A GUIDED TOUR OF AI: FROM FOUNDATIONS TO LATEST APPLICATION

Workshop -Image Processing

Sorbonne University Abu Dhabi (SUAD)

Introduction to Digital Image Processing

Digital Image Processing (DIP) deals with manipulation of digital images through a digital computer. The field of DIP focuses on developing computer system that is able to perform processing of an image. The input of that system is a digital image and the system employs efficient algorithms to output a processed image. The most common example of such a system is Adobe Photoshop.

3d world around us | captured by | a camera | and sent to | computer system to focus on the plant | to produce output as | a processed image

What is a digital image?

A digital image is nothing but a two-dimensional mathematical function ***f(x,y)***, where ***x*** and ***y*** are the **spatial** (plane) coordinates, and the amplitude of ***f*** at any pair of coordinates ***(x,y)*** is called the intensity of the image at that level.

If ***x,y*** and the ***amplitude*** values of ***f*** are ***finite*** and ***discrete quantities***, we call the image a ***digital image***. A digital image is composed of a finite number of elements called ***pixels*** (aka “picture elements”), each of which has a particular location and value.

First Digital Image Before Internet

In 1957, Russell Kirsch converted a photograph of his three-month-old son, Walden into a tiny digital file using an early computer. This was created by scanning an analogue photograph.

What we see in the image?

How the computer system interprets the image?

→

148	123	52	107	123	162	172	123	64	89	⋯
147	130	92	95	98	130	171	155	169	163	⋯
141	118	121	148	117	107	144	137	136	134	⋯
82	106	93	172	149	131	138	114	113	129	⋯
57	101	72	54	109	111	104	135	106	125	⋯
138	135	114	82	121	110	34	76	101	111	⋯
138	102	128	159	168	147	116	129	124	117	⋯
113	89	89	109	106	126	114	150	164	145	⋯
120	121	123	87	85	70	119	64	79	127	⋯
145	141	143	134	111	124	117	113	64	112	⋯
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	

Digital Image and Pixel

Pixel intensity value

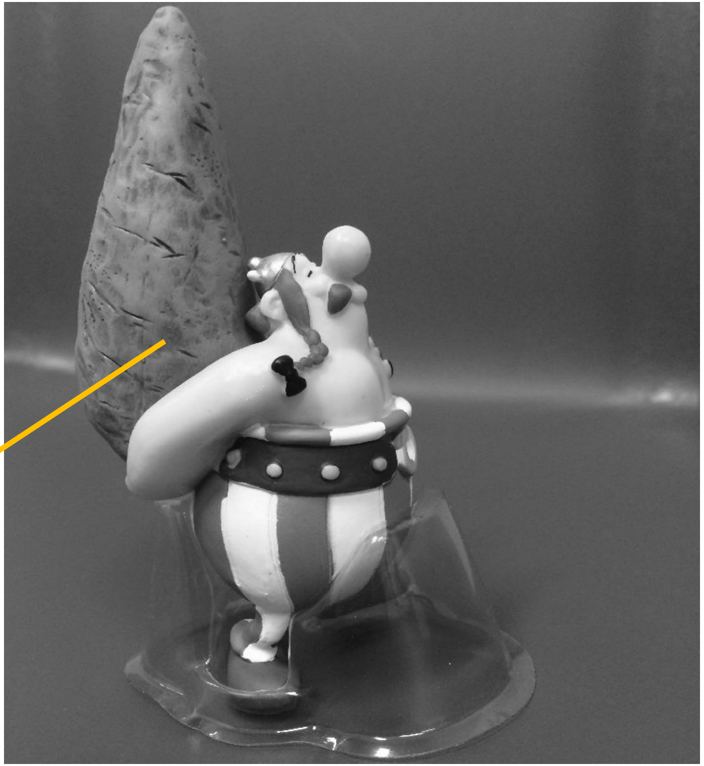
f(1,1) = 103

Pixel location

rows columns

f(645:650,1323:1328) =

83	82	82	82	82	82
82	82	82	81	81	81
82	82	81	81	80	80
82	82	81	80	80	79
80	79	78	77	77	77
80	79	78	78	77	77



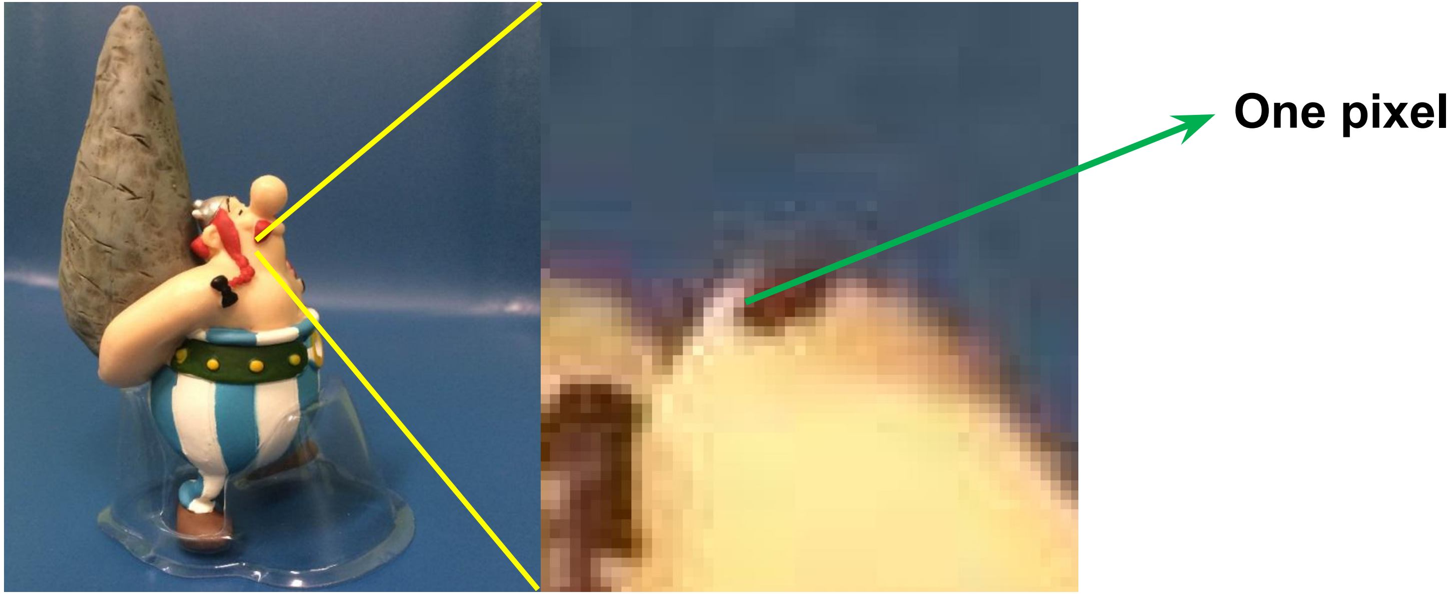
f(2724,2336) = 88

Consider the following image (2724x2336 pixels) to be 2D function or a **matrix** with **rows** and **columns**

In 8-bit representation Pixel intensity values change between 0 (Black) and 255 (White)

Digital image is an approximation of pixel

Remember *digitization* implies that a digital image is an *approximation* of a real scene



COLOR MODELS

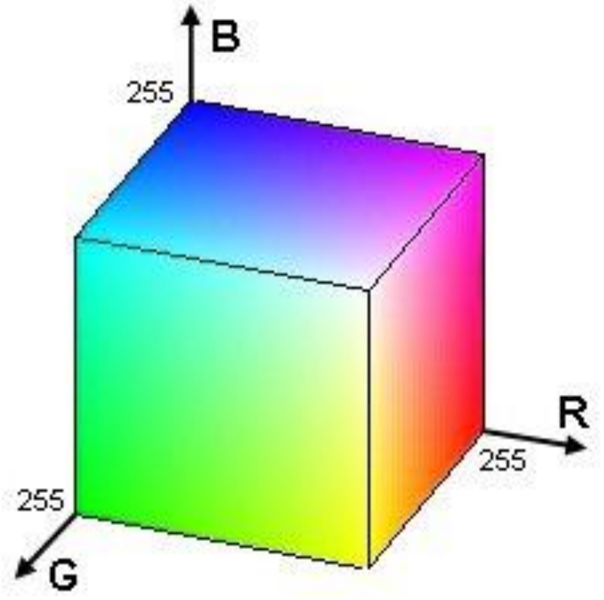
- Gray-level images

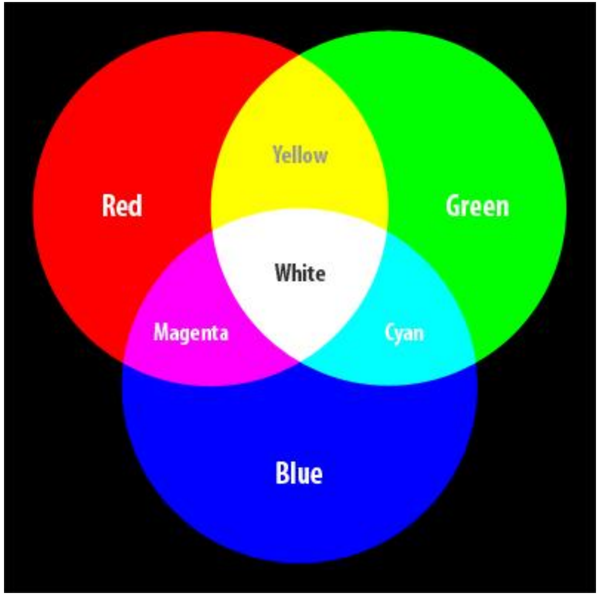
- Color images

COLOR SPACES

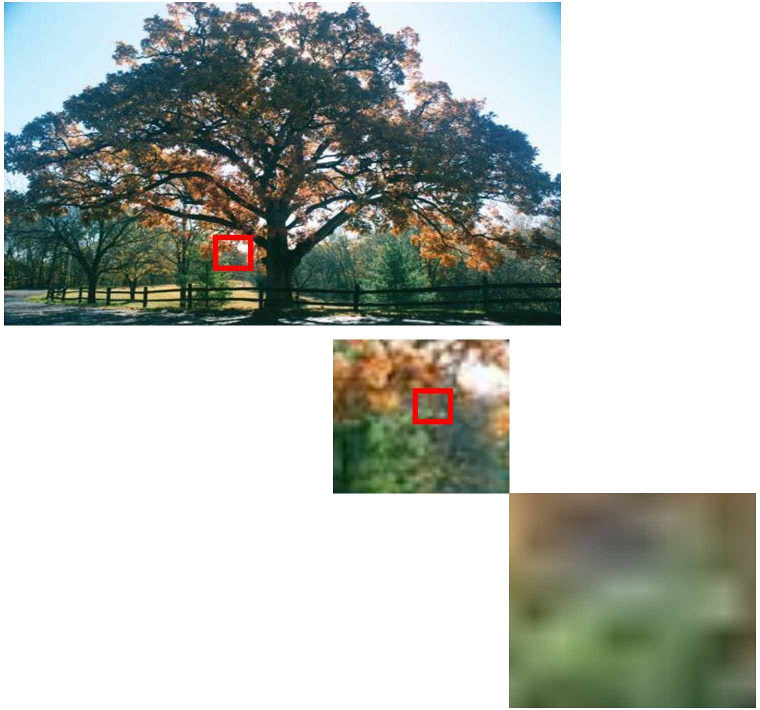
- How about the color components in the image ?
  - Described using color models
- Colors are represented as ordered triplets that define the color space
- Example : RGB color model – most popular
- Express every color as a weighted sum of the three component colors: Red, Blue and Green.

RGB Model

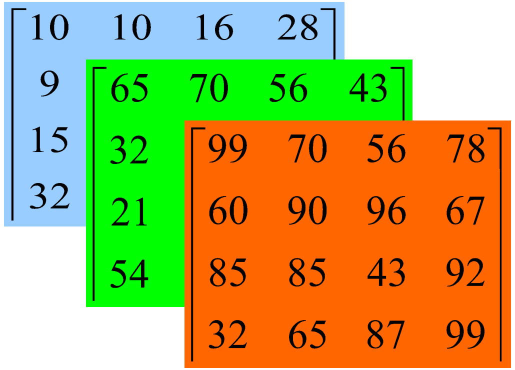




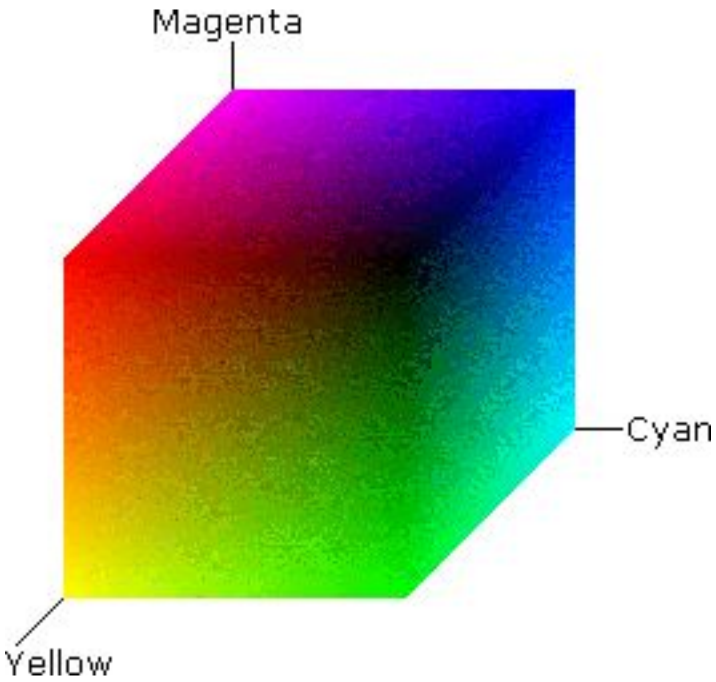
RGB Model



Each pixel contains a vector
representing red, green and
blue components.



CMY Model



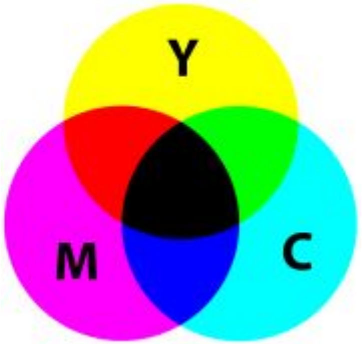


IMAGE FORMATS

BMP (Bitmap)

Uses RGB color model, without compression. Color depth of 24-bit

GIF (Graphics Interchange Format)

Compressed with some basic lossless compression techniques to 20 – 25% of original picture without loss. Supports 24-bit colors.

TIFF (Tagged Image File Format)

Supports grey levels, RGB, and CMY color model. Also supports lots of different compression methods. Additionally contains a descriptive part with properties a display should provide to show the image.

JPEG (JOINT PHOTOGRAPHICS EXPERT GROUP)

Color representation

JPEG applies to color and grey-scaled still images

Image content

Of any complexity, with any statistical characteristics

Properties of JPEG

State-of-the-art regarding compression factor and image quality

Image Attributes

Image Dimension: Number of rows x Number of columns

Note: Image Dimension = Total number of pixels in the image

Image Attributes

Image Resolution: Area covered by each pixel

Note: **Image Resolution is measured in Pixels Per Inch(PPI)**

Greater the PPI, higher the image resolution

Image Attributes

Number of Color Planes: Area covered by each pixel

Red, Green, and Blue

(RGB) Channel

Black and White

Channel

Image Attributes

Bit Depth: Color information stored in each pixel of the image

24 Bits

7 Bits

5 Bits

3 Bits

2 Bits

1 Bit

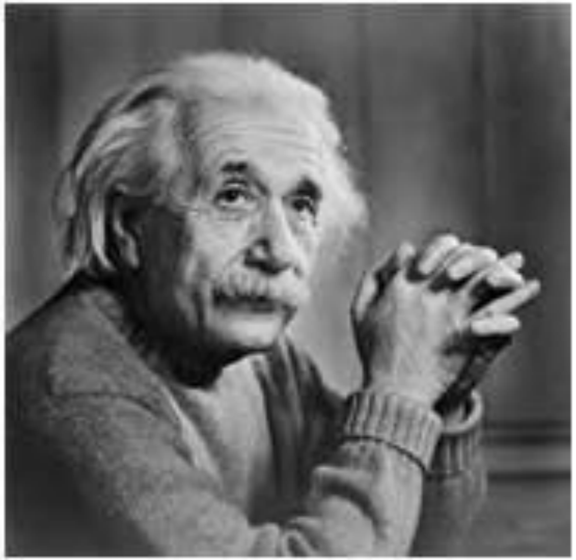
Relationship between Bits Per Pixel (BPP) and Color

Bits per pixel	Number of colors
1 bpp	2 colors
2 bpp	4 colors
4 bpp	16 colors
6 bpp	64 colors
5 bpp	32 colors
6 bpp	64 colors
8 bpp	256 colors
16 bpp	65536 colors
32 bpp	4294967296 colors (4294 million colors)

Number of colors = $(2)^{BPP}$
White color = $(2)^{BPP}-1$
For 1 bpp,
0 - Black, 1 - White
For 8 bpp,
0 - Black, 255 - White

Size of an Image

Image Size = Number of rows X Number of Columns X Number of color channels

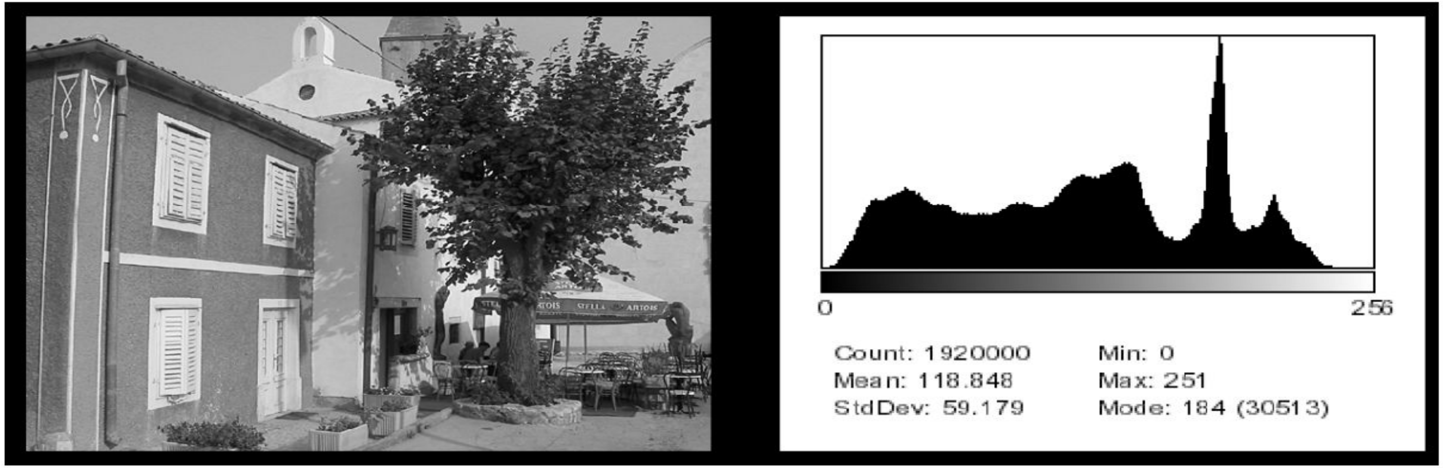


Number of rows = 182
Number of columns = 186
Number of color channels = 3
Size = 101556

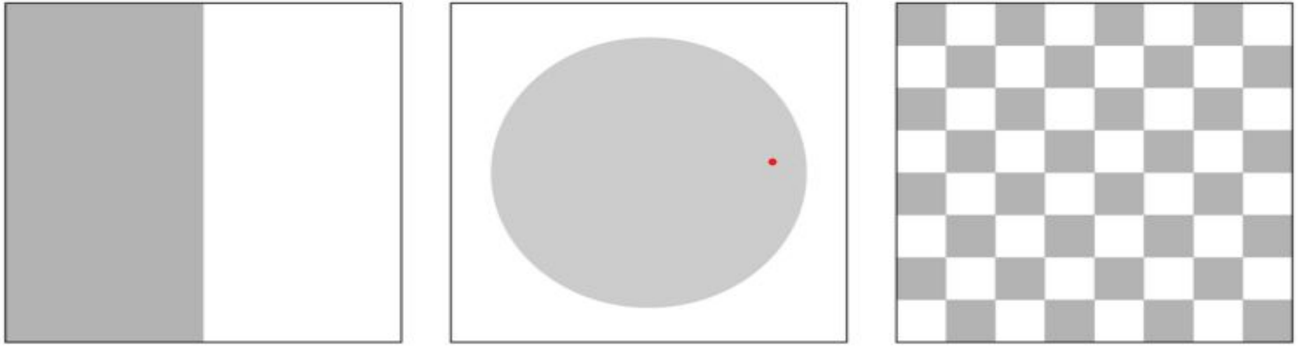
Histograms of Images

Histograms plots how many times (frequency) each intensity value in image occurs

Example: Image (left) has 256 distinct gray levels (8 bits) Histogram (right) shows frequency (how many times) each gray level occurs



- Different images can have **same** histogram
- 3 images below have same histogram



- Half of pixels are gray, half are white
  - Same histogram = same statisics
  - Distribution of intensities could be different
- Can we reconstruct image from histogram? No!

Why we need histograms of image?

- Problems with image can be identified on histogram
  - Over and under exposure
  - Brightness
  - Contrast
  - Dynamic Range
- Point operations can be used to alter histogram. E.g
  - Addition
  - Multiplication
  - Exp and Log
  - Intensity Windowing (Contrast Modification)

Image Brightness

- Brightness of a grayscale image is the **average intensity** of all pixels in image

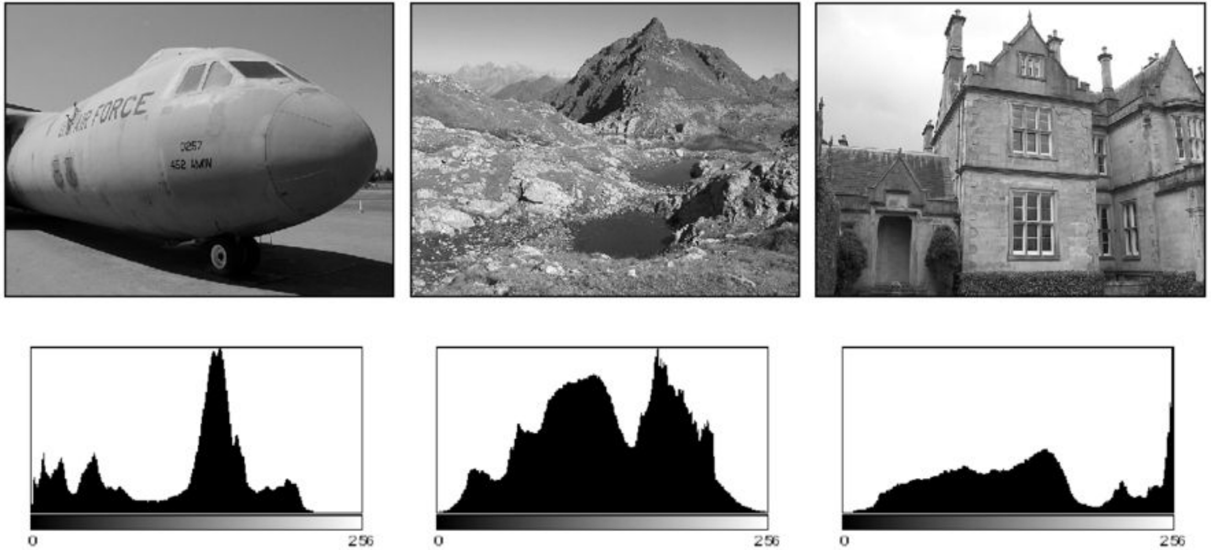
$$B(I) = \frac{1}{wh} \sum_{v=1}^{h} \sum_{u=1}^{w} I(u, v)$$

2. Divide by total number of pixels

1. Sum up all pixel intensities

Histogram and Exposure

Exposure? Are intensity values spread (good) out or bunched up (bad)



Image

Histogram

(a) (b) (c)

Underexposed | **Properly Exposed** | **Overexposed**

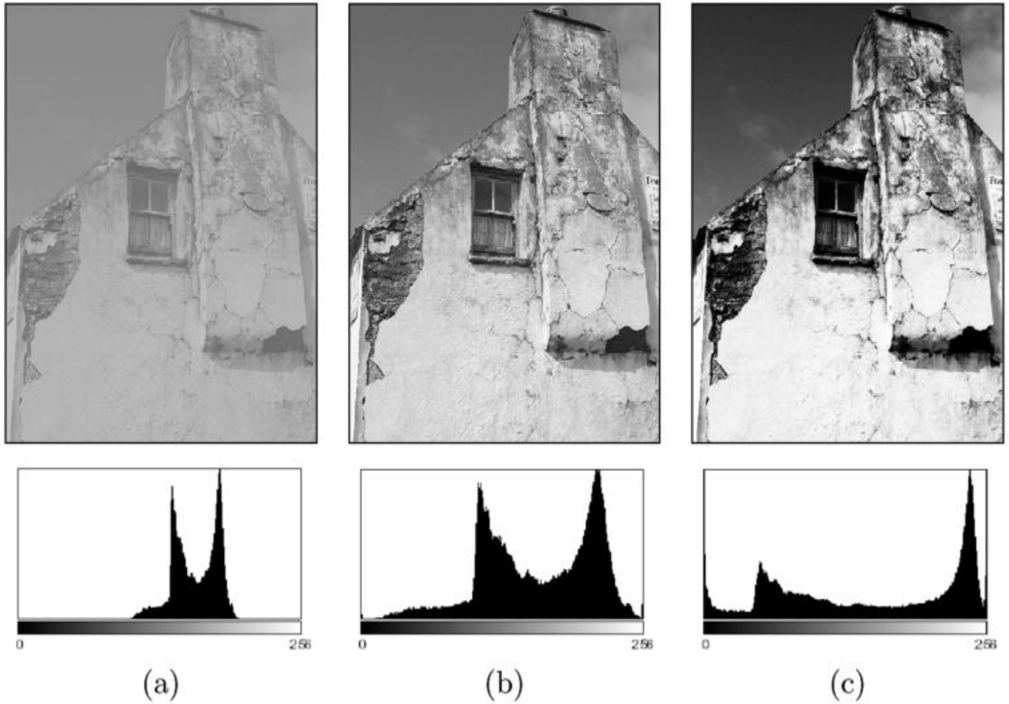
Image Contrast

- The contrast of a grayscale image indicates how easily objects in the image can be distinguished
- **High contrast image:** many distinct intensity values
- **Low contrast:** image uses few intensity values

Histogram and Contrast

Good Contrast?

Widely spread intensity values + large difference between min and max intensity values



Low contrast

Normal contrast

High contrast

Image

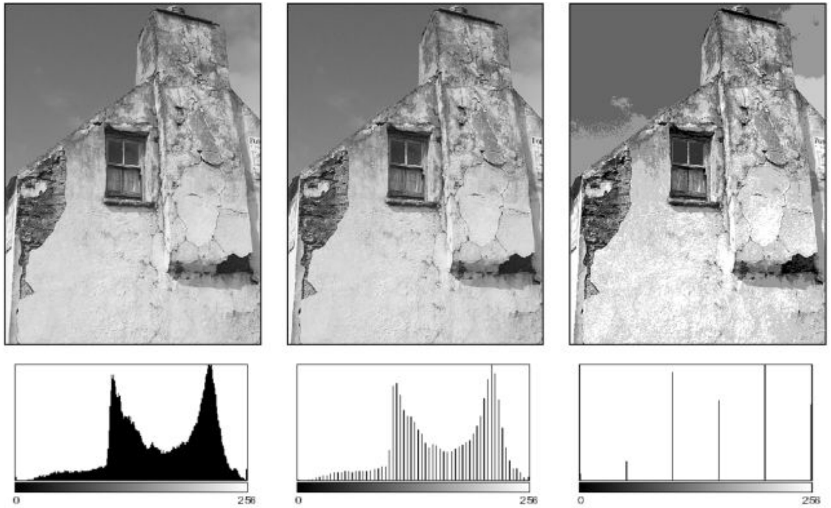
Histogram

High Dynamic Range Imaging

- High dynamic range means very bright and very dark parts in a single image (many distinct values)
- Dynamic range in photographed scene may exceed number of available bits to represent pixels
- Solution:
  - Capture multiple images at different exposures
  - Combine them using image processing

Histogram and Dynamic Range

Dynamic Range:
Number of distinct pixels in image



(a) (b) (c)

High Dynamic Range

Low Dynamic Range (64 intensities)

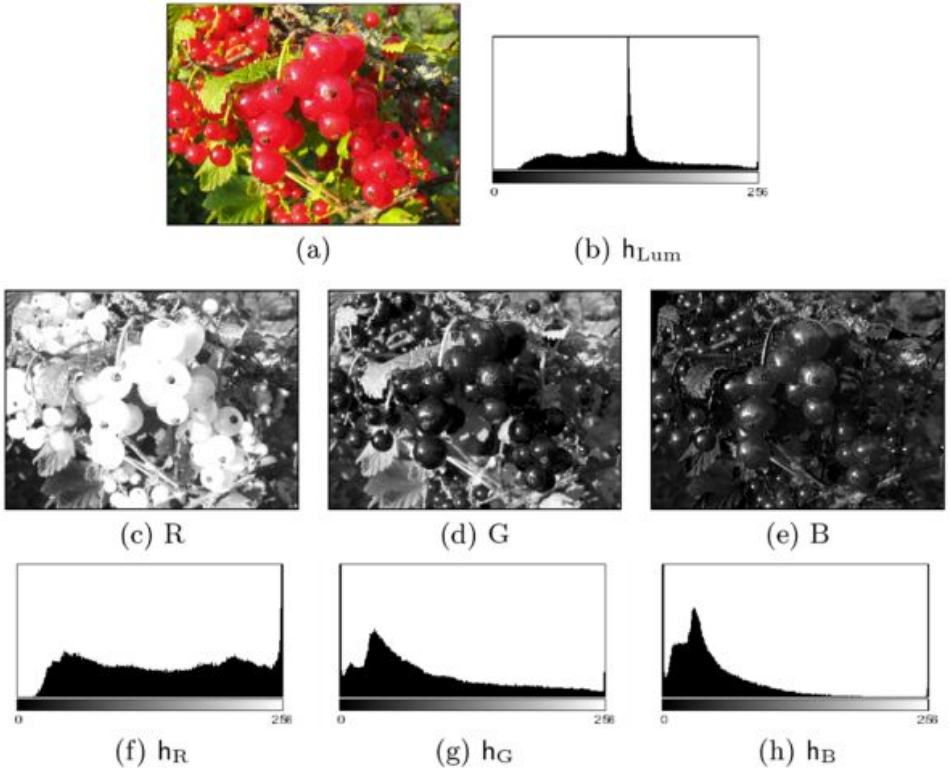
Extremely low Dynamic Range (6 intensity values)

- Difficult to increase image dynamic range (e.g. interpolation)
- HDR (12-14 bits) capture typical, then down-sample

Color Image Histograms

Two types:

1. **Intensity histogram:**
   - Convert color image to gray scale
   - Display histogram of gray scale
2. **Individual Color Channel Histograms:** 3 histograms (R,G,B)



Color Image Histograms

- Both types of histograms provide useful information about lighting, contrast, dynamic range and saturation effects
- No information about the actual color distribution!
- Images with totally different RGB colors can have same R, G and B histograms

Histogram Equalization

- Often images poorly use the full range of the gray scale
- Solution:

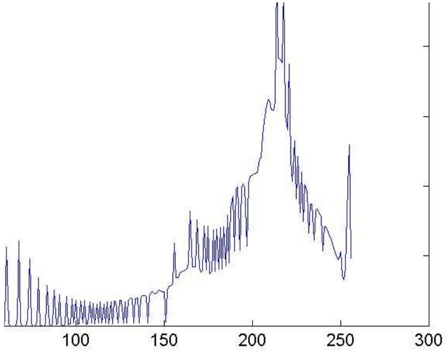
  Transform image such that its histogram is spread out more evenly in grayscale.
- Rather than guessing the parameters and the form of the transformation use original gray-scale distribution as the cue

Color Image Histograms

Source image

Equalized Image

Corresponding Histograms



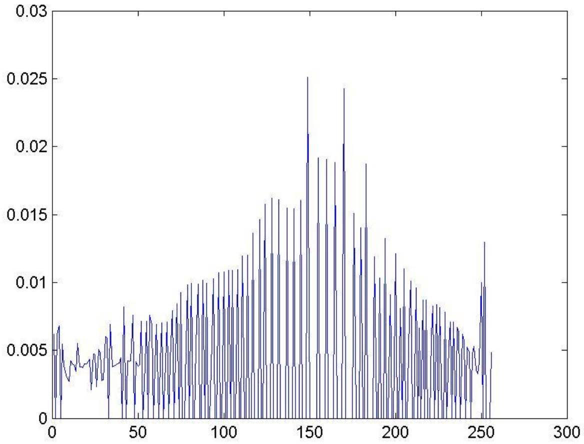


Image Filtering

Filtering is a fundamental signal processing operation, and often a pre-processing operation before further processing.

- **Applications:**
  - **Image denoising**
  - **Image enhancement (i.e., “make the image more vivid”)**
  - **Edge detection**

Basic Filters: Averaging Filters

- Basic idea: replace each pixel by the average of the pixels in a square window surrounding the pixel.
- Example: for a 3×3 averaging filter,

$$f'(x, y) = \frac{1}{9} \sum_{s=-1}^{1} \sum_{t=-1}^{1} f(x - s, y - t).$$

- Extends the idea of "moving average" for images.
- This means that the mask is constant, with all values equal to 1/9 in this case.

Basic Filters: Averaging Filters

- General case: For an $n \times n$ averaging filter,

$$w(x,y) = \frac{1}{n^2} \underbrace{\begin{bmatrix} 1 & \dots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \dots & 1 \end{bmatrix}}_{n \text{ columns}} \left.\vphantom{\begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}}\right\} n \text{ rows.}$$

  where, typically, n is an odd number.

- The elements of the mask must sum to one!

Averaging Filters Example

Averaging with a 3×3 averaging filter:

Original image

100	100	100	100	100
100	200	205	203	100
100	195	200	200	100
100	200	205	195	100
100	100	100	100	100

$\longrightarrow$

Filtered image

56	89	101	90	56
88	144	167	145	89
99	167	200	168	100
88	144	166	144	88
56	89	100	89	55

Averaging Filters Example

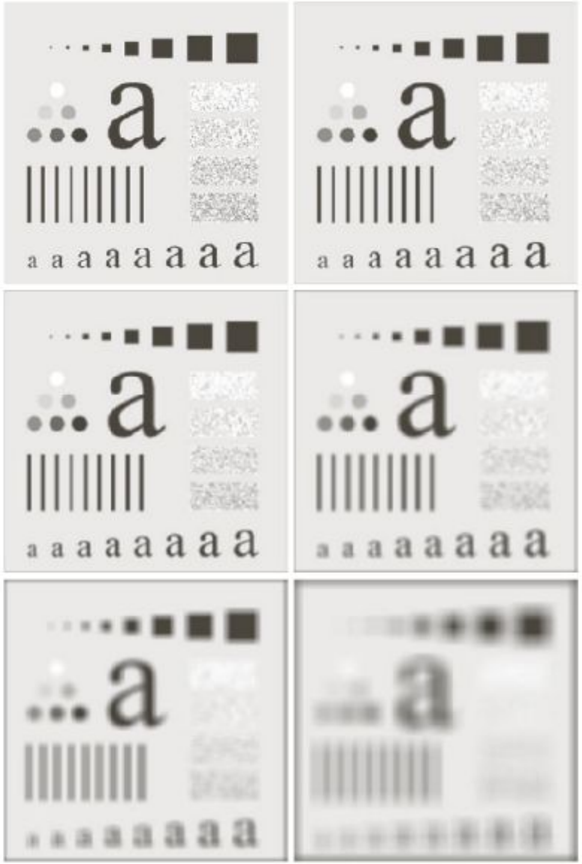


FIGURE 3.33 (a) Original image, of size 500×500 pixels. (b)–(f) Results of smoothing with square averaging filter masks of sizes $m = 3, 5, 9, 15$, and 35, respectively. The black squares at the top are of sizes 3, 5, 9, 15, 25, 35, 45, and 55 pixels, respectively; their borders are 25 pixels apart. The letters at the bottom range in size from 10 to 24 points, in increments of 2 points; the large letter at the top is 60 points. The vertical bars are 5 pixels wide and 100 pixels high; their separation is 20 pixels. The diameter of the circles is 25 pixels, and their borders are 15 pixels apart; their intensity levels range from 0% to 100% black in increments of 20%. The background of the image is 10% black. The noisy rectangles are of size 50×120 pixels.

Averaging Filter Summary

- Averaging filters can be applied for image denoising since the image pixel values change slowly but noise is a wide band signal (see previous figure).
- This filters blur image edges and other details.
  - This means that for image denoising there is a trade-off between noise remove capability and blurring of image detail.
  - Larger windows remove more noise but introduce more blur.
- Fundamentally, an averaging filter is a low-pass filter.

Weighted Averaging Filters

- Instead of averaging all the pixels in a window equally, give the pixels a weight inversely proportional to the distance to the center of the window.
- Example of a 3×3 weighted mask,

$$w(x, y) = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} .$$

  (Again, notice that weights sum to one.)
- Still, a low-pass filter. However, better behaved.

Generating Smoothing Filters

- Examples of smoothing filters:

$$\frac{1}{10}\begin{bmatrix}1 & 1 & 1\\ 1 & 2 & 1\\ 1 & 1 & 1\end{bmatrix}, \quad \frac{1}{16}\begin{bmatrix}1 & 2 & 1\\ 2 & 4 & 2\\ 1 & 2 & 1\end{bmatrix}, \quad \frac{1}{(b+2)^2}\begin{bmatrix}1 & b & 1\\ b & b^2 & b\\ 1 & b & 1\end{bmatrix}.$$

- Criteria for designing a smoothing filter:
  - $h(s,t) \geq 0$, so that it functions as averaging,
  - $\sum_{s=-s_0}^{s_1} \sum_{t=-t_0}^{t_1} h(s,t) = 1$, to preserve the dynamic range.

Generating Smoothing Filters

- Designing a Gaussian smoothing mask:
  1. Create a distance matrix to the center of the mask:

$$d(x,y) = \begin{bmatrix} \ddots & & \vdots & & \\ & \sqrt{2} & 1 & \sqrt{2} & \\ \cdots & 1 & 0 & 1 & \cdots \\ & \sqrt{2} & 1 & \sqrt{2} & \\ & & \vdots & & \ddots \end{bmatrix}.$$

  2. Apply the Gaussian function to the matrix,

$$h'(x,y) = \exp\left[-d(x,y)^2/(2\sigma^2)\right].$$

  3. Normalize: $h(x,y) = h'(x,y)/\left(\sum_s \sum_t h'(s,t)\right)$.

Nonlinear Filters

- For image denoising (and other applications), the blurring associated with linear filters is undesired.
- Moreover, linear filters are ineffective to remove some types of noise; e.g., impulsive noise.

- Solution: use nonlinear filters.

Image Edge Detection

Edge Detection is a method of segmenting an image into regions of discontinuity. **Edges** are significant local changes of intensity in a digital image. There are three types of edges:

- Horizontal edges
- Vertical edges
- Diagonal edges

Image Edge Detection

Edge Detection is a widely used technique in digital image processing like

- pattern recognition
- image morphology
- feature extraction

Image Edge Detection

Two popular methods

- Canny Edge Detection (smooth edges, based on non-maxima suppression, complex method, time consuming)
- Sobel Edge Detection (time-efficient, rough edges, based on gradients)

Image Edge Detection

Left: Original | Middle: Sobel | Right: Canny

Brief Introduction to Singular Value Decomposition

- If A is rectangular $m \times k$ matrix of real numbers, then there exists an $m \times m$ orthogonal matrix U and a $k \times k$ orthogonal matrix V such that

$$\underset{(m\times k)}{A} = \underset{(m\times m)}{U}\underset{(m\times k)}{\Lambda}\underset{(k\times k)}{V^T} \qquad UU^T = VV^T = I$$

  - Λ is an $m \times k$ matrix where the $(i,j)^{th}$ entry $\lambda_i, i = j = 1, 2, \ldots, min(m, k)$ and the other entries are zero. Physically, A equals rotation$\times$stretching$\times$rotation.
    - The positive constants λ_i are the singular values of A.
- If A has rank r, then there exists r positive constants $\lambda_1, \lambda_2, \ldots, \lambda_r$; r orthogonal $m \times 1$ unit vectors $u_1, u_2, \ldots, u_r$ and r orthogonal $k \times 1$ unit vectors $v_1, v_2, \ldots, v_r$ such that

$$A = \sum_{i=1}^{r} \lambda_i u_i v_i^T$$

Application of SVD

- SVD can be used to compute optimal low-rank approximations of arbitrary matrices.
- Face recognition
    - Represent the face images as eigenfaces and compute distance between the query face image in the principal component space.
- Data mining
    - Latent Semantic Indexing for document extraction.
- Image Compression

Image Compression

- Image compression is minimizing the size in bytes of a graphics file without degrading the quality of the image to an unacceptable level.
- The reduction in file size allows more images to be stored in a given amount of disk or memory space.
- It also reduces the time required for images to be sent over the Internet or downloaded from Web pages.

Application: Images sent over Whatsapp Messenger

Image Compression

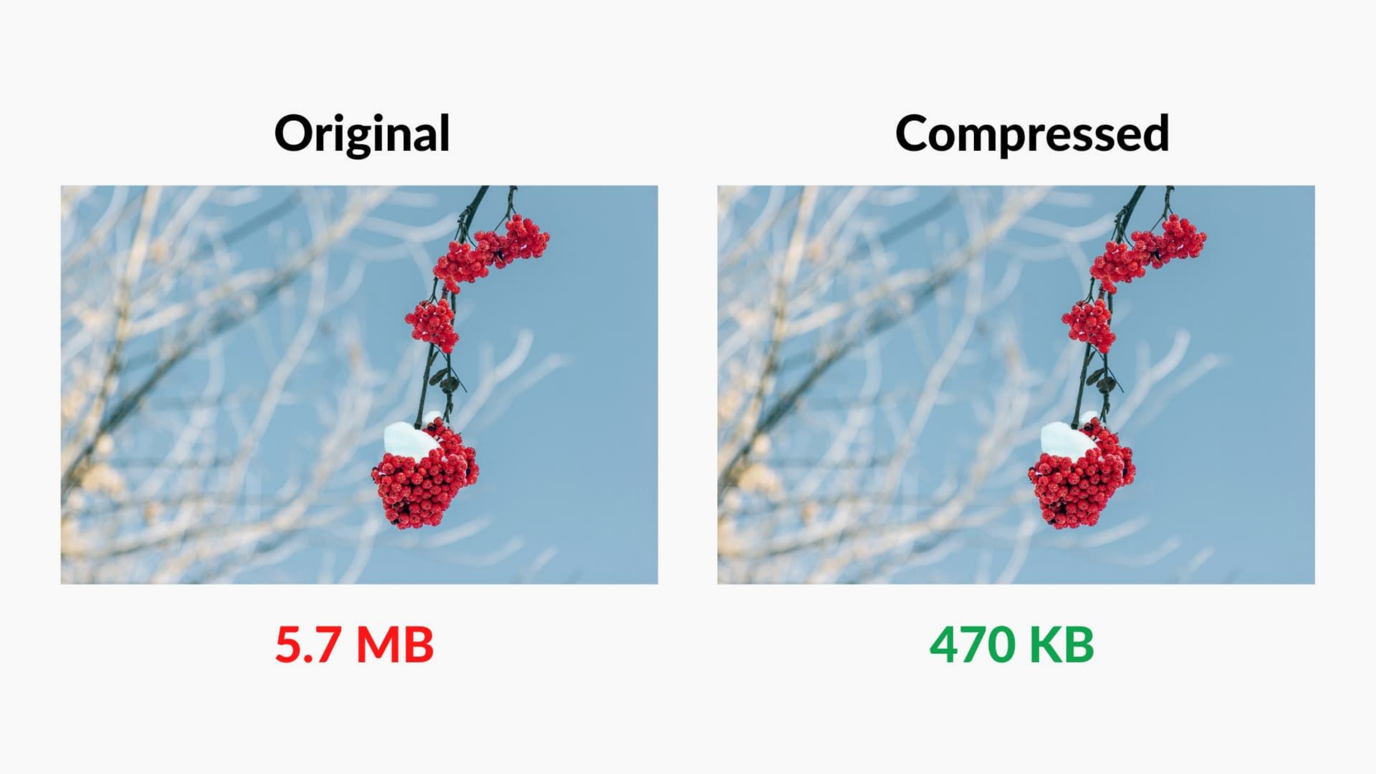


Image Compression Using SVD

- An image is stored as a 200×200 matrix M with entries between 0 and 1. The matrix M has rank 200.
- Select $r > 200$ as an approximation to the original M.
  - As r is increased from 1 all the way to 200 the reconstruction of M would improve i.e. approximation error would reduce
- Advantage
  - To send the matrix M, need to send $200 \times 200 = 40000$ numbers.
  - To send an $r = 35$ approximation of M, need to send $35 + 35 * 200 + 35 * 200 = 14035$ numbers
    - 35 singular values.
    - 35 left vectors, each having 200 entries.
    - 35 right vectors, each having 200 entries.

Thank You For Your Time Today

ANY QUESTIONS?