7-9 Monday – 309-GD2

Xử lý ảnh
INT3404 1

Giảng viên: TS. Nguyễn Thị Ngọc Diệp

Email: ngocdiep@vnu.edu.vn

Slide & code: https://github.com/chupibk/INT3404_1

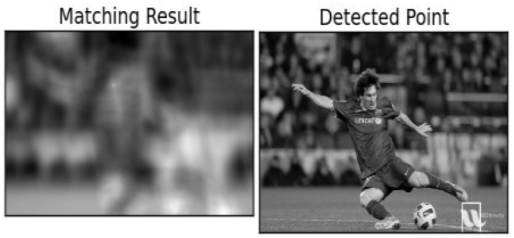
Final project - Registration form

- https://forms.gle/Qpp1hbX9QPG875gT8
- Speadsheet to edit:
  - https://docs.google.com/spreadsheets/d/1c13rgmfNdIpJEPD544qDY2sHxH-ZOxzobeW8fI1CI_w/edit#gid=82493842
  - → Edit with care!
- Final project schedule: week 13, 14, 15

Final project schedule

Group name	Problem	Schedule
HCA_image	trích xuất data trong bảng từ ảnh scan (excel) để trả về file excel	Week 14
Naive solver	Sudoku solver, mục tiêu: đọc được số từ ảnh chụp điện thoại	Week 14
gnissecorpegami	Đề tài chọn: Sudoku solver Mục tiêu: Viết một chương trình nhận ảnh chụp một câu đố Sudoku 9x9 và trả lại bức ảnh sau khi đã điền hết các ô trống còn lại	Week 14
SODOKU	Giải trò chơi Sodoku bằng phương pháp xử lý ảnh và học máy. Đầu vào sẽ chứ những hình ảnh chứa những câu đố Sodoku, đầu ra là ảnh Sodoku đã được giải.	Week 15
Resnet	Sudoku Solver - Đây là bài toán giải trò chơi Sudoku, với input là 1 image câu đố sudoku, sau khi cho qua mô hình của chúng em, output đầu ra sẽ là 1 image câu đố đã được giải. ^^	Week 15
Tứ kỵ sĩ	Sudoku solver	Week 13
UET_IP	Xây dựng 1 project phân đoạn đường cao tốc, nhận dạng làn đường, nhận dạng biển báo. Đầu vào là 1 video đường cao tốc, đầu ra là video đã được highlight đường, làn đường, biển báo.	Week 13
TicTacToe	Tic-tac-toe, mục tiêu: có thể đưa ra ảnh output xác định trạng thái trò chơi hiện tại (đang chơi, X thắng, O thắng)	Week 14
Gonzalez's Descendants	Mô hình cho phép nhận diện các khuôn mặt có trong bức ảnh cho trước	Week 13
allenGroup	recover 3D scene from single image!	Week 15
Naive Solver	Giải sudoku từ ảnh chụp được	Week 14
N-A-M-E-K	Đầu vào là ảnh game X-O, đầu ra là gợi ý cho nước đi tiếp theo.	Week 15
Team TTTN	Selected problem: Chess Board Reader. Goal: Correctly detect and identify a chessboard and the configuration of its pieces.	Week 14, Week 15
Untitled	Giải rubic 3x3x3 → Face morphing	Week 13
15gg	sudoku solver, giải bài toán sudoku qua hình ảnh	Week 15
Image Killer	Đề tài: Tự động chấm điểm bài thi trắc nghiệm. Mục tiêu: dùng các kỹ thuật xử lý ảnh đã học để có thể tìm được số câu trả lời đúng trong mẫu phiếu trả lời trắc nghiệm được dùng trong kỳ thi THPTQG.	Week 13
17020646 →17	lottery ticket recognition → nhận vào đồng tiền Việt Nam và đưa ra mệnh giá	Week 15

Recall week 11: Template matching

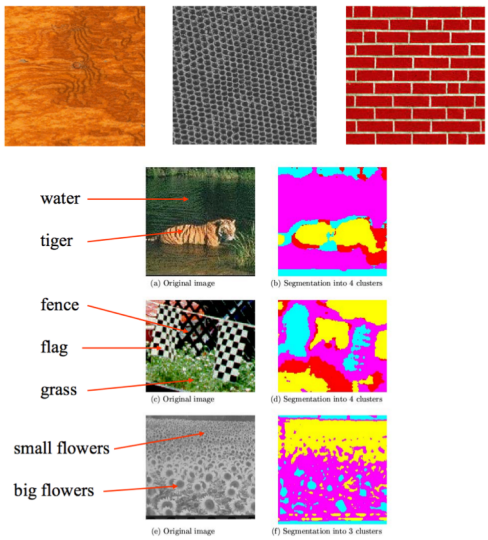


Without normalization

How do we do normalization?

With normalization

Recall week 11: Texture analysis



- Structural approach
- Statistical approach
- Fourier approach

Example of using texture energy features for segmentation

Homework 4: Making & Solving puzzle

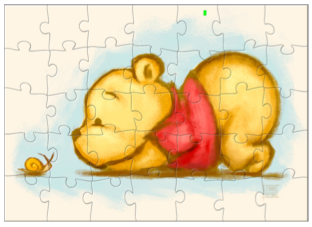
1. Find an image
2. Crop it into at least 6x6 pieces
3. Try to put the pieces to the whole image again

Note: It's okay to use the original image as a guide

Homework 4: Challenge mode

- For those who wants some extra points on final exam
- Note: all code in Python
- Maximum extra points: +5
- Implement the following (don't forget to comment the code):
  1. Create puzzle cut path (+1)
     Pieces must include "holes" (i.e., interlocking)
     Reference: https://blog.wolfram.com/2012/06/28/designing-jigsaw-puzzles-with-mathematica/
  2. Do cutting and display (+1)
  3. Solve the puzzle
     1. With original image guide (+1)
     2. Without original image (+2)
     3. With piece rotated (+1)

HW4 – challenge mode example



1. Cut path

2. Cut & display

3.2. Rotated pieces

Submission link

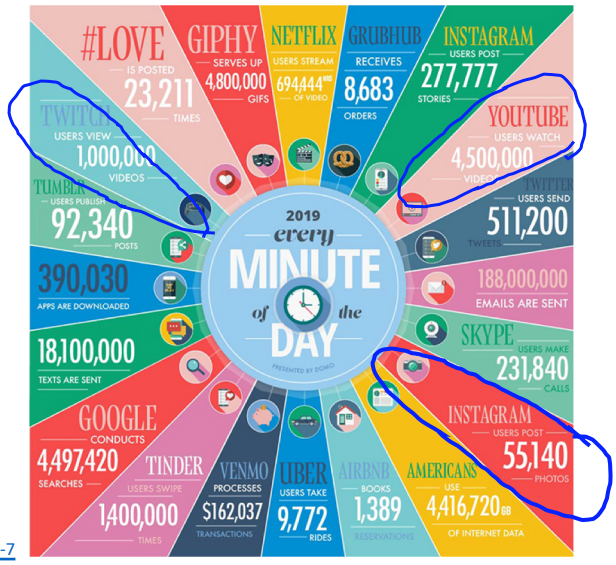
- https://forms.gle/s6xzWtYPoPWXD9698
- Deadline: 2 weeks
  - Nov 24, 2019 23:59
- Extended: Dec 1, 2019 23:59

Lịch trình

Tuần	Nội dung	Yêu cầu đối với sinh viên
1	Giới thiệu môn học Làm quen với OpenCV + Python	Cài đặt môi trường: Python 3, OpenCV 3, Numpy, Jupyter Notebook
2	Phép toán điểm (Point operations) – Điều chỉnh độ tương phản – Ghép ảnh	Làm bài tập 1: điều chỉnh gamma tìm contrast hợp lý
3	Histogram - Histogram equalization - Phân loại ảnh dùng so sánh histogram	Thực hành ở nhà
4	Phép lọc trong không gian điểm ảnh (linear processing filtering) - làm mịn, làm sắc ảnh	Thực hành ở nhà Tìm hiểu thêm các phép lọc
5	Tìm cạnh (edge detection)	Thực hành ở nhà
6	Các phép toán hình thái (Erosion, Dilation, Opening, Closing) - tìm biển số	Làm bài tập 2: tìm barcode
7	Chuyển đổi không gian - miền tần số (Fourier) - Hough transform	Thực hành ở nhà
8	Phân vùng (segmentation) - depth estimation - threshold-based - watershed/grabcut	Đăng ký thực hiện bài tập lớn
9	Mô hình màu Chuyển đổi giữa các mô hình màu	Làm bài tập 3: Chuyển đổi mô hình màu và thực hiện phân vùng
10	Mô hình nhiễu -Giảm nhiễu -Khôi phục ảnh -Giảm nhiễu chu kỳ - Ước lượng hàm Degration -Hàm lọc ngược, hàm lọc Wiener	Thực hành ở nhà
11	Template matching – Image Matching	Làm bài tập 4: puzzle
12	Nén ảnh	Thực hành ở nhà
13	Hướng dẫn thực hiện đồ án môn học	Trình bày đồ án môn học
14	Hướng dẫn thực hiện đồ án môn học	Trình bày đồ án môn học
15	Tổng kết cuối kỳ	Ôn tập

Why image compression?

Data amount generated every minute



Efficient Store, Transmit and Display

Source: https://www.domo.com/learn/data-never-sleeps-7

Image compression goal

- Store image data as efficiently as possible
- Ideally, want to
  - Maximize image quality
  - Minimize storage space and processing resources
- Can't have best of both worlds
- What are some good compromises

Storage space of pictures

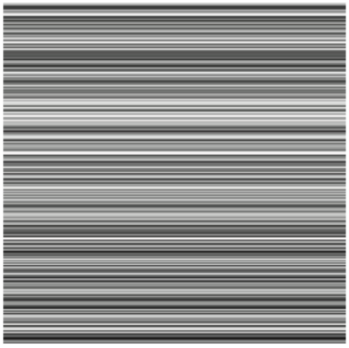
- A 1000x1000 picture with 24 bits per pixel takes up 3 megabytes.
- The Encyclopaedia Brittanica scannned at 300 pixels per inch and 1 bit per pixel requires 25,000 pages × 1,000,000 bytes per page = 25 gigabytes.
- Video is even bulkier: 90 minute movie at 640 ×480 resolution spatially, 24 bit per pixel, 24 frames per second, requires 90 ×60 ×24 ×640 ×480 ×3=120 gigabytes.
- Applications: HDTV, film, remote sensing and satellite image transmission, network communication, image storage, medical image processing, fax.

Why is it possible to compress images?

- Data != information/knowledge
- Data >> information
- Key idea in compression: only keep the info
- But why is data != info?
  - Answer: redundancy
- Statistical redundancy
  - Spatial redundancy and coding redundancy
- Psychovisual redundancy
  - Greyscale redundancy and frequency redundancy

Spatial redundancy

- Pixel values are not spatially independent
- High correlation among neighbor pixels



Coding redundancy

- Redundancy when mapping from the pixels (symbols) to the final compressed binary code (information theory)
- Example

r_k	$p_r(r_k)$	Code 1	$l_1(r_k)$	Code 2	$l_2(r_k)$
$r_{87} = 87$	0.25	01010111	8	01	2
$r_{128} = 128$	0.47	01010111	8	1	1
$r_{186} = 186$	0.25	01010111	8	000	3
$r_{255} = 255$	0.03	01010111	8	001	3
r_k for $k = 87, 128, 186, 255$	0	—	8	—	0

$$L_{\text{avg}} = 0.25(2) + 0.47(1) + 0.03(3) = 1.81 \text{ bits}$$

Redundancy vs Compression ratio

- Data redundancy

$$R = 1 - \frac{1}{C} = 1 - \frac{1}{4.42} = 0.774$$

- Compression ratio

$$C = \frac{b}{b'} = \frac{8}{1.81} \approx 4.42$$

b, b': number of bits (or information carrying units)

Psychovisual redundancy

- Human eye does not respond to all visual information with equal sensitivity
- Some data is simply of less relative importance → can be eliminated without introducing any significant difference to the human eye
- Example:
  - color
  - intensity

Frequency redundancy

- Human eye functions as a lowpass filter
  - High frequencies in an image can be “ignored” without the HVS noticing
  - Key issue in lossy image compression

How Compression works

- Compression is done by transforming and removing image data redundancies
- Mathematically this means transforming data to a statistically uncorrelated set

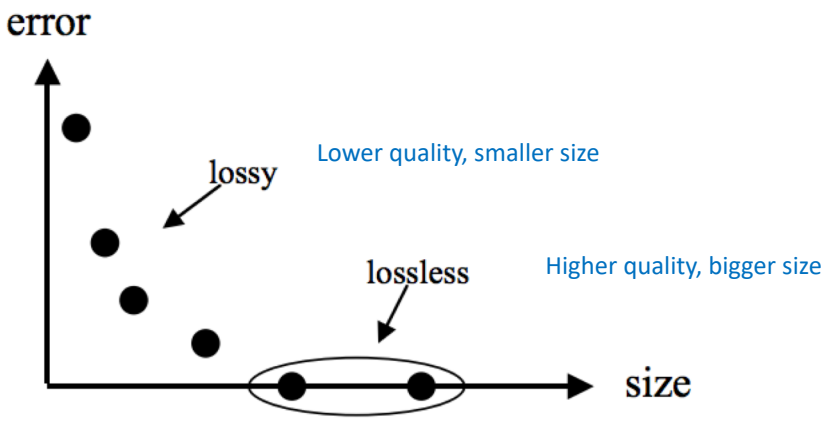
Two major types of Compression algorithms

- Lossless compression
  - The original data is reconstructed perfectly
  - Theoretical limit on maximal compression
  - Practical compression ratios of < 10:1 (still images)
  - Uses entropy coding only (or none at all)
  - Examples: BMP, TIFF, GIF
- Lossy compression
  - Decomposition results in an approximation of the original image
  - Maximal compression rate is a function of reconstruction quality
  - Practical compression ratios of > 10:1 (still images)
  - Uses both quantization and entropy coding
  - Usually involves transform into frequency or other domain
  - Examples: JPEG, JPEG-2000

Details

Overall

Lossy vs lossless



Lower quality, smaller size

Higher quality, bigger size

Compression applications

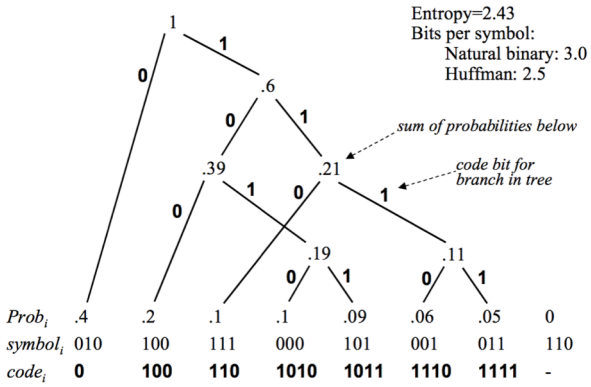
- Transmission compression
  - Transmission time and throughput limits computation time
  - Interactivity demands short response times
  - Bandwidth limitations may require constant compression ratios
- Storage compression
  - Data analysis may afford higher compression ratios

Common lossless compression algorithms

- Huffman
- Run length coding (RLE)
- Variable length coding
- Arithmetic coding
- LZW (Lempel-Ziv-Welch)

Huffman coding example

- These are prefix codes assigned to symbols depending on the probability of the symbols



Run-length coding

- Similar consecutives symbols are grouped and presented by a code
- Start from a specific code, following by a symbol and the number indicating the number of its copies
- It is a entropy coding technique used in H.264, CAVLC
- For example: aaaabbbddaaddccccccca
- Is is compressed to: 4a3b2d2a2d7c1a

BMP (bitmap)

- Use 3 bytes per pixel, one each for R, G, and B
- Can represent up to 2^24 = 16.7 million colors
- No entropy coding
- File size in bytes = 3*width*height, which can be very large
- Can use fewer than 8bits per color, but you need to store the color pallete
- Performs well with ZIP, RAR, etc.

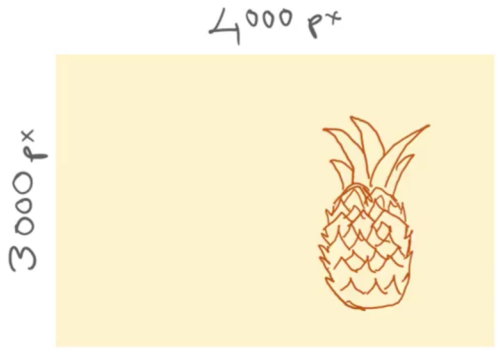
GIF (Graphics interchange format)

- Can use up to 256 colors from 24-bit RGB color space
  - If source image contains more than 256 colors, need to reprocess image to fewer colors
- Suitable for simpler images such as logos and textual graphics, not so much for photographs
- Uses LZW lossless data compression

JPEG (Joint Photographic Experts Group)

- Most dominant image format today
- Typical file size is about 10% of that of BMP (can vary depending on quality settings)
- Unlike GIF, JPEG is suitable for photographs, not so much for logos and textual graphics

Image file size



3000 x 4000 x 3 = 36,000,000

36 MB

IMG_4586.jpg

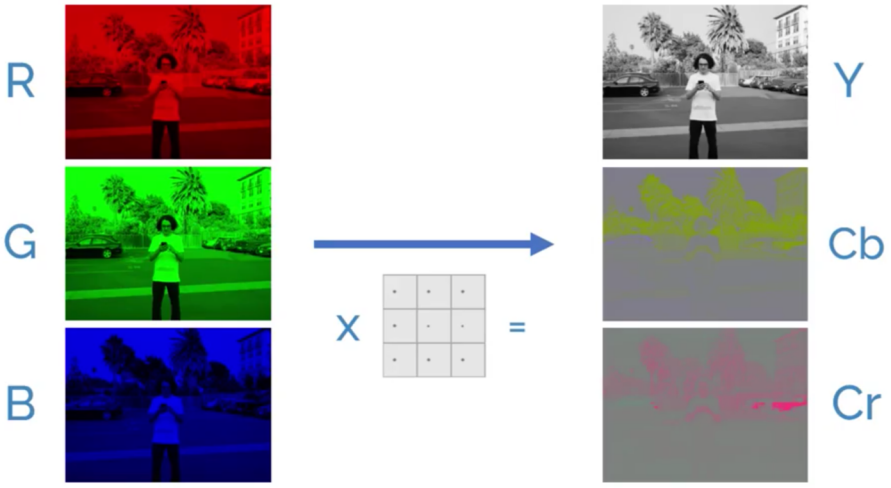
JPEG image - 1.9 MB
Created Friday, May 3, 2019 at 4:06 PM
Modified Friday, May 3, 2019 at 4:06 PM
Last opened Saturday, August 17, 2019 at 10:40 PM
Dimensions 4032 × 3024
Add Tags...

Only 1.9MB!

Lossy compression

The more information you discard, the worse the image quality gets

Color space conversion



Luminance

Chrominance

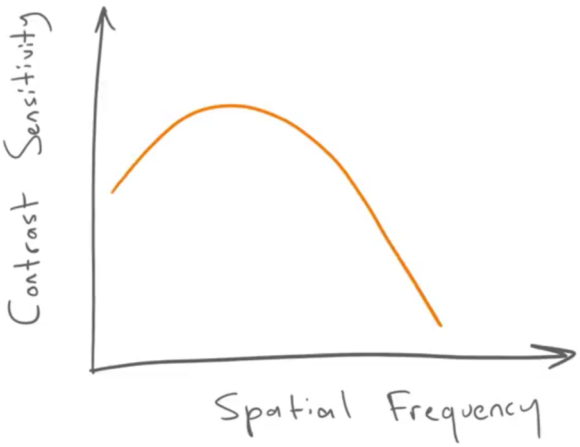
Our visual system is much more sensitive to the changes in brightness than color
→ We can safely downsample the chroma components to save some space

Chroma subsampling

Luminance

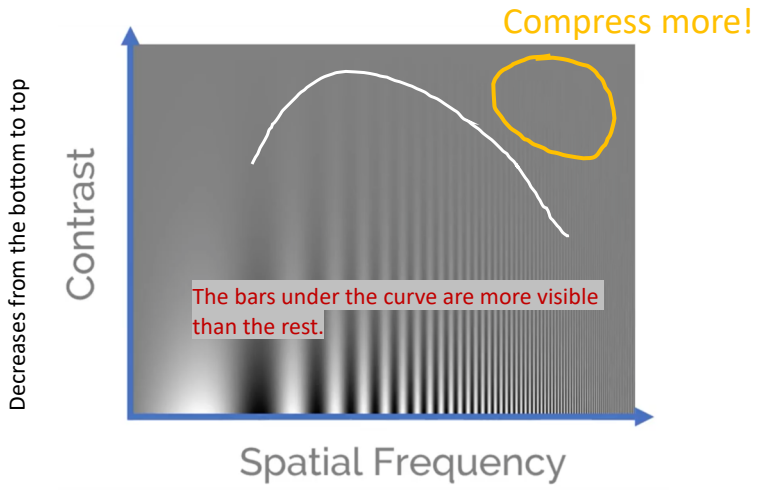
Chrominance

Frequency-dependent contrast sensitivity

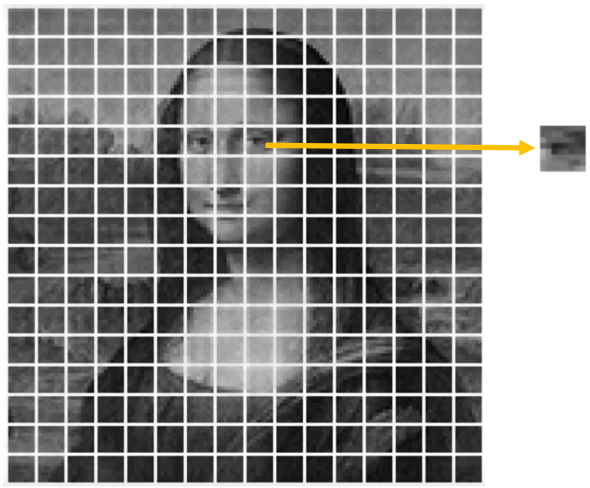


It's easier to miss small objects or fine details in a picture as compared to the large ones

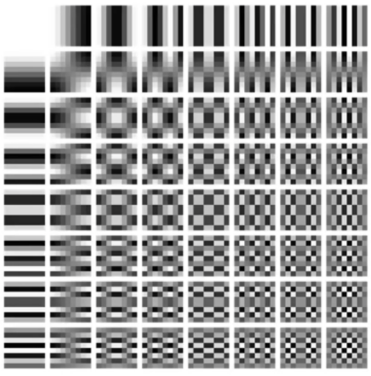
Frequency vs contrast



Increases from left to right



8x8 blocks



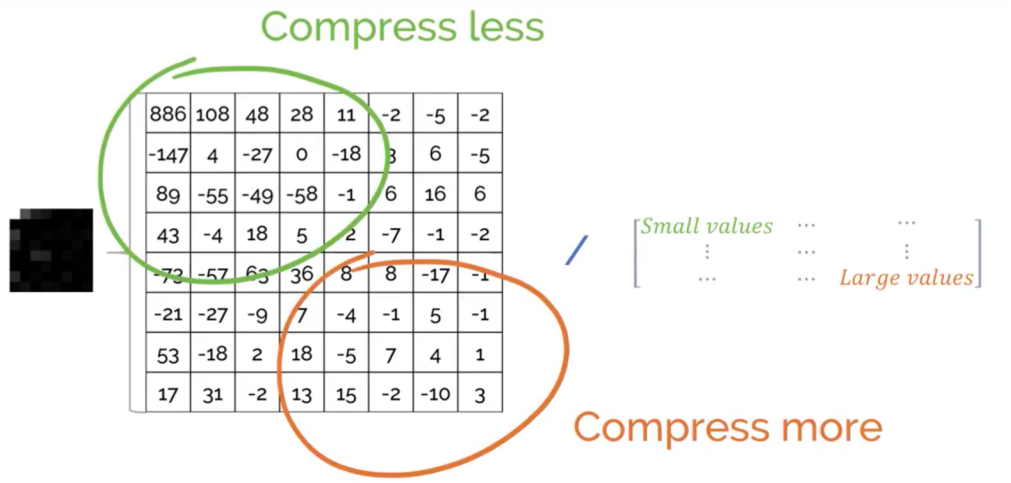
64 frequency patterns

JPEG compression

98	92	94	93	93	107	109	107
102	121	138	118	89	75	59	67
142	127	116	111	105	95	71	52
117	89	39	39	51	72	76	62
189	145	61	68	85	74	89	80
180	167	147	130	130	136	125	126
156	157	158	150	142	116	106	124
153	134	133	136	132	125	149	160

886	108	48	28	11	-2	-5	-2
-147	4	-27	0	-18	3	6	-5
89	-55	-49	-58	-1	6	16	6
43	-4	18	5	-2	-7	-1	-2
-73	-57	63	36	8	8	-17	-1
-21	-27	-9	7	-4	-1	5	-1
53	-18	2	18	-5	7	4	1
17	31	-2	13	15	-2	-10	3

Quantization

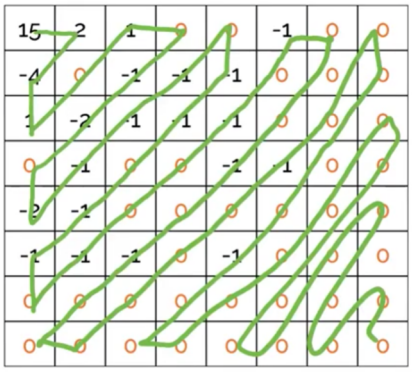


=

15	2	1	0	0	-1	0	0
-4	0	-1	-1	-1	0	0	0
1	-2	-1	-1	-1	0	0	0
0	-1	0	0	-1	-1	0	0
-2	-1	0	0	0	0	0	0
-1	-1	-1	0	-1	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Zig-zag arrangement

=



= [15, 2, -4, 1, 0, 1, …, 0, 0, 0, 0, 0, 0, 0, 0]

Lossless Encoding

= [15, 2, -4, 1, 0, 1, …, 0, 0, 0, 0, 0, 0, 0, 0]

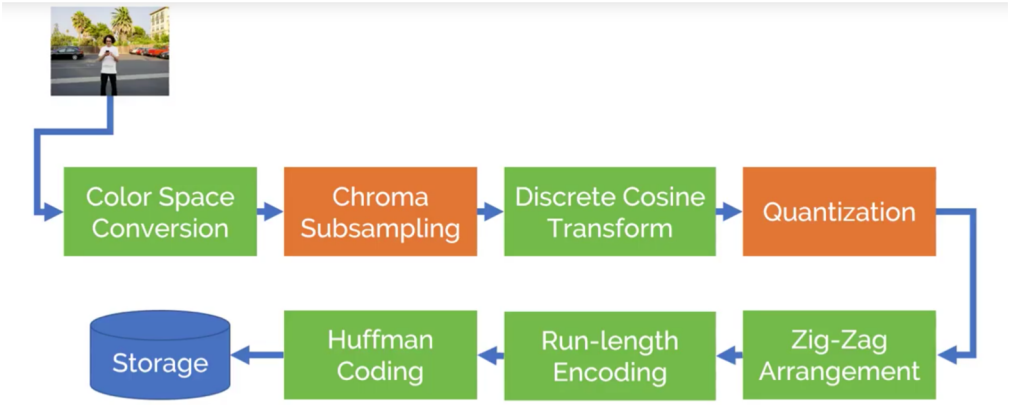
repeat(0, 18)

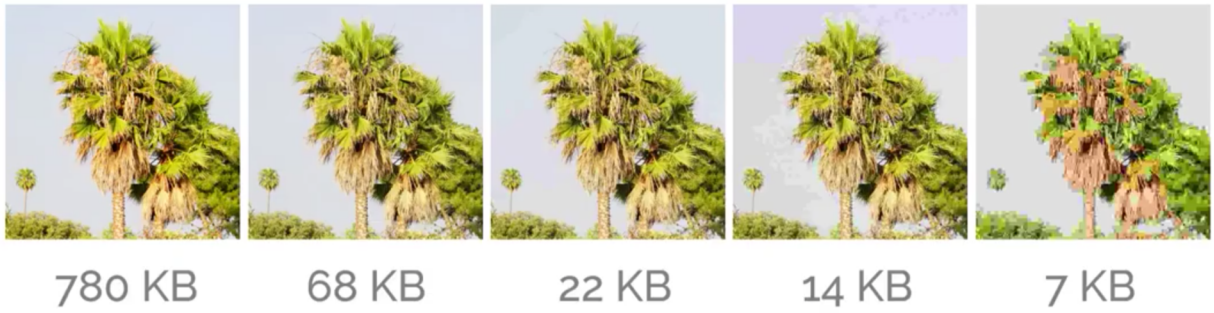
Run-length encoding

Huffman (entropy) encoding

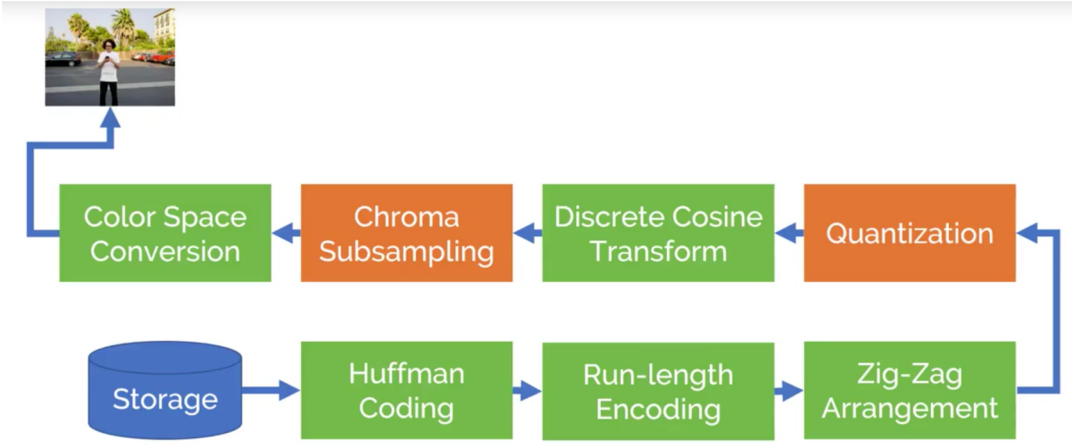
Encode the more frequent values with fewer bits and less frequent values with more bits

JPEG encoding



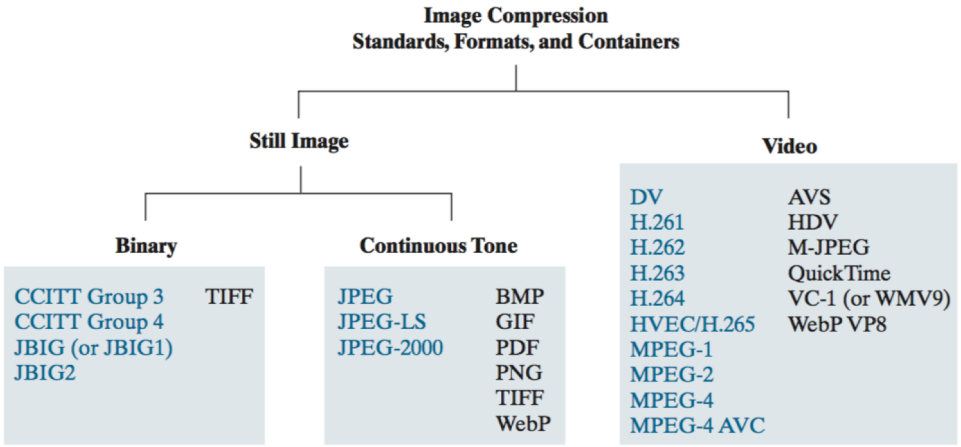


JPEG decoding



Popular compression standards

FIGURE 8.6
Some popular image compression standards, file formats, and containers. Internationally sanctioned entries are shown in blue; all others are in black.



References

- https://www.youtube.com/watch?v=Ba89cI9eIg8
- R. C. Gonzalez, R. E. Woods, “Digital Image Processing,” 4th edition, Pearson, 2018.