# CSC317 Digital Image Processing

# Class Note One - Basics

## Roadmap

1. Administration – Syllabus, Schedule, Roster (Name)
2. Math Needed for CSC317
3. Basics about Images
4. Installation of Anaconda Python 3.7, OpenCV
5. (Next Week) Python Programming Basics

## I. Administration

We will use the sequence in the outline file to cover the first part of a computer vision course – digital image processing. There are three parts for this course. Part I covers the basics about images and neighborhood processing (pixel-level processing), and geometric transformations. Part II covers the local image processing methods including convolutions, spatial-domain filters, and Fourier transforms & frequency-domain filters. Part III covers image analysis with morphological transformations, gradient operators, Laplacian operators, Hough Line Detection, and Image Segmentations.

Our class includes about 12-13 lectures(modules), three quizzes/exams, three homework assignments with programming problems and computational problems. For each *lecture*, i.e., a *D2L module*, we will begin with about concepts in the area of digital image processing and analysis, followed by python programming method. For the lecture, we will use materials from several resources. The textbook is the main resource for the lectures. Programming will be done with Anaconda Python and OpenCV. If you prefer C++, you may use it to complete all programming problems in the assignment. However, the lecture will only focus on python and OpenCV. When we refer to OpenCV, we are not limited to learning OpenCV. We will also use some other packages such as Numpy, Skimage, etc.

## II. Math Needed

We will be using the following math areas whenever needed. This list is for your reference only and is subject to changes.

1. Mean, Variance, Standard Deviation (Statistics) – concepts only.
2. Derivations (Calculus) Gradients (Calculus)
3. Polar Coordinates
4. Trigonometry
5. Matrix Operations, Determinants, and traces.
6. Linear Algebra: Matrix multiplications
7. Convolution: Discrete and Circular Convolutions
8. Fourier Transform: Complex Numbers, Euler Formula
9. Image Geometry: Scaling, Translation, Rotation, Shearing

## III. Basics about Images and Computer Vision

* Computer Vision integrates techniques from (EE) Digital Signal Processing, (MATH) Computer Geometry, (CS) Computer Graphics, and (CS) Machine Learning.
* In CSC317, we will focus on Digital Image Processing (Histogram, Convolution, and Gradient) as well as basics about Image Geometry (Scaling, Translation, Rotation, and Shearing), Morphology (Dilations, Erosions, Opening, and Closing), Segmentation and Line Detection.

(Section Numbers and contents are taken from the textbook.)

### 1.1 Images and Pictures

For our purpose, an image is a picture with foreground and background. Foreground is representing objects, when background represents the environment.

### 1.2 What is Image Processing?

## Image Processing involves changing the nature of an image for the following purposes:

1. Improve its pictorial information for human interpretation
2. Render it more suitable for autonomous machine perception

We will use a computer to change the nature of digital images. Here are some examples:

The left one is an original image of an alley with two pedestrian; the right one is a sharpened image.

Figure 1.1 Sharpening an image

The left one is a building with sale-and-pepper noises; the right one has dee-noised.

Figure 1.2 Remove noises

The left car has a fuzzy license plate; the right one has a clear license plate.

Figure 1.3 Removing the blur on the license plate

The left one has bricks and cylinders stacked together; the right image only shows the edges of these objects.

Figure 1.4 Display edges of bricks and cylinders

## THe left image is a buffalo; the right one uses blurring to remove the detail.

Figure 1.5 Blurring to remove details

### 1.3 Image Acquisition and Sampling - Issues

* CCD (Charged-Coupled Device) – Camera (will be discussed later)
* Flatbed Scanner – CCD Scanner
* X-Ray or MRI – Energy Sources

We will look into the CCD Camera model next week. We will discuss about Sampling today. Sampling refers to the process of digitizing a continuous function. For example, we sample the values of y for a range of given x values using the function:

y = sin(x) + (1/3)sin(3x)

If we take not enough samples, the shape of the function may be distorted. We call the situation as *under-sampling*. We can use R programming IDE R-Studio to draw the curve:

curve(sin(x)+(1/3)\*sin(3\*x), from =0, to=10, n=20)

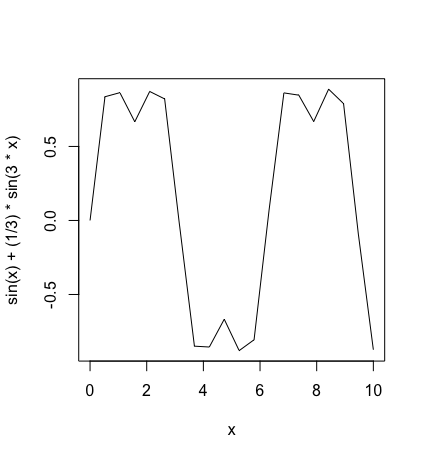


Fig. 1.6 Under Sampling (n=20)

Increased sample size (n=200)

Description automatically generated

Fig. 1.7 Increased sampling size

To improve the result of sampling, we need to increase the sampling size to 200 samples:

curve(sin(x)+(1/3)\*sin(3\*x), from =0, to=10, n=200)

Q: What is the difference between two sampled images?

A: The file size is enlarged. We will talk about CCD Camera next week.

## 1.4 Images and Digital Images

### Terms about Image Basics

* **Image *I***: A (digital) image is defined by integrating and sampling continuous (analog) data in a spatial domain. Each image consists of a rectangular array of pixels that are the “atomic elements” of an image.
* **Pixel**: A pixel is a triplet *(x, y, u)* in a 2D domain, each combining a *location* *(x, y)* ∈ Z2 and a value *u*, the *sample* at the pixel location *(x, y)*. We can represent the sampling at a pixel location as a function *p* *that maps the pixel location (x,y) to an integer u,* *p(x,y) = u.*
* **Carrier *Ω***: The rectangular set of pixel locations

***Ω*****= {*(x, y)* : 1 ≤ *x* ≤ *Ncols* ∧ 1 ≤ *y* ≤ *Nrows* } ⊂ Z2**

of *I* containing the *grid points or pixel locations for Ncols ≥ 1 and Nrows ≥ 1.*

* **Window *Wm,n\_p*(*I*):** A Window*Wm,n\_p (I)* is a sub-image I of size m x n positioned with respect to a reference point p (i.e., a pixel location). The default is that m = n is an odd number, and p is the central location in the window.
* **Channel:** one dimension of the pixel values. A *vector-valued image* has more than one channel or band, as it is the case for scalar images. Image values *(u1, . . . , uNchannels* ) are vectors of length *Nchannels*. For example, color images in the common RGB color

model have three channels, one for the red component, one for the green, and one for

the blue component. The values *ui* in each channel are in the set {*0, 1, . . . , Gmax*};

each channel is just a gray-level image.

A *digital image* differs from a *photo* in that the x, y, and f(x, y) (for the grayscale value) are all discrete. *Black is represented by 255, and white is represented by 0*. Also, in OpenCV, *white* objects are always considered *foreground*, and *black* pixels are referred to as *background*. In other words, a digital image in OpenCV consists of white objects on a black background.

### 1.5 Some Applications

1. Medicine
2. Agriculture
3. Industry
4. Law enforcement

### 1.6 Image Processing Operations

1. Image enhancement
2. Image restoration
3. Image segmentation
4. Image registration

### 1.7 Tasks or Design Issues about Image Processing/Computer Vision

1. Image Acquisition – Part I of this course
2. Image Preprocessing(Pixel, Local, and Global) – Part II of this course
3. Image Analysis (Segmentation) – Part III of this course
4. Recognition and Interpretation – future Computer Vision course.

### 1.8 Types of Images

* **Scalar image:** For each pixel, p(x,y) = v, where v: 0 to 255. Note, **white = 255, black = 0.**
* **Binary image:** For each pixel, p(x,y) = black or white. Traditionally, black is used for the foreground and white is used for the background. So, p(x, y) = 1 if black; 0 if white. However, *OpenCV always uses white for foreground, and black for background.* The concept will be critical when we discuss about mathematical morphology. Some textbook still uses 1 for white and 0 for black.
* **Vector-Valued or RGB image:** For each pixel, there are three channels, i.e., Red, Green, and Blue. A vector-valued image has more than one *channel* or *band*, as it is the case for scalar images. For example, color images in the common RGB color model has three channels, i.e., Red, Green, and Blue channels.

## 1.9 Image File Size

* Binary Image: 512 x 512 x 1 = 262,144 bits = 32,768 bytes = 0.033 MB
* Gray Scale: 512 x 512 x 1 = 262,144 bytes = 0.262 MB
* Color: 512 x 512 x 3 = 0.786 MB

## IV. Installations of Python and OpenCV

1. Anaconda Python version 3
2. OpenCV

Instead of downloading python from the official website of Python language, we are going to use [Anaconda Distribution](http://www.anaconda.com/download) for Python and OpenCV. The anaconda distribution has packaged OpenCV in its Python packages 2.7 and 3.7. We will use python 3.7 for our labs.

For Windows users, please refer to [Installation for Windows](https://docs.anaconda.com/anaconda/install/windows/).

For Mac OS X users, please refer to [Installation for Mac OS X](https://docs.anaconda.com/anaconda/install/mac-os/).

Complete the installation following the steps below:

1. Download [python 3.7](https://www.anaconda.com/download/) after bringing up the webpage at and clicking the button representing python 3.7.

A screenshot of a social media post

Description automatically generated

1. You will be enticed to download the “cheatsheet” for Anaconda. It is up to you if you want it.
2. After clicking the “Download 3.7” button, you will then be brought to the webpage as follows:

A screenshot of a cell phone

Description automatically generated

1. The download will begin in a moment.
2. After the download is completed, the screen for installation will be displayed.
3. On Mac, there is a warning message to be displayed as follows:

A screenshot of a cell phone

Description automatically generated

1. Then the installation begins:

A screenshot of a cell phone

Description automatically generated

You only need to click the “continue” button to complete the installation procedure.

1. The next step is to install OpenCV. For Windows users and Mac OS X users, the procedures are somewhat different.
2. (Windows 10) Once the installation is completed, you may use Anaconda Navigator to install OpenCV, or a GUI interface provided by the Anaconda Navigator to install opencv-python module.
3. If you use the Anaconda Navigator, you can bring it up first. Then you will see this window:

A screenshot of a cell phone

Description automatically generated

1. (Windows 10) If you bring up the Anaconda Prompt. Under Windows 10, you can click the “Start” > “Anaconda Prompt”. Then, you can enter the command “>conda install opencv” to install OpenCV.
2. You may click the “Environments” button on the left of the window, and bring up the window for installing OpenCV:

A screenshot of a cell phone

Description automatically generated

1. (Mac OS X) You need to choose an environment, e.g., base(root), and then click the “installed” dropdown button to choose “uninstalled”. Then find the modules libopencv, opencv, py-opencv to install the OpenCV modules for Python. For Windows 10 users, you may not be able to complete the installation of OpenCV this way.
2. (Windows 10) You want to bring up an Anaconda Command Prompt window and enter the command “conda install python=3.7 anaconda=custom”. After this, you can check your python version by entering a command “python –version”. It should show 3.7.
3. You can go back to step 12 above and install OpenCV-related packages.
4. Once the installation is completed, you can verify the installation using the command via the iPython console “import cv2”. If you do not see an
5. As of Fall 2022, we need to roll back Python 3.9 to Python 3.7 for installing OpenCV 2 in Anaconda Python. The command to use is “conda install python=3.7”. Without rolling back to Python 3.7, OpenCV cannot be installed successfully.
6. On MAC OS X, the Spyder may need to roll back to version 3.3.0. Otherwise, the OpenCV functions imshow() may freeze after displaying an image. You may click the “version” setting icon (that looks like a wheel) on the top right-hand corner of the Spyder icon in the Anaconda Navigator and choose the version 3.3.0. If your environment works without rolling back Spyer, you don’t have to.

## V. Next Week

1. Section 1.3 Image Acquisition with CCD Camera
2. Python Programming