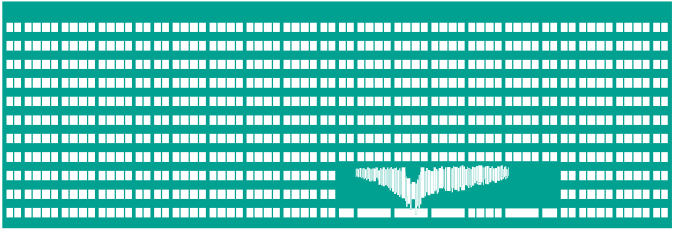
VŠB TECHNICKÁ UNIVERZITA OSTRAVA

VSB TECHNICAL UNIVERSITY OF OSTRAVA



www.vsb.cz

PNG, JPEG

LZ algorithms

Michal Vasinek

VSB – Technical University of Ostrava

name.surname@vsb.cz

May 23, 2019

VSB TECHNICAL UNIVERSITY OF OSTRAVA

Overview

- Digital Image
- PNG
- Discrete cosine transform
- JPEG
- Future

Digital Image

- A digital image is a rectangular array of dots, or picture elements, arranged in m rows and n columns.
- The expression $m \times n$ is called the resolution of the image.
- Picture elements are called pixels.
- dpi - dots per inch, pixels per inch.
- RGB standard - 8bit per colour component.

resolution	size[bytes]
1024 x 1024	3,145,728
2048 x 2048	12,582,912
4656 x 3496(16Mpx)	48,832,128

Image compression inevitable, lossless and lossy methods.

Basic principle of image compression

The Principle of Image Compression

If we select a pixel in an image at random, there is a good chance that its neighbors will have the same color or very similar colors.

Image compression is therefore based on the fact that neighboring pixels are highly correlated. This correlation is also called spatial redundancy.

PNG

- Portable network graphics - PNG.
- Design finalized in 1996.
- Aim to overcome patent issues related to GIF.

PNG

- Supports images with 1,2,4,8,16 bitplanes.
- Sophisticated color matching.
- A transparency feature - alpha channel.
- Lossless compression using Deflate combined with pixel prediction.
- Extensibility: meta-information can be added to an image file without creating incompatibility with existing applications.

PNG - encoding

- The first step - delta filtering (five options - next slides).
- Filtering calculates predicted value for each pixel and replaces the pixel with the difference between the pixel and its predicted value.
- Filtering is done on each image row.
- Differences encoded using Deflate.

PNG - filtering

- Each row starts with a byte indicating filtering method for the row.
- The heuristic recommended by PNG for adaptive filtering is to apply all five filtering types to the row of pixels and select the type that produces the smallest sum of absolute values of outputs.

Type 0

No filtering.

Type 1 - sub

Filtering for byte $B_{i,j}$ of the image:

$$B_{i,j} = (B_{i,j} - B_{i-t,j}) \mod 256$$

where t is colour component. For instance for RGB 8 bit bitplane, each colour component repeats per three bytes, hence $t = 3$.

PNG - filtering

Type 2 - up

Upper byte.

$$B_{i,j} = B_{i,j} - B_{i,j-1} \mod 256$$

Type 3 - average

Average of left and up neighbouring byte.

$$B_{i,j} = (B_{i,j} - [B_{i-t,j} + B_{i,j-1}]/2) \mod 256$$

Type 4 - Paeth predictor

$$B_{i,j} = B_{i,j} - PaethPredictor[B_{i-t,j}, B_{i,j-1}, B_{i-t,j-1}]) \mod 256$$

```
function PaethPredictor (a, b, c)
begin
; a=left, b=above, c=upper left
p:=a+b-c ;initial estimate
pa := abs(p-a) ; compute distances
pb := abs(p-b) ; to a, b, c
pc := abs(p-c)
; return nearest of a,b,c,
; breaking ties in order a,b,c.
if pa<=pb AND pa<=pc then return a
else if pb<=pc then return b
else return c
end
```

GIF - interlacing

- Remember: GIF uses LZW (patent issues).
- Mid 90's: need for a format to replace GIF.
- GIF demonstration

PNG - interlacing

- Divides the image into blocks 8x8 pixels each.
- Each block is displayed in seven steps - interlacing using Adam7.

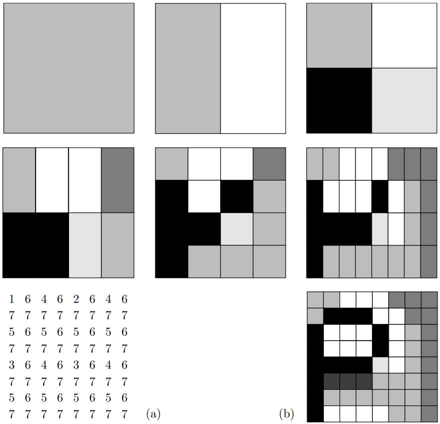


Figure: PNG - interlacing, Salomon.

PNG - interlacing

Figure: PNG vs. GIF - interlacing, https://nuwen.net/png.html

PNG - interlacing

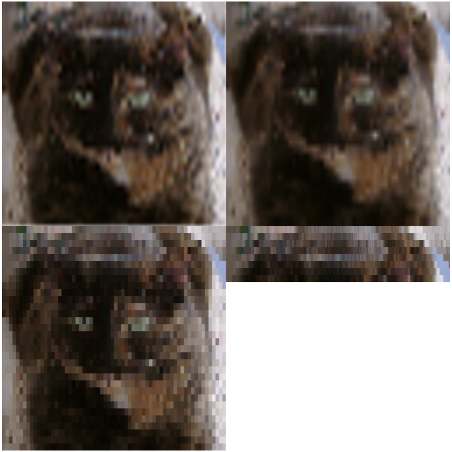


Figure: PNG vs. GIF - interlacing, https://nuwen.net/png.html

Figure: PNG vs. GIF - interlacing, https://nuwen.net/png.html

PNG - interlacing

Figure: PNG vs. GIF - interlacing, https://nuwen.net/png.html

PNG - interlacing

Figure: PNG vs. GIF - interlacing, https://nuwen.net/png.html

Figure: PNG vs. GIF - interlacing, https://nuwen.net/png.html

PNG - interlacing

Figure: PNG vs. GIF - interlacing, https://nuwen.net/png.html

Lossy image compression - metrics

Developers and implementers of lossy image compression methods need a standard metric to measure the quality of reconstructed images P compared with the original ones O.

- Two methods:
  - Mean squared error (MSE):

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(P_i - Q_i)^2$$

  - Peak signal to noise ratio (PSNR):

$$PSNR = 20\log_{10}\frac{max_i|P_i|}{\sqrt{MSE}}$$

JPEG

- Joint Photographic Experts Group (JPEG).
- The work on the format started in June 1987 and the first proposal of the stadard produced in 1991.

Seven JPEG steps

1. Color transformation RGB to YC_bC_r i.e. luminance/chrominance color space.
2. Downsampling of chrominance components. Yields approx. 1/2 of the original size.
3. Reorganization of the image into groups of 8x8 pixels called data units.
4. Discrete cosine transform to each data unit.
5. Frequency components are divided by quantization coefficients.
6. Quantized values encoded by combination of RLE and Huffman.
7. Formation of the final format. Adding headers and JPEG required parameters.

JPEG decoders does these steps in reverse order.

- Three components:
  - Y - brightness of the pixel.
  - C_b - blue chrominance.
  - C_r - red chrominance.

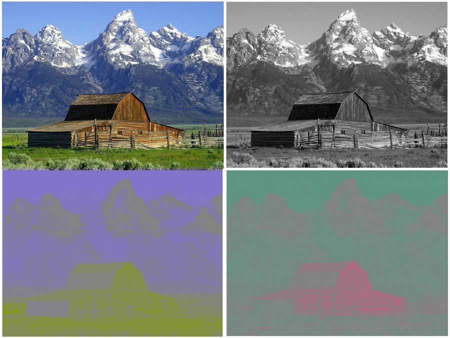


Figure: YCbCr - https://en.wikipedia.org/wiki/YCbCr

Downsampling

- Reduction of chrominance components.
- Usually one of three ratios:
  - 4:4:4 - no downsampling.
  - 4:2:2 - reduction by a factor of 2 in the horizontal direction.
  - 4:2:0 - reduction by a factor of 2 in the horizontal and vertical directions (The most common).

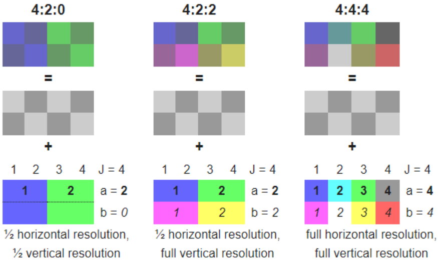


Figure: Chroma subsampling - https://en.wikipedia.org/wiki/Chroma_subsampling

Block Splitting

- Each channel is divided into 8 x 8 blocks of values.
- For different chroma subsampling this yields Minimum Coded Unit(MCU):
  - 8x8 - (4:4:4)
  - 16x8 - (4:2:2)
  - 16x16 - (4:2:0)
- Each data unit is encoded separately.

DCT

- Each block is converted into frequency domain using normalized two-dimensional discrete cosine transformation (DCT).
- Before computing DCT the block is shifted so that the range of values, typically [0;255], is turned into the range [-128;127].

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

Figure: DCT preprocessing - https://en.wikipedia.org/wiki/JPEG

- Expression for two dimensional DCT:

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

- u is the horizontal spatial frequency.
- v is the vertical spatial frequency.
- $\alpha(u)$ is $1/\sqrt{2}$ if $u = 0$ and 1 otherwise.
- $g(x, y)$ is the value of pixel.
- $G_{u,v}$ are coefficients of DCT.

$$
G = \begin{bmatrix}
-415.38 & -30.19 & -61.20 & 27.24 & 56.12 & -20.10 & -2.39 & 0.46 \\
4.47 & -21.86 & -60.76 & 10.25 & 13.15 & -7.09 & -8.54 & 4.88 \\
-46.83 & 7.37 & 77.13 & -24.56 & -28.91 & 9.93 & 5.42 & -5.65 \\
-48.53 & 12.07 & 34.10 & -14.76 & -10.24 & 6.30 & 1.83 & 1.95 \\
12.12 & -6.55 & -13.20 & -3.95 & -1.87 & 1.75 & -2.79 & 3.14 \\
-7.73 & 2.91 & 2.38 & -5.94 & -2.38 & 0.94 & 4.30 & 1.85 \\
-1.03 & 0.18 & 0.42 & -2.42 & -0.88 & -3.02 & 4.12 & -0.66 \\
-0.17 & 0.14 & -1.07 & -4.19 & -1.17 & -0.10 & 0.50 & 1.68
\end{bmatrix}
$$

Figure: DCT result - https://en.wikipedia.org/wiki/JPEG

$$G_f = \sqrt{\frac{2}{n}} C_f \sum_{t=0}^{n-1} p_t cos\left[\frac{(2t+1)f\pi}{2n}\right]$$

where C_f is $1/\sqrt{2}$ if $f = 0$ and 1 otherwise.
The inverse DCT:

$$p_t = \sqrt{\frac{2}{n}} \sum_{j=0}^{n-1} C_j G_j \cos\left[\frac{(2t+1)j\pi}{2n}\right]$$

DCT 1D example

The following experiment illustrates the power of the DCT in one dimension. We start with the set of eight correlated data items $\mathbf{p} = (12, 10, 8, 10, 12, 10, 8, 11)$, apply the DCT in one dimension to them, and find that it results in the eight coefficients

$$28.6375,\ 0.571202,\ 0.46194,\ 1.757,\ 3.18198,\ -1.72956,\ 0.191342,\ -0.308709.$$

These can be fed to the IDCT and transformed by it to precisely reconstruct the original data (except for small errors caused by limited machine precision). Our goal, however, is to compress the data by quantizing the coefficients. We first quantize them to $28.6, 0.6, 0.5, 1.8, 3.2, -1.8, 0.2, -0.3$, and apply the IDCT to get back

$$12.0254,\ 10.0233,\ 7.96054,\ 9.93097,\ 12.0164,\ 9.99321,\ 7.94354,\ 10.9989.$$

We then quantize the coefficients even more, to $28, 1, 1, 2, 3, -2, 0, 0$, and apply the IDCT to get back

$$12.1883,\ 10.2315,\ 7.74931,\ 9.20863,\ 11.7876,\ 9.54549,\ 7.82865,\ 10.6557.$$

Finally, we quantize the coefficients to $28, 0, 0, 2, 3, -2, 0, 0$, and still get back from the IDCT the sequence

$$11.236,\ 9.62443,\ 7.66286,\ 9.57302,\ 12.3471,\ 10.0146,\ 8.05304,\ 10.6842,$$

where the largest difference between an original value (12) and a reconstructed one (11.236) is 0.764 (or 6.4% of 12).

Quantization

- The human eye is good at seeing small differences in brightness over a relatively large area, but not so good at distinguishing the exact strength of a high frequency brightness variation.
- This allows one to greatly reduce the amount of information in the high frequency components.
- Divide each value of $G_{u,v}$ by a predefined constant and round to the nearest integer.

$$B_{j,k} = round(\frac{G_{j,k}}{Q_{j,k}})$$

Quantization

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

Figure: DCT quantization matrix - https://en.wikipedia.org/wiki/JPEG

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

Figure: DCT after quantization - https://en.wikipedia.org/wiki/JPEG

- Zigzag ordering of JPEG components.
- Aims to produce sequence of zeros => Run Length Encoding followed by Huffman coding.

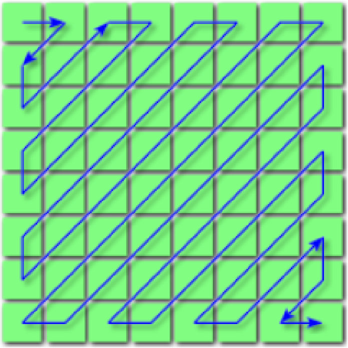


Figure: zigzag - https://en.wikipedia.org/wiki/JPEG

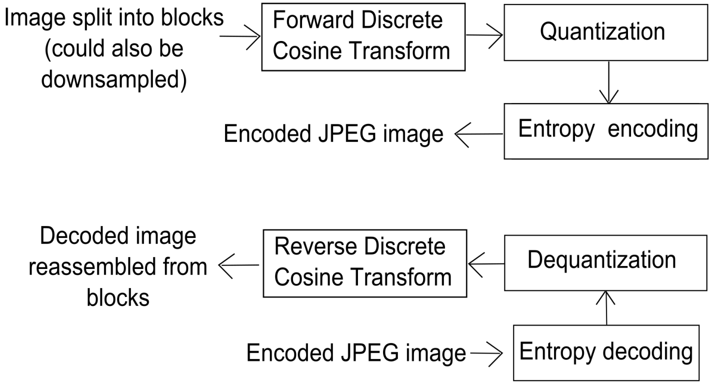


Figure: JPEG - https://en.wikipedia.org/wiki/JPEG

Future

- Many scientists believe the future of image compression lies in inpainting methods.
- Based on interpolation of missing data.
- Selection of pixels bearing the most information, the rest of pixels interpolated.
- Inpainting presentation link

Future - Inpainting

original | data near edges kept | inpainted

Figure: Inpainting reconstruction based on edges.

Thank you for your attention

Michal Vasinek

VSB – Technical University of Ostrava

name.surname@vsb.cz

May 23, 2019

VSB TECHNICAL UNIVERSITY OF OSTRAVA