CAP 4453 Robot Vision

Dr. Gonzalo Vaca-Castaño

Gonzalo.vacacastano@ucf.edu

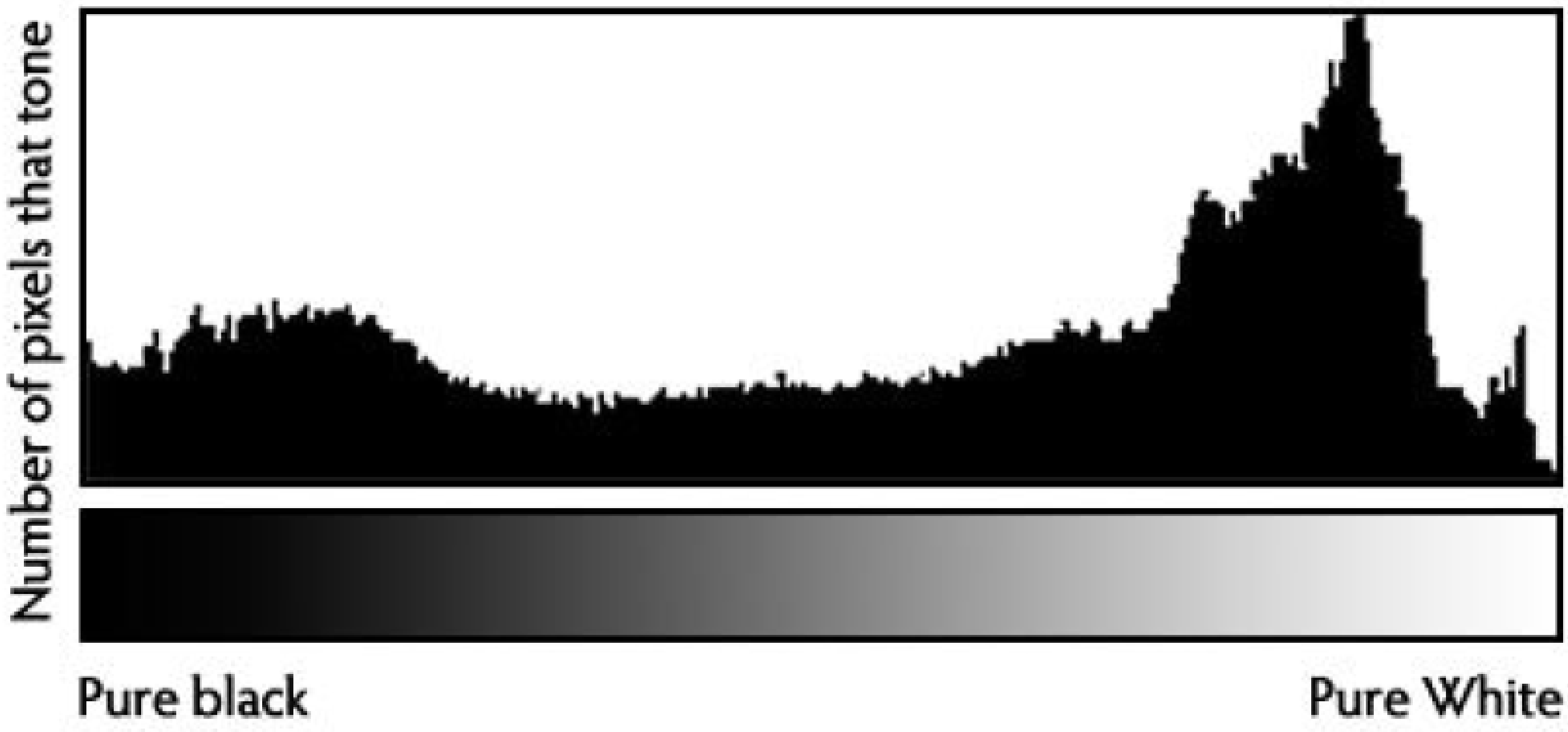
Administrative details

- REU Program 2023REUFlyer.pdf
- Homework 1 issues ?

Histogram

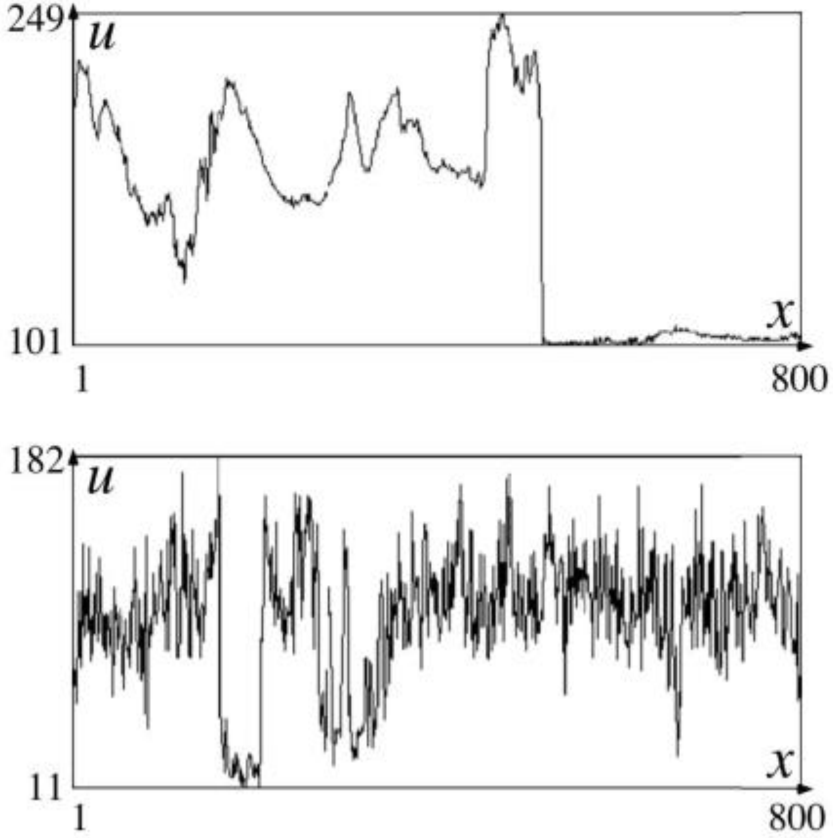
- A histogram is a graphic representation of numerical data that shows the data distribution. When the number of observations is large, and the bin's size is small, the histogram will be similar to the distribution density chart.
- How to create a histogram?
    1. Find the range of the numerical data Range = Max - Min.
    2. Choose the number of bins you prefer to present.
    3. Calculate the bin size: Bin size = Range/number of bins.
    4. For every bin count, the total number of observations falls in the bin.
    5. Present the data s a column chart, where each column represents the number of observations in a bin.

Image Histogram

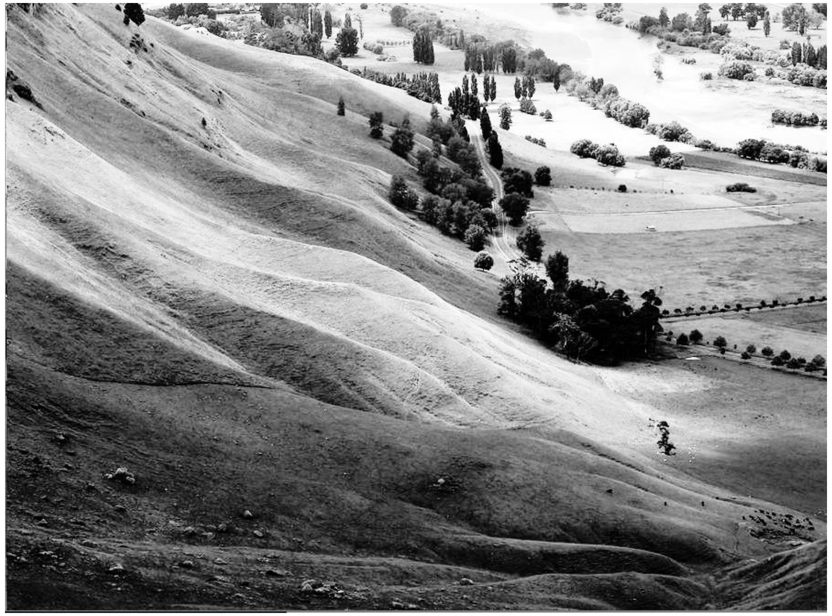


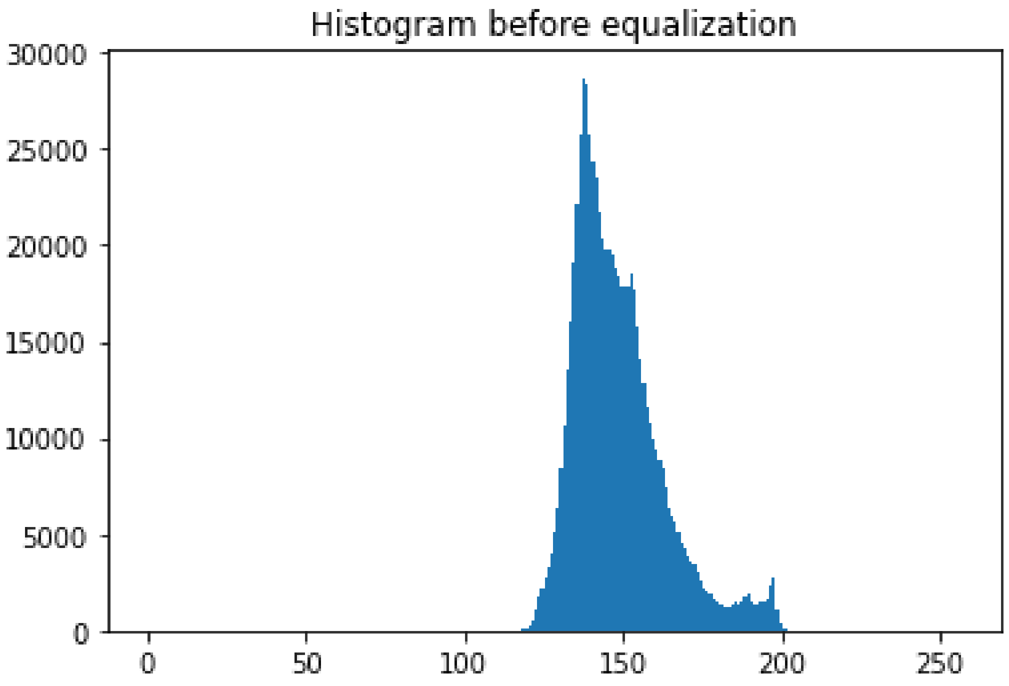
Histogram Example

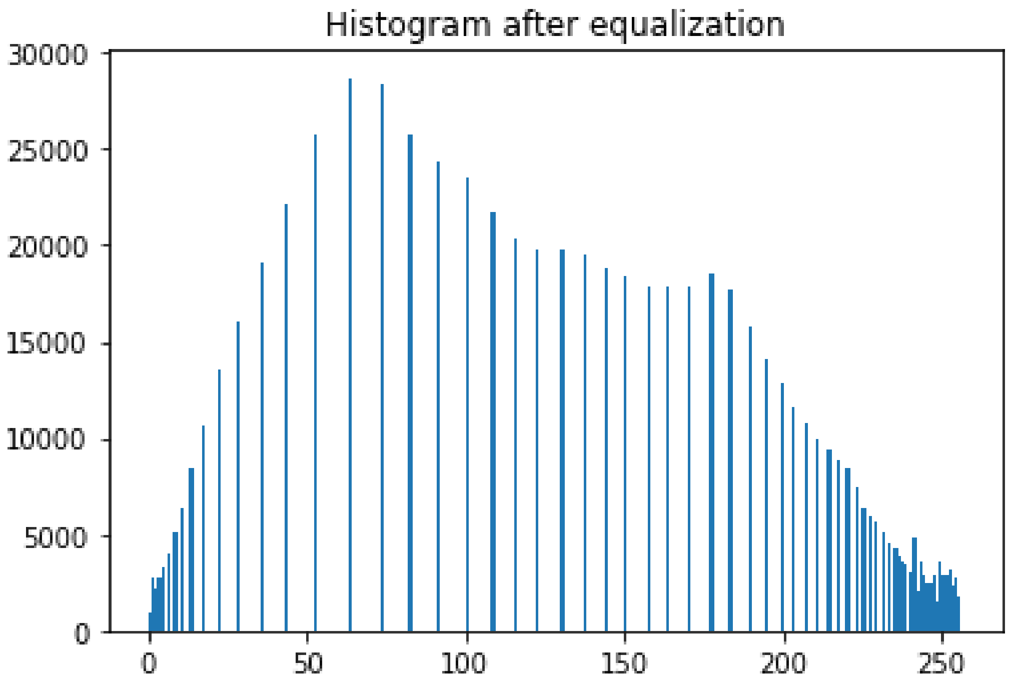
Intensity profiles for selected (two) rows



```python
Created on Tue Jan 17 08:31:55 2023

@author: gonza
"""

import cv2
import numpy as np
import matplotlib.pylab as plt


# reading an image using imreadmethod
my_img = cv2.imread('madanmohan_temple.jpg', 0)
my_img = cv2.imread('Unequalized_Hawkes_Bay_NZ.jpg',0)
equ = cv2.equalizeHist(my_img)

# stacking both the images side-by-side orientation
res = np.hstack((my_img, equ))

# getting histograms
plt.hist(my_img.ravel(),bins = 256, range = [0,256])
plt.title('Histogram before equalization')
plt.show()

plt.hist(equ.ravel(),bins = 256, range = [0,256])
plt.title('Histogram after equalization')
plt.show()


# showing image input vs output
cv2.imshow('image', res)
cv2.waitKey(0)
cv2.destroyAllWindows()
```




Histogram before equalization
30000
25000
20000
15000
10000
5000
0
0
50
100
150
200
250


Histogram after equalization
30000
25000
20000
15000
10000
5000
0
0
50
100
150
200
250

Questions?

Robot Vision

3. Image Filtering

Credits

- Some slides comes directly from:
  - Yogesh S Rawat (UCF)
  - Noah Snavely (Cornell)
  - Ioannis (Yannis) Gkioulekas (CMU)
  - Mubarak Shah (UCF)
  - S. Seitz
  - James Tompkin
  - Ulas Bagci

Outline (next 2 weeks)

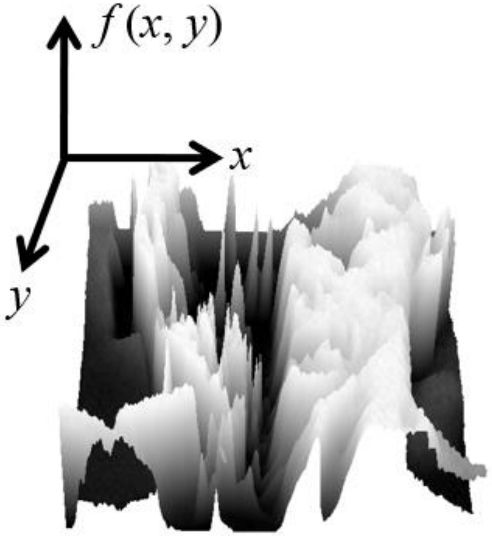
- ~~Image as a function~~
  - Linear algebra
- Extracting useful information from Images
  - ~~Histogram~~
  - Noise
  - Filtering (linear)
  - Smoothing/Removing noise
  - Convolution/Correlation
  - Image Derivatives/Gradient
  - Edges
- Colab Notes/ homeworks
- Read Szeliski, Chapter 3.
- Read/Program CV with Python, Chapter 1.

From last class

What is an image?

- We can think of a (grayscale) image as a **function**, *f*, from R^2 to R:
  - $f(x,y)$ gives the **intensity** at position (x,y)

snoop

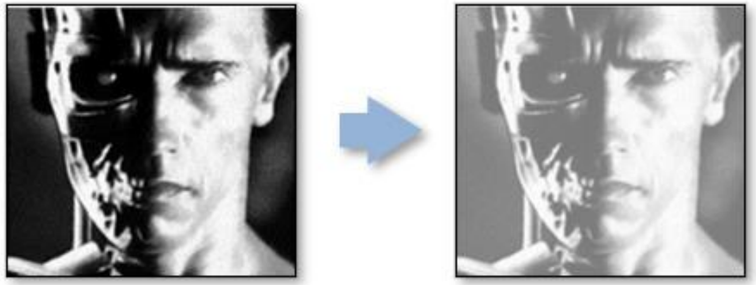


3D view

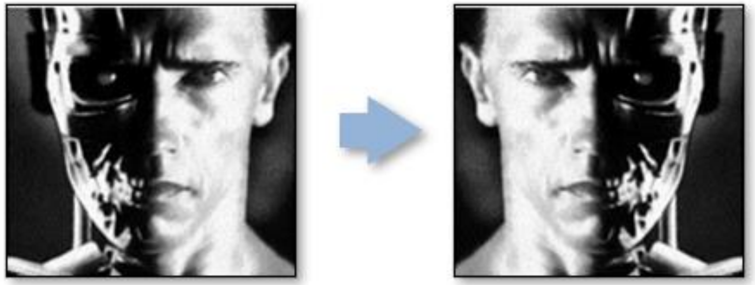
  - A **digital** image is a discrete (**sampled**, **quantized**) version of this function

Image transformations

- As with any function, we can apply operators to an image



$g(x,y) = f(x,y) + 20$



$g(x,y) = f(-x,y)$

- Today we'll talk about a special kind of operator, *convolution* (linear filtering)

Basic Linear Algebra

Linear Algebra basics

- Vectors
  - Operations
- Matrix
  - Operations

Linear Algebra basics
Vector

- Scalar: $x \in \mathbb{R}$
- Vector: $\boldsymbol{x} \in \mathbb{R}^N$
  - Row Vector v $\in \mathbb{R}^{1\times n}$

    $$\boldsymbol{x} = [x_1 \quad x_2 \quad \cdots \quad x_n]$$

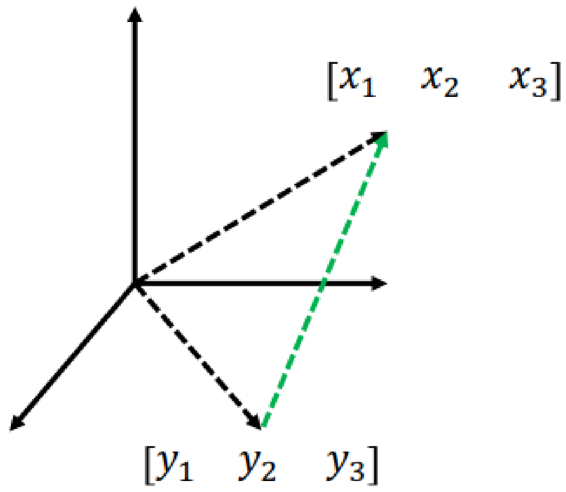
  - Column vector v $\in \mathbb{R}^{n\times 1}$: $\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = [x_1 \quad x_2 \quad \cdots \quad x_n]^{\boldsymbol{T}}$
  - Transpose

Linear Algebra Basics

Vectors - use

- Store data in memory
  - Feature vectors
  - Pixel values
  - Any other data for processing
- Any point in coordinate system
  - Can be n dimensional
- Difference between two points

$$[x_1 - y_1 \quad x_2 - y_2 \quad x_3 - y_3]$$



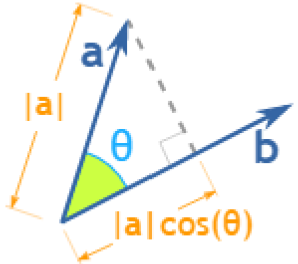
Linear Algebra Basics

Vector operations

- Norm – size of the vector
- p-norm $$\|x\|_p = \left(\sum_i |a_i|^p \right)^{\frac{1}{p}} \qquad p \geq 1$$
- Euclidean norm $$\|x\|_2 = \left(\sum_i |a_i|^2 \right)^{1/2}$$
- L1-norm $$\|x\|_1 = \left(\sum_i |a_i| \right)$$
- L-infinity $$||\boldsymbol{x}||_\infty = \max_i |x_i|$$

Linear Algebra Basics
Vector operations

- Inner product (dot product)
  - Scalar number
  - Multiply corresponding entries and add



$$\boldsymbol{x}^T\, \boldsymbol{y} = \begin{bmatrix} x_1 & x_2 & \cdots & x_n \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \sum_k^n x_k\ y_k$$

[Dot Product (mathsisfun.com)](https://www.mathsisfun.com)

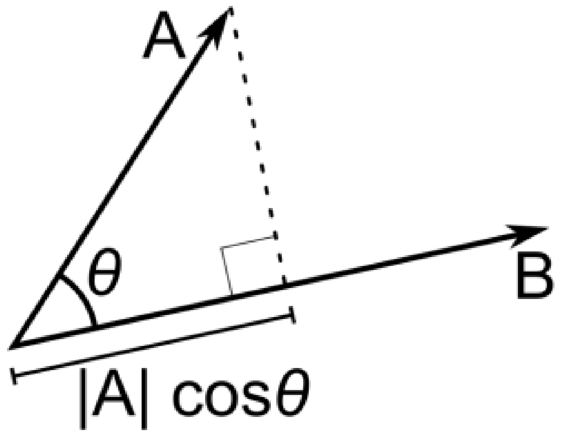
Linear Algebra Basics

Vector operations

- Inner product (dot product)

$$\boldsymbol{x}_i^T \boldsymbol{x}_i = \sum_k^n \left(x_k^i\right)^2 = \text{squared norm of } \boldsymbol{x}_i$$

- x.y is also |x||y|cos(angle between x and y)



- If B is a unit vector, A.B gives projection of A on B

Linear Algebra Basics
Vector operations

- Outer product

$$\boldsymbol{x}_i \boldsymbol{x}_j^T = \begin{bmatrix} x_1^i x_1^j & x_1^i x_2^j & \cdots & x_1^i x_n^j \\ x_2^i x_1^j & x_2^i x_2^j & \cdots & x_2^i x_2^j \\ \vdots & \vdots & \ddots & \vdots \\ x_n^i x_1^j & x_n^i x_2^j & \cdots & x_n^i x_m^j \end{bmatrix} \text{ (a matrix)}$$

Linear Algebra Basics
Matrix

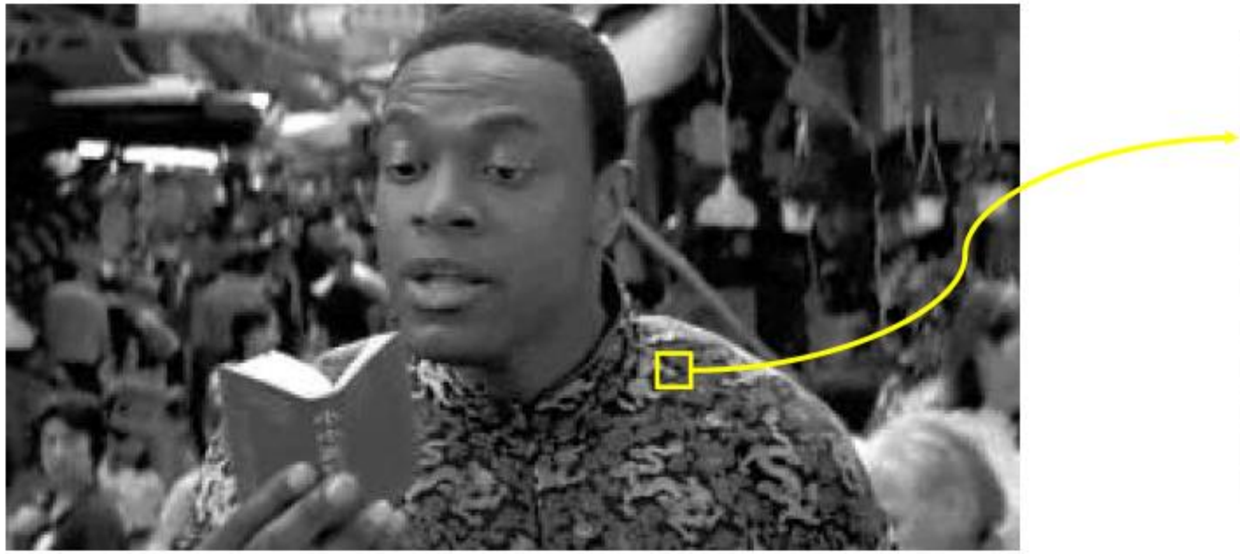
- Array $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ of numbers with shape m by n,
  - m rows and n columns

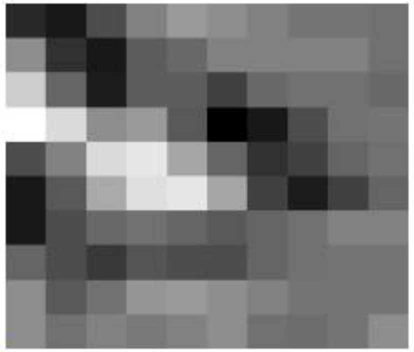
$$\boldsymbol{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}$$

- A row vector is a matrix with single row
- A column vector is a matric with single column

Linear Algebra Basics
Matrix - use

- Image representation – grayscale
  - One number per pixel
  - Stored as nxm matrix



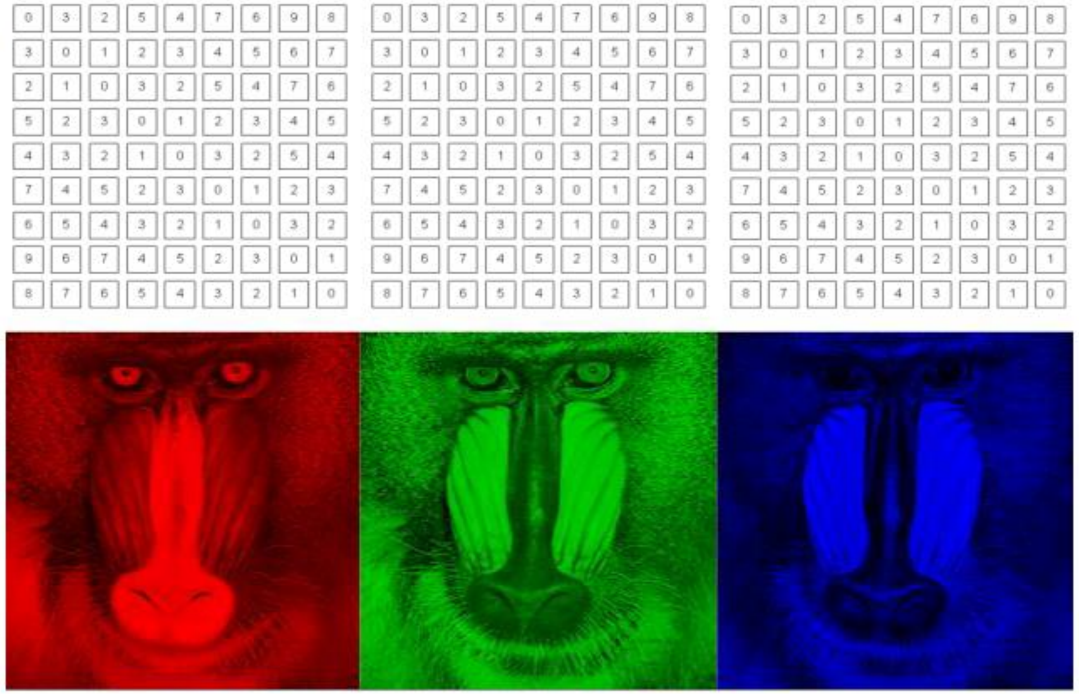


0	3	2	5	4	7	6	9	8
3	0	1	2	3	4	5	6	7
2	1	0	3	2	5	4	7	6
5	2	3	0	1	2	3	4	5
4	3	2	1	0	3	2	5	4
7	4	5	2	3	0	1	2	3
6	5	4	3	2	1	0	3	2
9	6	7	4	5	2	3	0	1
8	7	6	5	4	3	2	1	0

Linear Algebra Basics

Matrix - use

- Image representation – RGB
  - 3 numbers per pixel
  - Stored as nxmx3 matrix



Linear Algebra Basics
Matrix operations

- Addition

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} + \begin{bmatrix} e & f \\ g & h \end{bmatrix} = \begin{bmatrix} a+e & b+f \\ c+g & d+h \end{bmatrix}$$

- Both matrices should have same shape, except with a scalar

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} + 2 = \begin{bmatrix} a+2 & b+2 \\ c+2 & d+2 \end{bmatrix}$$

- Same with subtraction

Linear Algebra Basics

Matrix operations

- Scaling

$$s \text{ x} \begin{bmatrix} a & b \\ c & d \end{bmatrix} = \begin{bmatrix} s\text{x}a & s\text{x}b \\ s\text{x}c & s\text{x}d \end{bmatrix}$$

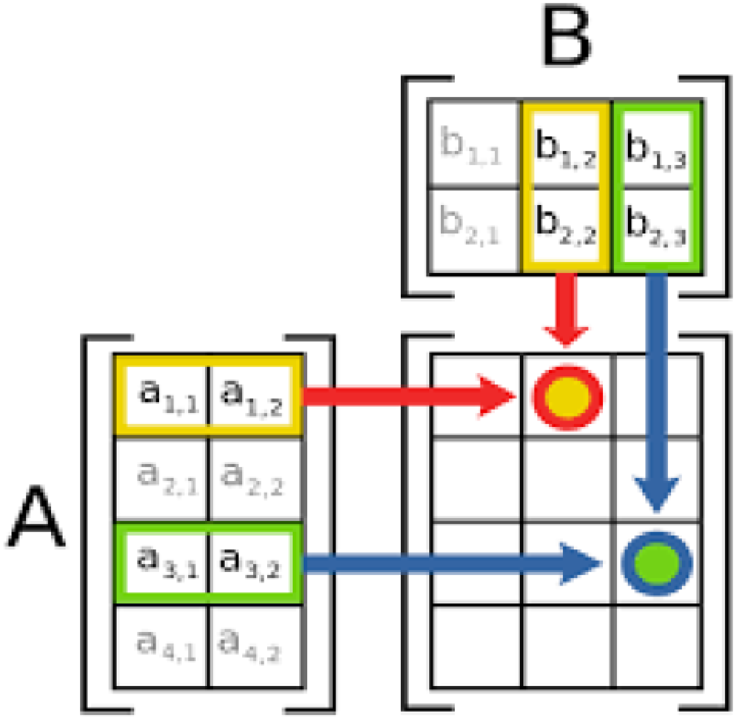
- Hadamard product

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \odot \begin{bmatrix} e & f \\ g & h \end{bmatrix} = \begin{bmatrix} a\text{x}e & b\text{x}f \\ c\text{x}g & d\text{x}h \end{bmatrix}$$

Linear Algebra Basics

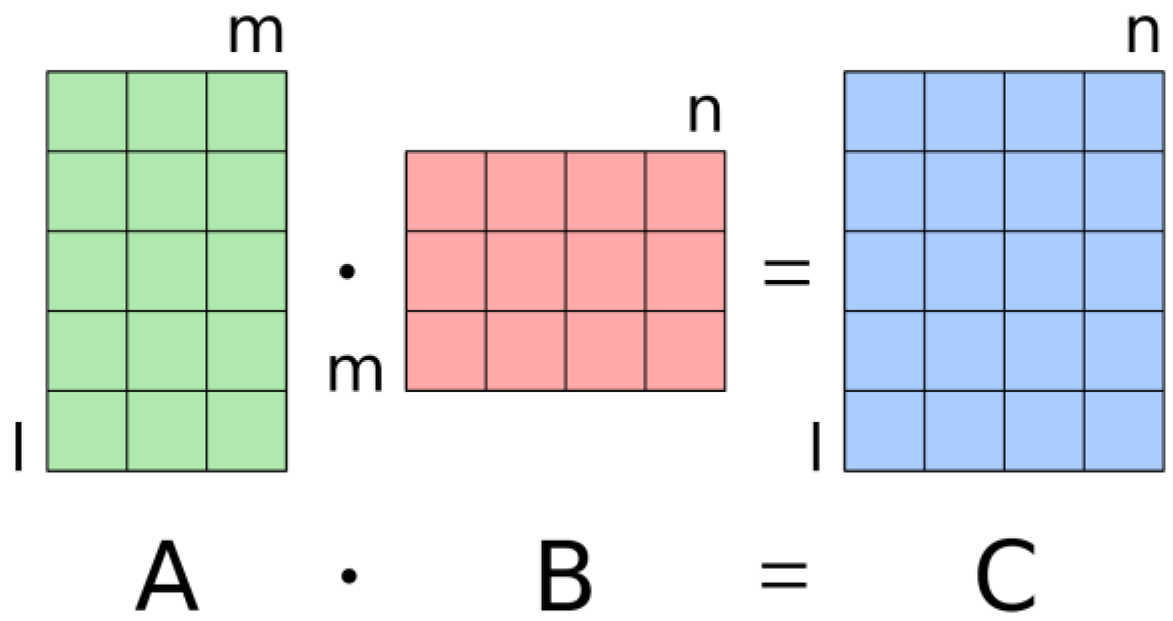
Matrix operation

- Matrix Multiplication
  - Compatibility?
  - mxn and nxp
  - Results in mxp matrix



Linear Algebra Basics

Matrix operation



Linear Algebra Basics

Matrix operation

- Transpose

$$\boldsymbol{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}$$

$$\boldsymbol{A}^T = \begin{bmatrix} a_{11} & a_{21} & \cdots & a_{m1} \\ a_{12} & a_{22} & \cdots & a_{m2} \\ \vdots & \vdots & \ddots & \vdots \\ a_{1n} & a_{2n} & \cdots & a_{mn} \end{bmatrix}$$

Linear Algebra Basics

Matrix operation

- Inverse
  - Given a matrix A, its inverse A^{-1} is a matrix such that

$$\mathbf{AA^{-1} = A^{-1}A = I}$$

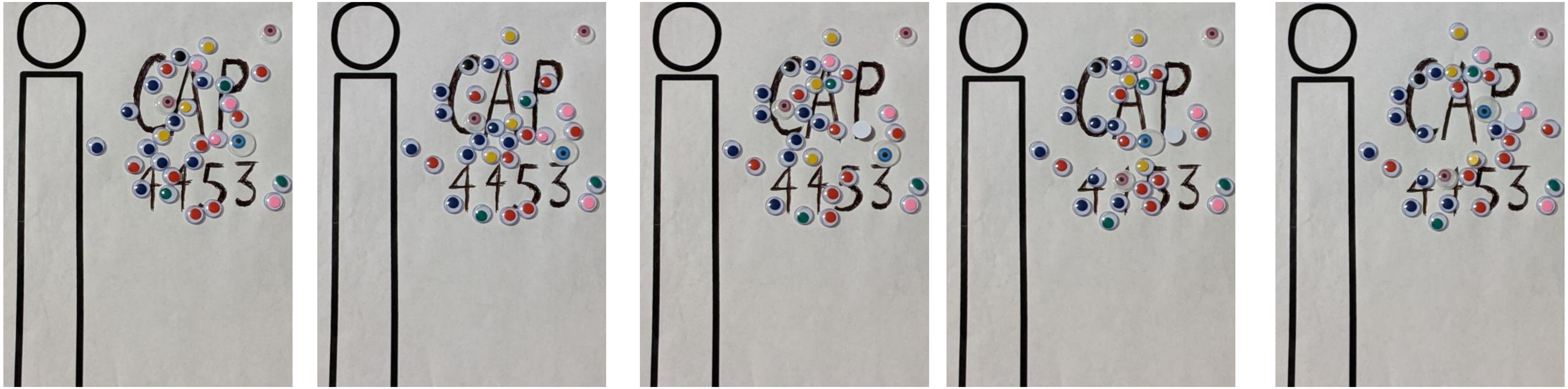
- Inverse does not always exist
  - Singular vs non-singular
- Properties
  - $(A^{-1})^{-1} = A$
  - $(AB)^{-1} = B^{-1}A^{-1}$

Linear Algebra Basics

MORE WILL BE INTRODUCED DURING THE COURSE AS IT IS NEEDED

Question: Noise reduction

- Given a camera and a still scene, how can you reduce noise?

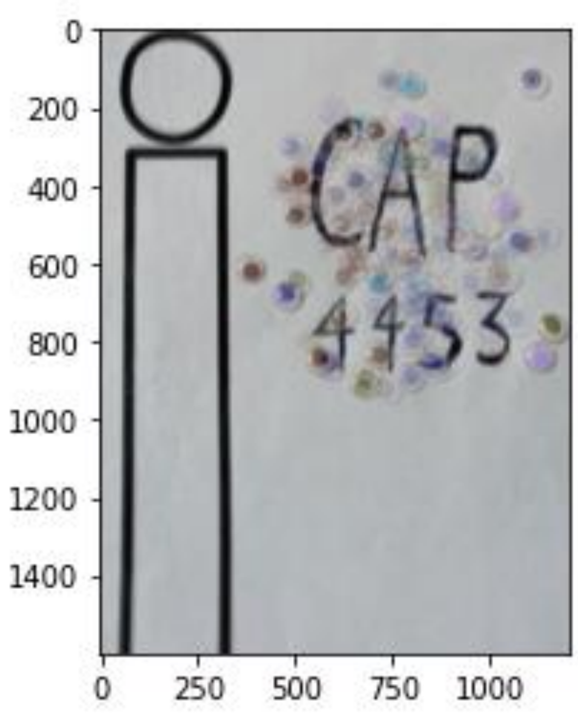


Take lots of images and average them!

Source: S. Seitz

Question: Noise reduction

- Given a camera and a still scene, how can you reduce noise?



Take lots of images and average them!

Can we something else?

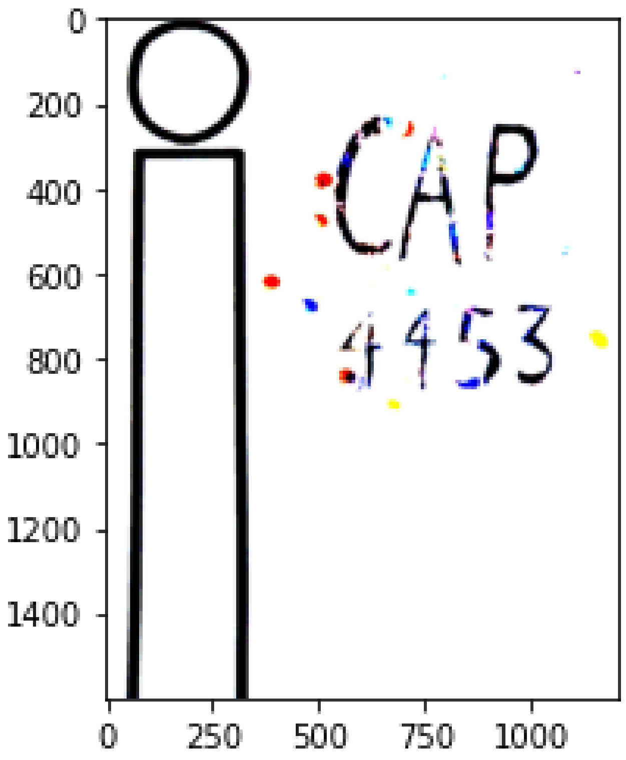
Source: S. Seitz

Thresholding !

$$g(m,n) = \begin{cases} 255, & f(m,n) > A \\ 0 & otherwise \end{cases}$$

Question: Noise reduction

- This is not a gray scale image




```python
import cv2
import os
import numpy as np
import matplotlib.pyplot as plt

folder='C:/Users/gonza/OneDrive/Teaching/CAP4453/class3/'
list_dir = [fil for fil in os.listdir(folder ) if fil[-3:]=='jpg']

for iFile, fname in enumerate(list_dir):
    if iFile == 0:
        sumFile = cv2.imread(folder + fname)
        sumFile = sumFile.astype(np.float)
    else:
        sumFile = sumFile + cv2.imread(folder + fname).astype(np.float)

sumFile = sumFile/len(list_dir)
sumFile[sumFile>90]=255
sumFile[sumFile<=90]=0

plt.imshow(sumFile.astype(np.uint8))
```


Source: S. Seitz