Lecture 6

Video Basics

Video Analysis

- Object Detection/Recognition
- Motion Detection/Recognition
- Activity Detection/Recognition

Video is a sequence of Images displayed at certain rate to stimulate motion!

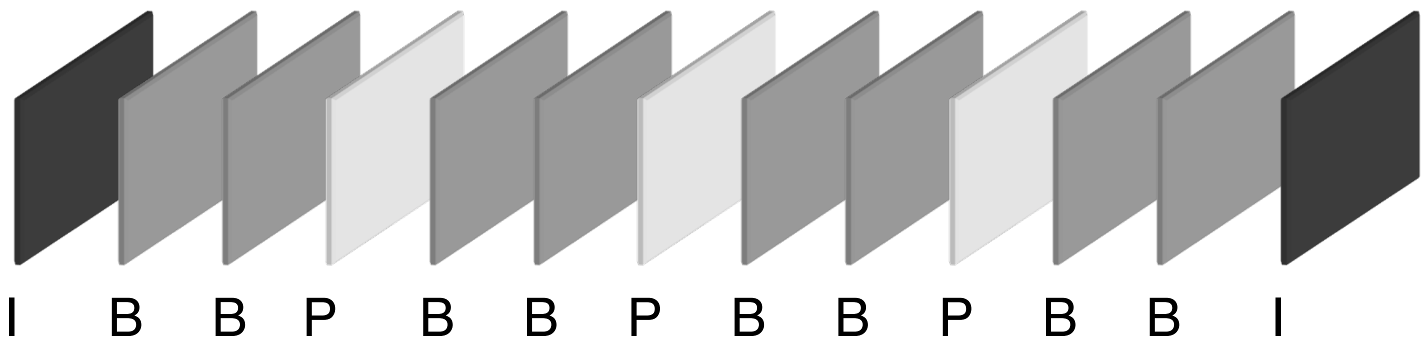
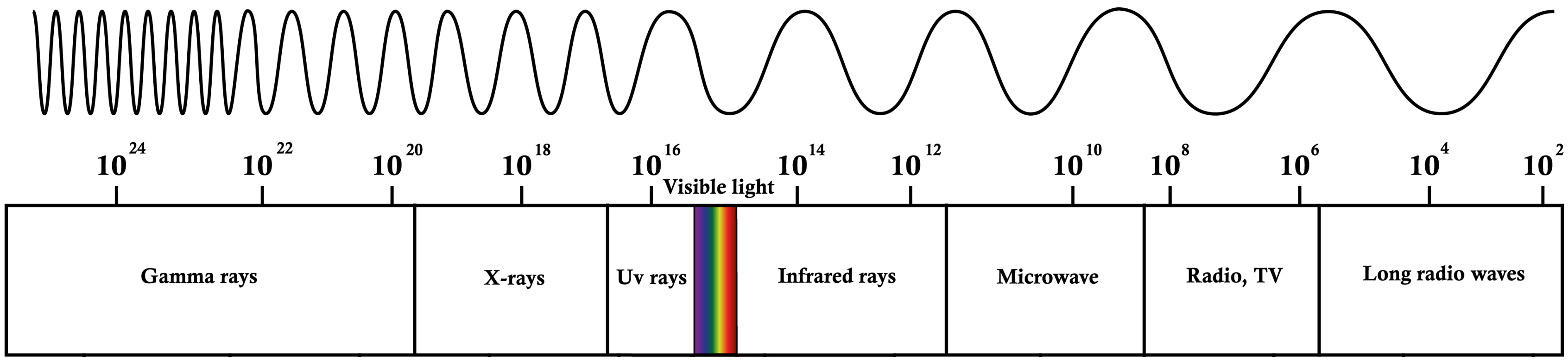
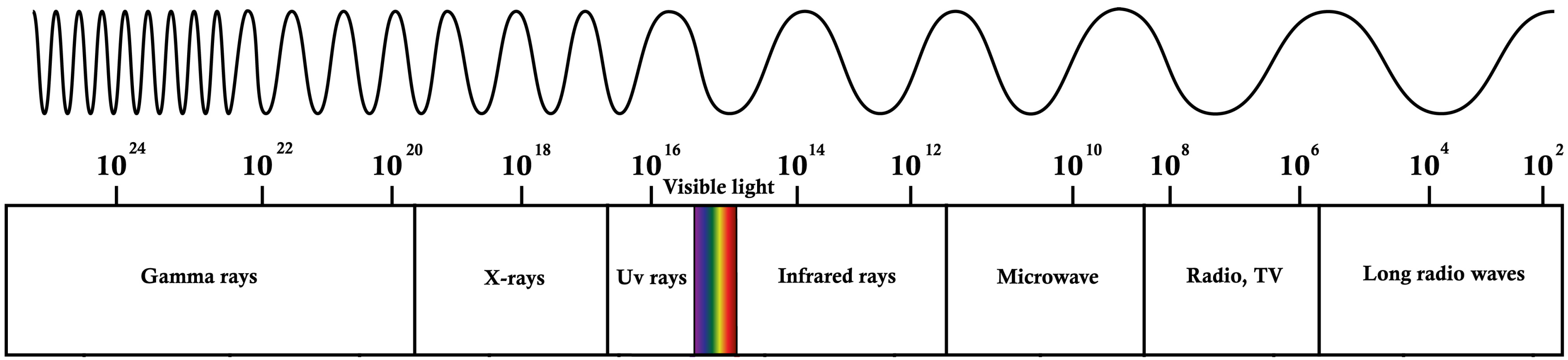


Image acquisition is the process of capturing EM energy radiated/passed by objects!



- Gamma-Ray Imaging
- X-Ray Imaging
- Infrared Imaging
- Visible Spectrum Imaging

Image source: https://s-media-cache-ak0.pinimg.com/originals/9a/19/9e/9a199eb6c2b084694f3f0a36b71b9269.jpg



Visible light is an electromagnetic wave in the 400 nm - 700 nm range!

Image source: https://s-media-cache-ak0.pinimg.com/originals/9a/19/9e/9a199eb6c2b084694f3f0a36b71b9269.jpg

Image Sensing

- Sensors are used to measure reflected energy
- Light is converted to voltage
- ADC is used to convert analog to digital voltage

Single sensor?

Move the sensor over the surface and measure the reflected energy!

Sensor Strip

Move the strip over the surface!

Example?

Sensor Array

- A array of sensors captures the light
- No mechanical movement
- CMOS and CCD are popular sensors

Sampling and Quantization

Digitizing the coordinate values is called sampling!

Digitizing the amplitude values is called quantization!

Image Terminology

- Pixels -- picture elements in digital images
- Image Resolution -- number of pixels in a digital image
  - width x height (e.g., 640X480)
- Bits/pixel – also contributes to the quality of the image

Image Representation

24-bit color Image

Size=786 kb

512*512 pixels

8-bit Gray Image

Size=263 kb

512*512 pixels

1-bit BW Image

Size=33 kb

512*512 pixels

Which image is bigger size?

Size=786 kb

Size=263 kb

8-bit color-map Image

Size=263 kb

512*512 pixels

Color Look-up Tables

- The idea is to store only the code value for each pixel.
- If a pixel stores the value 25, the meaning is to go to row 25 in a color look-up table (LUT).

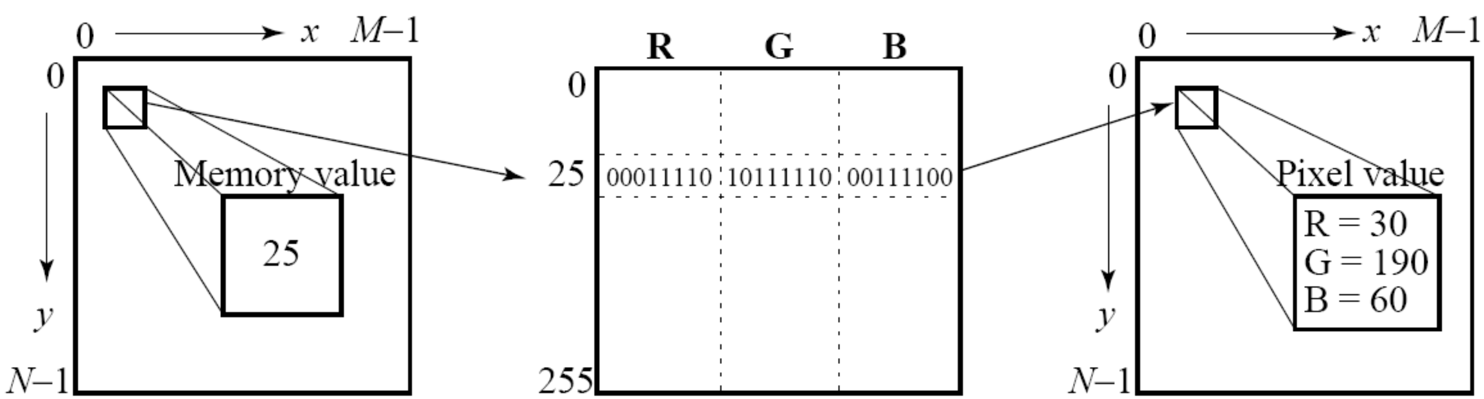


Image File Formats

- Some formats are restricted to particular hardware / operating system platforms.
- There are applications that convert formats from one system to another.
- Most image formats incorporate compression, lossless or lossy.

Popular Formats

- 8-bit GIF: one of the most important formats because of its historical connection to the WWW and HTML markup language as the first image type recognized by net browsers.
- JPEG: currently the most important common file format.
- PNG: most popular lossless image format.
- TIFF: flexible file format due to the addition of tags.
- EXIF: allows the addition of image metadata.
- PS and PDF: vector based language, popular in publishing and academia

Microsoft Formats

- Vectored: WMF
- Non-vectored: BMP

Human Visual System

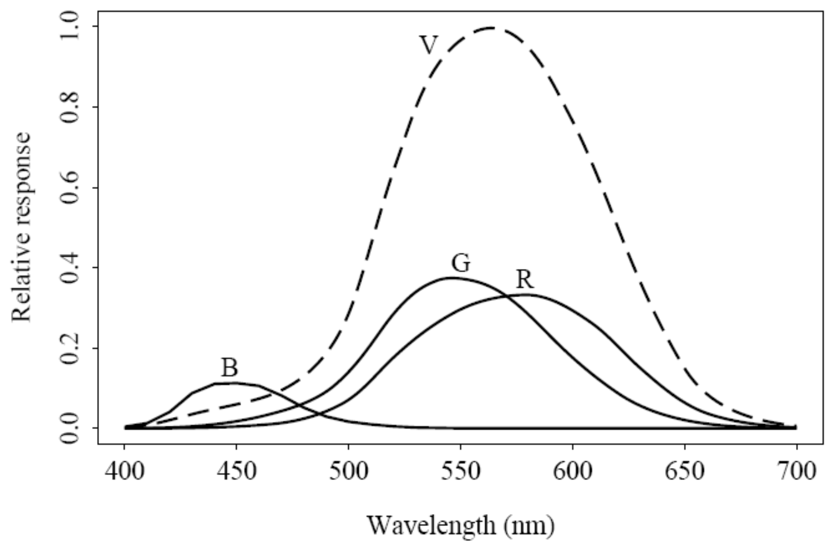
- The eye works like a camera, with the lens focusing an image onto the retina (upside-down and left-right reversed).
- The retina consists of an *array* of *rods* and three kinds of cones.
- The rods come into play when light levels are low and produce a image in shades of gray ("all cats are gray at night!").
- For higher light levels, the cones each produce a signal. Because of their differing pigments, the three kinds of cones are most sensitive to red (*R*), green (*G*), and blue (*B*) light.

Spectral Sensitivity of the Eye

- The eye is most sensitive to light in the middle of the visible spectrum.
- The Blue receptor sensitivity is much smaller than the curves for Red or Green — Blue is a late addition, in evolution.
- Statistically, Blue is the favourite color of humans, regardless of nationality — perhaps for this reason: Blue is a latecomer and thus is a bit surprising!

Spectral Sensitivity of the Eye

- The rod sensitivity curve looks like the luminous-efficiency function V(λ) but is shifted to the red end of the spectrum.



R,G, and B cones, and Luminous Efficiency curve V(λ).

Color Models

- RGB
- YCbCr
- YUV
- YIQ
- CMY
- HSV

CMY - Cyan, Magenta, Yellow

$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

RGB to CMY

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} C \\ M \\ Y \end{bmatrix}$$

CMY to RGB

RGB to YUV

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.299 & -0.587 & 0.886 \\ 0.701 & -0.587 & -0.114 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

RGB to YIQ

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.595879 & -0.274133 & -0.321746 \\ 0.211205 & -0.523083 & 0.311878 \end{bmatrix} = \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

RGB to YCbCr

$$\begin{bmatrix} Y \\ C_b \\ C_r \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.168736 & -0.331264 & 0.5 \\ 0.5 & -0.418688 & -0.081312 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} + \begin{bmatrix} 0 \\ 0.5 \\ 0.5 \end{bmatrix}$$

JPEG Image Compression

The JPEG Standard

- JPEG is an image compression standard that was developed by the "Joint Photographic Experts Group". JPEG was formally accepted as an international standard in 1992.

- JPEG is a lossy image compression method. It employs a transform coding method using the DCT (*Discrete Cosine Transform*).

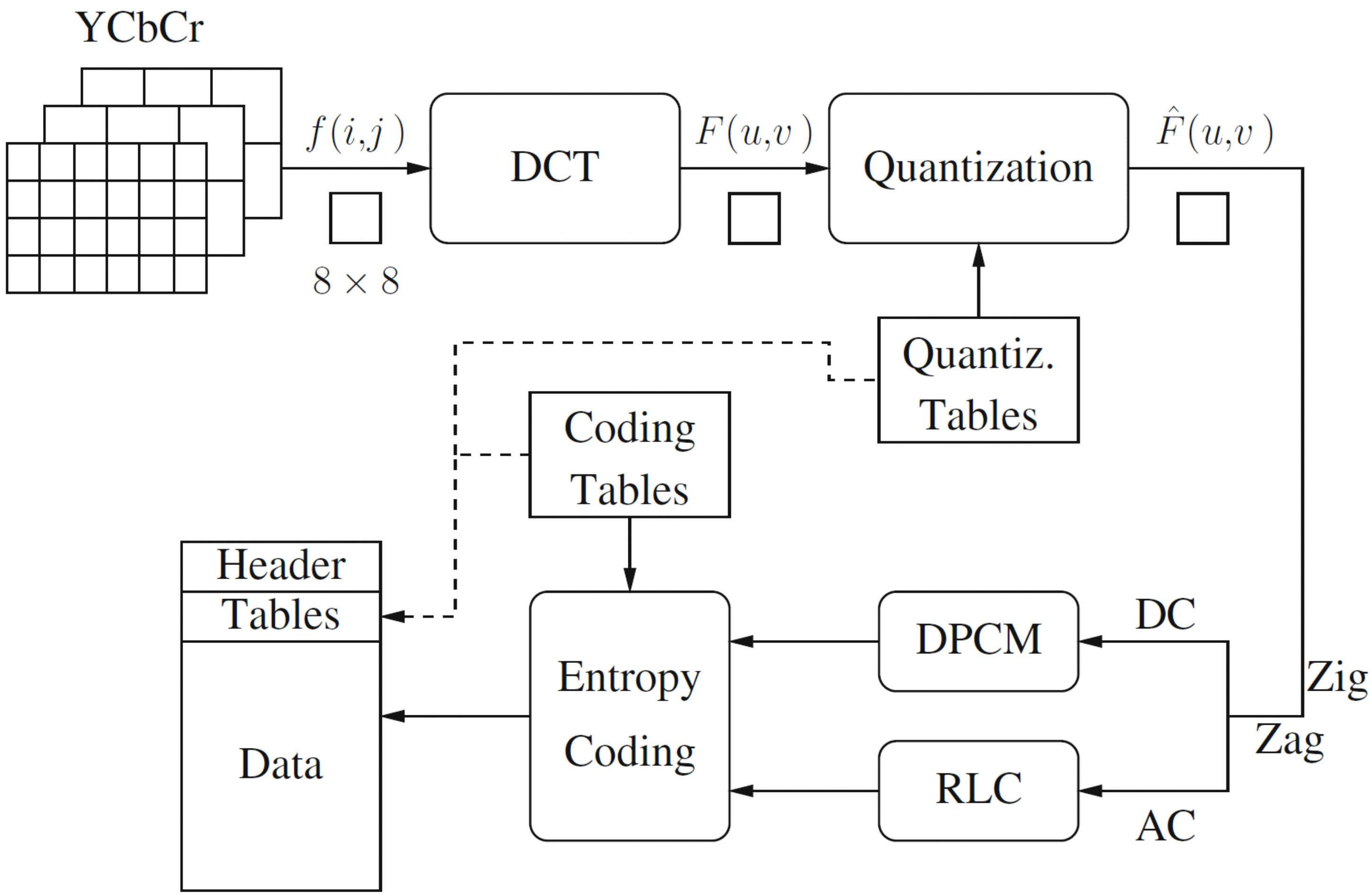
- An image is a function of i and j (or conventionally x and y) in the *spatial domain*. The 2D DCT is used as one step in JPEG in order to yield a frequency response which is a function $F(u, v)$ in the *spatial frequency domain*, indexed by two integers u and v.

The effectiveness of the DCT transform coding method in JPEG relies on 3 major observations!

Observation 1: Useful image contents change relatively slowly across the image.

Observation 2: Humans are much less likely to notice the loss of very high spatial frequency components than the loss of lower frequency components.

Observation 3: Visual acuity (accuracy in distinguishing closely spaced lines) is much greater for gray ("black and white") than for color.



Block diagram for JPEG encoder

1. Much of the information in an image is repeated, hence “spatial redundancy”.
2. The spatial redundancy can be reduced by largely reducing the high spatial frequency contents.
3. Chroma subsampling (4:2:0) is used in JPEG.

Main Steps in JPEG Image Compression

- Transform RGB to YCbCr and subsample color.
- DCT on image blocks.
- Quantization.
- Zig-zag ordering and run-length encoding.
- Entropy coding.

DCT on image blocks

- Each image is divided into 8 × 8 blocks. The 2D DCT is applied to each block image $f(i, j)$, with output being the DCT coefficients $F(u, v)$ for each block.
- Using blocks, however, has the effect of isolating each block from its neighboring context. This is why JPEG images look choppy ("blocky") when a high compression ratio is specified by the user.

Quantization

$$\hat{F}(u,v) = round\left(\frac{F(u,v)}{Q(u,v)}\right)$$

- ***F(u, v)*** represents a DCT coefficient, ***Q(u, v)*** is a "quantization matrix" entry, and $\hat{F}(u,v)$ represents the *quantized DCT coefficients* which JPEG will use in the succeeding entropy coding.
- The quantization step is the main source for loss in JPEG compression.
- The entries of ***Q(u, v)*** tend to have larger values towards the lower right corner. This aims to introduce more loss at the higher spatial frequencies — a practice supported by Observations 1 and 2.

Quantization Tables

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

Luminance

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

Chrominance

An 8 × 8 block from the Y image of 'Lena'

200	202	189	188	189	175	175	175
200	203	198	188	189	182	178	175
203	200	200	195	200	187	185	175
200	200	200	200	197	187	187	187
200	205	200	200	195	188	187	175
200	200	200	200	200	190	187	175
205	200	199	200	191	187	187	175
210	200	200	200	188	185	187	186

$f(i, j)$

515	65	-12	4	1	2	-8	5
-16	3	2	0	0	-11	-2	3
-12	6	11	-1	3	0	1	-2
-8	3	-4	2	-2	-3	-5	-2
0	-2	7	-5	4	0	-1	-4
0	-3	-1	0	4	1	-1	0
3	-2	-3	3	3	-1	-1	3
-2	5	-2	4	-2	2	-3	0

$F(u, v)$

JPEG compression for a smooth image block.

$$\begin{matrix} 32 & 6 & -1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{matrix}$$

$\hat{F}(u, v)$

$$\begin{matrix} 512 & 66 & -10 & 0 & 0 & 0 & 0 & 0 \\ -12 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -14 & 0 & 16 & 0 & 0 & 0 & 0 & 0 \\ -14 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{matrix}$$

$\tilde{F}(u, v)$

$$\begin{matrix} 199 & 196 & 191 & 186 & 182 & 178 & 177 & 176 \\ 201 & 199 & 196 & 192 & 188 & 183 & 180 & 178 \\ 203 & 203 & 202 & 200 & 195 & 189 & 183 & 180 \\ 202 & 203 & 204 & 203 & 198 & 191 & 183 & 179 \\ 200 & 201 & 202 & 201 & 196 & 189 & 182 & 177 \\ 200 & 200 & 199 & 197 & 192 & 186 & 181 & 177 \\ 204 & 202 & 199 & 195 & 190 & 186 & 183 & 181 \\ 207 & 204 & 200 & 194 & 190 & 187 & 185 & 184 \end{matrix}$$

$\tilde{f}(i, j)$

$$\begin{matrix} 1 & 6 & -2 & 2 & 7 & -3 & -2 & -1 \\ -1 & 4 & 2 & -4 & 1 & -1 & -2 & -3 \\ 0 & -3 & -2 & -5 & 5 & -2 & 2 & -5 \\ -2 & -3 & -4 & -3 & -1 & -4 & 4 & 8 \\ 0 & 4 & -2 & -1 & -1 & -1 & 5 & -2 \\ 0 & 0 & 1 & 3 & 8 & 4 & 6 & -2 \\ 1 & -2 & 0 & 5 & 1 & 1 & 4 & -6 \\ 3 & -4 & 0 & 6 & -2 & -2 & 2 & 2 \end{matrix}$$

$(i, j) = f(i, j) - \tilde{f}(i, j)$

JPEG compression for a smooth image block

Another 8 × 8 block from the Y image of 'Lena'

70	70	100	70	87	87	150	187
85	100	96	79	87	154	87	113
100	85	116	79	70	87	86	196
136	69	87	200	79	71	117	96
161	70	87	200	103	71	96	113
161	123	147	133	113	113	85	161
146	147	175	100	103	103	163	187
156	146	189	70	113	161	163	197

$f(i, j)$

-80	-40	89	-73	44	32	53	-3
-135	-59	-26	6	14	-3	-13	-28
47	-76	66	-3	-108	-78	33	59
-2	10	-18	0	33	11	-21	1
-1	-9	-22	8	32	65	-36	-1
5	-20	28	-46	3	24	-30	24
6	-20	37	-28	12	-35	33	17
-5	-23	33	-30	17	-5	-4	20

$F(u, v)$

JPEG compression for a smooth image block.

$$\begin{array}{rrrrrrrr}
-5 & -4 & 9 & -5 & 2 & 1 & 1 & 0 \\
-11 & -5 & -2 & 0 & 1 & 0 & 0 & -1 \\
3 & -6 & 4 & 0 & -3 & -1 & 0 & 1 \\
0 & 1 & -1 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 \\
0 & -1 & 1 & -1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}$$

$\hat{F}(u, v)$

$$\begin{array}{rrrrrrrr}
-80 & -44 & 90 & -80 & 48 & 40 & 51 & 0 \\
-132 & -60 & -28 & 0 & 26 & 0 & 0 & -55 \\
42 & -78 & 64 & 0 & -120 & -57 & 0 & 56 \\
0 & 17 & -22 & 0 & 51 & 0 & 0 & 0 \\
0 & 0 & -37 & 0 & 0 & 109 & 0 & 0 \\
0 & -35 & 55 & -64 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}$$

$\tilde{F}(u, v)$

$$\begin{array}{rrrrrrrr}
70 & 60 & 106 & 94 & 62 & 103 & 146 & 176 \\
85 & 101 & 85 & 75 & 102 & 127 & 93 & 144 \\
98 & 99 & 92 & 102 & 74 & 98 & 89 & 167 \\
132 & 53 & 111 & 180 & 55 & 70 & 106 & 145 \\
173 & 57 & 114 & 207 & 111 & 89 & 84 & 90 \\
164 & 123 & 131 & 135 & 133 & 92 & 85 & 162 \\
141 & 159 & 169 & 73 & 106 & 101 & 149 & 224 \\
150 & 141 & 195 & 79 & 107 & 147 & 210 & 153
\end{array}$$

$\tilde{f}(i, j)$

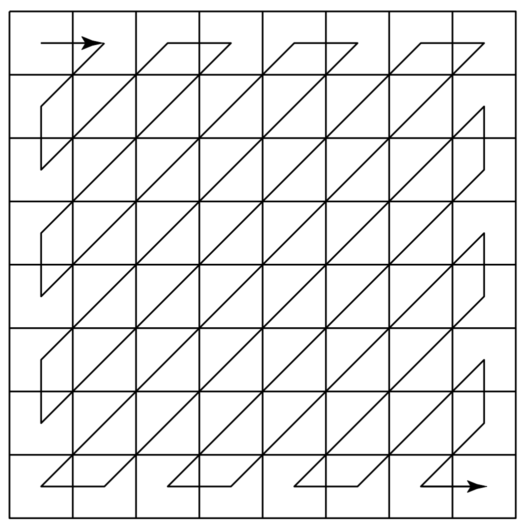
$$\begin{array}{rrrrrrrr}
0 & 10 & -6 & -24 & 25 & -16 & 4 & 11 \\
0 & -1 & 11 & 4 & -15 & 27 & -6 & -31 \\
2 & -14 & 24 & -23 & -4 & -11 & -3 & 29 \\
4 & 16 & -24 & 20 & 24 & 1 & 11 & -49 \\
-12 & 13 & -27 & -7 & -8 & -18 & 12 & 23 \\
-3 & 0 & 16 & -2 & -20 & 21 & 0 & -1 \\
5 & -12 & 6 & 27 & -3 & -2 & 14 & -37 \\
6 & 5 & -6 & -9 & 6 & 14 & -47 & 44
\end{array}$$

$(i, j) = f(i, j) - \tilde{f}(i, j)$

JPEG compression for a textured image block.

Run-length Coding (RLC) on AC coefficients

- RLC aims to turn the $\hat{F}(u,v)$ values into sets {*#-zeros-to-skip*, *next non-zero value*}.
- To make it most likely to hit a long run of zeros: a *zig-zag scan* is used to turn the 8×8 matrix $\hat{F}(u,v)$ into a *64-vector*.



Zigzag scan in JPEG.

DPCM on DC coefficients

- The DC coefficients are coded separately from the AC ones. *Differential Pulse Code modulation* (*DPCM*) is the coding method.

- If the DC coefficients for the first 5 image blocks are 150, 155, 149, 152, 144, then the DPCM would produce 150, 5, -6, 3, -8, assuming $d_i = DC_{i+1} - DC_i$, and $d_0 = DC_0$.

Entropy Coding

- The DC and AC coefficients finally undergo an entropy coding step to gain a possible further compression.
- Use DC as an example: each DPCM coded DC coefficient is represented by (SIZE, AMPLITUDE), where SIZE indicates how many bits are needed for representing the coefficient, and AMPLITUDE contains the actual bits.
- In the example we're using, codes 150, 5, −6, 3, −8 will be turned into
    - (8, 10010110), (3, 101), (3, 001), (2, 11), (4, 0111) .
- SIZE is Huffman coded since smaller SIZEs occur much more often. AMPLITUDE is not Huffman coded, its value can change widely so Huffman coding has no appreciable benefit.