

CSC317 Visual Computing

Class Note One - Basics

Roadmap

- I. Administration – Syllabus, Schedule, Roster (Name)
- II. Math Needed for CSC317
- III. Basics about Images
- IV. Installation of Anaconda Python 3.6, OpenCV
- V. Python Programming Basics

Administration

Please sign the roster sheet – with your name only.

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). Part II covers the image transformations including geometric transformations, morphological transformations, and local processing with convolutions. Part III covers image analysis with gradient including Fourier transforms, Hough Line Detection, and Image Segmentations.

Our class includes about 12 lectures, three paper-based exams (with essay questions), three homework assignments with programming problems and computational problems. For each lecture, we will begin with about one-hour of concepts in the area of digital image processing and analysis, followed by about another one-hour of programming methods. For the lecture, we will use materials from several resources. The textbook is the main resource for the lectures. But it is not enough for some subjects for which hand-outs will be provided. 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.

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 - representing a line with " $x \cos\theta + y \sin\theta = r$ " in (r, θ) domain
4. Trigonometry - $\cos(x + y) = \cos x \cos y - \sin x \sin y$; $\sin(x + y) = \sin x \cos y + \cos x \sin y$;
5. Matrix Operations, Determinants, and traces.
6. Linear Algebra: Eigenvalues, Eigenvectors
7. Convolution: Discrete and Circular Convolutions
8. Fourier Transform: Complex Numbers ($i = \sqrt{-1}$, $i^2 = -1$, $i^3 = -i$, $i^4 = 1$, what is i^{135} ?)
9. Euler Formula: $e^{i\theta} = \cos \theta + i \sin \theta$

10. Image Geometry: Scaling, Translation, Rotation, Shearing (matrix operations and trigonometry)

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) with basics about Image Geometry (Scaling, Translation, Rotation), Morphology (Dilations, Erosions, Opening, and Closing), Segmentation and Line Detection.

Tasks or Design Issues about Image Processing/Computer Vision

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

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:



(a) The original image



(b) Result after “sharpening”



(a) The original image



(b) After removing noise



(a) The original image



(b) After removing the blur



(a) The original image



(b) Its edge image



(a) The original image



(b) Blurring to remove detail

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)
```

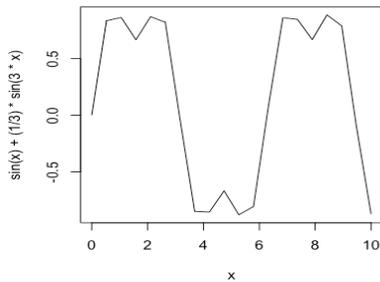


Fig. 1.6

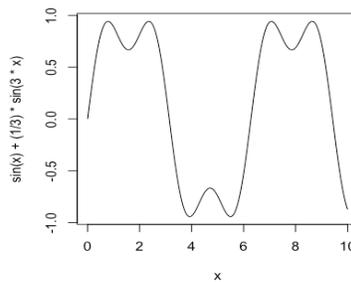


Fig. 1.7

To improve the result of sampling, we need to increase the sampling size:
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.
Today, we will just jump ahead to Section 1.9 of our textbook.

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* $(x, y) \in \mathbb{Z}^2$ 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

$$\Omega = \{(x, y) : 1 \leq x \leq N_{cols} \wedge 1 \leq y \leq N_{rows}\} \subset \mathbb{Z}^2$$

of I containing the *grid points or pixel locations* for $N_{cols} \geq 1$ and $N_{rows} \geq 1$.

- **Window $W_{m,n,p}(I)$:** A Window $W_{m,n,p}(I)$ is a sub-image I of size $m \times n$ 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 $(u_1, \dots, u_{N_{channels}})$ are vectors of length $N_{channels}$. 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 u_i in each channel are in the set $\{0, 1, \dots, G_{max}\}$; each channel is just a grey-level image.

Types of Images

- **Scalar image:** For each pixel, $p(x, y) = v$, where $v: 0..255$. Note, **white = 255, black = 0**.
- **Binary image:** For each pixel, $p(x, y) = \text{black or white}$. 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: $512 \times 512 \times 1 = 262,144$ bits = $32,768$ bytes = 0.033 MB
- Grey Scale: $512 \times 512 \times 1 = 262,144$ bytes = 0.262 MB
- Color: $512 \times 512 \times 3 = 0.786$ MB

Installations of Python and OpenCV

1. Anaconda Python 3.6
2. OpenCV

Next Week

1. (1.3) Image Acquisition with CCD Camera
2. Python Programming