



## Advanced Image Processing

JPEG compression

TP Class N° 4

May 2, 2019

### JPEG compression

JPEG compression main steps:

1. RGB color space to YCbCr color space conversion:

**YCbCr** (aka **YCC**) is the color space used in digital formats JPEG and MPEG.

**Range:**  $R, G, B, Y \in [0, 255]$

**Conversion RGB-YCbCr:**

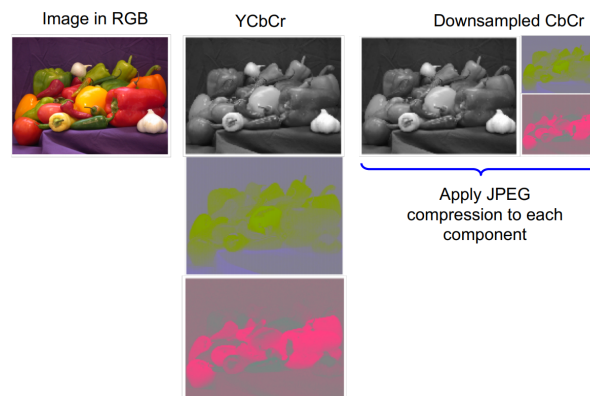
$$\begin{aligned} Y &= 0.299 R + 0.587 G + 0.114 B \\ Cb &= -0.1687 R - 0.3313 G + 0.5 B + 128 \\ Cr &= 0.5 R - 0.4187 G - 0.0813 B + 128 \end{aligned}$$

$$\begin{aligned} R &= Y + 1.402 (Cr - 128) \\ G &= Y - 0.34414 (Cb - 128) - 0.71414 (Cr - 128) \\ B &= Y + 1.772 (Cb - 128) \end{aligned}$$

2. Keep the  $Y$  component.

Downsample the  $Cb$  and  $Cr$  components in 2 times.

Apply JPEG compression to each component.



3. Preprocessing for DCT transformation

- Split the image into  $8 \times 8$  non-overlapping blocks
- In each block subtract global mean computed as  $2^{k-1}$ , where  $k$  is the number of gray levels in the image

4. DCT transformation per block:  $T(u, v)$

5. Block coefficients quantization:

$$\bar{T}(u, v) = \text{round}\left(\frac{T(u, v)}{Z(u, v)}\right)$$

16	11	10	16	24	40	51	61;
12	12	14	19	26	28	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;]

6. Symbols encoding:

- Zig-zag scanning
- Huffman coding (see *Hint* below)

Zig-zag scanning							
0	1	5	6	14	15	27	28
2	4	7	13	16	26	29	42
3	8	12	17	25	30	41	43
9	11	18	24	31	40	44	53
10	19	23	32	39	45	52	54
20	22	33	38	46	51	55	60
21	34	37	47	50	56	59	61
35	36	48	49	57	58	62	63

### Exercise 1.

- Implement by yourself the JPEG compression algorithm described above.
- Take a colour image *test.png*. Compress this image by your JPEG algorithm implementation.
- Perform the image reconstruction.
- Display the original and reconstructed images. Compare those images based on the *MSE*. Make a conclusion about the compression efficiency based on the visual images quality and the files size.

### Hint

For Huffman coding use the Matlab implementation <https://ch.mathworks.com/help/comm/ug/huffman-coding-1.html> and see [https://moodle.unige.ch/pluginfile.php/241361/mod\\_folder/content/0/IN%20Theme%2010%20-%20lossless%20image%20compression.pdf](https://moodle.unige.ch/pluginfile.php/241361/mod_folder/content/0/IN%20Theme%2010%20-%20lossless%20image%20compression.pdf) page 22.

### Submission

Please archive your report and codes in “Name\_Surname.zip” (replace “Name” and “Surname” with your real name), and upload to “Assignments/TP4: JPEG compression” on <https://chamilo.unige.ch> before **Wednesday, May 15 2019, 23:59 PM**. Note, **the assessment is mainly based on your report, which should include your answers to all questions and the experimental results.**