Filters in Image Processing

Image Compression Standards

David Svoboda

email: svoboda@fi.muni.cz
Centre for Biomedical Image Analysis
Faculty of Informatics, Masaryk University, Brno, CZ

November 8, 2019

Outline

Image Compression Standards

- Binary image compression
  - CCITT Group 3 & 4 – *run length encoding*
    (Consultative Comittee of the International Telephone and Telegraphy)
  - JBIG – *arithmetic compression*
    (Joint Bilevel Imaging Group)
- Continuous tones still image compression standards
  - JPEG – *discrete cosine transform*
    (Joint Photographics Experts Group)
  - JPEG 2000 – *discrete wavelet transform*
- Video compression standards
  - MPEG-1, MPEG-2 – *DCT and predictive coding*
    (Moving Pictures Experts Group)
  - MPEG-4 – *Wavelet based coding*

CCITT Group 3 1D

Specification

- Proposed for fax machines (black-and-white bitmaps).
- For encoding a combination of Huffman and RLE is used.
- Often called *Modified Huffman coding*.
- Trained on eight documents:
  - typed business letter (English)
  - circuit diagram (hand drawn)
  - printed and typed invoice (French)
  - densely types report (French)
  - printed technical article including figures and equations (French)
  - graph with printed captions (French)
  - dense document (Kanji)
  - handwritten memo with very large white-on-black letters (English)
- Code words for lengths 1–2560 pixels are stored in publicly known table.

Part of table with CCITT Group 3 codes

run length	white code word	black code word
0	00110101	0000110111
1	000111	010
2	0111	11
3	1000	10
4	1011	011
5	1100	0011
6	1110	0010
7	1111	00011
⋮	⋮	⋮
1344	011011010	0000001010011
1408	011011011	0000001010100
1472	010011000	0000001010101
⋮	⋮	⋮

CCITT Group 3 1D

Specification

- The documents are scanned line by line.
- *Termination codes* . . . Short run lengths (1–63 pixels)
- *Make-up codes* . . . Long run lengths (64 and more pixels)
- Each run length (white/black) is split into sequence of code words
- Some examples:
  12 white pixels → 001000
  76 white pixels (=64+12) → 11011|001000
  2561 black pixels (=2560+1) → 000000011111|010
  8800 black pixels (=2560+2560+2560+1088+32) →
  00000001111|000000011111|000000011111|0000011110101|0000011010
- Each line end is encoded with a special EOL code word:
  000000000001

CCITT Group 3 2D

Specification

- 2-dimensional coding
  - Images are divided into several groups of K lines.
  - K is typically 2 or 4.
  - The first line of each group is encoded using CCITT Group 3 1D method.
  - The rest of lines is encoded using a differential scheme where new code words specific for 2D encoding are introduced.
- Typical compression ratio $1:10 \sim 1:20$
- Pay attention to error propagation!
- The "K-factor" allows more error-free transmission

CCITT Group 4 2D

Specification

- Designed for encoding data on disk drives → no built-in transition error detection/correction.
- No EOL codes (not needed).
- K set to infinity.

Use of CCITT G3 & G4

- TIFF compression Type 2 . . . G3 1D
- TIFF compression Type 3 . . . G3 2D
- TIFF compression Type 4 . . . G4 2D

JPEG

What does JPEG mean?

- JPEG = Joint Photographic Experts Group
- The popular image file format is "JPEG File Interchange Format (JFIF)"

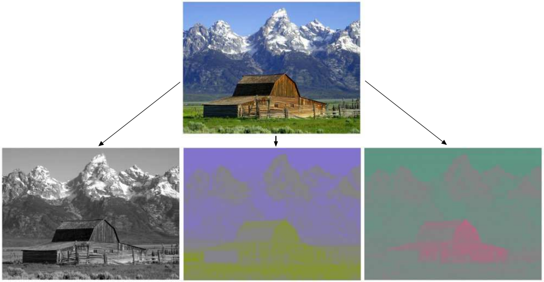
The aims of JPEG:

- storage of continuous tone grayscale and color images
- recognizable image at 0.083 bit/pixel
- useful image at 0.25 bit/pixel
- feasibility of 64 kbit/s (ISDN)

JPEG

1. color component transform 8-bit RGB $\rightarrow$ YCrCb
   (Y is the luminance component and Cb and Cr are the blue and red chrominance components)

$$\begin{pmatrix} Y \\ Cb \\ Cr \end{pmatrix} = \begin{pmatrix} 0.299 & 0.587 & 0.114 \\ -0.169 & -0.331 & 0.500 \\ 0.500 & -0.419 & -0.081 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix}$$



JPEG

2. reduce resolution of Cr and Cb by factor 2
3. process Y, Cr and Cb independently
4. split the image into 8×8 blocks
5. apply 8×8 forward DCT on each block
6. quantization using zonal thresholding
   (left – luminance table; right – chrominance table)

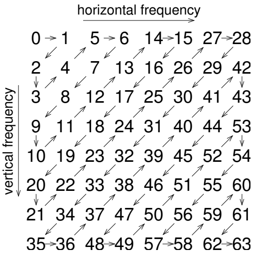
16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

17	18	24	47	99	99	99	99
18	21	26	66	99	99	99	99
24	26	56	99	99	99	99	99
47	66	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99

7. compression quality is driven by Q-scale factor (1–100%) that multiplies the items from the tables given above

JPEG

8. apply lossless predictive coding to quantized DC (lowest frequency) coefficients from DCT
9. read remaining quantized values from DCT in zigzag pattern



10. locate sequences of zero coefficients and use RLE
11. apply Huffman/arithmetic coding on zero run-lengths and magnitude of AC values

1. cut the image into tiles of size greater than $[128 \times 128]$ to prevent blocking artifacts but enabling ROI selection
2. DC level shift for channel of s bit depth:

$$[0, 2^s - 1] \quad \rightarrow \quad [-2^{s-1}, 2^{s-1} - 1]$$

3. color transform: RGB $\rightarrow$ YCrCb
   - irreversible component transform (lossy):

$$\begin{pmatrix} Y \\ Cb \\ Cr \end{pmatrix} = \begin{pmatrix} 0.299 & 0.587 & 0.114 \\ -0.169 & -0.331 & 0.500 \\ 0.500 & -0.419 & -0.081 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix}$$

   - reversible component transform (lossless):

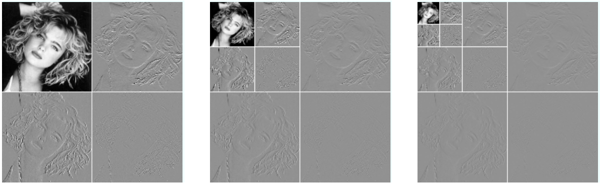
$$\begin{aligned} Y &= \left\lfloor \frac{R + 2G + B}{4} \right\rfloor \\ Cr &= B - G \\ Cb &= R - G \end{aligned}$$

4. apply DWT via lifting scheme to each tile independently
   - Daubechies 9/7 filter (floating-point wavelet – lossy):

     LoD = [0.027 -0.017 -0.078 0.267 0.603 0.267 -0.078 -0.017 0.027]

     HiD = [0.046 -0.029 -0.296 0.558 -0.296 -0.029 0.046]
   - Le Gall 5/3 filter (integer wavelet – lossless):

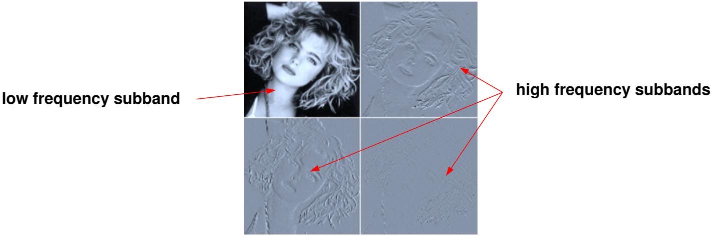
     LoD = [-0.125 0.25 0.75 0.25 -0.125]
     HiD = [-0.25 0.5 -0.25]
5. level of detail of DWT is a parameter of compression

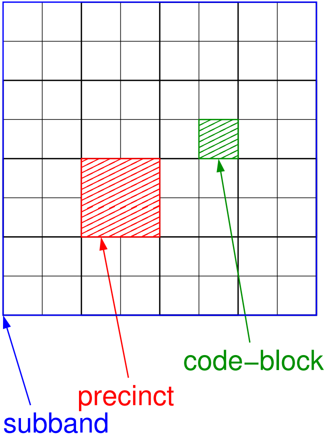


6. quantize DWT coefficients for each subband b individually:

$$q_b(x,y) = sign(I_b(x,y)) \left\lfloor \frac{|I_b(x,y)|}{\Delta_b} \right\rfloor$$

I_b ... original intensity values in the processed subband b
q_b ... matrix with quantized values
Δ_b ... quantization step b (for lossless compression $\Delta_b = 1$)

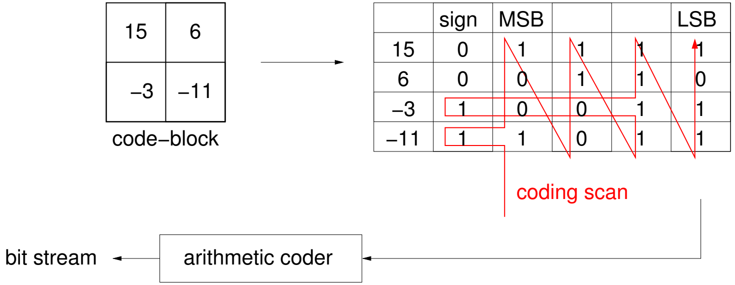




7. divide individual DWT subbands into rectangular blocks called *precincts*
8. each precinct is further divided into non-overlapping rectangles called *code-blocks*

Notice: Each code-block forms the input to entropy encoder and is encoded independently

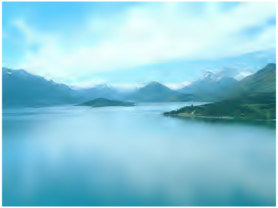
9. the coefficients in a code block are separated into bitplanes and read in the order from MSB to LSB (sign bit is the second one)
10. the sequence of bit is coded using arithmetic encoder (probability of each symbol is adaptively derived from its neighbours)



- an image is partitioned into:
  tiles – precincts – code-blocks $\leftrightarrow$ coarse – medium – fine
  - memory efficient implementations
  - streaming
  - easy direct access to specified positions
- simple modifications (rotation, crop, ...) of an image do not require decompression of an image
- reading of ROIs instead of decompressing the whole images
- progressive access
- elimination of blocking artifacts typical for JPEG

original (979 kB)

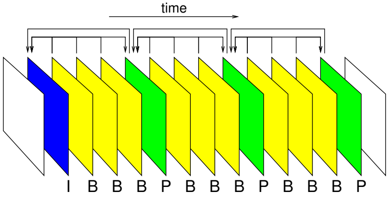
JPEG (6.21 kB)



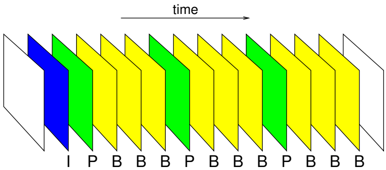
JPEG2000 (1.83 kB)

- I-frames . . . intraframe/independent frame
  frame encoded using JPEG-like compression scheme
  (in MPEG-1 & MPEG-2)
- P-frames . . . predictive frame
  the difference (measure of correlation) to a previous P- or I-frame;
  variable length coded
- B-frames . . . bidirectional frame (the most common frames)
  difference between the current frame and a prediction of it based on
  the previous I- or P-frame and the next P-frame
  $\rightarrow$ when movement of an object gradually uncovers a background area

Display order of frames:

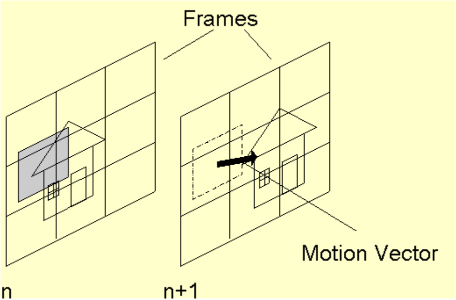


Coding order:



Motion Vector

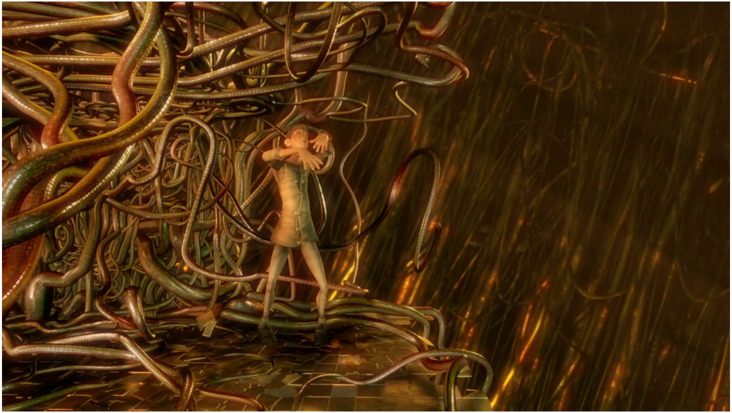
$$(\Delta_x, \Delta_y) = (block_x^{n+1} - block_x^n, block_y^{n+1} - block_y^n)$$



Notice: Correct detection of *motion vector* influences the compression quality.

MPEG

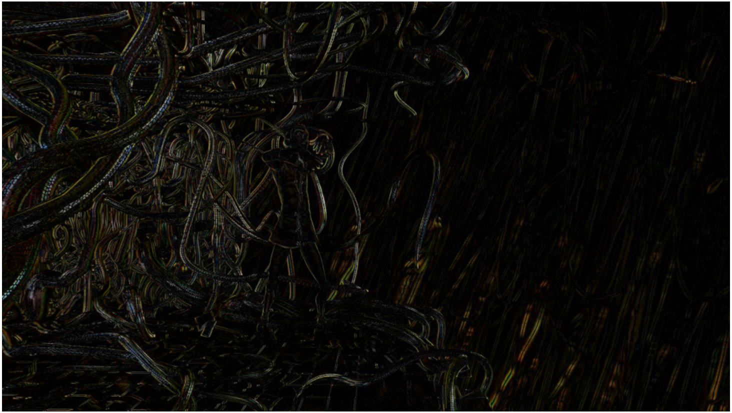
Motion compensation (an example)



Original frame

MPEG

Motion compensation (an example)



Difference between original and the next frame

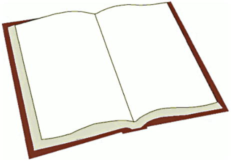
MPEG

Motion compensation (an example)

Difference between original and the next frame shifted by 2 pixels to the right

Bibliography

- Salomon, D. Data Compression, The Complete Reference, 4th edition, Springer, London, 2007, ISBN-1846286025
- Gonzalez, R. C., Woods, R. E. Digital image processing / 2nd ed., Upper Saddle River: Prentice Hall, 2002, pages 793, ISBN 0201180758
- Rabbani, M., Joshi R. An overview of the JPEG 2000 still image compression standard, Signal Processing: Image Communication, Volume 17, Issue 1, January 2002, Pages 3-48, ISSN 0923-5965



You should know the answers . . .

- Why do we convert RGB into YCrCb in JPEG and JPEG2000 standard?
- Explain the meaning of luminance/chrominance table.
- Explain the origin of the phenomenon called *blocking artifact*.
- How and why do we use predictive coding in JPEG standard?
- Describe the frame ordering in MPEG scheme.
- What is a difference between I, P, and B frames?
- How do we loose the information when using JPEG/JPEG2000?
- Encode the sequence [101010101010] using CCITT G3 1D scheme.