Introduction to Signal and Image Processing

Dr Pradipta Biswas, PhD (Cantab)
Assistant Professor
Indian Institute of Science
https://cambum.net/

Signal Processing

- Humans are the most advanced signal processors
  - speech and pattern recognition, speech synthesis,…
- We encounter many types of signals in various applications
  - Electrical signals: voltage, current, magnetic and electric fields,…
  - Mechanical signals: velocity, force, displacement,…
  - Acoustic signals: sound, vibration,…
  - Other signals: pressure, temperature,…
- Most real-world signals are analog
  - They are continuous in time and amplitude
  - Convert to voltage or currents using sensors and transducers
- Analog circuits process these signals using
  - Resistors, Capacitors, Inductors, Amplifiers,…
- Analog signal processing examples
  - Audio processing in FM radios
  - Video processing in traditional TV sets

Why we need in research

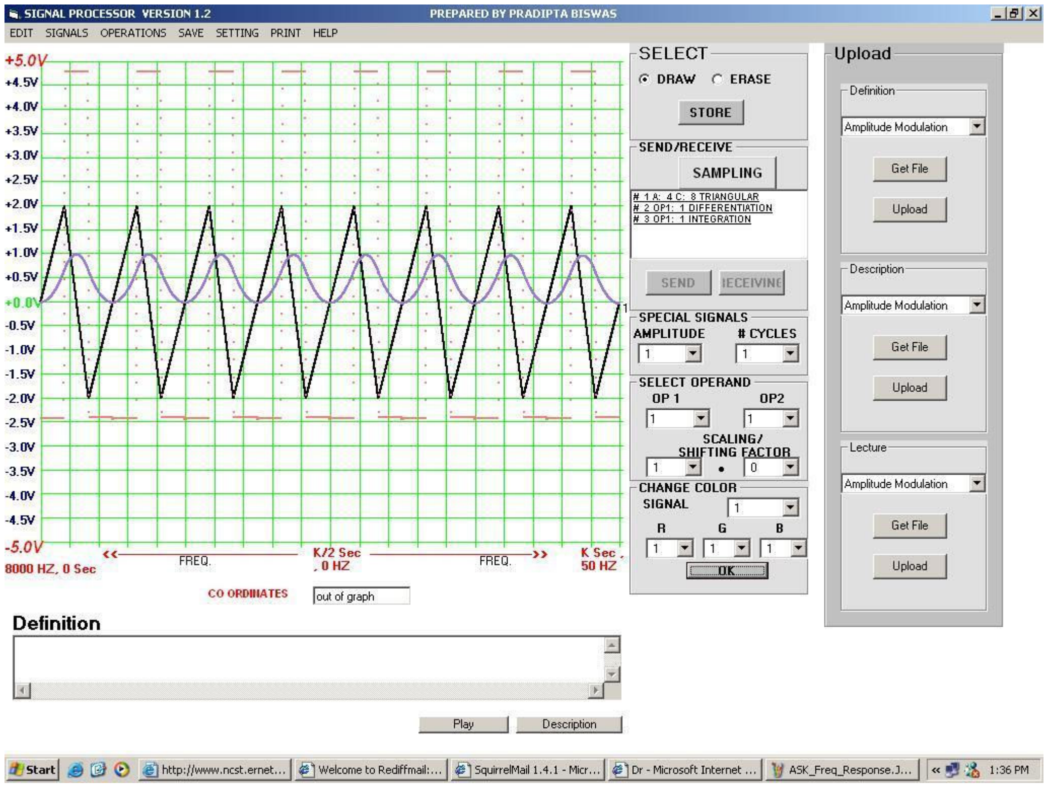
- We measure parameters for a time duration
  - We can consider it a signal f(t)
- We work with images or videos
  - We can consider it a signal f(x,y) or f(x,y,t)
- We need to find
  - Correlation
  - Outlier
  - Sudden change in signal
  - Repeating pattern and so on

Contents

- Analog and Digital Signal
- Basic Signal Operations
- Digital Signal Processing
- Time and Frequency Domain Representations
- Filtering Techniques
- Masking

Basic Signal Operations

- Arithmetic operations
- Differentiation / Integration
- Time shifting / scaling

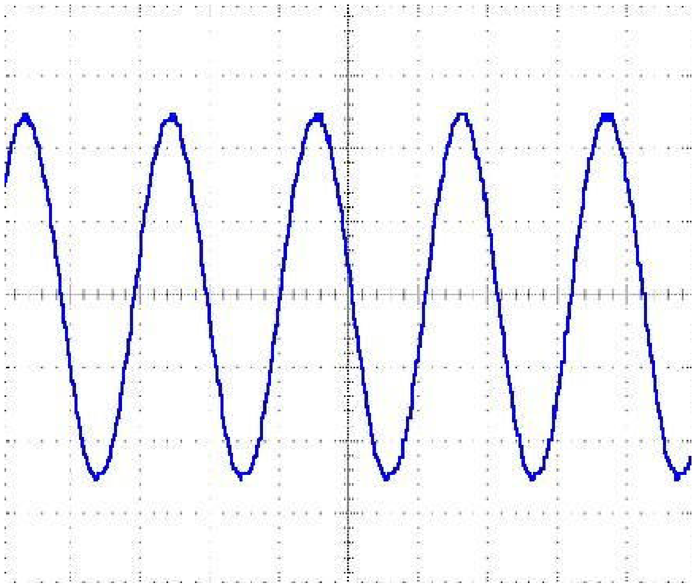


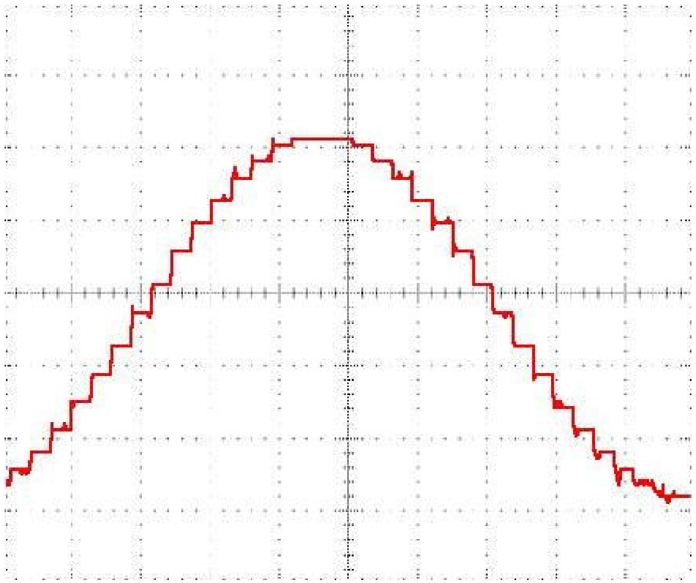
Limitations of Analog Signal Processing

- Accuracy limitations due to
  - Component tolerances
  - Undesired nonlinearities
- Limited repeatability due to
  - Tolerances
  - Changes in environmental conditions
    - Temperature
    - Vibration
- Sensitivity to electrical noise
- Limited dynamic range for voltage and currents
- Inflexibility to changes
- Difficulty of implementing certain operations
  - Nonlinear operations
  - Time-varying operations
- Difficulty of storing information

Digital Signal Processing

- Converting a continuously changing waveform (analog) into a series of discrete levels (digital)

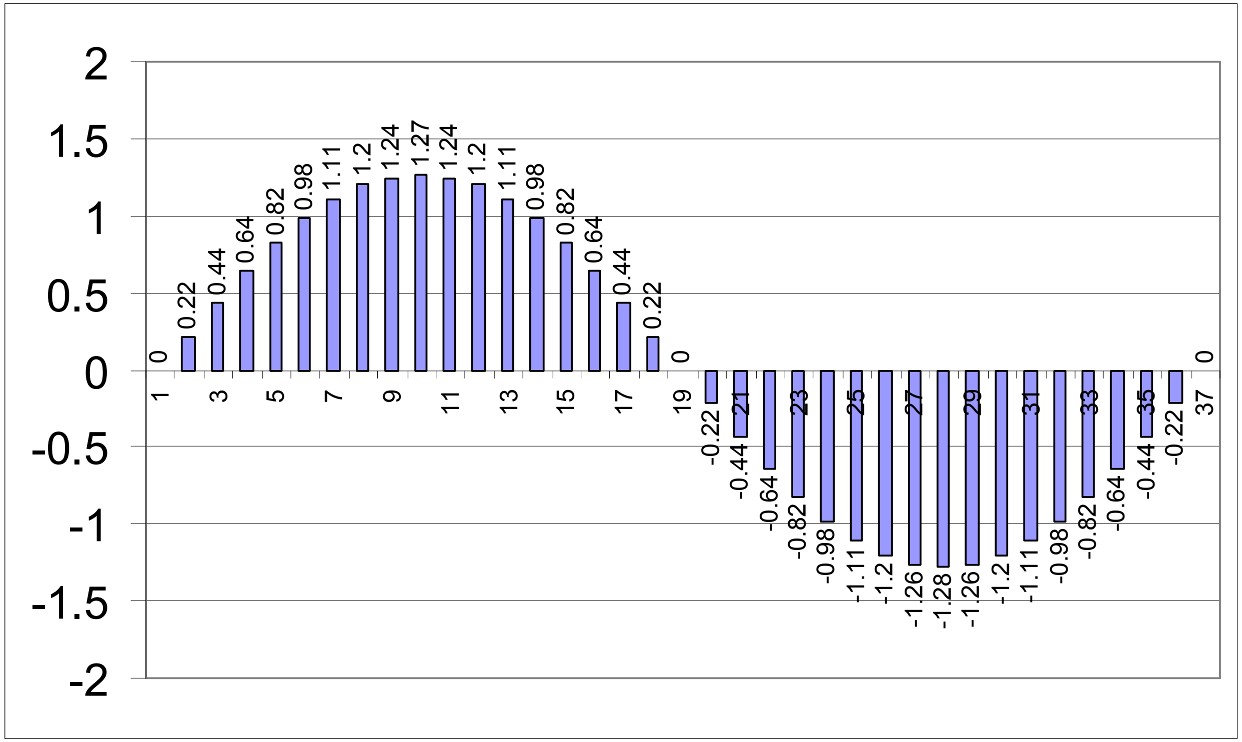




What is DSP?

- The analog waveform is sliced into equal segments and the waveform amplitude is measured in the middle of each segment

- The collection of measurements make up the digital representation of the waveform



Sampling

- Sampling is a process of converting a signal (for example, a function of continuous time and/or space) into a numeric sequence (a function of discrete time and/or space)

- If a function x(t) contains no frequencies higher than B hertz, it is completely determined by giving its ordinates at a series of points spaced 1/(2B) seconds apart

- A sufficient sample-rate is therefore 2B samples/second, or anything larger. Equivalently, for a given sample rate fs, perfect reconstruction is guaranteed possible for a bandlimit B < fs/2

- The two thresholds, 2B and fs/2 are respectively called the Nyquist rate and Nyquist frequency

Aliasing

- Aliasing is an effect that causes different signals to become indistinguishable (or aliases of one another) when sampled

- If the sampling rate is B Hz, any frequency over B/2 HZ in the original signal will create aliasing effect

- Aliasing causes
  - –Distortion in audio signal
  - –Moiré patterns in digital image
  - –Wagon wheel effect in video

A moiré pattern, formed by two sets of parallel lines, one set inclined at an angle of 5° to the other

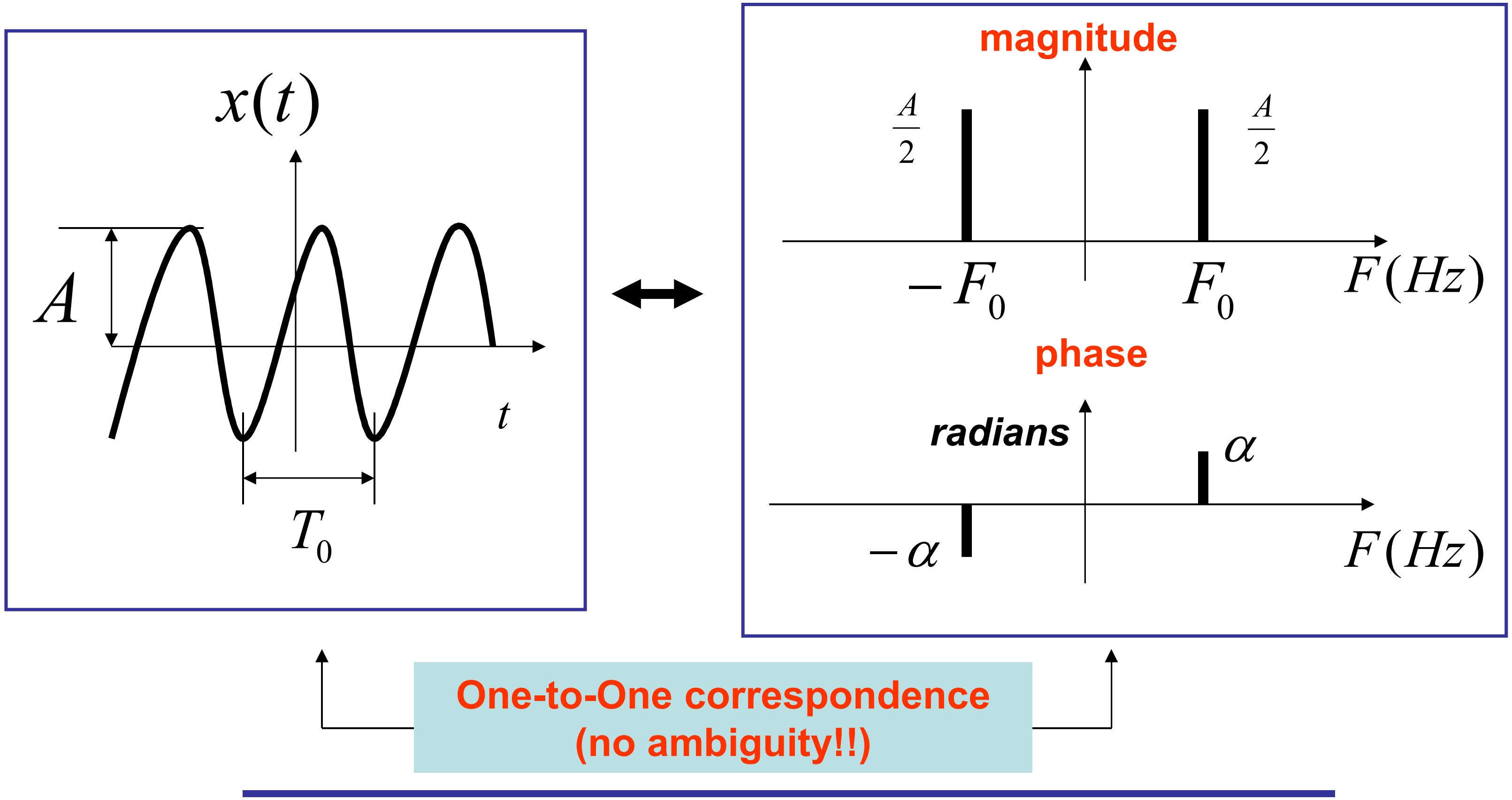
The fine lines that make up the sky in this image create moiré patterns when shown at some resolutions for the same reason that photographs of televisions exhibit moiré patterns: the lines are not absolutely level.

Resolution Trade-offs

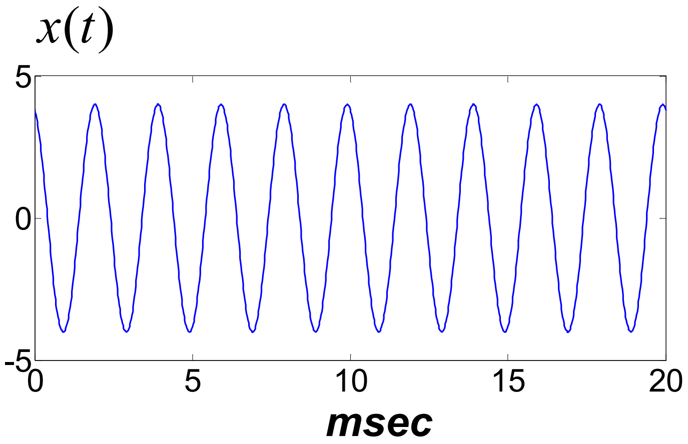
Bit Resolution	High Bit Count	Good Duplication	Slow
	Low Bit Count	Poor Duplication	Fast
Sample Rate	High Sample Rate	Good Duplication	Slow
	Low Sample Rate	Poor Duplication	Fast

Continuous Time and Frequency Domain

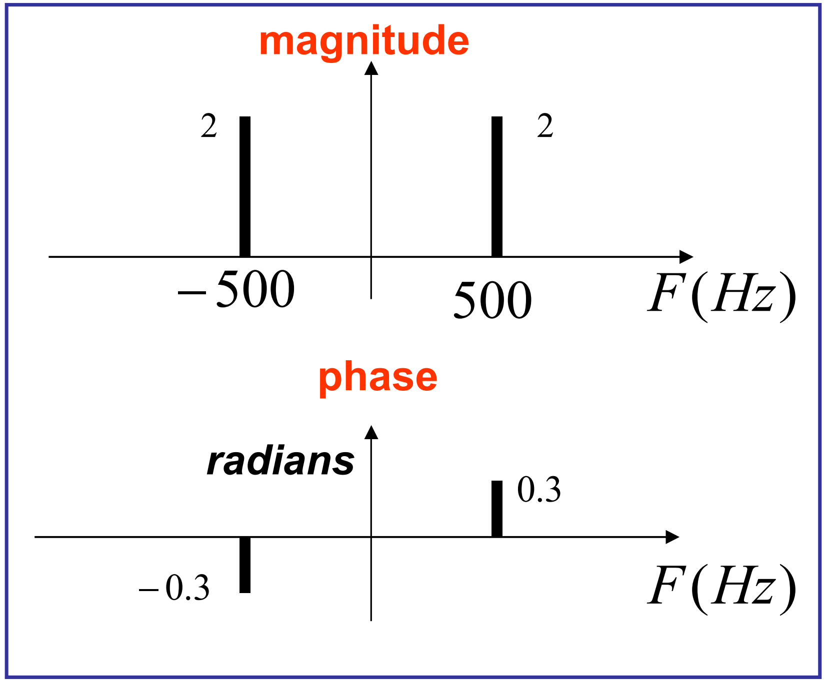
In continuous time, there is a one to one correspondence between a sinusoid and its frequency domain representation:



Example



Given this sinusoid, its frequency, amplitude and phase are unique



Jean Baptiste Joseph Fourier

Fourier was born in Auxerre,

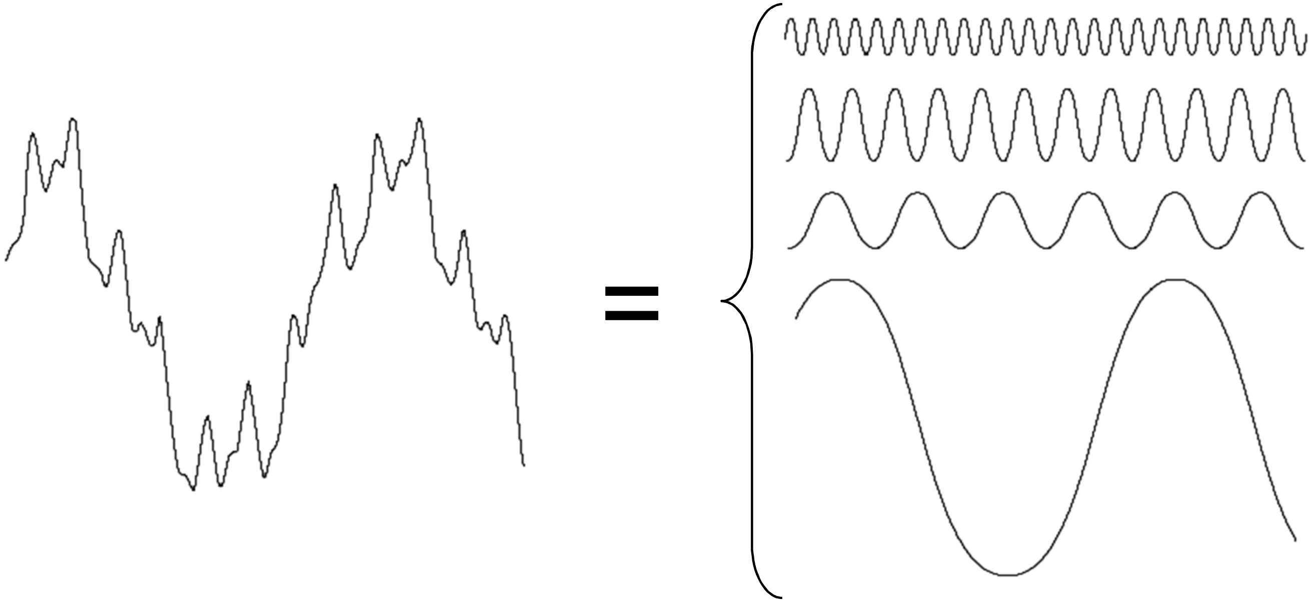
France in 1768

- Most famous for his work "*La Théorie Analitique de la Chaleur*" published in 1822
- Translated into English in 1878: "*The Analytic Theory of Heat*"

Nobody paid much attention when the work was first published

One of the most important mathematical theories in modern engineering

The Big Idea



Any function that periodically repeats itself can be expressed as a sum of sines and cosines of different frequencies each multiplied by a different coefficient – a *Fourier series*

The Discrete Fourier Transform (DFT)

The *Discrete Fourier Transform* of $f(x, y)$, for x = 0, 1, 2…M-1 and y = 0,1,2…N-1, denoted by $F(u, v)$, is given by the equation:

$$F(u,v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) e^{-j2\pi(ux/M + vy/N)}$$

for u = 0, 1, 2…M-1 and v = 0, 1, 2…N-1.

The Inverse DFT

It is really important to note that the Fourier transform is completely **reversible**

The inverse DFT is given by:

$$f(x, y) = \frac{1}{MN} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u, v) e^{j2\pi(ux/M + vy/N)}$$

for x = 0, 1, 2…M-1 and y = 0, 1, 2…N-1

Laplace Transform

The Laplace Transform of a function, f(t), is defined as;

$$L[f(t)] = F(s) = \int_0^\infty f(t) e^{-st} dt$$

The Inverse Laplace Transform is defined by

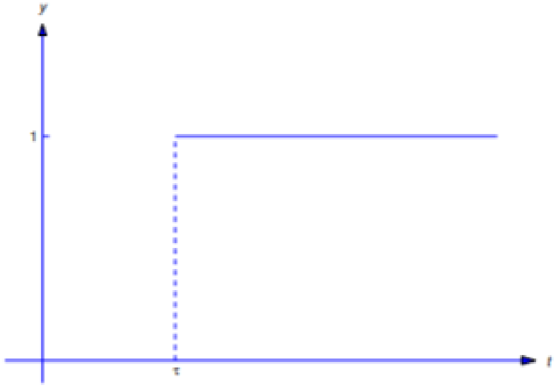
$$L^{-1}[F(s)] = f(t) = \frac{1}{2\pi j} \int_{\sigma - j\infty}^{\sigma + j\infty} F(s) e^{ts} ds$$

Laplace Transform – Unit Step

Laplace Transform of the unit step.

$$u(t) = \begin{cases} 0, & t < 0 \\ 1, & t \geq 0 \end{cases}$$

$$L[u(t)] = \int_0^{\infty} 1 e^{-st} dt = \frac{-1}{s} e^{-st} \Big|_0^{\infty}$$



$$L[u(t)] = \frac{1}{s}$$

The Laplace Transform of a unit step is: $\frac{1}{s}$

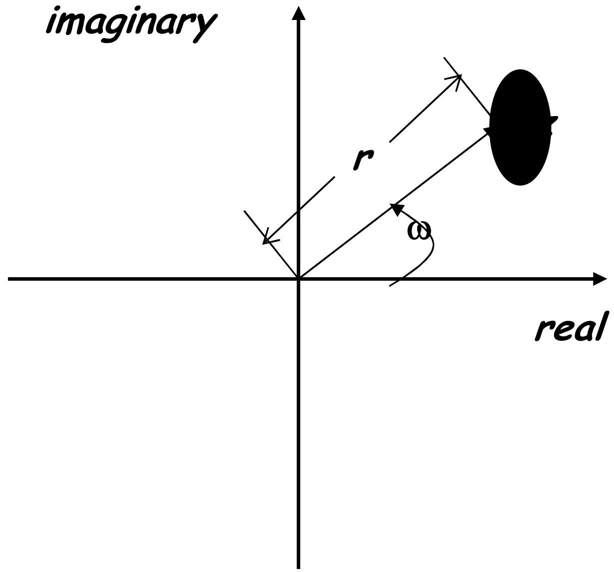
More Examples: https://math.libretexts.org/Bookshelves/Differential_Equations/Book%3A_Elementary_Differential_Equations_with_Boundary_Value_Problems_(Trench)/08%3A_Laplace_Transforms/8.04%3A_The_Unit_Step_Function

Z-Transform

- Discrete-time signals

- Definition:

$$X(z) = \sum_{n=-\infty}^{\infty} x(n) z^{-n}$$



  – z is a complex variable:

$$z = re^{j\omega}$$
$$= r\cos\omega + jr\sin\omega$$

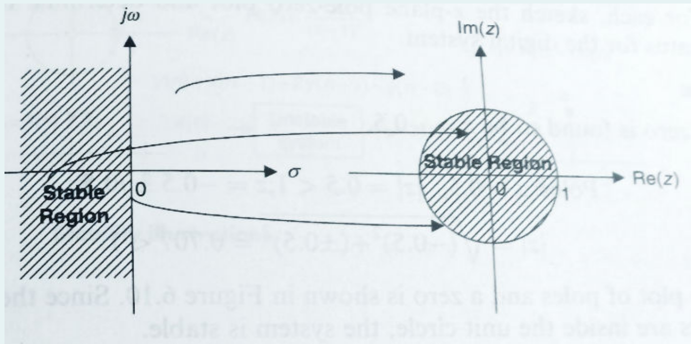
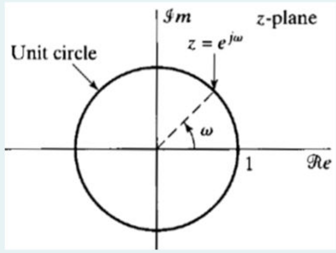
Z-transform

- What is z^{-n} or z^{n}?

$$z^{-n} = r^{-n} e^{-j\omega n} = \underbrace{r^{-n} \cos \omega n}_{\text{real part}} - \underbrace{j r^{-n} \sin \omega n}_{\text{imaginary part}}$$

$$z^{n} = r^{n} e^{j\omega n} = \underbrace{r^{n} \cos \omega n}_{\text{real part}} + \underbrace{j r^{n} \sin \omega n}_{\text{imaginary part}}$$

- rate of decay (or growth) is determined by *r*
- frequency of oscillation is determined by ω

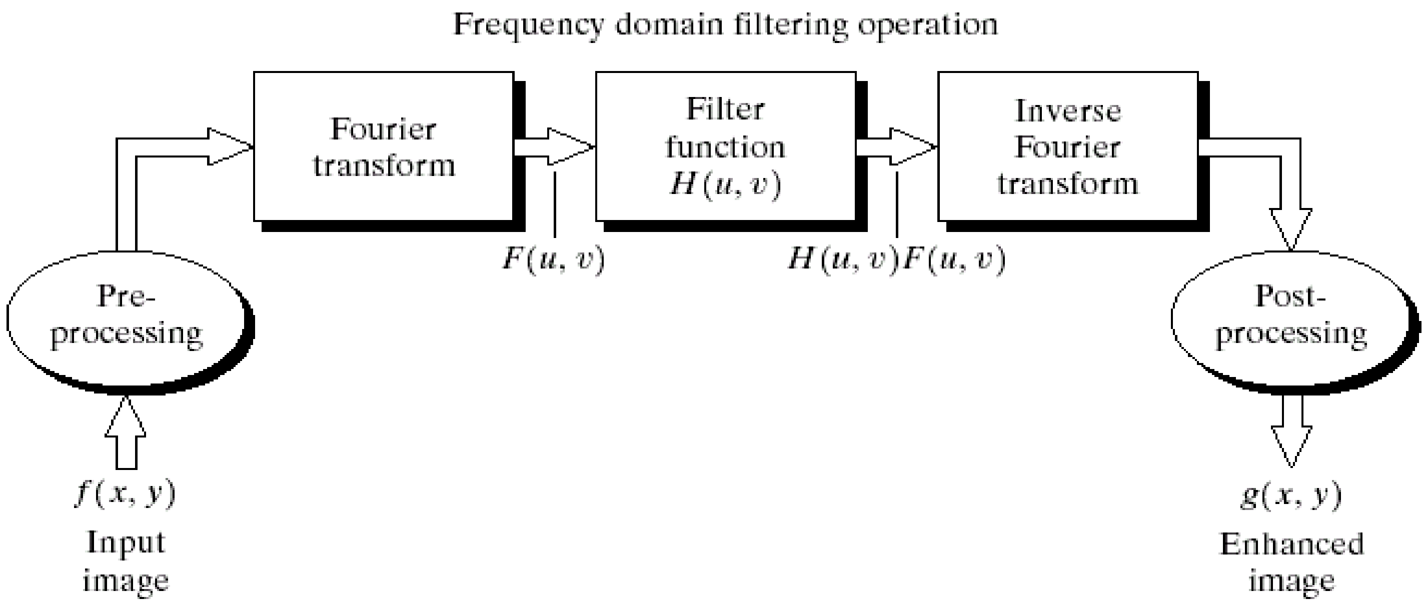
Laplace Transform	Z – Transform	Fourier Transform
Applicable for continuous time system analysis • Used to solve differential equation. • Stability of the system can be determined by the location of poles in the s-plane. (A system is stable if all poles lie to the left of s-plane).  Mapping between s-plane and z-plane. FT exists in the stable region. Can be defined for both stable and unstable systems.	For describing and analyzing digital systems • Used to solve difference equations • Pole-zero plot provides a graphical tool to investigate the characteristics of a digital system (eg. Digital filter).  Location of poles in Z-plane determines the stability of a system (stable if poles are inside the unit circle amplitude). • Used to analyze the transient and steady state responses of a LTI system. • Can be defined for both stable and unstable systems.	Gives a relationship between time domain and frequency domain representation of a signal. Few application areas are: • Used for amplitude and power spectrum analysis • Used for signal frequency analysis • Used for signal processing, digital filter design. • Can be defined only for stable system

Laplace Transform (LT)	Z – Transform	Fourier Transform (FT)
Represents continuous time domain signal in s-domain. Used to investigate the properties of a continuous time linear system.	Counter part of Laplace transform for discrete time signals. Used for analysis of discrete time signals.	Provides frequency domain representation of a time domain signal. Both Laplace and Z transforms have similarity relations with FT.
2 sided Laplace transform: $$X(s) = \int_{t=-\infty}^{\infty} x(t)e^{-st}dt$$ Inverse LT: $$x(t) = (-\frac{j}{2\pi})\int_{\sigma-j\omega}^{\sigma+j\omega} X(s)e^{st}d\omega$$	$$X(Z) = \sum_{n=-\infty}^{\infty} x(n)z^{-n}$$ Z transform exists only for those values of z for which the series converges. This is represented by ROC. Inverse Z transform: $$x(n) = (1/2\pi j)\oint X(z)z^{n-1}dz$$	For continuous time signals, $$X(\omega) = \int_{t=-\infty}^{\infty} x(t)e^{-j\omega t}dt$$ Its inverse FT is $$x(t) = (\frac{1}{2\pi})\int_{\omega=-\pi}^{\pi} X(\omega)e^{j\omega t}d\omega$$ (ω is frequency in radians/s and x(t) is a time domain signal) For Discrete signals, Analysis equations is $X(k) = \sum_{k=0}^{N-1} x(n)e^{-j2\pi kn}$ Synthesis equation is $x(n) = (1/N)\sum_{k=0}^{N-1} X(k)e^{j2\pi kn}$ (N is the number of samples in the period)
s = $\sigma + j\omega$; If s in Laplace transform is only imaginary (σ=0), LT and FT are both same.	Z = $re^{j\Omega}$. If r = 1, Z transform is same a discrete FT.	Cannot manage unstable systems. For FT to exists, the signal should be absolutely summable. FT is a special case of LT (s=jw) and Z transform (r=1).

The DFT and Image Processing

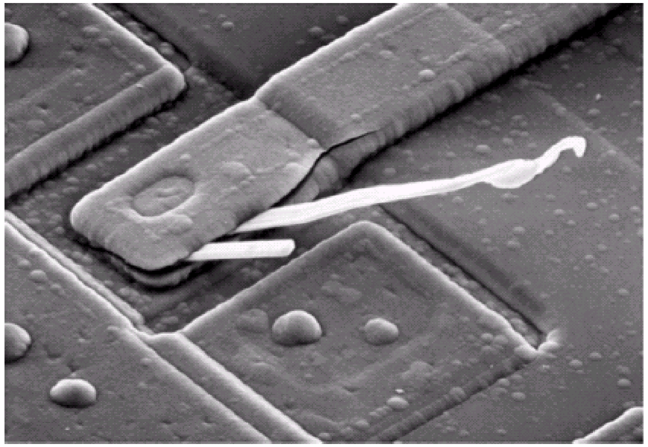
To filter an image in the frequency domain:

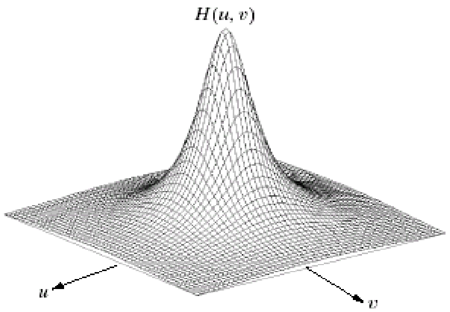
1. Compute $F(u,v)$ the DFT of the image
2. Multiply $F(u,v)$ by a filter function $H(u,v)$
3. Compute the inverse DFT of the result

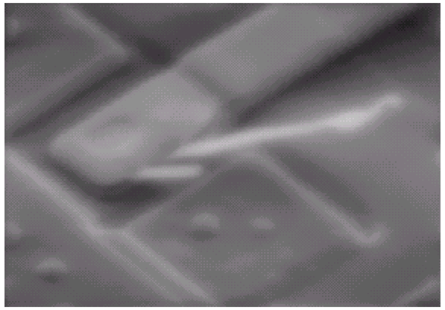


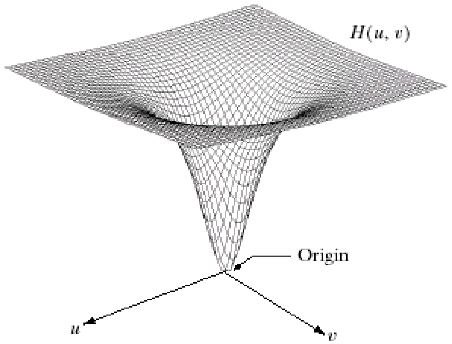
Some Basic Frequency Domain Filters

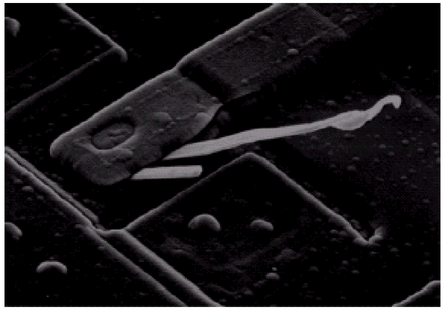
Low Pass Filter











High Pass Filter

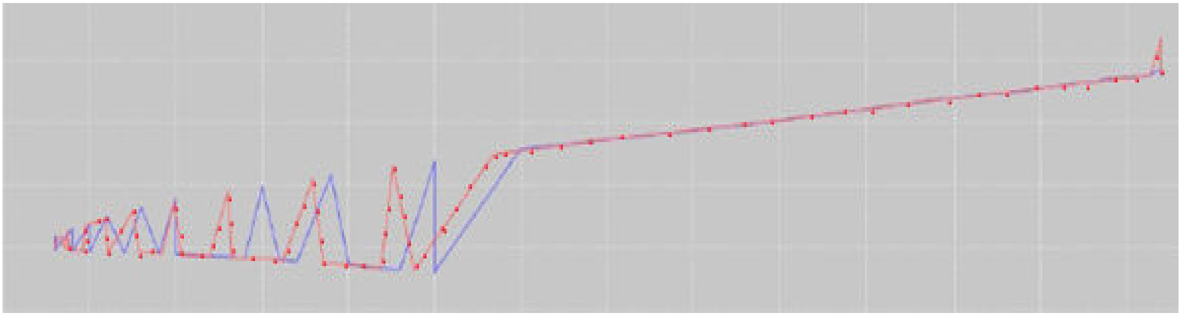
Filtering

- Outlier / sudden change detection
- Smoothing
- Enhancement / Noise Cancellation
- Special purpose – edge detection

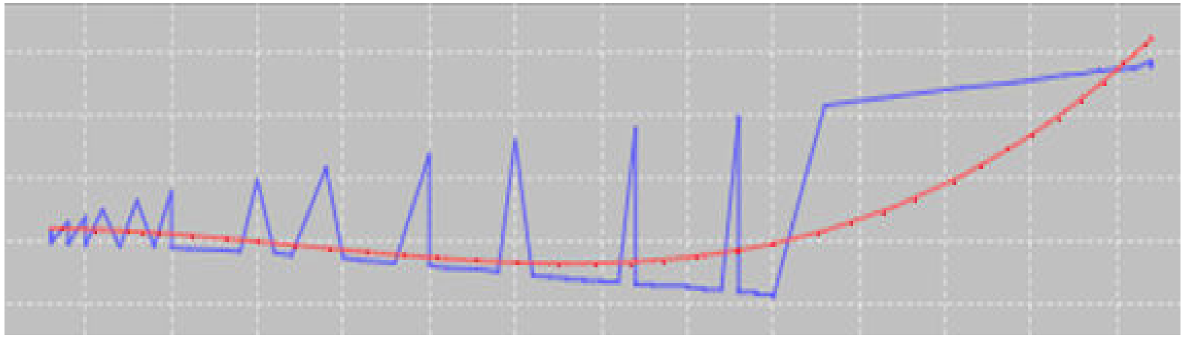
Filtering Example – Cursor Trace Smoothing

- Moving cursor in a vibrating place or by people with motor impairment create jitter in movement

- Filtering can remove jitter to facilitate pointing in a GUI



Kalman Filter

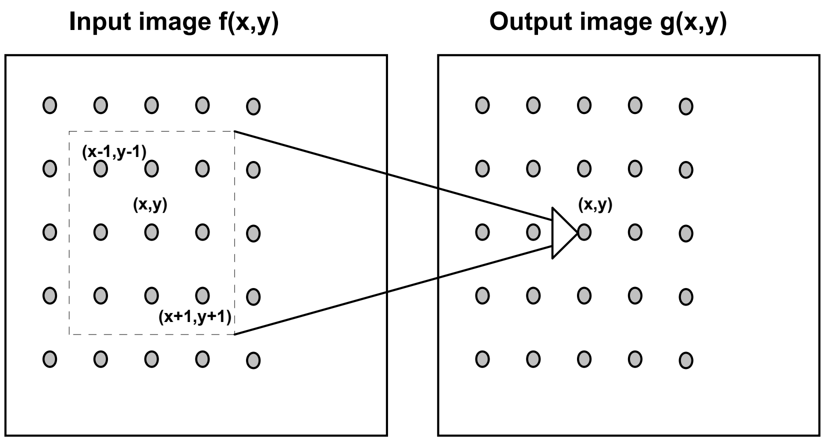


Polynomial Filter

N. Blow and P. Biswas, A pointing facilitation system for motor-impaired users combining polynomial smoothing and time weighted gradient target prediction models, **Assistive Technology** 29(1), ISSN: 1040-0435, Taylors and Francis, 2016

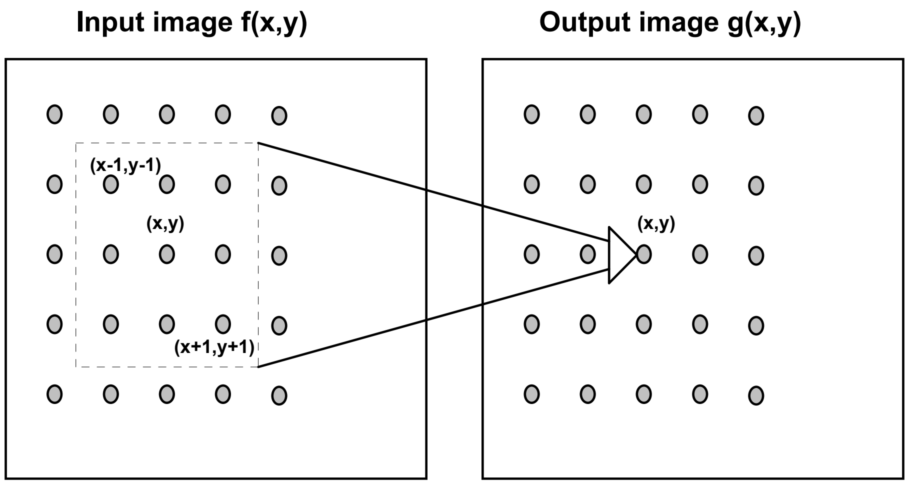
Image Filtering

- The output *g(x,y)* can be a linear or non-linear function of the set of input pixel grey levels {*f(x-M,y-M)…f(x+M,y+M*}.



Linear filtering and convolution

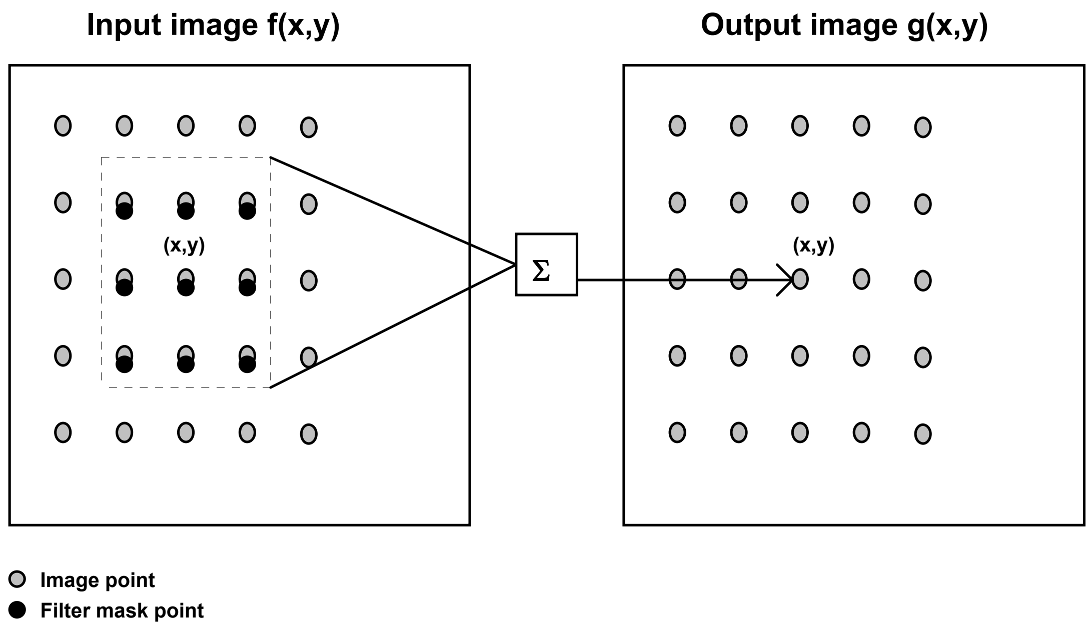
- Example
  - 3x3 arithmetic mean of an input image (ignoring floating point byte rounding)
  - Simple arithmetic averaging
  - Useful for smoothing images corrupted by additive broad band noise



Averaging Filter

- Convolution involves 'overlap – multiply – add' with 'convolution mask'

$$H = \begin{pmatrix} \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \end{pmatrix}$$



Linear filtering and convolution

- We can define the convolution operator mathematically
  - Defines a 2D convolution of an image *f(x,y)* with a filter *h(x,y)*

$$g(x,y) = \sum_{x'=-1}^{1} \sum_{y'=-1}^{1} h(x',y') f(x-x', y-y')$$

$$= \frac{1}{9} \sum_{x'=-1}^{1} \sum_{y'=-1}^{1} f(x-x', y-y')$$

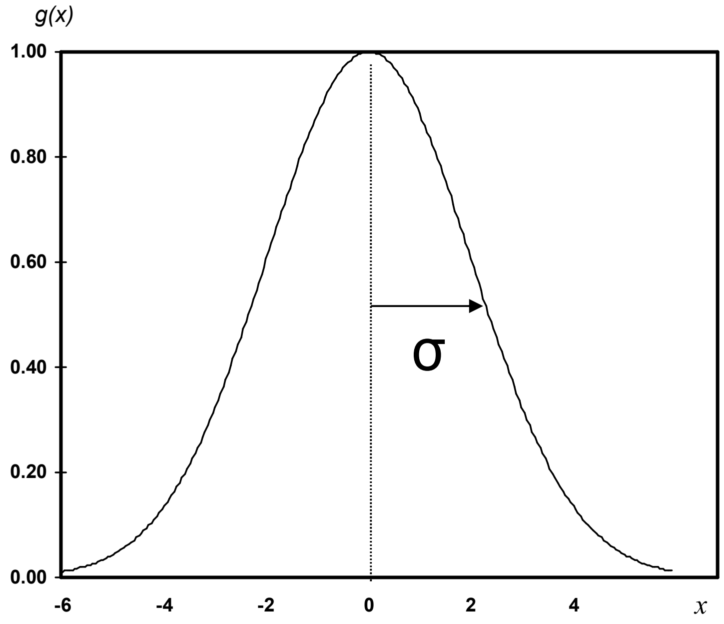
Gaussian filter

- Example – convolution with a *Gaussian* filter kernel
  - σ determines the width of the filter and hence the amount of smoothing

$$g(x,y) = exp(-\frac{(x^2+y^2)}{2\sigma^2})$$

$$= g(x)g(y)$$

$$g(x) = exp(-\frac{x^2}{2\sigma^2})$$



Example Images

Original

Noisy

Filtered

σ=1.5

Filtered

σ=3.0

Edge Detection

- Edge detection filter
  - Simple differencing filter used for enhancing edged
  - Has a bandpass frequency response

$$H = \begin{pmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{pmatrix}$$

The Sobel Operators

- Better approximations of the gradients exist
  - The *Sobel* operators below are commonly used

$$s_x = \frac{1}{8}\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \qquad s_y = \frac{1}{8}\begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

Comparing Edge Operators

Gradient: $$\nabla f = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right]$$

Good Localization
Noise Sensitive
Poor Detection

Roberts (2 x 2):

0	1
-1	0

1	0
0	-1

Sobel (3 x 3):

-1	0	1
-1	0	1
-1	0	1

1	1	1
0	0	0
-1	-1	1

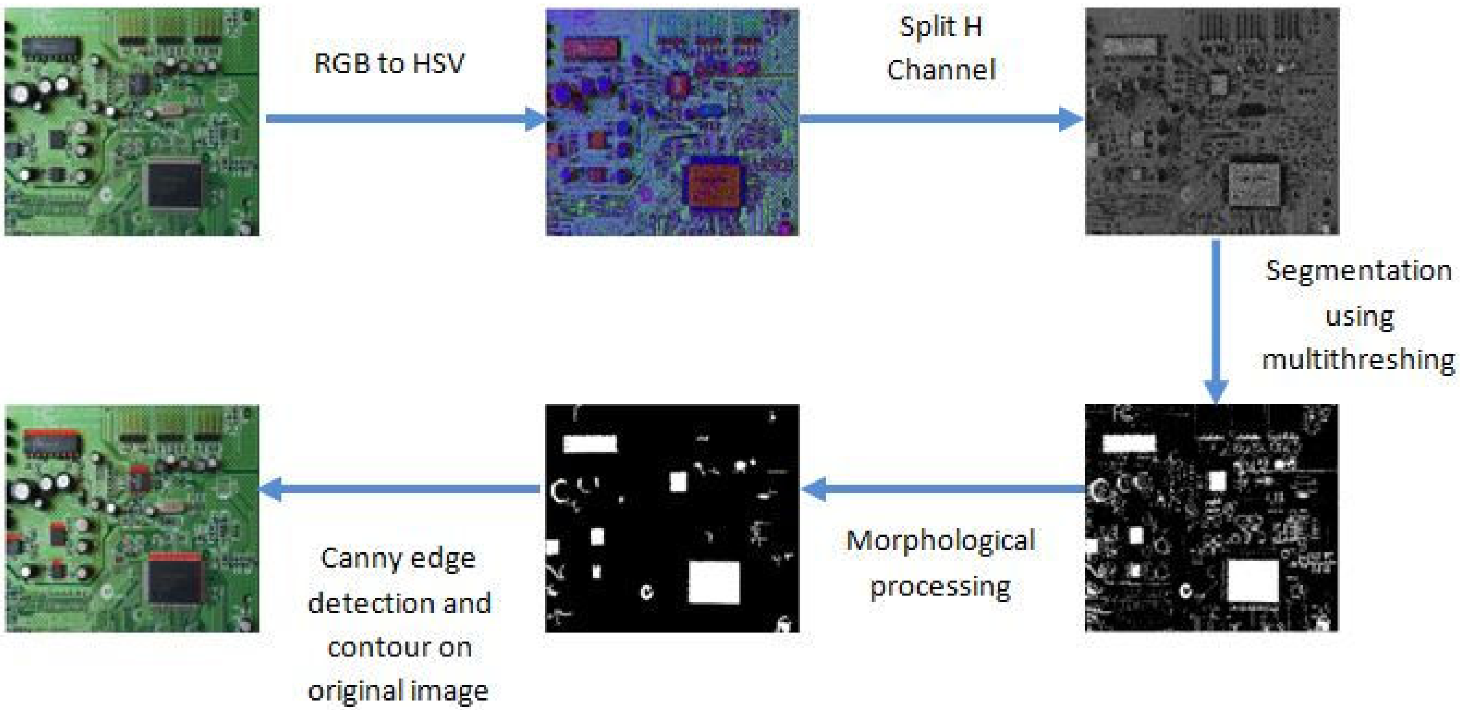
Sobel (5 x 5):

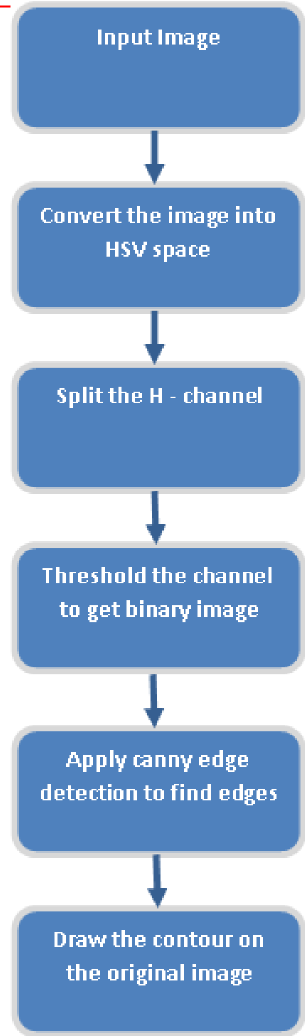
-1	-2	0	2	1
-2	-3	0	3	2
-3	-5	0	5	3
-2	-3	0	3	2
-1	-2	0	2	1

1	2	3	2	1
2	3	5	3	2
0	0	0	0	0
-2	-3	-5	-3	-2
-1	-2	-3	-2	-1

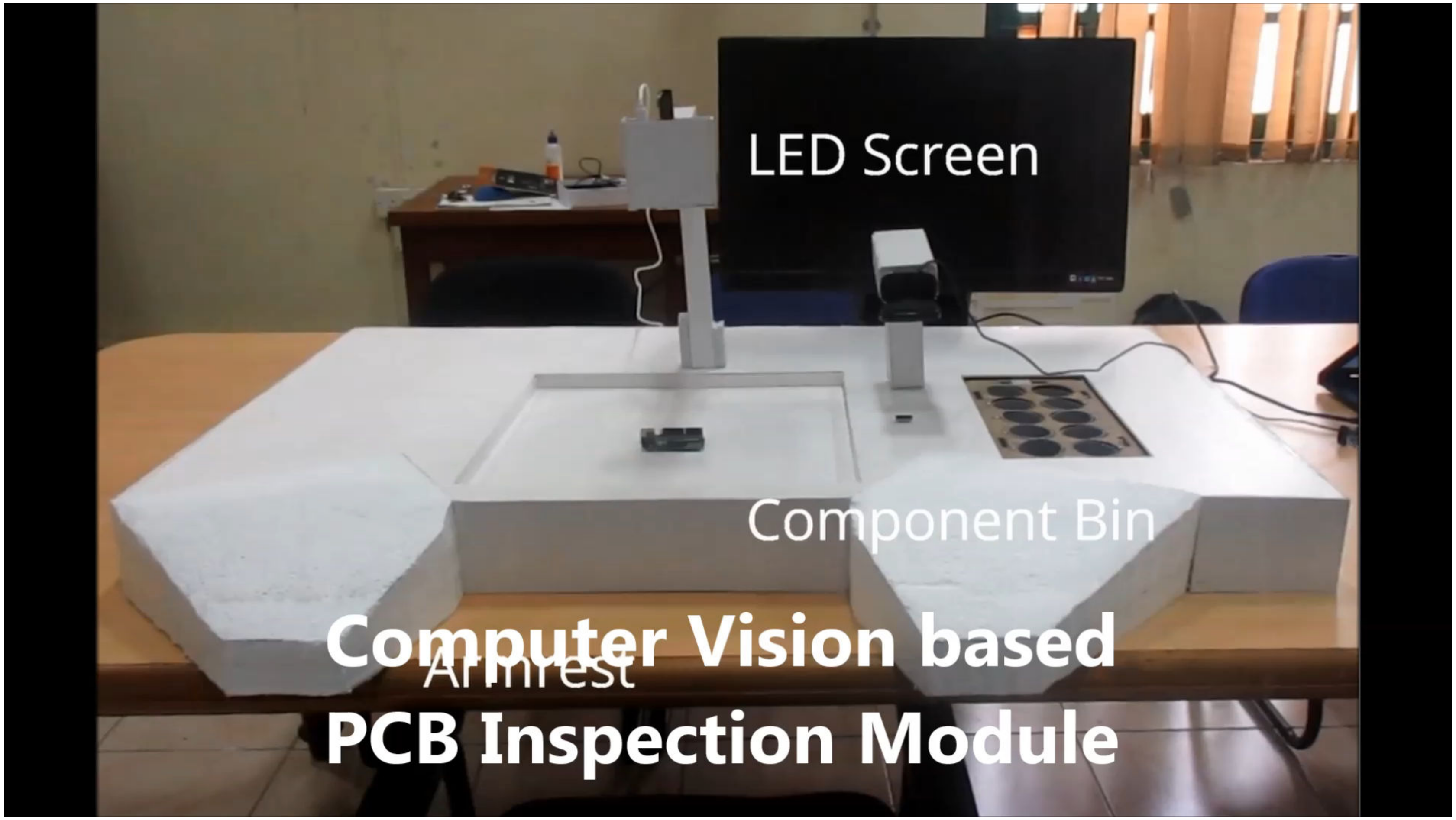
Poor Localization
Less Noise Sensitive
Good Detection

Case Study - PCB Inspection





Video Demonstration – PCB Inspection



Take Away Points

- Representing data as a signal
- Doing elementary signal manipulation
- Time and frequency domain conversions
- Filtering, Smoothing and Masking operations