COS 791

Lecture 2

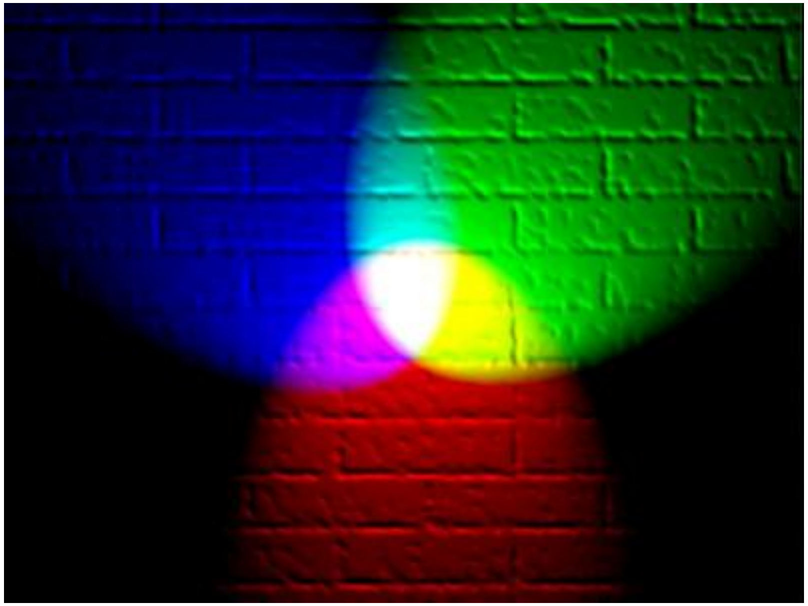
Colour models, image formats and compression

RGB Colour model

- All colours in the RGB model are defined as a combination of Red, Green and Blue (RGB)
- Additive colour model:
  - Red, green and blue light are added together to form a colour
- Full intensity of all three colours = white
- Low intensity of all three colours = black / dark

RGB colour model

- Uses light to display colour
- Used for computer and television display
  - Cathode Ray Tube (CRT)
  - Liquid Crystal Display (LCD)
  - Plasma display
- Is device dependent



Colour representation

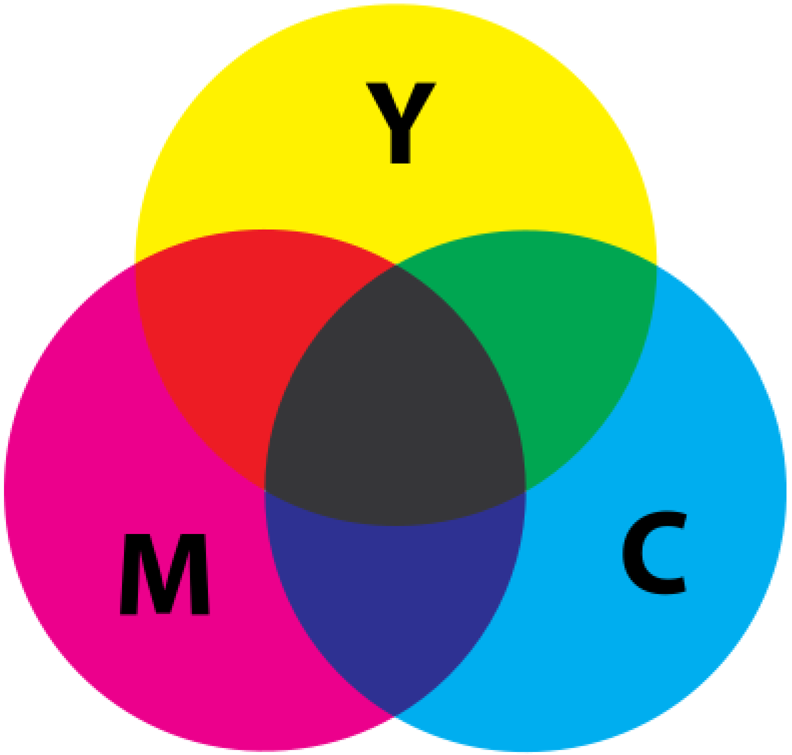
- Other colour models:
  - CMYK
  - YUV
  - YC_rC_b

CMYK colour model

- Subtractive colour model
- Colours created by absorption and reflected light
- When you subtract red, green and blue in turn from white you get:
  - Cyan, Magenta and Yellow (CMY)
  - Black (K) is added to the model

CMYK colour model

- Result of reflected light
- Used for printed material
- Combination of three should give black, but since the black is often not pure, black is added to the model



Colour representation

- RGB and CMYK not economical for transmission of colour information
- YUV was developed for transmission of colours for colour television
- Luminance / Chrominance:
  - Luminance (Y) = brightness
  - Chrominance (U and V) = colour

Colour representation

- Luminance (Y)
  - Measure of overall brightness
  - Weighted linear combination of RGB channels determined by sensitivity of the human eye to the three RGB colours
  - For example:
    - $Y = 0.299R + 0.587G + 0.114B$

Colour representation

- Chrominance components:
  - Difference between red and blue respectively and luminance
  - $U = R - Y$
  - $V = B - Y$

Colour representation

- RGB colours are represented in the range {0…255}
- When transformed to YUV luminance Y share the same range
- U and V fall into the range {-179,…,179}.
- To adjust all three components to the same range representable with 8 bits the chrominance components are further linearly transformed to C_r and C_b, obtaining thus the colour model YC_rC_b

Colour representation

- Human eyes are less sensitive to changes in chrominance than in luminance
  - Chrominance signals are often represented with fewer bits without introducing visual distortion to the image.
  - Utilized in JPEG compression and TV signals
- YC_rC_b model – IIF, TIFF, JPEG and MPEG

Colour representation

- YUV and YC_rC_b are not absolute colour models, rather they are ways of encoding RGB information

Digital image representation

- To a computer, an image is a collection of numbers that constitute different light intensities in different areas of the image
- Individual points are referred to as pixels
- Pixels form a rectangular map of where each pixel is located and its colour

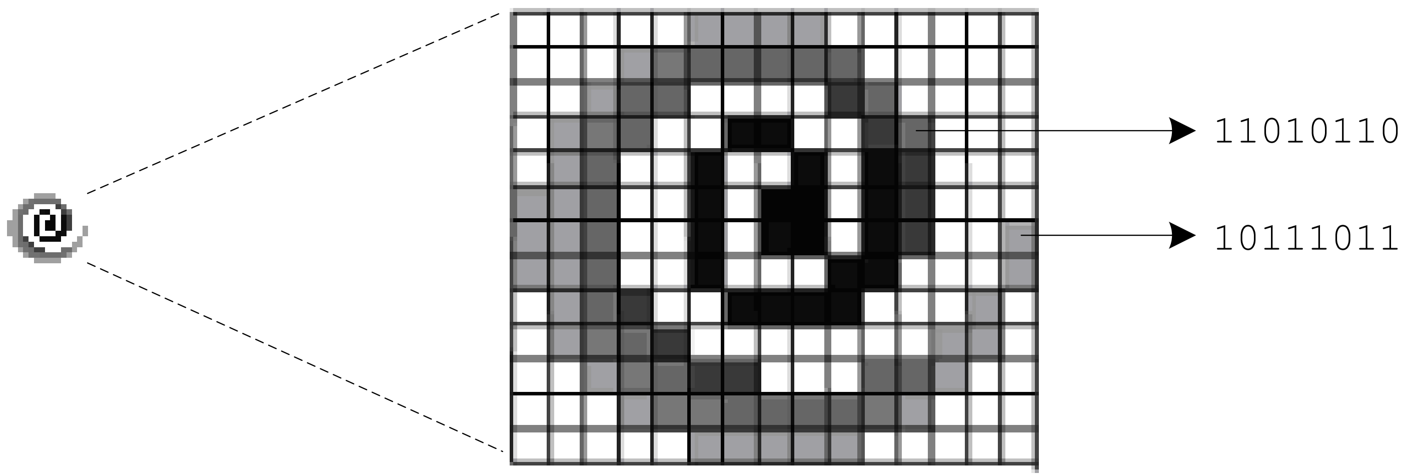
Digital image representation

- Pixels make up the raster data of an image
- A common image size is 640 x 480 pixels.
  - Such an image would contain about 300,000 pixels

Digital image representation

- Bit depth:
  - Number of bits used to store the colour of each pixel
- Bit depth 8:
  - Maximum of 8 bits are used to decribe the colour of each pixel
  - Can thus display 256 different colours
  - For example, greyscale

Digital image representation

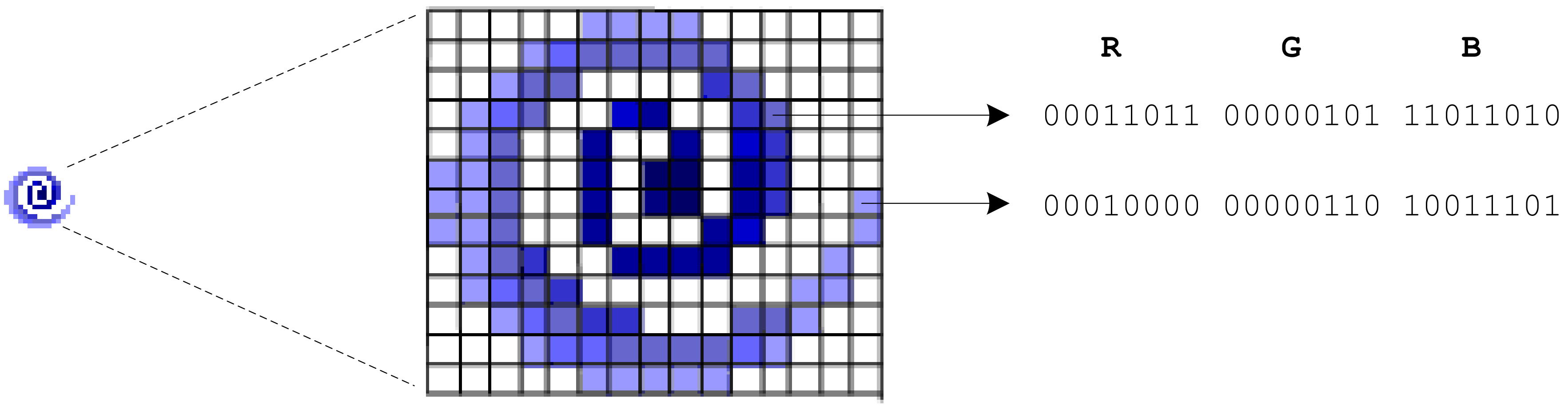


- Greyscale image with bit depth 8 that can display 256 different intensities of grey
- Each pixel is a numerical value converted to a bit pattern

Digital image representation

- Colour images are often stored as 24-bit images
- Uses RGB model
  - Each colour component is represented by 8 bits
  - Darkest intensity 0 and brightest value 255
    - White = (255, 255, 255)

Digital image representation



- Each pixel is represented by three 8-bit values: one red, one green and one blue
- States the quantity of each primary colour as a numeric value that is converted to a bit pattern

Digital image representation

- In one pixel, there can be 256 different quantities of red, green and blue,
- Adds up to more than 16-million combinations
- In comparison:
  - a quality offset printing press can print about 4,000 colours,
  - a traditional film photograph can contain in the region of 6-million colours
  - the average human eye can recognise between 7 and 10-million colours

Compression

- Image filesizes have to be realistic
- Compression:
  - Techniques that make use of mathematical formulas to analyse and condense image data, resulting in smaller file sizes

Compression

- Two types of compression:
  - Lossless compression
  - Lossy compression

Lossless compression

- Represents data in mathematical formulas while not removing any information from the original image.
- Original image’s integrity is maintained
- Decompressed image output is bit-by-bit identical to the original image input

Lossy compression

- Compresses by discarding excess image data from the original image
- Removes details that are too small for the human eye to differentiate
- Results in close approximations of the original image, although not an exact duplicate
- Creates smaller filesizes

Image formats

- Images fall into one of two categories:
  - Spatial domain formats
  - Transform domain formats

Spatial domain formats

- Colours are represented as a sufficiently dense rectangular grid
- Image formats that fall into spatial domain:
  - Raster images
  - Palette images

Raster images

- Image data is stored in a row by row manner with one or more bytes per pixel depending on the bit depth
- Most common formats are BMP, TIFF and PNG
- Compression is optional, if compression is used it is lossless
  - BMP typically use Run-length encoding (RLE) as an optional extra

Run-length encoding

- Replaces a sequence of identical values

  $v_1, v_2, \ldots, v_n$

  by a pair of values (v, n) which indicates that the value v is replicated n times

Run-length encoding

- On pixel level:
  - Look for replicated pixels
  - For example:

Red	**Green**	**Blue**
(10100110	11000100	00001100)
(10100110	11000100	00001100)
(10100110	11000100	00001100)

- Can be compressed to:

3 10100110 11000100 00001100

Run-length encoding

- Can also compress each colour plane separately
- This results in better compression rate

Palette images

- More suitable for images with low colour depth
  - Computer-generated graphics, line drawings and cartoons
- Most popular is GIF (Graphical Interchange Format)
- By definition a GIF image cannot have a bit depth greater than 8, thus the maximum number of colours that can be used is 256
- Image data is a rectangular M x N array of 8-bit pointers to a colour palette

Transform domain formats

- People perceive natural images as a collection of segments filled with texture rather than as matrices of pixels
- Highly inefficient to store images as rectangular matrices of colour

Transform domain formats

- Transform domain formats use lossy compression to create an approximation of the original uncompressed image
- Compression gets rid of excess data

Transform domain formats

- Mathematical operation (called the transform) converts image from spatial (2D) domain to transform domain
- Most commonly used transforms:
  - Discrete Cosine Transform (DCT)
  - Discrete Wavelet Transform (DWT)
  - Discrete Fourier Transform (DFT)

Transform domain formats

- JPEG images
  - Joint Photographic Expert Group
  - Popular on Internet because of small filesizes
  - Uses DCT
- DWT is used in JPEG2000
- We will only look at DCT and JPEG

JPEG compression

- High frequency image components which contribute less to the overall picture are compressed at a higher level than low frequency components
  - Low frequency components = pixel values that change slowly over space
  - High frequency components = pixel values that change rapidly over space

JPEG compression

- Human visual system is less sensitive to quantization errors in high frequency

JPEG compression

- Five steps for JPEG compression:
  - Colour transformation
  - Division into blocks and subsampling
  - DCT transform
  - Quantization
  - Encoding

1. Colour transformation

- Colour transformed from RGB colour model to YC_rC_b
- Strictly not necessary, but enables higher compression rates

2. Division into blocks and subsampling

- Each channel (Y, C_r and C_b) is divided into 8 x 8 blocks
- For example, the following 8 x 8 bit subimage:

$$\text{Original 8x8 block} = \begin{bmatrix} 52 & 55 & 61 & 66 & 70 & 61 & 64 & 73 \\ 63 & 59 & 55 & 90 & 109 & 85 & 69 & 72 \\ 62 & 59 & 68 & 113 & 144 & 104 & 66 & 73 \\ 63 & 58 & 71 & 122 & 154 & 106 & 70 & 69 \\ 67 & 61 & 68 & 104 & 126 & 88 & 68 & 70 \\ 79 & 65 & 60 & 70 & 77 & 68 & 58 & 75 \\ 85 & 71 & 64 & 59 & 55 & 61 & 65 & 83 \\ 87 & 79 & 69 & 68 & 65 & 76 & 78 & 94 \end{bmatrix}.$$

Examples from wikipedia

2. Division into blocks and subsampling

- Human eyes are more sensitive to changes in brightness (the Y component) than changes in colour
- To achieve a higher compression rate chrominance C_r and C_b can be subsampled

Chrominance subsampling

- To achieve greater compression ratios one can measure chrominance only over every 2x2 block or even 4x4 block
- This result in fewer chrominance values stored in comparison with luminance values

JPEG compression

- Next is the DCT transform
- However, first range should be adjusted:
  - Y, Cr and Cb components are in the range 0...255
  - The DCT transform is designed to work on pixel values of between -128 and 127
  - Subtract 128 from all the values in the 8 x 8 block to adjust the range

Example after range adjustment

$$
\begin{array}{cc}
 & \overset{x}{\longrightarrow} \\
g = & \begin{bmatrix}
-76 & -73 & -67 & -62 & -58 & -67 & -64 & -55 \\
-65 & -69 & -73 & -38 & -19 & -43 & -59 & -56 \\
-66 & -69 & -60 & -15 & 16 & -24 & -62 & -55 \\
-65 & -70 & -57 & -6 & 26 & -22 & -58 & -59 \\
-61 & -67 & -60 & -24 & -2 & -40 & -60 & -58 \\
-49 & -63 & -68 & -58 & -51 & -60 & -70 & -53 \\
-43 & -57 & -64 & -69 & -73 & -67 & -63 & -45 \\
-41 & -49 & -59 & -60 & -63 & -52 & -50 & -34
\end{bmatrix} \Big\downarrow y.
\end{array}
$$

3. DCT Transform

- Basic idea:
  - The changes in values over an 8 x 8 block are rather mild
  - Instead of recording the individual values, we could record, the average values and how much each pixel differs from this average value.
  - Is many cases these differences would be rather small and could be safely ignored

3. DCT Transform

- In an 8 x 8 block, rapid changes in values (in other words high frequency components) will be averaged and basically ignored

3. DCT Transform

- The YC_rC_b signals from each block are transformed from the spatial domain to the transform domain with the DCT.
- For an 8 x 8 block of luminance (or chrominance) values of the 8 x 8 block of DCT coefficients are computed as follows:

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

Where:

- u is the horizontal spatial value, for the integers 0 <= u <= 8
- v is the vertical spatial value, for the integers 0 <= v <= 8
- $\alpha(u)$ = 1/√2, if u = 0 or 1 otherwise
- $g_{x,y}$ is the pixel value at coordinates (x,y)
- $G_{u,v}$ is the DCT coefficient at coordinates (u,v)

Example after DCT transform

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

3. DCT Transform

- The larger coefficient $G[0,0]$ is called the DC coefficient
  - The DC coefficient defines the basic hue of the whole block
- The remaining 63 coefficients are called the AC coefficients
- Most of the signal is aggregated in one corner of the result – this aids in encoding later on

3. DCT Transform

- DCT temporarily increases the bit depth of the data since the transform works on real numbers
- This may force the codec to temporarily use 16-bits to store the coefficients, however this is only done on a small part of the image at a time

4. Quantization

- Human eye is good at seeing small differences in brightness over a relatively large area, but not good at measuring the exact strength of the high frequency brightness
- Higher frequency components can be reduced

4. Quantization

- Reduces the number of possible values
- Enables representation of DCT coefficients using fewer bits
- Results in loss of information

= the **lossy** part of the algorithm

4. Quantization

- DCT coefficients $G[u,v]$ are quantized by dividing them by an integer value from a quantization table $Q[u,v]$ and rounding the result to the nearest integer.

$B[u,v]$ = *round*($G[u,v]$ / $Q[u,v]$)

4. Quantization

- Luminance and chrominance signals may use different quantization tables.
- Larger values of quantization produce higher compression but introduce more perceptual distortion.
- JPEG standard recommends a set of quantization tables indexed by a quality factor

Quantization tables

- Quality level of 50;

$$Q_{50} = \begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix}$$

Quantization tables

- Quality level of 10:

$$Q_{10} = \begin{bmatrix} 80 & 60 & 50 & 80 & 120 & 200 & 255 & 255 \\ 55 & 60 & 70 & 95 & 130 & 255 & 255 & 255 \\ 70 & 65 & 80 & 120 & 200 & 255 & 255 & 255 \\ 70 & 85 & 110 & 145 & 255 & 255 & 255 & 255 \\ 90 & 110 & 185 & 255 & 255 & 255 & 255 & 255 \\ 120 & 175 & 255 & 255 & 255 & 255 & 255 & 255 \\ 245 & 255 & 255 & 255 & 255 & 255 & 255 & 255 \\ 255 & 255 & 255 & 255 & 255 & 255 & 255 & 255 \end{bmatrix}$$

4. Quantization

- The larger the quantization the fewer bits can be allocated to each DCT and the larger the loss of information and with it visual distortion.
- Trade-off between image quality and degree of quantization

Comparison of compression rates

Q = 100

Q = 10

Q = 50

Q = 1

Q = 100 vs Q = 50

Q = 50 vs Q = 10

Quantization

- The image quantized at 50% quality uses 4.3% of the storage space of the original image, but has little noticeable loss of detail.

Example after quantization at 50%

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

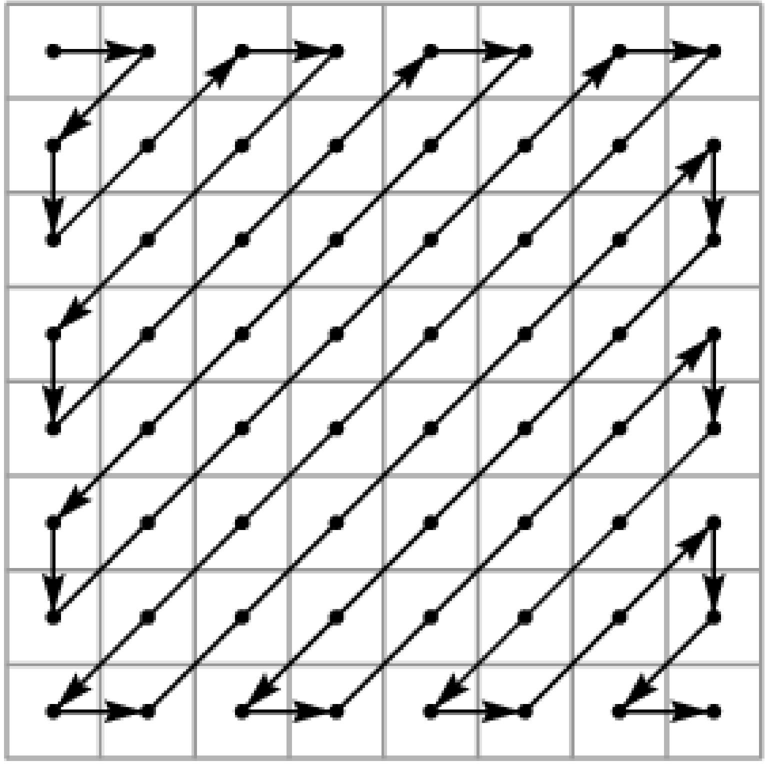
5. Encoding

- Quantized DCT coefficients are encoded to bits as compactly as possible
- Losslessy compressed with a variant of Huffman encoding
- Writes compressed data to file

5. Encoding

- Before storage all coefficients are converted by a encoder into a stream of binary data
- After quantization it is quite common for most coefficients to equal zero
- Coefficients are thus encoded in a zig-zag manner to try and group large streams of zeros together

5. Encoding



5. Encoding

- This block can then be encoded as:

20 -7 1 -1 0 -1 1 0 0 0 0 0 0 0 -2 1 1 0 0 ...

- But instead of recording all zeros, use RLE

20	−7	−1	1	−2	1	0	0
1	0	0	0	1	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
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Decompression

- During decompression JPEG recovers the quantized DCT coefficients from the compressed data stream, takes the inverse and displays the image.
- The whole process is reversible, except the quantization

Decoding

- The compressed stream is decoded into 64 DCT coefficients

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

Decompression

- The decoded 8 x 8 block is then multiplied with the standard quantization table with which is was compressed:

$$\begin{bmatrix} -416 & -33 & -60 & 32 & 48 & -40 & 0 & 0 \\ 0 & -24 & -56 & 19 & 26 & 0 & 0 & 0 \\ -42 & 13 & 80 & -24 & -40 & 0 & 0 & 0 \\ -42 & 17 & 44 & -29 & 0 & 0 & 0 & 0 \\ 18 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Comparison of block during decompression

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

$$\text{During decompression} = \begin{bmatrix} -416 & -33 & -60 & 32 & 48 & -40 & 0 & 0 \\ 0 & -24 & -56 & 19 & 26 & 0 & 0 & 0 \\ -42 & 13 & 80 & -24 & -40 & 0 & 0 & 0 \\ -42 & 17 & 44 & -29 & 0 & 0 & 0 & 0 \\ 18 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Decompression

- An inverse DCT function is then applied to the 8 x 8 block and the results are rounded:

$$\begin{bmatrix} -66 & -63 & -71 & -68 & -56 & -65 & -68 & -46 \\ -71 & -73 & -72 & -46 & -20 & -41 & -66 & -57 \\ -70 & -78 & -68 & -17 & 20 & -14 & -61 & -63 \\ -63 & -73 & -62 & -8 & 27 & -14 & -60 & -58 \\ -58 & -65 & -61 & -27 & -6 & -40 & -68 & -50 \\ -57 & -57 & -64 & -58 & -48 & -66 & -72 & -47 \\ -53 & -46 & -61 & -74 & -65 & -63 & -62 & -45 \\ -47 & -34 & -53 & -74 & -60 & -47 & -47 & -41 \end{bmatrix}$$

- And the range adjustment is reversed (back to 0…255)

Compression of original and decompressed image

Original image = $$\begin{bmatrix} 52 & 55 & 61 & 66 & 70 & 61 & 64 & 73 \\ 63 & 59 & 55 & 90 & 109 & 85 & 69 & 72 \\ 62 & 59 & 68 & 113 & 144 & 104 & 66 & 73 \\ 63 & 58 & 71 & 122 & 154 & 106 & 70 & 69 \\ 67 & 61 & 68 & 104 & 126 & 88 & 68 & 70 \\ 79 & 65 & 60 & 70 & 77 & 68 & 58 & 75 \\ 85 & 71 & 64 & 59 & 55 & 61 & 65 & 83 \\ 87 & 79 & 69 & 68 & 65 & 76 & 78 & 94 \end{bmatrix}.$$

Decompressed image = $$\begin{bmatrix} 62 & 65 & 57 & 60 & 72 & 63 & 60 & 82 \\ 57 & 55 & 56 & 82 & 108 & 87 & 62 & 71 \\ 58 & 50 & 60 & 111 & 148 & 114 & 67 & 65 \\ 65 & 55 & 66 & 120 & 155 & 114 & 68 & 70 \\ 70 & 63 & 67 & 101 & 122 & 88 & 60 & 78 \\ 71 & 71 & 64 & 70 & 80 & 62 & 56 & 81 \\ 75 & 82 & 67 & 54 & 63 & 65 & 66 & 83 \\ 81 & 94 & 75 & 54 & 68 & 81 & 81 & 87 \end{bmatrix}.$$

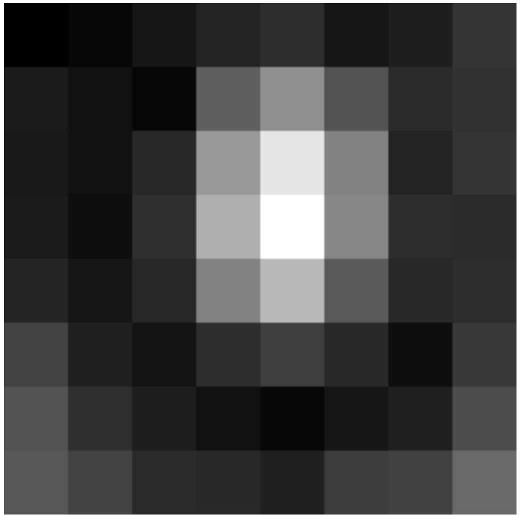
Decompression

- Numerical differences between original image and decompressed image:

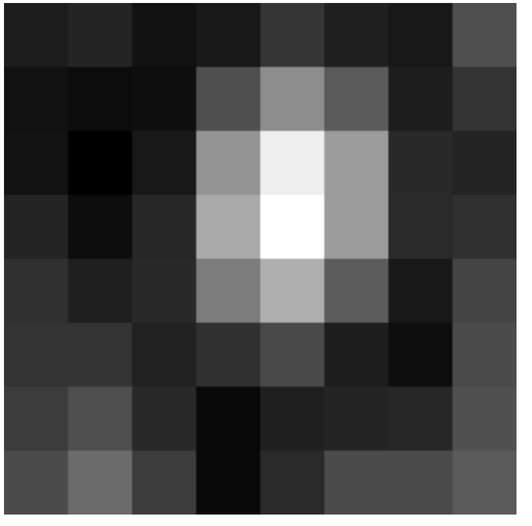
$$\begin{bmatrix} -10 & -10 & 4 & 6 & -2 & -2 & 4 & -9 \\ 6 & 4 & -1 & 8 & 1 & -2 & 7 & 1 \\ 4 & 9 & 8 & 2 & -4 & -10 & -1 & 8 \\ -2 & 3 & 5 & 2 & -1 & -8 & 2 & -1 \\ -3 & -2 & 1 & 3 & 4 & 0 & 8 & -8 \\ 8 & -6 & -4 & -0 & -3 & 6 & 2 & -6 \\ 10 & -11 & -3 & 5 & -8 & -4 & -1 & -0 \\ 6 & -15 & -6 & 14 & -3 & -5 & -3 & 7 \end{bmatrix}$$

Decompression

- Visual differences between original image and decompressed image:



Original image



Decompressed image

Typical usage of JPEG

- Best for photographs or paintings of realistic scenes
- Not good for images with sharp contrast, for example line drawings
- Not well suited for images that will undergo multiple edits since image quality will be lost with every re-edit, for example cropped or resized