	3.5
−31.3	17.9	−5.5	−12.4	14.3	−6.0	11.5	−6.0
−0.9	−11.8	12.8	0.2	28.1	12.6	8.4	2.9
4.6	−2.4	12.2	6.6	−18.7	−12.8	7.7	12.0
−10.0	11.2	7.8	−16.3	21.5	0.0	5.9	10.7

DCT matrix F(u,v))

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

Quantization matrix Q(u,v)

10	4	2	5	1	0	0	0
3	9	1	2	1	0	0	0
−7	−5	1	−2	−1	0	0	0
−3	−5	0	−1	0	0	0	0
−2	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

Task 4.1: Quantization

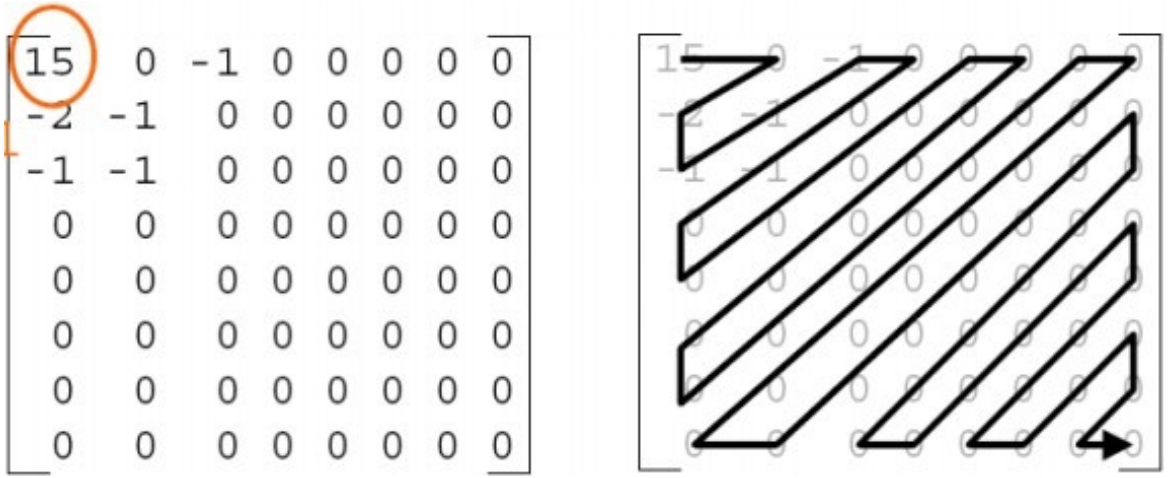
- **Advantages**
  - Compression: Save bits by restricting value range
    - Reduce frequency components that are neglibible for the image information
- **Disadvantages**
  - Less possible values → worst quality
    - Lossy operation
  - Quantization error (noise) occurs → edge artifacts

Task 5.1: Entropy Coding

- **Purpose of entropy coding**
  - Last step of JPEG compression
  - Reduce redundancy in quantized coefficients
  - Lossless
- **Main contribution of RLE to data compression**
  - Run-length encoding helps reducing data by compressing series of identical bits
  - A block must be handled first by a zig-zag scan

Task 5.1: Entropy Coding: Zig-Zag

- **The Zig-Zag scan maps an 8x8 matrix to a 1x64 vector**
  - Group low frequency coefficients at vector‘s front and high frequency coefficients at bottom.

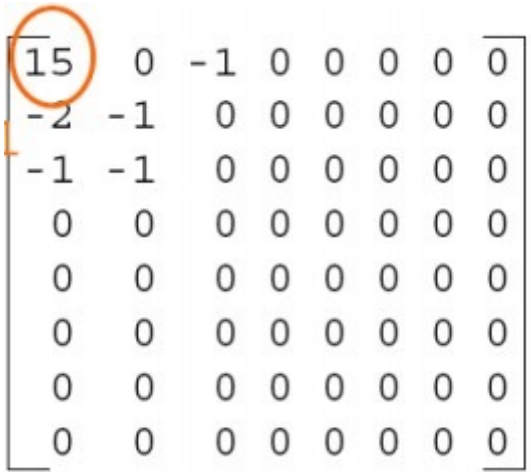


15 0 -2 -1 -1 -1 0 0 -1 0 0 0 0 0 0 0 . . . 0

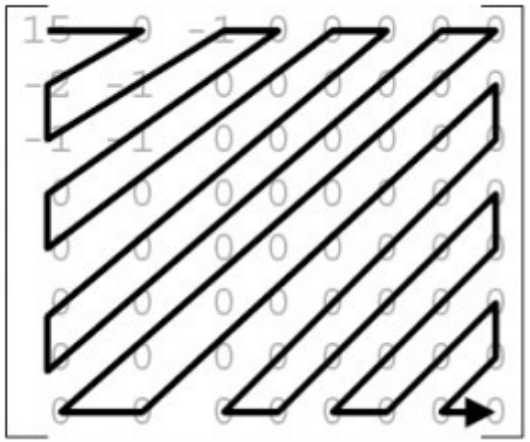
Task 5.2: Entropy Coding: Different Coding

- **DC- and AC-coefficients are entropy coded differently**
- **RLE (and zig-zag) is applied only for AC coefficients**
  - Due quantization step, the 1x64 vectors have a lot of zeros in them
  - Keeps *skip* and *value*, where *skip* is the number of zeros and *value* is the next non-zero component.

  0 -2 -1 -1 -1 0 0 -1 0 0 0 0 0 0 0… → **1**,-2,**0**,-1,**0**,-1,**0**,-1,**2**,-1 **56**

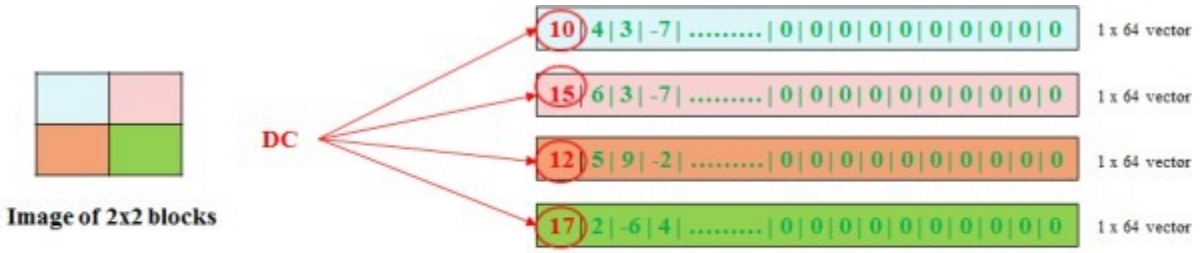


15	0	-1	0	0	0	0	0
-2	-1	0	0	0	0	0	0
-1	-1	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0



Task 5.3: Entropy Coding: Different Coding

- **For DC coefficients:**
  - Difference between the coefficients of neighboring blocks is used for getting higher compression rates
    - Neighboring DC coefficients are similar
    - Differences between them are very small



$\Rightarrow$ 10, 5, -3, 5, ..