EE 569 Discussion

Zhiruo Zhou

01/22/2020

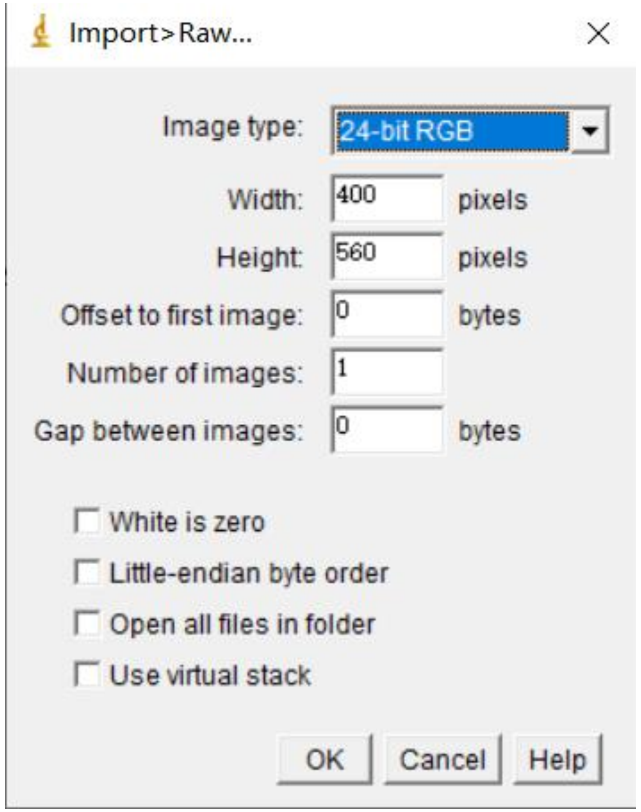
Before you start doing the homework

- Read all the following files carefully
  - EE569_Homework_Guidelines.pdf
  - EE569_MATLAB_Function_Guidelines.pdf
  - EE569_2021Spring_hw1.pdf

- Check the discussion board regularly

Image I/O data type – RAW

- Image file format: refer to "EE569_Homework_Guidelines.pdf"



- Free software:
  - ImageJ: http://rsb.info.nih.gov/ij/
- If you cannot open the raw file
  - Check the height, width and bits
  - "8-bit" for gray images
  - "24-bit RGB" for color images
- Use this to view images instead of staring at those on hw pdf files

Image I/O data type – RAW

- For programming purpose
  - You need to implement “readraw” and “writeraw” functions for reading and writing RAW images
  - Readraw: RAW to matrix
  - Writeraw: matrix to RAW

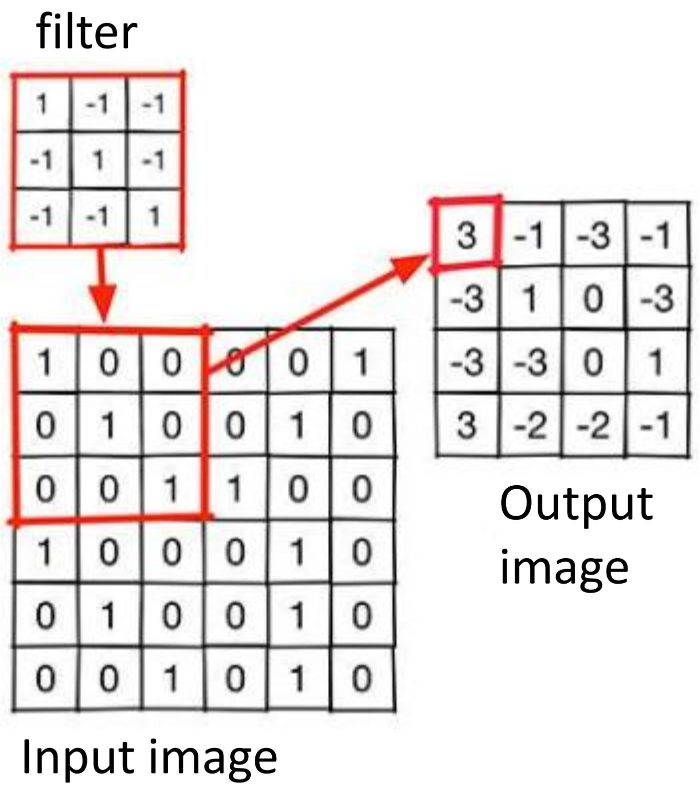
  - Reference code uploaded on DEN
  - Your functions should work for arbitrary image size, gray images, color images…
  - How to check whether your code is correct?
    - Could you imshow the output of readraw correctly?
    - Could ImageJ open the output of writeraw correctly?

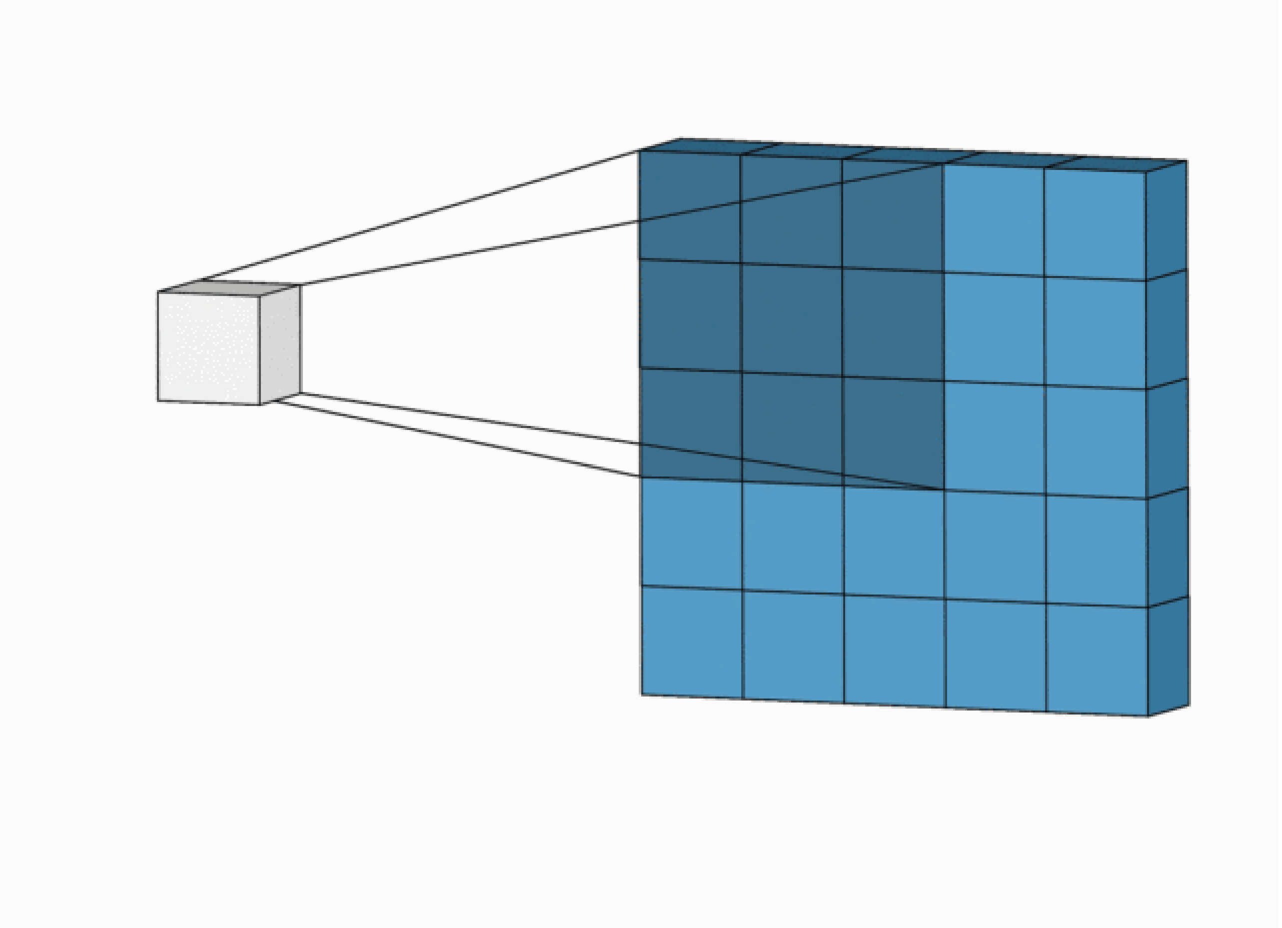
Convolution

- Also named filtering in image processing

1. Elementwise multiplication

2. Summation

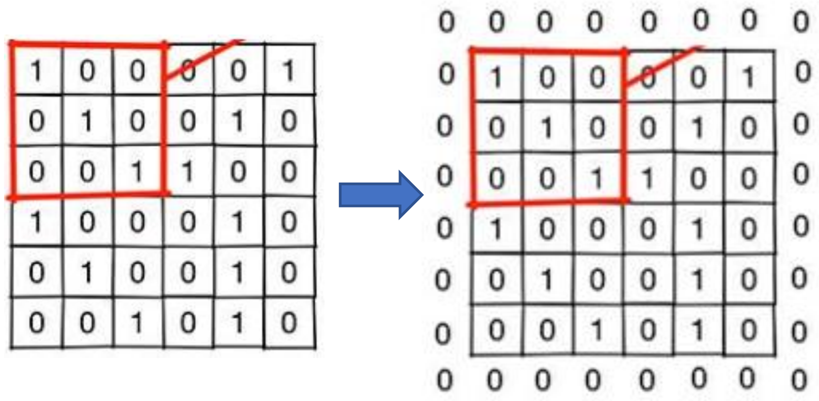


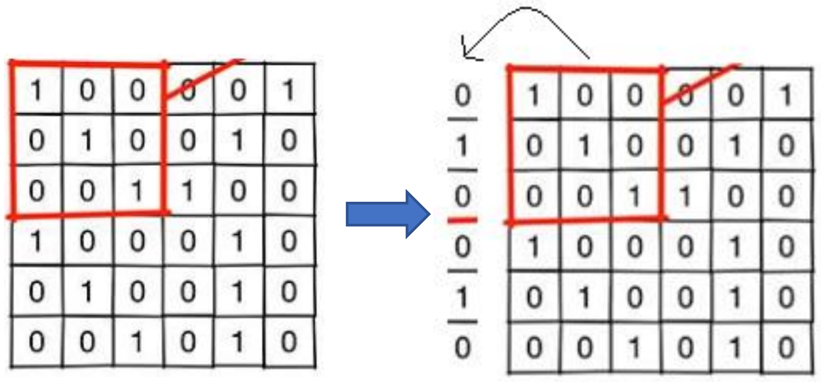


Padding (boundary extension)

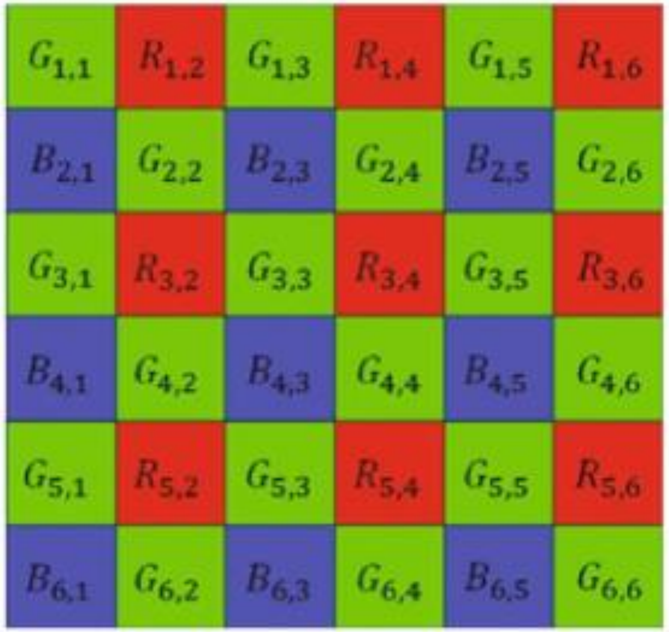
- To make the output size the same with that before convolution

- Mainly two types of padding
  - Zero padding
  - Mirror reflection
    - Row first or column first? Both are fine.

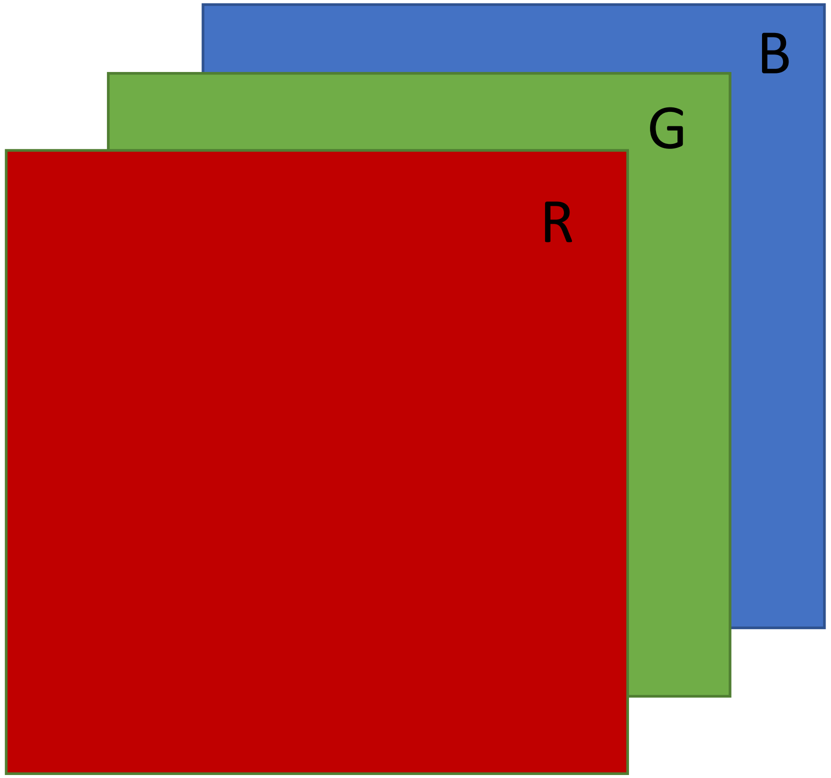




- Idea: interpolation



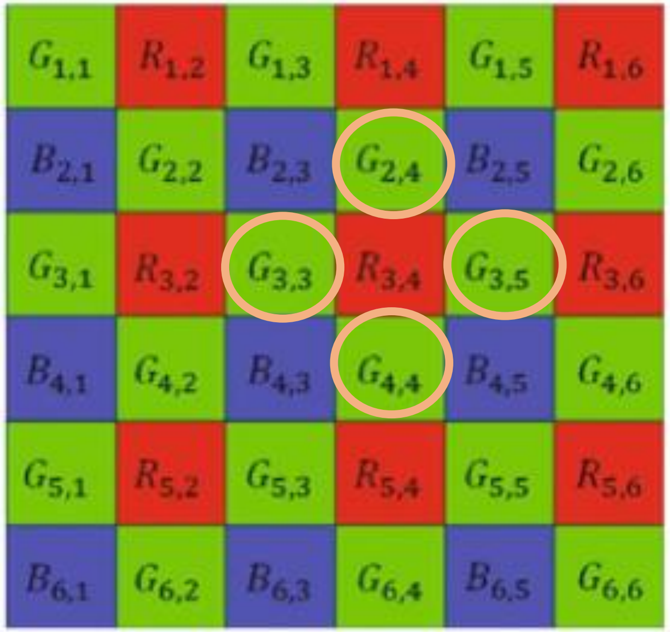
Input: H*W



Output: H*W*3

HW1 - P1 Demosaicing

- Bilinear interpolation



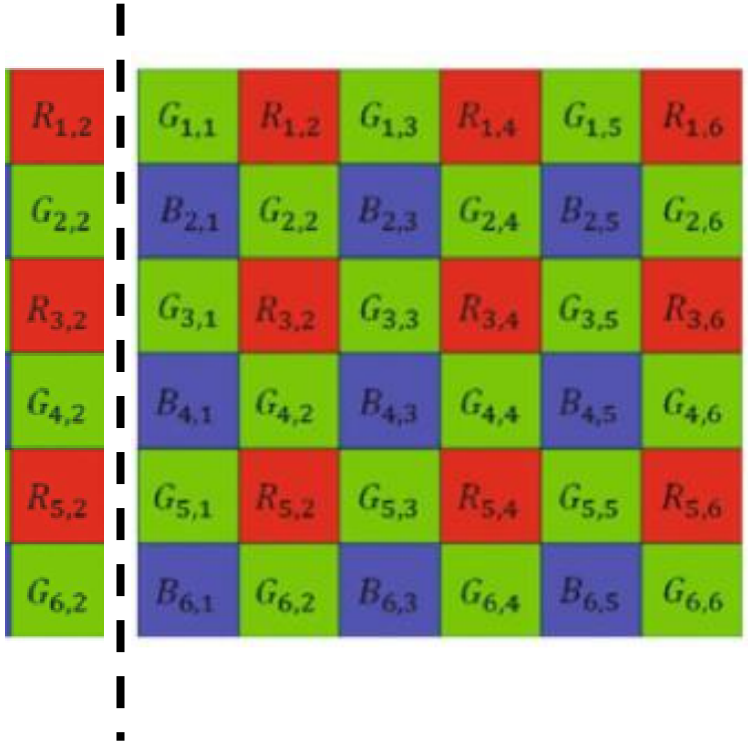
$$\hat{G}_{3,4}^{bl} = \frac{1}{4}\left(G_{3,3} + G_{2,4} + G_{3,5} + G_{4,4}\right)$$

What's the corresponding filter?

$$\begin{bmatrix} 0 & 1/4 & 0 \\ 1/4 & 0 & 1/4 \\ 0 & 1/4 & 0 \end{bmatrix}$$

HW1 - P1 Demosaicing

- Use “mirror reflection” for padding



HW1 – P1 Histogram Manipulation

- **Method A**: the transfer-function-based histogram equalization method
- **Method B**: the cumulative-probability-based histogram equalization method (bucket-filling method)

- **Note**
  - Generally, histogram manipulation can only cope with gray images
  - For RGB images, process each channel separately

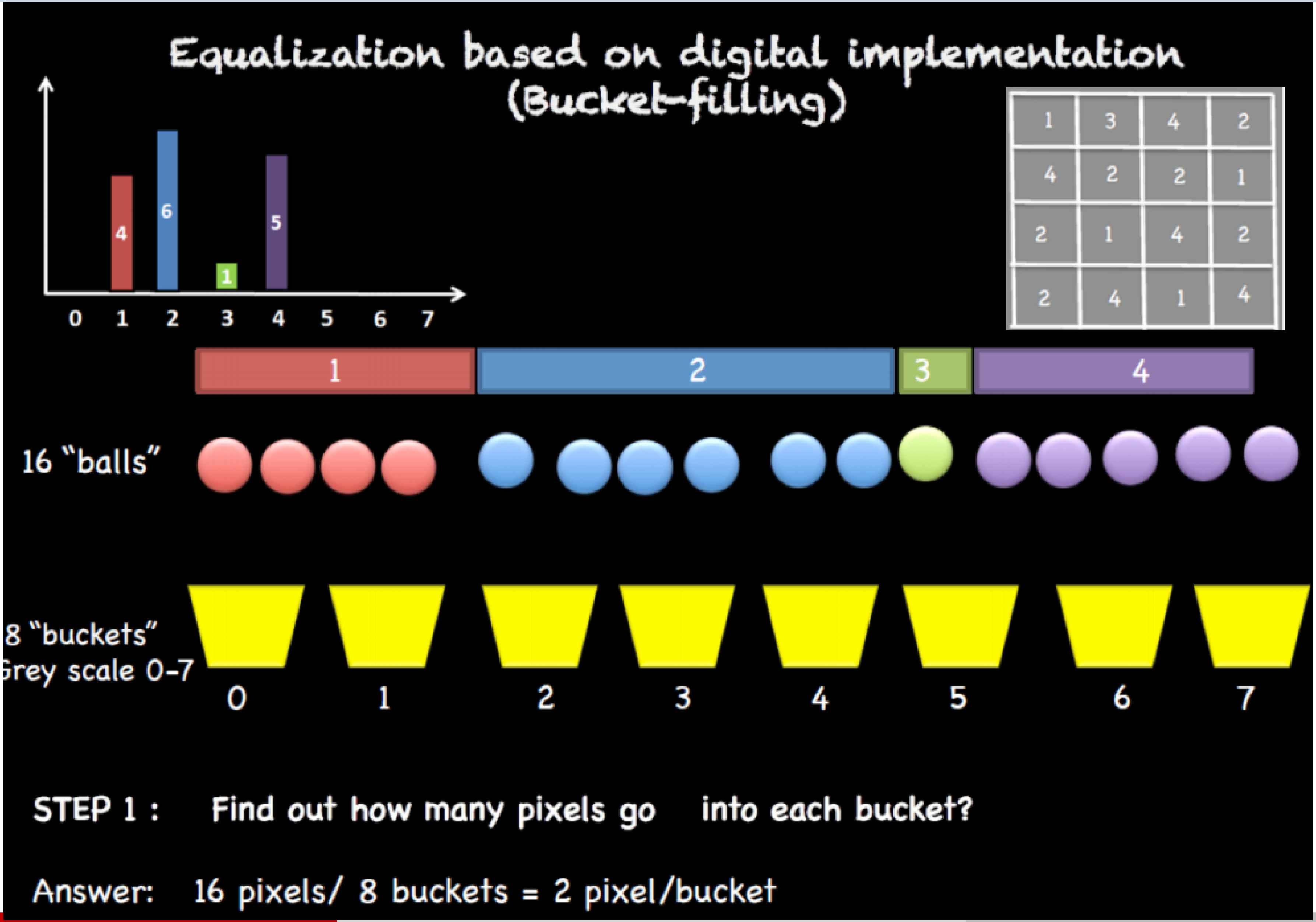
Method A: TRANSFER-FUNCTION-BASED

- Step-by-step Procedure:
  - Step 1: Obtain the histogram
    - Count the frequency of pixels of each grayscale value (0~255)
  - Step 2: Calculate the normalized probability histogram
    - Divide the histogram by total number of pixels
  - Step 3: Calculate the CDF
  - Step 4: Create the mapping-table
    - Mapping rule: x to CDF(x)* 255
    - Transfer function: $x \to 255 * CDF(x)$

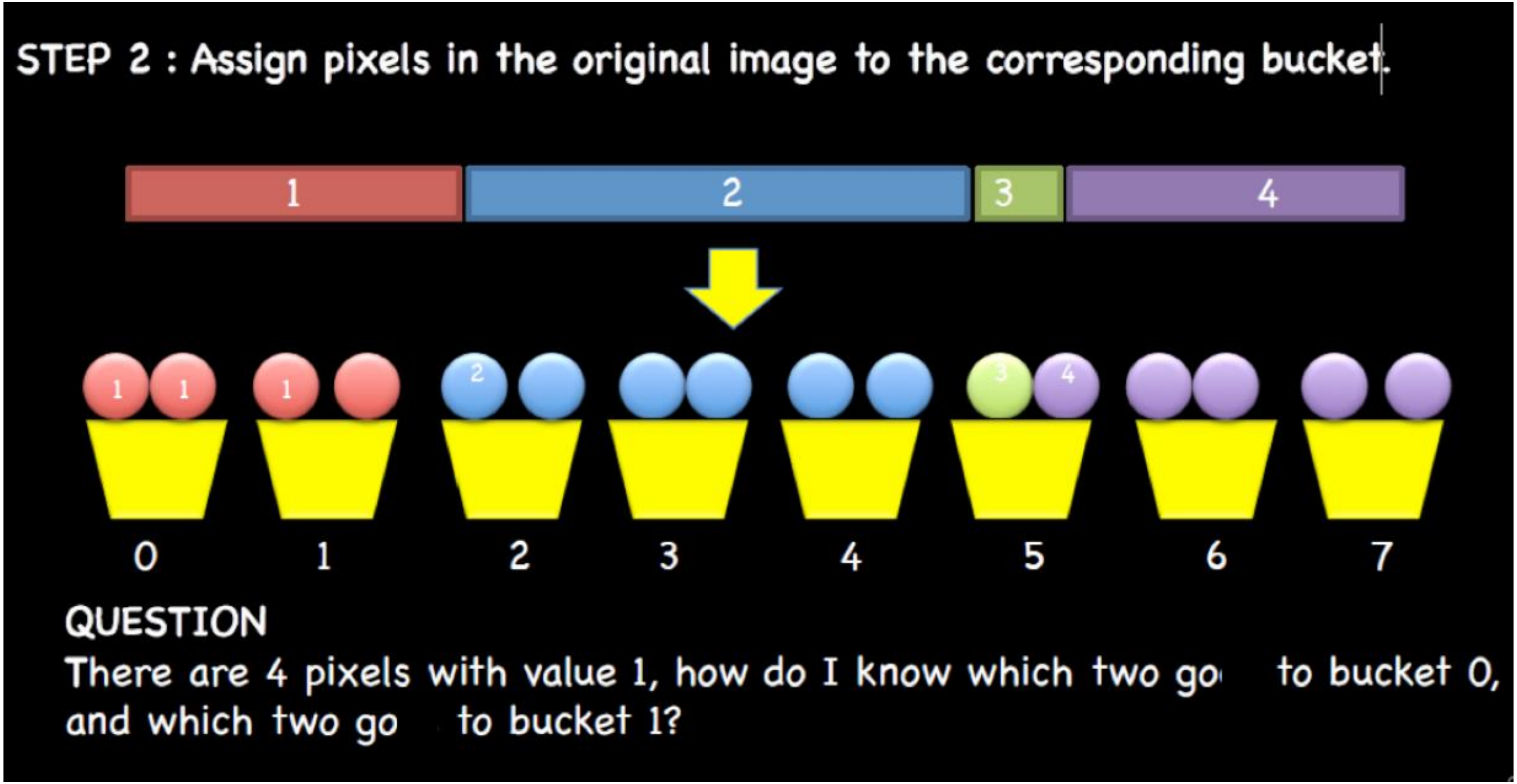
- Intuitive Tutorial Video available at

https://www.youtube.com/watch?v=PD5d7EKYLcA

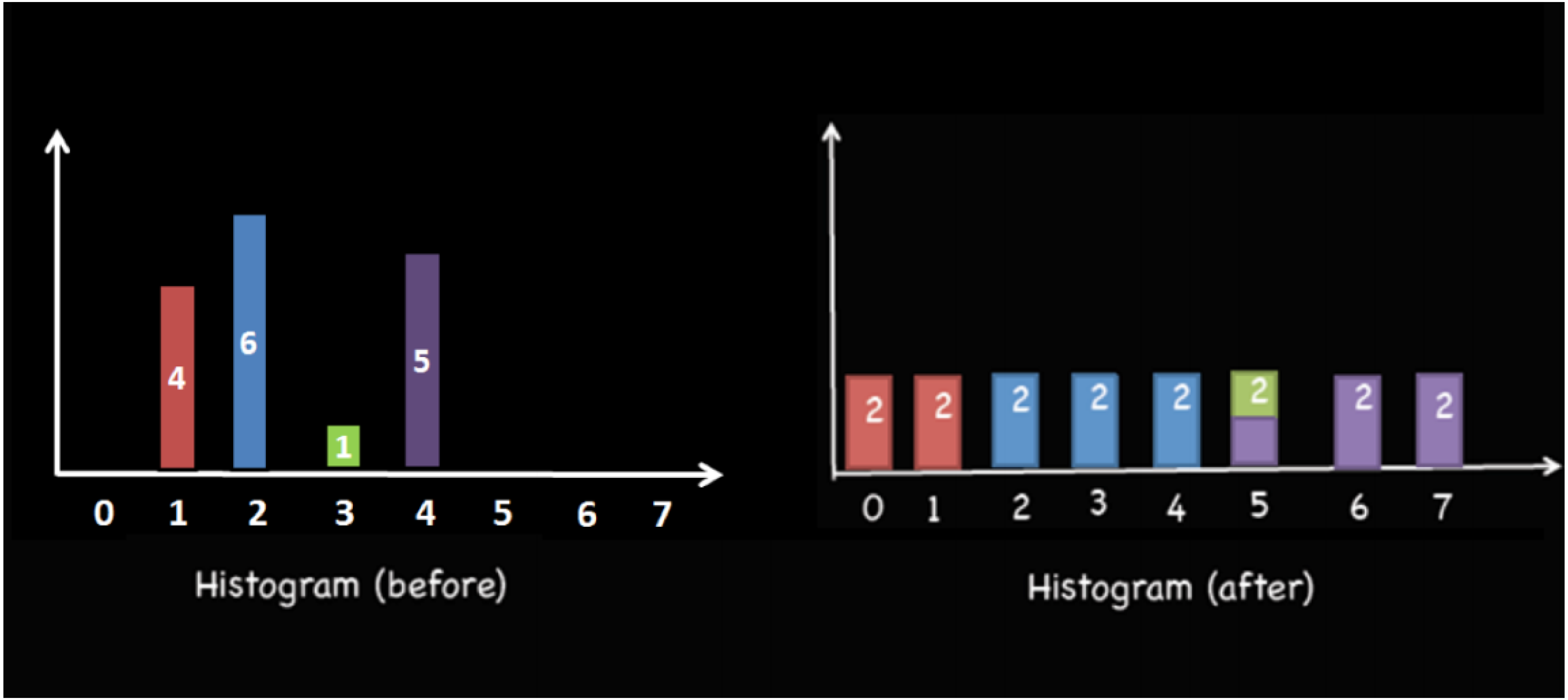
Method B: BUCKET FILLING



Method B: BUCKET FILLING



Method B: BUCKET FILLING



HW1 – P1 Histogram Manipulation

Original　　Method A　　Method B

HW1 – P1 Histogram Manipulation

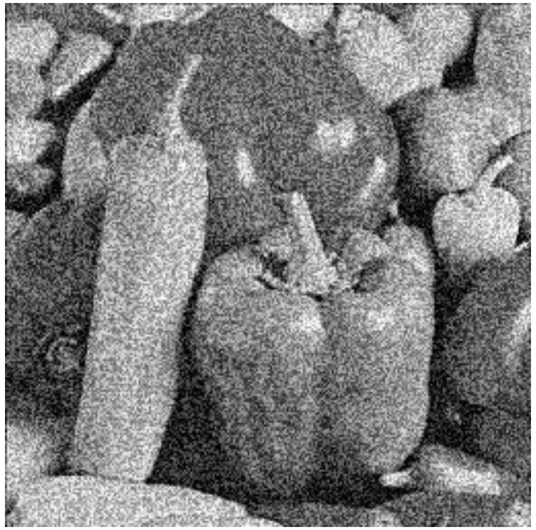
- Note
  - Histogram value needs to be computed by your code
  - Do NOT use Photoshop, ImageJ or MATLAB functions imhist()/hist() to obtain histogram
  - Figure plotting can be done by any tools - Matlab or MS Excel

HW1 – P2 Denoising

- Different types of noises
  - Uniform noise
  - Gaussian noise
  - Impulse noise (“pepper and salt”)

How to differentiate uniform noise and Gaussian noise?

- Draw the histogram of noise and see the distribution



Uniform (or Gaussian)

Impulse

HW1 – P2 Denoising

- Different denoising methods:
  - Low pass filter (mean, Gaussian)
  - Median filter
  - Bilateral filter
  - Non local mean (NLM) filter
  - BM3D
  - - …

HW1 – P2 Denoising

- Low pass filter (mean)

1	1	1
1	1	1
1	1	1

3x3 Mean kernel

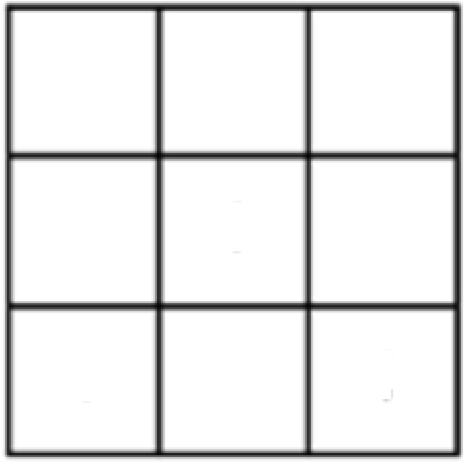
$$Y(i,j) = \frac{\sum_{k,l} I(k,l)w(i,j,k,l)}{\sum_{k,l} w(i,j,k,l)}$$

$$w(i,j,k,l) = \frac{1}{w_1 \times w_2}$$

where (k,l) is the neighboring pixel location within the window of size w_1 x w_2 centered around (i,j), I is the noisy image, Y is the output image.

HW1 – P2 Denoising

- Low pass filter (Gaussian)



$$Y(i,j) = \frac{\sum_{k,l} I(k,l)\,w(i,j,k,l)}{\sum_{k,l} w(i,j,k,l)}$$

$$w(i,j,k,l) = \frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{(k-i)^2+(l-j)^2}{2\sigma^2}\right)$$

where σ is the standard deviation of Gaussian distribution.

HW1 – P2 Denoising

- Bilateral filter

$$Y(i,j) = \frac{\sum_{k,l} I(k,l)w(i,j,k,l)}{\sum_{k,l} w(i,j,k,l)}$$

$$w(i,j,k,l) = exp\left(-\frac{(i-k)^2 + (j-l)^2}{2\sigma_c^2} - \frac{\|I(i,j) - I(k,l)\|^2}{2\sigma_s^2}\right)$$

where σ_c and σ_s are parameters of your choice.

1. Preserve sharp edges
2. Weights depend not only on Euclidean distance of pixels, but also on the difference on the pixel values

HW1 – P2 Denoising

- Non local mean filter

where $N_{x,y}$ is the window centered around location (x,y), and h is the filtering parameter. $\aleph$ denotes the local neighborhood centered at the origin, n_1, $n_2 \in \aleph$ denotes the relative position in the neighborhood window. a is the standard deviation of the Gaussian kernel.

$$w(i,j,k,l) = \exp\left(-\frac{\left\|I(N_{i,j}) - I(N_{k,l})\right\|_{2,a}^2}{h^2}\right)$$

$$\left\|I(N_{i,j}) - I(N_{k,l})\right\|_{2,a}{}^2 = \sum_{n_1,n_2\in\aleph} G_a(n_1,n_2)\Big(I(i-n_1,j-n_2) - I(k-n_1,l-n_2)\Big)^2$$

$$G_a(n_1,n_2) = \frac{1}{\sqrt{2\pi}a}\exp\left(-\frac{n_1^2+n_2^2}{2a^2}\right)$$

where $N_{x,y}$ is the window centered around location (x,y), and h is the filtering parameter. $\aleph$ denotes the local neighborhood centered at the origin, n_1, $n_2 \in \aleph$ denotes the relative position in the neighborhood window. a is the standard deviation of the Gaussian kernel.

HW1 – P2 Denoising

- Non local mean filter
  - Interpretation:

  takes Gaussian weighted Euclidean distance between the block centered the target pixel and the neighboring block

  - Good denoising performance
  - Computationally intensive

  - You can use online source code for NLM

How to write a good report

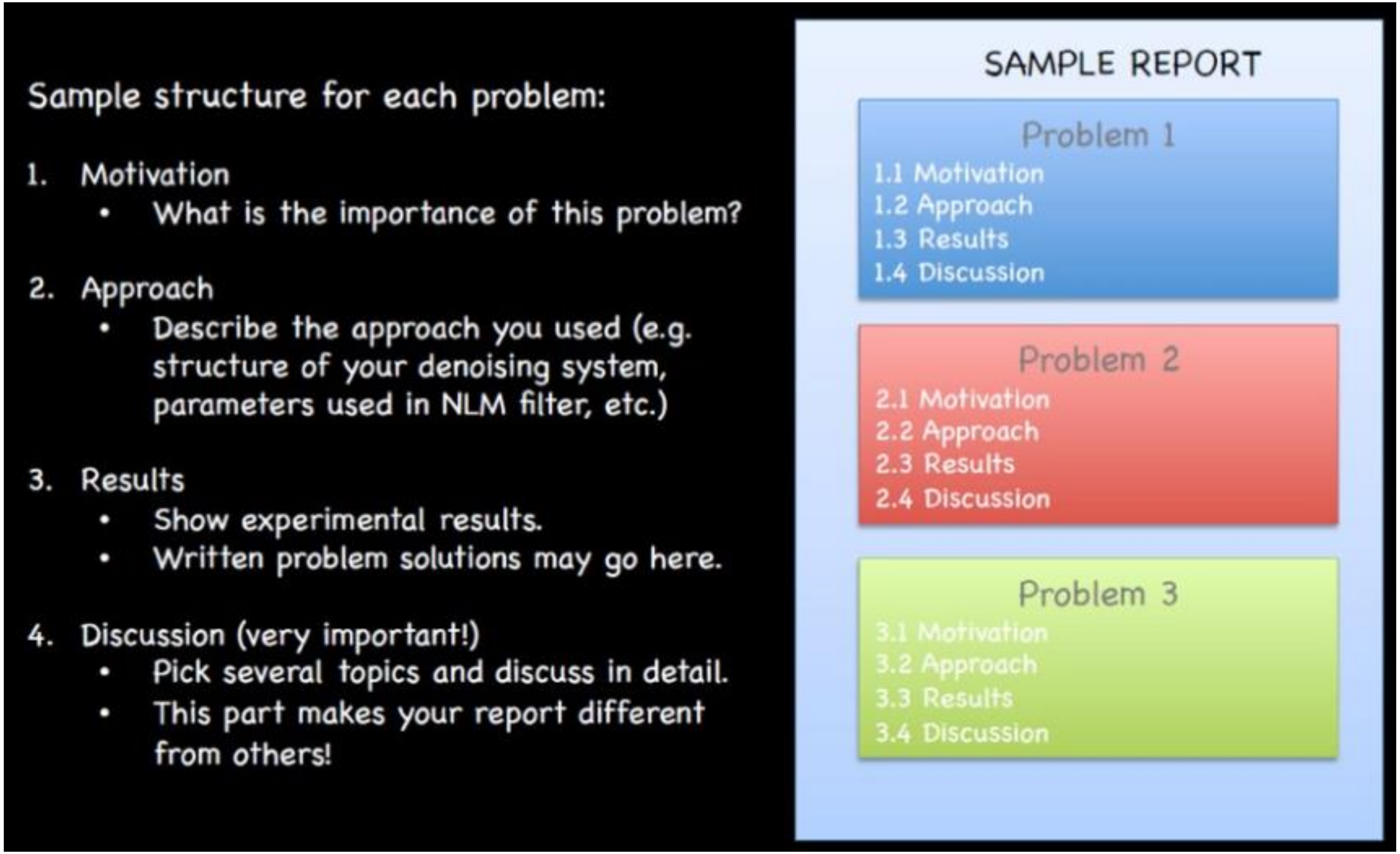
- **How many pages should you write?**
  - Short answer: it depends (font/figure size)
  - Longer report != better report
  - Suggestions: 1) sufficient length, 2) with some in depth discussion

- **Emphasize your idea/contribution**. That's what makes your work different

- **Use figures/tables/graphs/examples to strengthen your points**. Sometimes they are more powerful than words.

Do **NOT** copy problem descriptions into your report!

How to write a good report



Coding style

- Google Style Guides
  - Style guides for Google-originated open-source projects
  - https://google.github.io/styleguide/
  - C/C++, Python…

- Guidelines for writing clean and fast code in MATLAB
  - https://www.mathworks.com/matlabcentral/fileexchange/22943-guidelines-for-writing-clean-and-fast-code-in-matlab

- Generally
  - Modularized functions
  - Reasonable function arguments
  - Less hard coding


Q
A