NCKU
Tainan Taiwan

Special Topics on Visual Intelligence with Deep Learning
智慧視覺深度學習專論

Course Overview

Jar-Ferr Yang

Institute of Computer and Communication Engineering,
Department of Electrical Engineering,
National Cheng Kung University, Tainan, Taiwan

Objective: AI Research Training and Finding

Help New Students to Start Their Researches!

Selected Topics:

1. Pure AI Related Research: Performance Improvement/Fast Learning
2. Applications in 3D and AR/VR Related Research
3. Intelligent Visualization: Autonomous Driving
4. **AR/VR for medical applications**
5. Other

Help New Students to Fill Their Research Gaps!

Teaching materials will depend on the requests of course students. The teaching course will be the selected topics in intelligence visual researches and advanced digital signal processing:

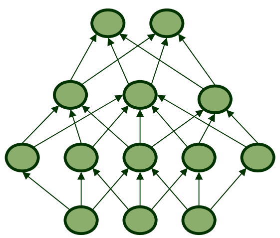
1. Advanced Linear Optimization
2. Adaptive Linear Optimization
3. Others (Requested by Students)

History of Deep Learning (Neural Networks 1986)

Neural network
Back propagation

1986

- Solve general learning problems
- Tied with biological system



But it was given up…

- Hard to train
- Insufficient computational resources
- Small training sets
- Does not work well

The linearization optimal (adaptive) problems for Weiner solution have been changed to nonlinear problems!

History of Deep Learning (Machine Learning before 2006)

Neural network
Back propagation

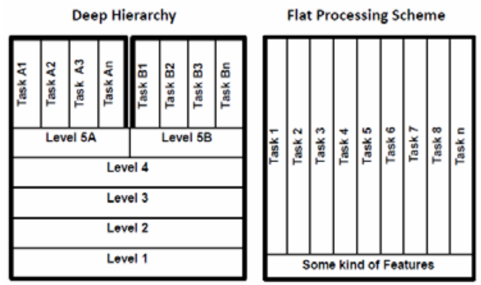
Machine Learning

1986 2006

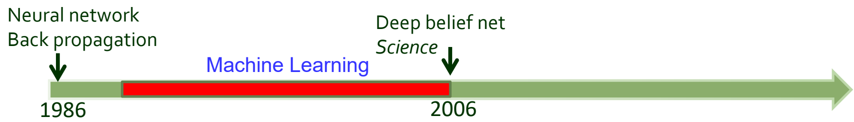
- **SVM**
- **Boosting**
- **Decision tree**
- **KNN**
- **…**

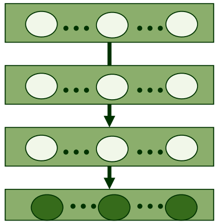
- Loose tie with biological systems
- Shallow model
- Specific methods for specific tasks
  - Hand crafted features (LBP, SIFT, HOG, …..)

Machine learning uses many new features with reasonable new processing techniques followed by learnt classification methods for object recognitions.



History of Deep Learning (2006 Deep Belief Net)

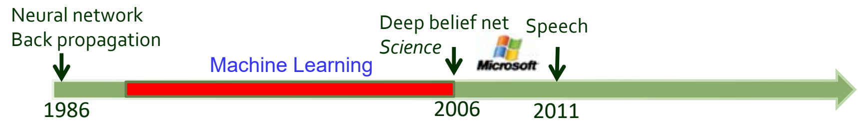




- Unsupervised & Layer-wised pre-training
- Better designs for modeling and training (normalization, nonlinearity, dropout)
- Feature learning
- New development of computer architectures
  - GPU
  - Multi-core computer systems
- Large scale databases

Deep network and large database for much better results!

History of Deep Learning (2011 Art in Speech)



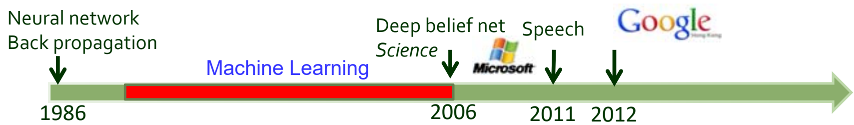
deep learning results (error rate)

task	hours of training data	DNN-HMM	GMM-HMM with same data
Switchboard (test set 1)	309	18.5	27.4
Switchboard (test set 2)	309	16.1	23.6
English Broadcast News	50	17.5	18.8
Bing Voice Search (Sentence error rates)	24	30.4	36.2
Google Voice Input	5,870	12.3	
Youtube	1,400	47.6	52.3

Deep Networks Advance State of Art in Speech Microsoft

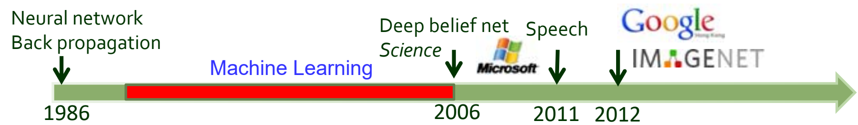
Deep Learning leads to breakthrough in speech recognition at MSR.

History of Deep Learning



How many computers to identify a **Cat**? **16000 CPU cores!**

History of Deep Learning



Rank	Name	Error rate	Description
1	**U. Toronto**	0.15315	Deep learning
2	U. Tokyo	0.26172	Hand-crafted features and learning models. Bottleneck.
3	U. Oxford	0.26979	
4	Xerox/INRIA	0.27058	

Object recognition over 1,000,000 images and 1,000 categories (2 GPU)

Artificial Intelligence (AI) becomes the dominated research topics in many application areas

Grading Policy

NCKU, Tainan Taiwan

Research Motivations - An Introduction (Oral Presentation) --------------- 10%
The report powerpoint file should be sent to teacher one day before the presentation (after 3 weeks of class begin) about **March 2**

Survey Reports – Past Existed Researches (Oral Presentation) -------------- 15%
The report powerpoint file should be sent to teacher one day before the presentation (after another 3 weeks of class) about **March 30**

Research Designs – New Research Idea (Oral Presentation) ------------------ 20%
The report powerpoint file should be sent to teacher one day before the presentation (after another 3 weeks of class) aabout **April 27**

Discussion and Question During Presentation ------------------------------------ 20%
During the other students' reports, the student must raise at least one question or suggestion, the scores are based on technical contributions

Final Report: (Course Summary) -- 35%
A final report WORD file in IEEE paper format and its powerpoint file should be given after one week of course closing date: **1. Powerpoint file in Chinese**
2. Word file in English (At least 4 pages like a conference paper)

Course Outline

NCKU, Tainan Taiwan

智慧視覺深度學習專論
Special Topics on Visual Intelligence with Deep Learning

Topics:

0. **Invited Talk:** AI-based Multimedia Detection, Extraction, and Transformation
1. 視覺信號處理 (Fundamental of Visual Signal Processing)
2. 線性代數和視覺特徵空間(Linear Algebra and Visual Feature Space)
3. 視覺數據分析與回歸(Fundament of Visual Data Analyses and Regression)
4. 統計分析和推理(Statistical Analyses and Inference)
5. 目標優化之自適應濾波(Adaptive Filtering for Target Optimization)
6. 支持向量機器學習(Support Vector Machine Learning)
7. 類神經網路之後向傳播學習(Back Propagation Learning for Neural Networks)
8. 深度學習和轉移學習(Deep Learning and Transfer Learning)
9. 深度積捲式神經網絡(Deep Convolutional Neural Networks)
10. 深度神經網絡的應用(Application of Deep Neural Networks)
11. 深度神經網絡之研究設計(Designs of Deep Neural Networks)

Good Resources about Deep Learning

NCKU, Tainan Taiwan

- **Webpages:**
  - Geoffrey E. Hinton's readings (with source code available for DBN) http://www.cs.toronto.edu/~hinton/csc2515/deeprefs.html
  - Notes on Deep Belief Networks http://www.quantumg.net/dbns.php
  - MLSS Tutorial, October 2010, ANU Canberra, Marcus Frean http://videolectures.net/mlss2010au_frean_deepbeliefnets/
  - Deep Learning Tutorials http://deeplearning.net/tutorial/
  - Hinton's Tutorial, http://videolectures.net/mlss09uk_hinton_dbn/
  - Fergus's Tutorial, http://cs.nyu.edu/~fergus/presentations/nips2013_final.pdf
  - CUHK MMlab project : http://mmlab.ie.cuhk.edu.hk/project_deep_learning.html
- **People:**
  - Geoffrey E. Hinton's http://www.cs.toronto.edu/~hinton
  - Andrew Ng http://www.cs.stanford.edu/people/ang/index.html
  - Ruslan Salakhutdinov http://www.utstat.toronto.edu/~rsalakhu/
  - Yee-Whye Teh http://www.gatsby.ucl.ac.uk/~ywteh/
  - Yoshua Bengio www.iro.umontreal.ca/~bengioy
  - Yann LeCun http://yann.lecun.com/
  - Marcus Frean http://ecs.victoria.ac.nz/Main/MarcusFrean
  - Rob Fergus http://cs.nyu.edu/~fergus/pmwiki/pmwiki.php

NCKU, Tainan Taiwan

FTP Information for Materials Download

from Moodle

NCKU
Tainan Taiwan

Special Topics on Visual Intelligence with Deep Learning

智慧視覺深度學習專論

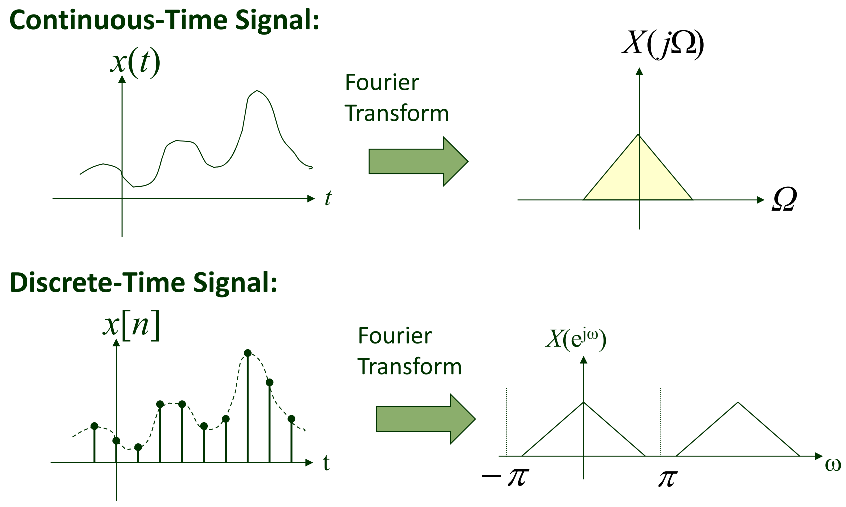
Part 1. Fundamental of Visual Signal Processing

視覺信號處理

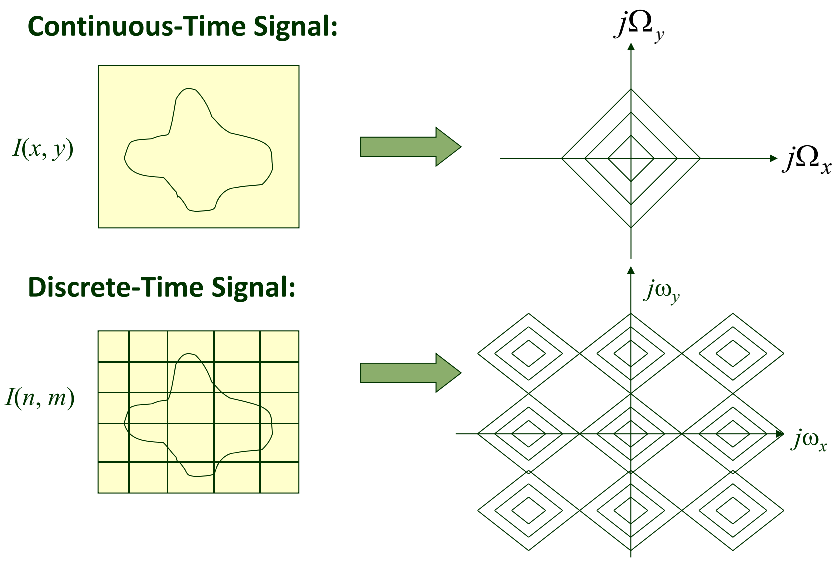
Jar-Ferr Yang

Institute of Computer and Communication Engineering,
Department of Electrical Engineering,
National Cheng Kung University, Tainan, Taiwan

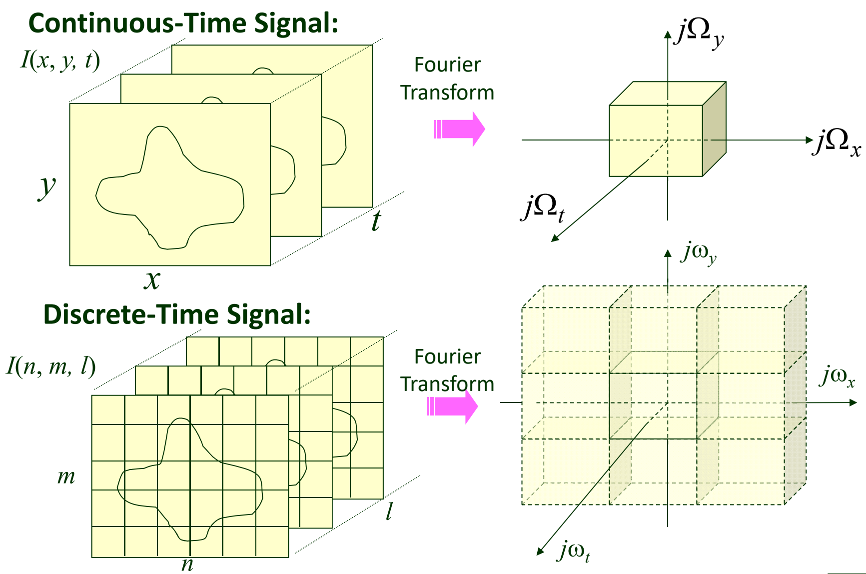
1-D Signal and Its Spectral



2-D Signal and Its Spectral



3-D Signal and Its Spectral



z-Transform of Discrete Signals

$$x[n] \Rightarrow X(z) = \sum_{n=-\infty}^{\infty} x[n] z^{-n}, \text{ROC} = \text{Region of Converge}$$

$$x[n,m] \Rightarrow X(z_1,z_2) = \sum_{n=-\infty}^{\infty}\sum_{m=-\infty}^{\infty} x[n,m] z_1^{-n} z_2^{-m}, \text{ ROC}$$

$$x[n,m,l] \Rightarrow X(z_1,z_2,z_3) = \sum_{n=-\infty}^{\infty}\sum_{m=-\infty}^{\infty}\sum_{l=-\infty}^{\infty} x[n,m,l] z_1^{-n} z_2^{-m} z_3^{-l}, \text{ ROC}$$

z-transform versus Fourier Transform (ROC $\supset$ unit circuit)

$$X(e^{j\omega}) = X(z)|_{z=e^{j\omega}} = \sum_{n=-\infty}^{\infty} x[n] e^{-j\omega n}$$

$$X(e^{j\omega_1}, e^{j\omega_2}) = X(z_1,z_2)|_{z_1=e^{j\omega_1}, z_2=e^{j\omega_2}} = \sum_{n=-\infty}^{\infty}\sum_{m=-\infty}^{\infty} x[n,m] e^{-j\omega_1 n} e^{-j\omega_2 m}$$

$$X(e^{j\omega_1}, e^{j\omega_2}, e^{j\omega_3}) = X(z_1,z_2,z_3)|_{z_1=e^{j\omega_1}, z_2=e^{j\omega_2}, z_3=e^{j\omega_3}}$$

$$= \sum_{n=-\infty}^{\infty} x[n,m,l] e^{-j\omega_1 n} e^{-j\omega_2 m} e^{-j\omega_3 l}$$

Fourier Transform of Discrete Signals

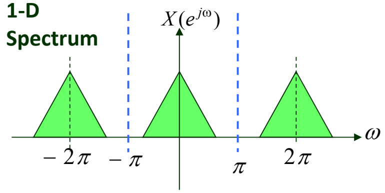
$$x[n] \longrightarrow X(e^{j\omega}) = \sum_{n=-\infty}^{\infty} x[n] e^{-j\omega n}$$

$$x[n,m] \longrightarrow X(e^{j\omega_1}, e^{j\omega_2}) = \sum_{n=-\infty}^{\infty}\sum_{m=-\infty}^{\infty} x[n,m] e^{-j\omega_1 n} e^{-j\omega_2 m}$$

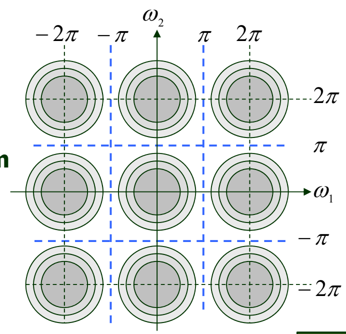
$$x[n,m,l] \longrightarrow X(e^{j\omega_1}, e^{j\omega_2}, e^{j\omega_3}) = \sum_{n=-\infty}^{\infty}\sum_{m=-\infty}^{\infty}\sum_{l=-\infty}^{\infty} x[n,m,l] e^{-j\omega_1 n} e^{-j\omega_2 m} e^{-j\omega_3 l}$$

$$\omega_i \to 2\pi + \omega_i \Rightarrow e^{j\omega_i} = e^{j(\omega_i + 2\pi)} = e^{j\omega_i} e^{j2\pi} \quad (e^{j2\pi} = 1)$$

1-D Spectrum



2-D Spectrum



Discrete Fourier Transform (DFT)

1D DFT: $$x[n] \Rightarrow X[k] = \sum_{n=0}^{N-1} x[n] e^{-j\frac{2\pi kn}{N}}$$

1D IDFT: $$X[k] \Rightarrow x[n] = \frac{1}{N}\sum_{k=0}^{N-1} X[k] e^{+j\frac{2\pi kn}{N}}$$

2D DFT: $$x[n,m] \Rightarrow X[k,l] = \sum_{n=0}^{N-1}\sum_{m=0}^{M-1} x[n,m] e^{-j\frac{2\pi kn}{N}} e^{-j\frac{2\pi lm}{M}}$$

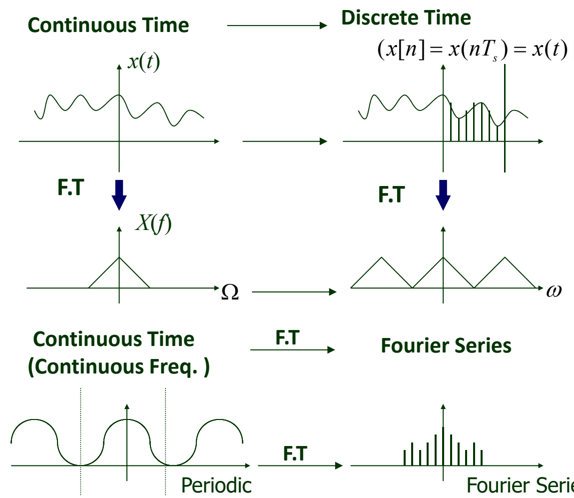
2D IDFT: $$X[k,l] \Rightarrow x[n,m] = \frac{1}{NM}\sum_{k=0}^{N-1}\sum_{l=0}^{M-1} X[k,l] e^{+j\frac{2\pi kn}{N}} e^{+j\frac{2\pi lm}{M}}$$

3D DFT: $$x[n,m,l] \Rightarrow X[k,p,q] = \sum_{n=0}^{N-1}\sum_{m=0}^{M-1}\sum_{l=0}^{L-1} x[n,m,l] e^{-j\frac{2\pi kn}{N}} e^{-j\frac{2\pi pm}{M}} e^{-j\frac{2\pi ql}{L}}$$

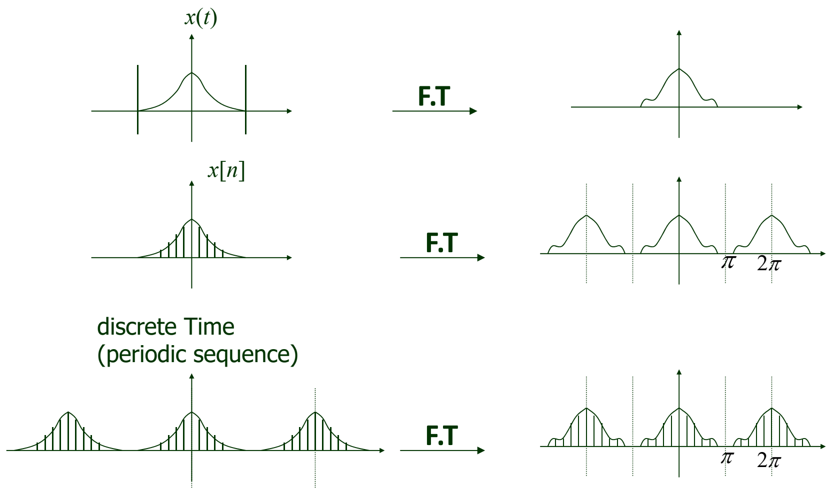
3D IDFT: $$X[k,p,q] \Rightarrow x[n,m,l] = \frac{1}{NML}\sum_{k=0}^{N-1}\sum_{p=0}^{M-1}\sum_{q=0}^{L-1} X[k,p,q] e^{+j\frac{2\pi kn}{N}} e^{+j\frac{2\pi pm}{M}} e^{+j\frac{2\pi ql}{L}}$$

DFT: Sampling of Fourier Transform

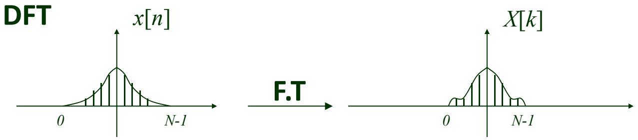
Discrete-Frequency Fourier Transform



Fourier Transform Versus DFT



DFT: Computing Fourier Transform



Finite duration data : M-point nonzero data

$$X(e^{j\omega}) = \sum_{n=0}^{9} x[n]e^{-j\omega n} \qquad \text{(M=9)}$$

16-point DFT

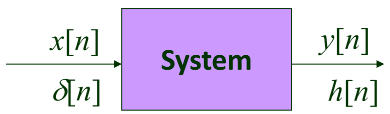
$$X[k] = \sum_{n=0}^{N-1} x[n]e^{-j\frac{2\pi nk}{N}} = \sum_{n=0}^{9} x[n]e^{-j\frac{2\pi nk}{N}}$$

$$(N = 16 \geq M = 10\ ) \qquad = X(e^{j\omega})\big|_{\omega=\frac{2\pi k}{N}}$$

Keep 9-point data by padding extra zeros

If N becomes larger, the frequency resolution becomes higher

Discrete Systems



Types of Systems:

1.Memory or Memoryless System?

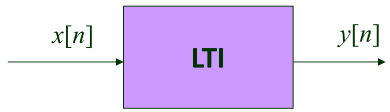
2.Linear or Nonlinear System?

3.Time-variant or Time-invariant System?

4.Causal or Noncausal System:

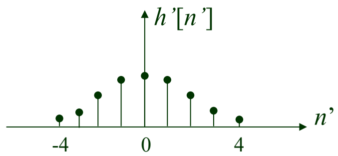
5.Stable or Unstable System?

Discrete LTI Systems



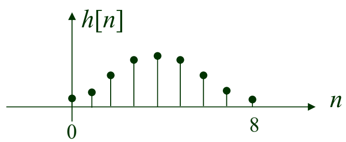
$$H(z) = \sum_{n=-\infty}^{\infty} h[n]z^{-n}$$

Linear and Time-Invariant



If the system is causal,
$h[n] = 0$ for $n < 0$

can be transferred to!



$z^{-4}H'(z) = H(z)$

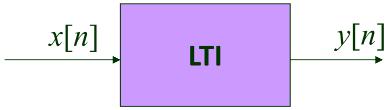
since

4 unit delays obtain gain,

$e^{-j4\omega}$, in Fourier domain

$= \begin{cases} \text{Linear Phase Delay} \\ \text{Unit Gain} \end{cases}$

Discrete LTI Systems



$$H(z)=\sum_{n=-\infty}^{\infty}h[n]z^{-n}$$

System Characteristics:

1. $h[n]$: Impulse Response
2. $H(e^{j\omega})$: Fourier transform of $h[n]$
3. $H(z)$: z transform of $h[n]$

Input and Output Relations:

1. Time domain: $y[n] = h[n] * x[n]$
2. Fourier domain: $Y(e^{j\omega})=H(e^{j\omega})X(e^{j\omega})$
3. Z-domain: $Y(z)=H(z)X(z)$

System and Its Stability

NCKU, Tainan Taiwan

BIBO Stable System:

$|x[n]| \le B_x < \infty$ for all n

⟺ $|y[n]| \le B_y < \infty$ for all n

For an LTI system ($h[n]$ is given), **if and only if**

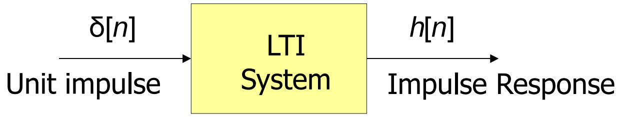
$$S = \sum_{k=-\infty}^{\infty}|h[k]| < \infty$$ ⟺ ROC must contain the unit circle

For realizable system, the causal condition gives the "**outward**" ROC

The poles must be **inside** the unit circle to assure a **stable system**.

Convolution of LTI Systems

NCKU, Tainan Taiwan



⋮ → ⋮

$x[1]\delta[n-1]$ → $x[1]\,h[n-1]$

$x[2]\delta[n-2]$ → $x[2]\,h[n-2]$

⋮ → ⋮

$x[k]\,\delta[n-k]$ → $x[k]\,h[n-k]$

+) ⋮ → ⋮

$x[n]=\sum_{k=-\infty}^{\infty}x[k]\delta[n-k]$ → $y[n]=\sum_{k=-\infty}^{\infty}x[k]h[n-k]$

Properties of Convolution Sum

NCKU, Tainan Taiwan

Convolution Sum:

$$\therefore\ y[n]=\sum_{k=-\infty}^{\infty}x[k]h[n-k]=x[n]*h[n]$$

Convolution Integral:

$$y(t)=\int_{-\infty}^{\infty}x(\tau)h(t-\tau)d\tau=x(t)*h(t)$$

Facts:

1. By using convolution formula, this system should be linear and time-invariant (LTI)。
2. But not assure if this system is memoryless or memory / causal or noncaual / stable or unstable?

Well-known Systems (Filters)

All-pole system:

$$H(z) = \frac{z^{-k}}{B(z)} \quad \text{zeros at } \{0, \pm\infty\} \qquad A(z) = z^{-k}$$

Minimum Phase system

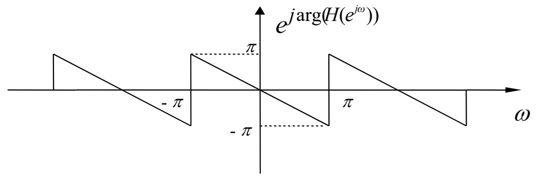
If all zeros and poles are inside the unit circle

Maximum phase system

If all zeros are outside the unit circle (poles much inside the unit circle)

Linear phase system: $H(e^{j\omega}) = A(e^{j\omega}) \bullet e^{j\arg(H(e^{j\omega}))}$

$$Arg\ H(e^{j\omega}) = \angle H(e^{j\omega}) = -\alpha_0\omega$$



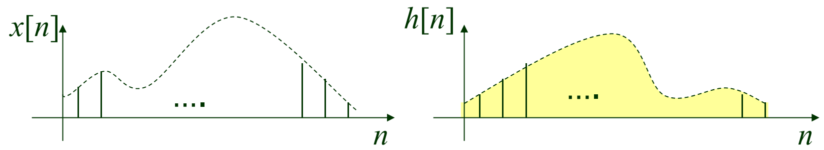
1-D Convolution Procedures

Convolution Sum:

$$y[n] = \sum_{k=-\infty}^{\infty} x[k]h[n-k] = \sum_{k=-\infty}^{\infty} h[k]x[n-k]$$

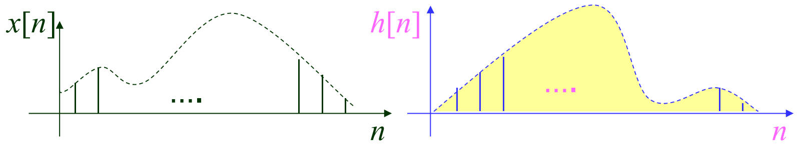
Convolution Procedures:

1. Reflecting $h[k]$ about the origin to obtain $h[-k]$.
2. Shifting the origin of the reflected sequence to $k = n$.
3. Multiplying $x[k]$ and $h[-k]$ and summing the results for all k.
4. Trying all $-\infty \le n \le \infty$

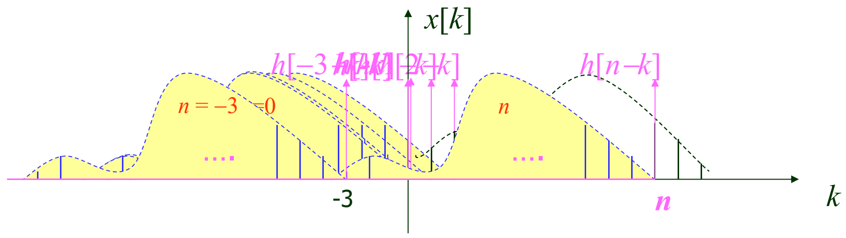


1D Convolution Procedures in Graphics

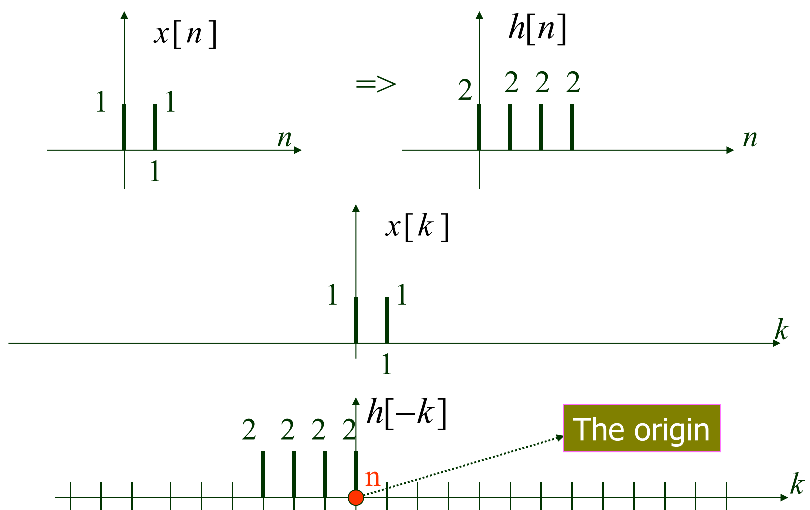
$$y[n] = \sum_{k=-\infty}^{\infty} x[k]h[n-k]$$



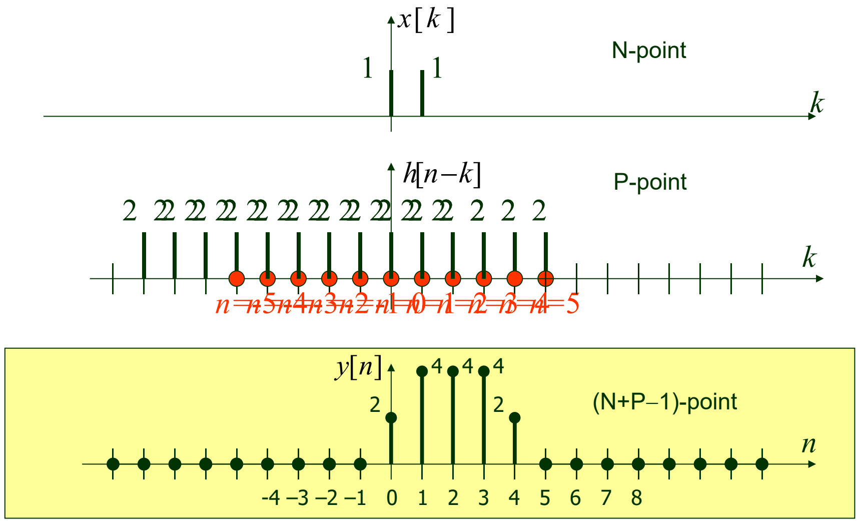
Computation in the k-domain



Convolution Examples (Two finite sequences)



Convolution Examples (Two finite sequences)



Practical Filtering Processes (P-point)

Filters inflate the data size to N+P-1 or M+P-1!!!

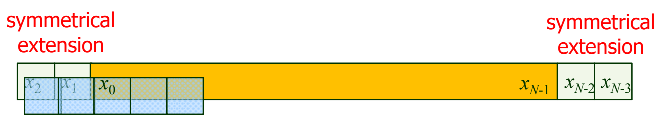
Filter coefficients (P=odd): $\{h_0, h_1, h_2, h_3, h_4\}$,
normally with $h_0=h_4$, $h_1= h_3$ (symmetrical)

Filter coefficients (P=even): $\{h_0, h_1, h_2, h_3\}$,
normally with $h_0=h_3$, $h_1= h_2$ (symmetrical)

How to get the same size of data vector (matrix)?

Practical Filtering Processes (P-coefficients)

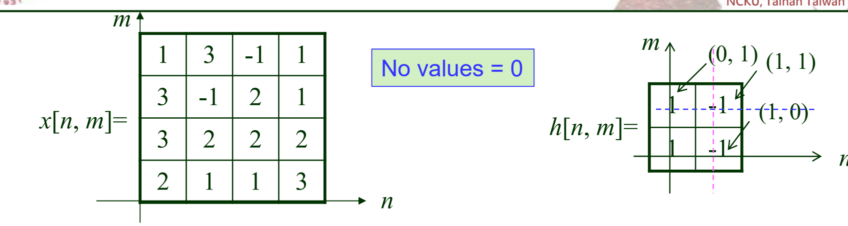
Filter coefficients (P=odd): $\{h_0, h_1, h_2, h_3, h_4\}$,
normally with $h_0=h_4$, $h_1= h_3$ (symmetrical)



Filter coefficients (P=even): $\{h_0, h_1, h_2, h_3\}$,
normally with $h_0=h_3$, $h_1= h_2$ (symmetrical)

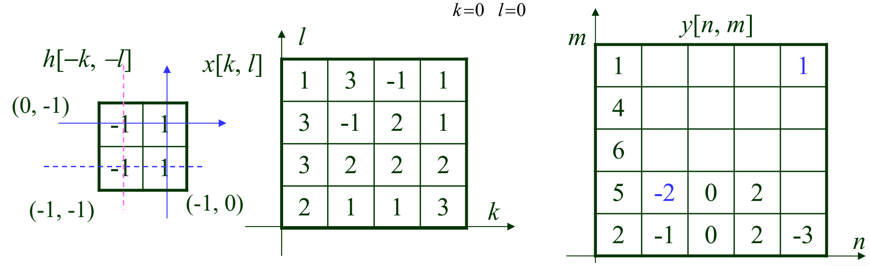
Proper data symmetrical extension and process N points only

2-D Convolution – Expanded Size (Theory)



Find the 2-D convolution: $y[n, m]=x[n, m] * h[n, m]=?$

$$y[n,m]=x[n,m]*h[n,m]=\sum_{k=0}^{N-1}\sum_{l=0}^{M-1}x[k,l]h[n-k,m-l]$$



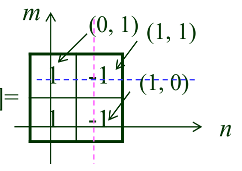
2-D Convolution – Same Size (Practical Case)

$x[n, m]=$

1	3	-1	1
3	-1	2	1
3	2	2	2
2	1	1	3

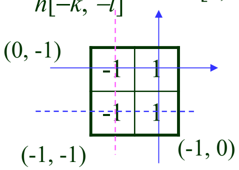
Boundary values by symmetrical padding

$h[n, m]=$



Find the 2-D convolution: $y[n, m]=x[n, m] * h[n, m]=?$

$h[-k, -l]$



$x[k, l]$

1	1	3	-1	1	1
1	1	3	-1	1	1
3	3	-1	2	1	1
3	3	2	2	2	2
2	2	1	1	3	3
2	2	1	1	3	3

$y[n, m]$

0	0		1
0			
0	-2	0	2
0	-2	2	0

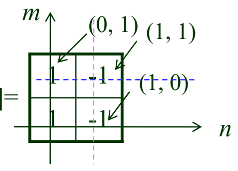
2-D Convolution – Reduced Size (AI-Application)

$x[n, m]=$

1	3	-1	1
3	-1	2	1
3	2	2	2
2	1	1	3

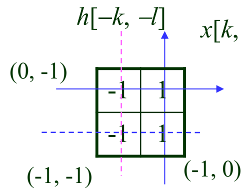
Removed boundary values

$h[n, m]=$



Find the 2-D convolution: $y[n, m]=x[n, m] * h[n, m]=?$

$h[-k, -l]$



$x[k, l]$

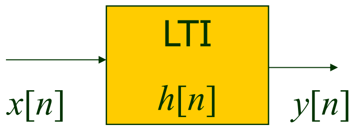
1	3	-1	1
3	-1	2	1
3	2	2	2
2	1	1	3

$y[n, m]$

-2		
1		
-2	0	2

The size of NxM image convolutes to a PxP filter will be reduced to (N-P)x(M-P)!

General Linear Time-Invariant System



$$H(z)=\frac{\sum_{k=0}^{M} b_k z^{-k}}{1-\sum_{k=1}^{N} a_k z^{-k}}=\frac{Y(z)}{X(z)}$$

$$\left[\sum_{k=0}^{M} b_k z^{-k}\right]X(z)=\left[1-\sum_{k=1}^{N} a_k z^{-k}\right]Y(z)$$

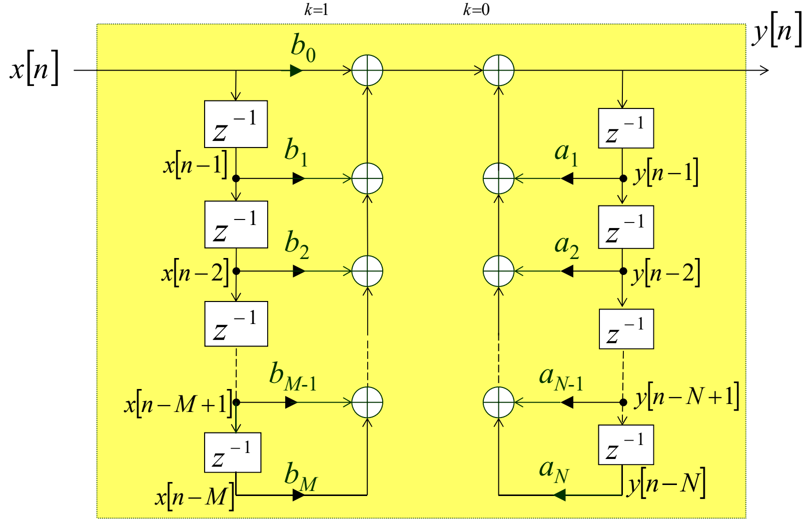
Construct a Difference Equation from $H(z)$:

$$\sum_{k=0}^{M} b_k\left[z^{-k}X(z)\right]=Y(z)-\sum_{k=1}^{N} a_k\left[z^{-k}Y(z)\right]$$

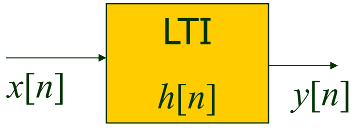
$$y[n]=\sum_{k=0}^{M} b_k x[n-k]+\sum_{k=1}^{N} a_k y[n-k]$$

Form-I Digital Filter Structure

$$y[n]=\sum_{k=1}^{N} a_k y[n-k]+\sum_{k=0}^{M} b_k x[n-k]$$



Linear Time-Invariant System



$$H(z)=\frac{\sum_{k=0}^{M} b_k z^{-k}}{1-\sum_{k=1}^{N} a_k z^{-k}}=\frac{Y(z)}{X(z)}$$

$$Y(z)=\left[\sum_{k=0}^{M} b_k z^{-k}\right]W(z) \quad \text{and} \quad X(z)=\left[1-\sum_{k=1}^{N} a_k z^{-k}\right]W(z)$$

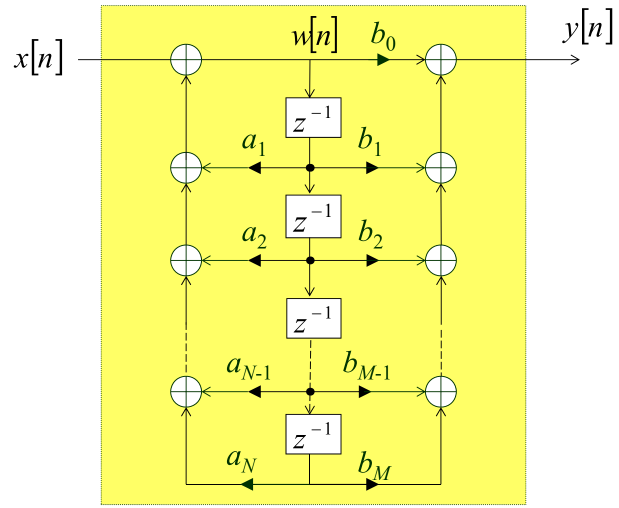
Two Difference Equations:

$$Y(z)=\sum_{k=0}^{M} b_k\left[z^{-k}W(z)\right] \quad \text{and} \quad X(z)=W(z)-\sum_{k=1}^{N} a_k\left[z^{-k}W(z)\right]$$

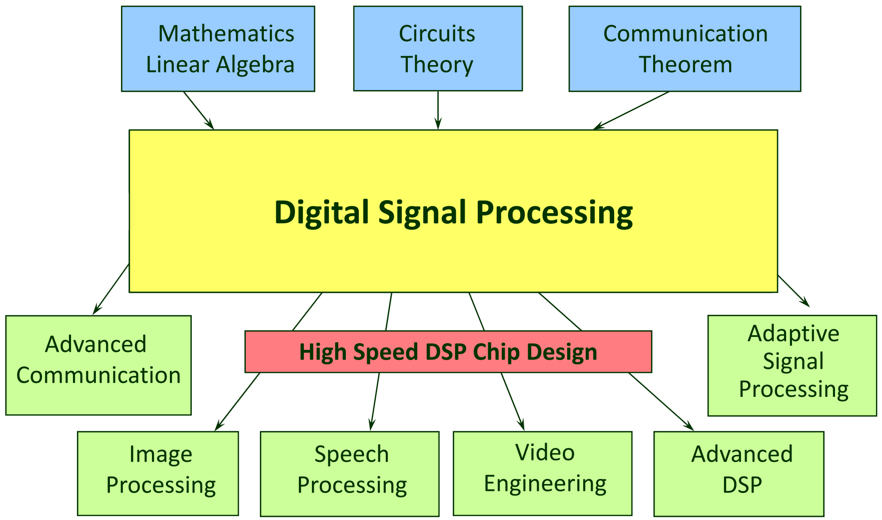
$$y[n]=\sum_{k=0}^{M} b_k w[n-k] \quad \text{and} \quad x[n]=w[n]-\sum_{k=1}^{N} a_k w[n-k]$$

Canonic Form Digital Filter

$$y[n]=\sum_{k=0}^{M} b_k w[n-k] \quad \text{and} \quad w[n]=x[n]+\sum_{k=1}^{N} a_k w[n-k]$$



DSP Related Courses



Signal Processing Coverage

Signal Processing can be further applied to:

Control system;
Spectrum Estimation;
Pattern Recognition
Tomography;
System Identification;
Communications (Detection and Estimation) ;
Information Theorem and Coding;
Medical Engineering

Media (Visual) Signal Processing:

Visual signal processing in 2D or 3D signals needs some processing techniques (mathematics) to deal with operations or analysis of signals.

Advanced Digital Signal Processing:

Present a series of relevant mathematical and statistical tools such that we could use the so-called advanced visual signal processing fundamentals to establish a solid backgrounds for advanced researches and realization practices.

Fundamentals for Advanced DSP

Mathematical Topics for Signal Processing:

- Linear System, Signal, Transformation;
- Vector Spaces and Linear Algebra;
- Probability and Stochastic Processes;
- Optimization (Constrained);
- Statistical Decision and Estimation Theorem
- Iterative Methods

Mathematical Models:

Linear Signal Models:
Linear Discrete-time Models; Stochastic MA and AR Models;

Adaptive Filtering Models:
System Identification; Inverse System Identification;
Adaptive Predictors; Interference cancellation

Gaussian Random Models:
Random probability model: Normal, Poisson, …

Hidden Markov Models:
Speech and text recognition, …

MediaCore

Special Topics on Visual Intelligence with Deep Learning
智慧視覺深度學習專論

Important Features for Visual Intelligence (HOG, SIFT, LBP)

Jar-Ferr Yang

Institute of Computer and Communication Engineering,
Department of Electrical Engineering,
National Cheng Kung University, Tainan, Taiwan

Histograms of Oriented Gradients (HOG)

HOG is an edge orientation histograms based on the orientation of the gradient in localized region that is called cells. Therefore, it is easy to express the rough shape of the object and is robust to variations in geometry and illumination changes. On the other hand, rotation and scale changes are not supported.

HOG Feature Vector Extraction

1. Color image converted to grayscale
2. Luminance gradient calculated at each pixel
3. Create a histogram of gradient orientations for each cell.
   - ✓ Feature quantity becomes robust to changes in form
4. Normalization and Descriptor Blocks
   - ✓ Feature quantity becomes robust to changes in illumination

Gradient Computation

Luminance gradient is calculated at each pixel

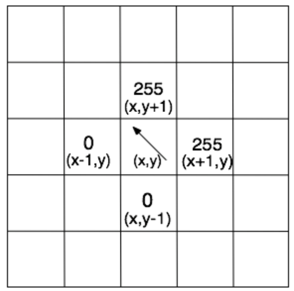
✓ The luminance gradient is a vector with magnitude $m(x,y)$ and orientation $\theta(x,y)$ represented by the change in the luminance.

$$m(x,y)=\sqrt{(L(x+1,y)-L(x-1,y))^2+(L(x,y+1)-L(x,y-1))^2}$$

$$\theta(x,y)=\tan^{-1}\left(\frac{L(x,y+1)-L(x,y-1)}{L(x+1,y)-L(x-1,y)}\right)$$

$$-\frac{\Pi}{2}<\theta<\frac{\Pi}{2}$$

where $L(x,y)$ denotes the luminance value of pixel at (x,y).

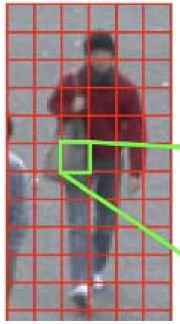


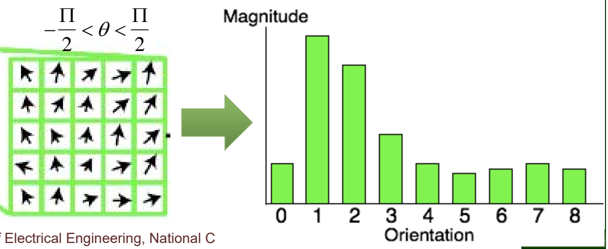
HOG Feature Vector

To create a histogram of gradient orientations for each cell (5×5 pixels) using the gradient magnitude and orientation of the calculated.

✓The orientation bins are evenly spaced over 0°– 180° and divided by nine of 20°. By adding the magnitude of the luminance gradient for each orientation, generation a histogram.

Orientation bin is $\left(\theta+\frac{\Pi}{2}\right)\div\Pi\times 9$



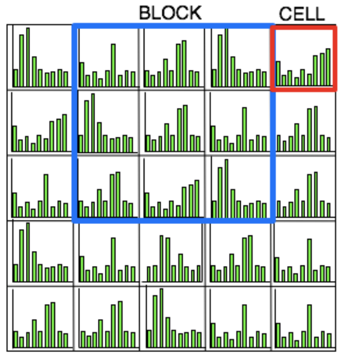


Normalization of HOG Feature Vector

Normalization and Descriptor Blocks

✓ Normalization is performed by:

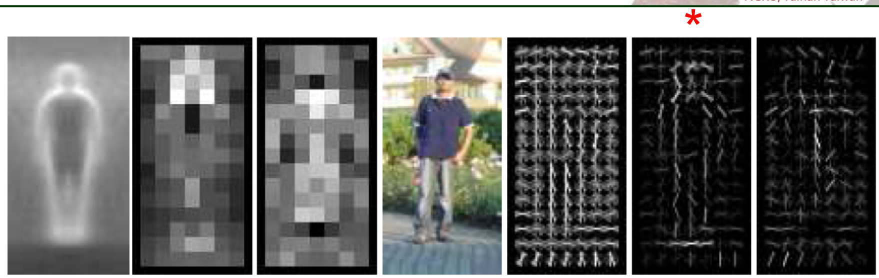
$$v(n)=\frac{v(n)}{\sqrt{\left(\sum_{k=1}^{3\times3\times9}v(k)^2\right)+1}}$$



$v(n)$ denotes the n^{th} accumulated magnitudes for each bin direction

Block (3 × 3 cells) is performed by moving one cell to the entire region.

Pictorial Examples



(a) (b) (c) (d) (e) (f) (g)

(a) average gradient image over training examples
(b) each "pixel" shows max positive SVM weight in the block centered on that pixel
(c) same as (b) for negative SVM weights
(d) test image
(e) its HOG descriptor
(f) HOG descriptor weighted by positive SVM weights
(g) HOG descriptor weighted by negative SVM weights

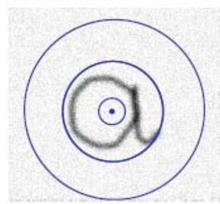
Scale Invariant Feature Transform (SIFT)

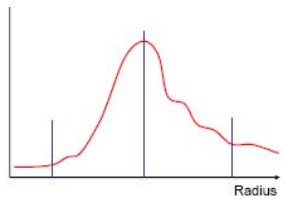
Image content is transformed into local SIFT feature coordinates that are invariant to translation, rotation, scale, and other imaging parameters

Scale Selection Principle

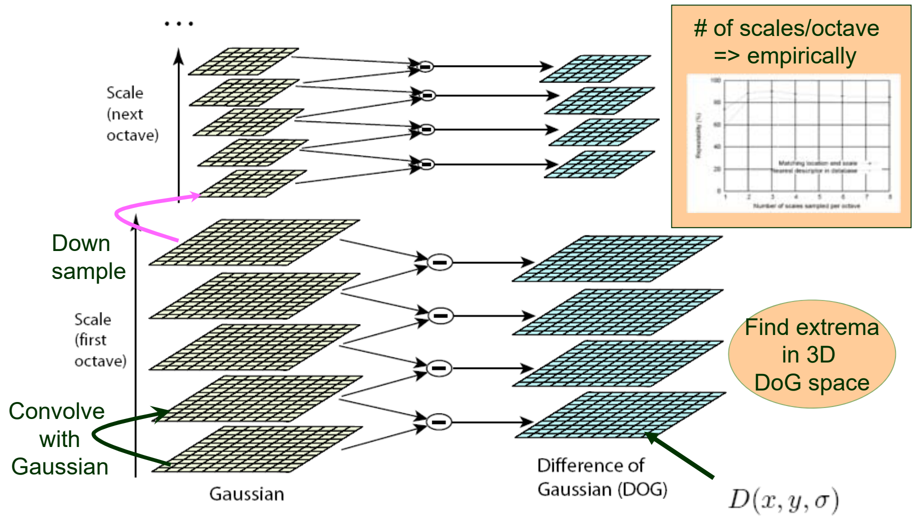
In the absence of other evidence, assume that a scale level, at which combination of normalized derivatives assumes a local maximum over scales, can be treated as reflecting a characteristic length of a corresponding structure in the data.

➔ **Maxima/minima of Difference of Gaussian**





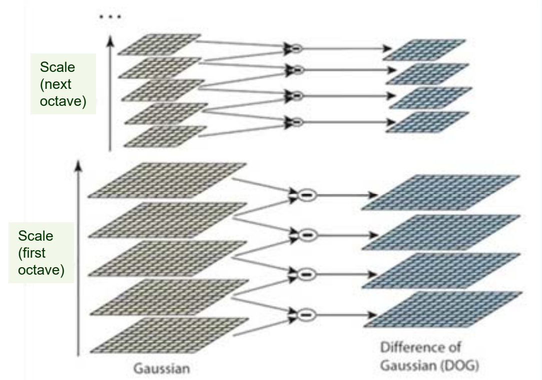
Finding Keypoints – Scale, Location



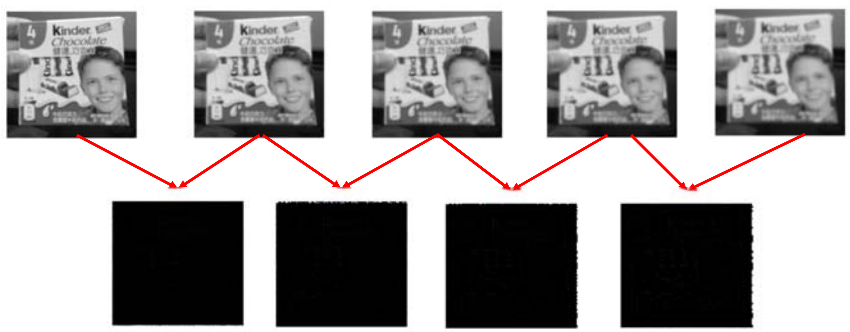
Difference of Gaussian (1/2)

Estimate the Gaussian difference of different σ

$$D(x,y,\sigma) = (G(x,y,k\sigma) - G(x,y,\sigma)) * I(x,y)$$
$$= L(x,y,k\sigma) - L(x,y,\sigma)$$



Difference of Gaussian (2/2)

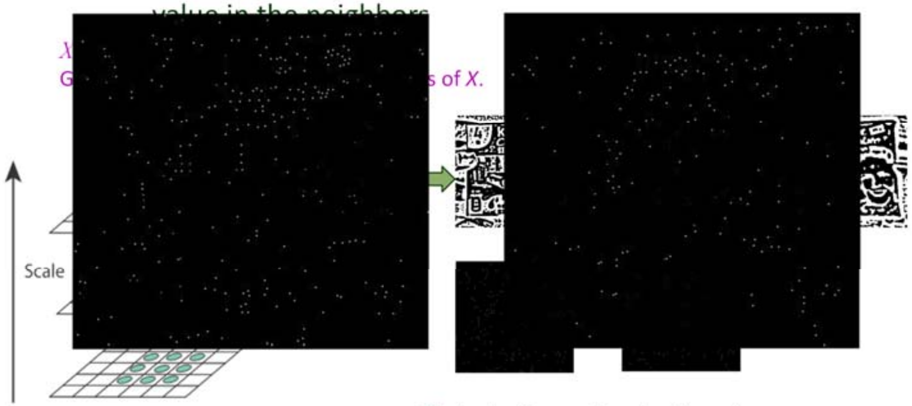


Difference of Gaussian binary images

Check- Local Minima and Maxima

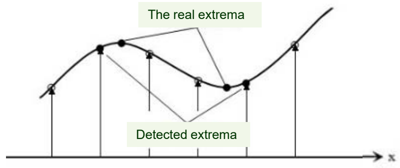
Find the local minima and maxima
and check if X is the minima and maxima
value in the neighbors

X
G
s of X.



The local minima and maxima binary images

Discard Low-contrast Feature Points



- **Taylor Expansion of DoG:**

$$D(x) = D + \frac{\partial D^T}{\partial x} x + \frac{1}{2} x^T \frac{\partial^2 D}{\partial x^2} x$$

- **Let D(x) = 0, the offset $\hat{x}$ is :**

$$\hat{x} = -\left(\frac{\partial^2 D^{-1}}{\partial x^2}\right)\left(\frac{\partial D}{\partial x}\right) \quad \text{remove } \left|\hat{x}\right| > T_x$$

$$D(\hat{x}) = D + \frac{1}{2}\left(\frac{\partial D^T}{\partial x}\right)\hat{x} \quad \text{remove } \left|D(\hat{x})\right| < T_D$$

Related Work – Scale-Invariant Feature Transform

Eliminate edge responses

Hessian matrix $H = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix}$

$$Tr(H) = D_{xx} + D_{yy} = \alpha + \beta$$

$$Det(H) = D_{xx} D_{yy} - D_{xy}^2 = \alpha\beta$$

Principal curvature R: let $\alpha = \gamma\beta$

$$R = \frac{Tr(H)^2}{Det(H)} = \frac{(\alpha+\beta)^2}{\alpha\beta} = \frac{(\gamma\beta+\beta)^2}{\gamma\beta^2} = \frac{(\gamma+1)^2}{\gamma}$$

Remove the edge point by $R > \frac{(\gamma_{th}+1)^2}{\gamma_{th}}$

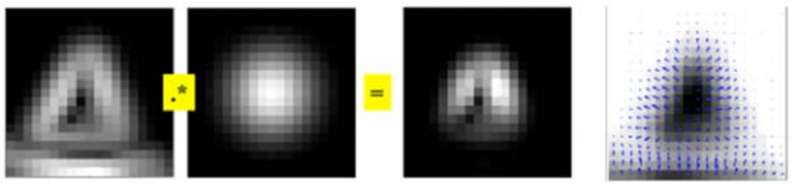
[illegible]

Feature Point Orientations

The gradient magnitude $m(x, y)$ and orientation $\theta(x, y)$

$$m(x,y) = \sqrt{(L(x+1,y) - L(x-1,y))^2 + (L(x,y+1) - L(x,y-1))^2}$$

$$\theta(x,y) = \tan^{-1}(L(x,y+1) - L(x,y-1), L(x+1,y) - L(x-1,y))$$



gradient magnitude · weighted by 2D Gaussian kernel · weighted by 2D gradient magnitude · orientation

Generate the Descriptor of SIFT

Form an orientation histogram with 8 bits (45 degree per bit)

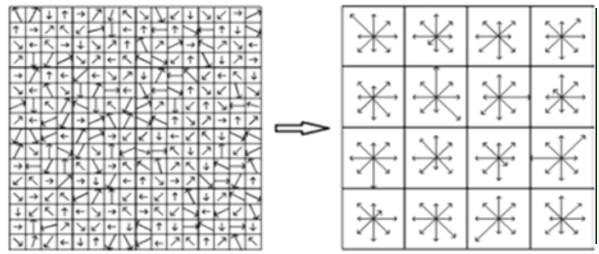
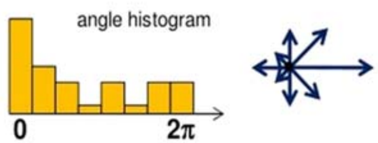


Image gradient · Key-point descriptor

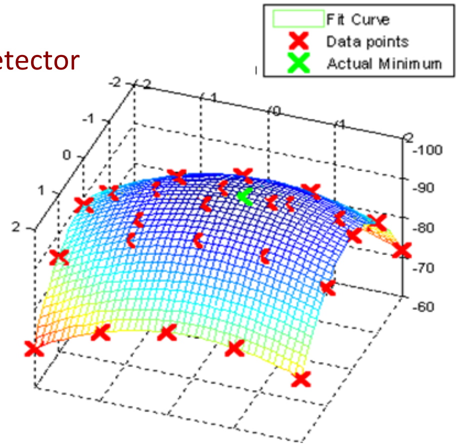


Finding Keypoints – Scale, Location

- **Sub-pixel Localization**
  - Fit Tri-variate quadratic to find sub-pixel extrema
- **Eliminating edges**
  - Similar to Harris corner detector

$$\mathbf{H} = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix}$$

$$\frac{\mathrm{Tr}(\mathbf{H})^2}{\mathrm{Det}(\mathbf{H})} < \frac{(r+1)^2}{r}$$



Finding Keypoints – Scale, Location

NCKU, Tainan Taiwan

- **Key issue: Stability (Repeatability)**
- **Alternatives**
  - ✓ Multi-scale Harris corner detector
  - ✓ Harris-Laplacian
  - ✓ Kadir & Brady Saliency Detector
  - ✓ …
  - ✓ Uniform grid sampling
  - ✓ Random sampling

Descriptor	Grid	Random	Saliency [4]	DoG [7]
11 × 11 Pixel	64.0%	47.5%	45.5%	N/A
128-dim Sift	65.2%	60.7%	53.1%	52.5%

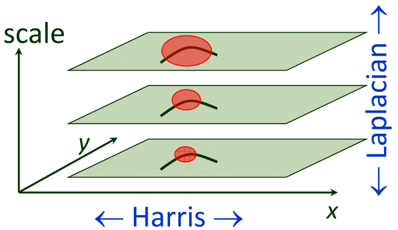
** Important Note ** Their application was scene classification NOT correspondence matching

Department of Electrical Engineering, National Cheng Kung University
MediaCore

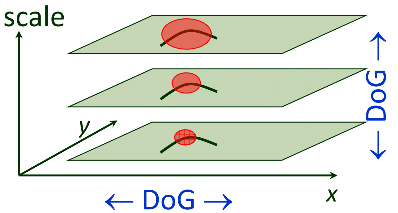
Finding Keypoints – Scale, Location

NCKU, Tainan Taiwan

- **Harris-Laplacian**[1]
  Find local maximum of:
  - Laplacian in scale
  - Harris corner detector in space (image coordinates)



- **SIFT**[2]
  Find local maximum of:
  - Difference of Gaussians in space and scale



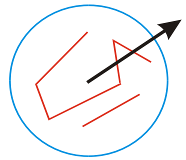
[1] K.Mikolajczyk, C.Schmid. "Indexing Based on Scale Invariant Interest Points". ICCV 2001
[2] D.Lowe. "Distinctive Image Features from Scale-Invariant Keypoints". IJCV 2004

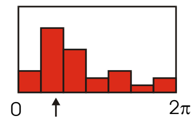
Department of Electrical Engineering, National Cheng Kung University
MediaCore

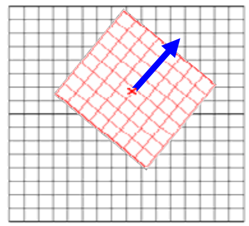
Finding Keypoints – Orientation

NCKU, Tainan Taiwan

- Create histogram of local gradient directions computed at selected scale
- Assign canonical orientation at peak of smoothed histogram
- Each key specifies stable 2D coordinates (x, y, scale, orientation)
- Assign dominant orientation as the orientation of the keypoint



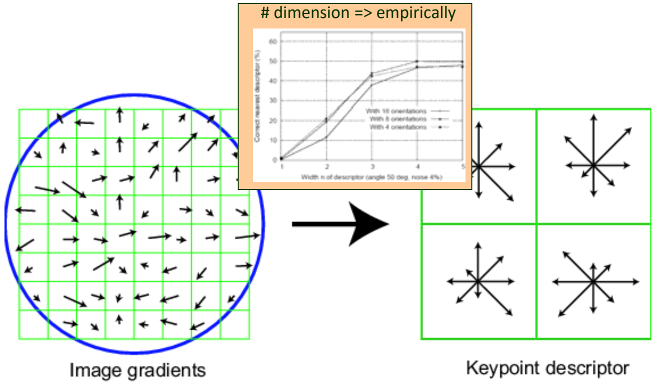




Department of Electrical Engineering, National Cheng Kung University
MediaCore

Creating Signature

- Thresholded image gradients are sampled over 16x16 array of locations in scale space
- Create array of orientation histograms
- 8 orientations x 4x4 histogram array = 128 dimensions



Comparison with HOG

- Histogram of Oriented Gradients
- General object class recognition (Human)
  - Engineered for a different goal
- Uniform sampling
- Larger cell (6-8 pixels)
- Fine orientation binning
  - 9 bins/180^{O} vs. 8 bins/360^{O}
- Both are well engineered

More about SIFT

Comparison with MOPS:

- Multi-Image Matching using Multi-Scale Orientated Patches (CVPR ’05)
- Simplified SIFT
  - ✓ Multi-scale Harris corner
  - ✓ No Histogram in orientation selection
  - ✓ Smoothed image patch as descriptor
- Good performance for panorama stitching

Conclusion about SIFT:

- Histogram of Oriented Gradients are becoming more popular
- SIFT may not be optimal for general object classification

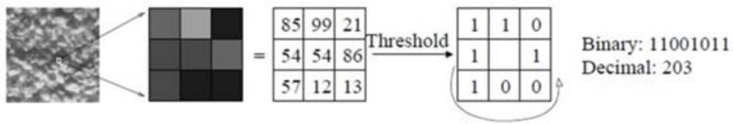
Local Binary Pattern (LBP)

Why LBP for Visual Description?

Object can be seen as a composition of micro-patterns which can be well described by LBP operator.

Basic LBP operator

The LBP operator was originally designed for texture description. The operator assigns a label to every pixel of an image by thresholding the 3x3-neighborhood of each pixel with the center pixel value and considering the result as a binary number.

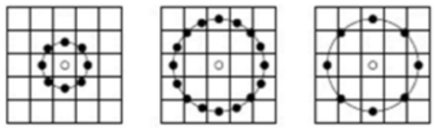


The histogram of the labels used as a texture descriptor.

LBP Operators for Different Scales

Notation for LBP Operator: $\mathrm{LBP}_{P,R}^{u2}$

- Extended LBP operators to use larger neighborhoods of different sizes, where $u2$ stands for using only uniform patterns and the subscript (P, R) represents using the operator in a neighborhood with P points and R radius
- Defining the local neighborhood as a set of sampling points evenly spaced on a circle centered at the pixel to be abled to allow any radius and number of sampling points.
- If a sampling point does not fall in the center of a pixel using Bilinear interpolation.



(8,1) (16,2) (8,2)

Uniform LBP Patterns and Histogram of LBP

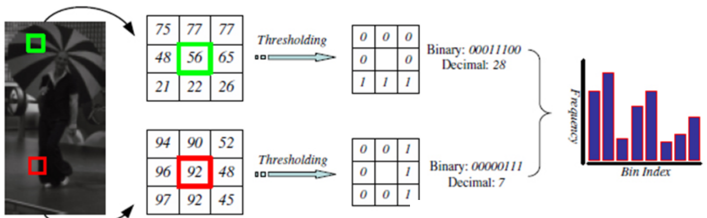
Uniform Pattern:

A local binary pattern is called uniform if the binary pattern contains at most two bitwise transitions from 0 to 1 or vice versa when the bit pattern is considered circular. The patterns with 00000000 (0 transitions), 01110000 (2 transitions) and 11001111 (2 transitions) are uniform. The patterns with 11001001 (4 transitions) and 01010011 (6 transitions) are not uniform.

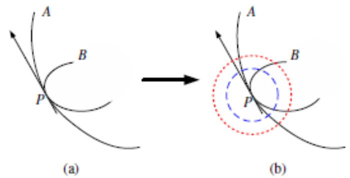
Histogram of LBP:

- For the LBP histogram, uniform patterns are used so that the histogram has a separate bin for every uniform pattern and all non-uniform patterns are assigned to a single bin.
- Ojala *et al.* noticed that in their experiments with texture images, uniform patterns account for a bit less than 90 % of all patterns when using the (8,1) neighborhood and for around 70 % in the
- (16,2) neighborhood. We have found that 90.6 % of the patterns in the (8,1) neighborhood and 85.2 % of the patterns in the (8,2) neighborhood are uniform in case of preprocessed FERET facial images.

Histogram of LBP Bins



LBP versus Gradient:



- Two same gradients may correspond to rather different local structures, thus ambiguous.
- The unexpected noises will drastically degrade the performance. Only gradient is insufficient to judge useful points and outliers. The "uniform LBP" provides the possibility to effectively remove outliers.
- The gradient typically drops color information, for it is difficult to define a metric for colors similar to intensity gradient.

Image Data Vectors

- Assume there are C subjects characterized by training facial images used for identity recognition. For C identities, the i^{th} identity is with N training facial images in size of $p\times q$ pixels and the k^{th} channel of j^{th} face image of i^{th} identity is represented by $V_{i,j,k} \in R^{p\times q}$, $i = 1, 2, \ldots, C$, $j = 1, 2, \ldots, N$ and $k = 1,2, \ldots K$.
- To achieve subspace-based classification, the facial image, $V_{i,j} \in R^{p\times q\times K}$ is further reformed by converting it to grayscale domain and cascading its q column vectors into a larger column vector, $x_{i,j} \in R^{d\times 1}$, where $d = p\times q$.

Grayscale Transformation and Cascading:

$$\boldsymbol{V}_{i,j} \in R^{p\times q\times K} \rightarrow \boldsymbol{G}_{i,j} \in R^{p\times q} \rightarrow \boldsymbol{x}_{i,j} \in R^{d\times 1}$$

Linear Regression Classification (LRC)

Training Phase | **Testing Phase**

Transformation and Cascading

$V_{i,j} \to G_{i,j} \to x_{i,j} \in R^{d\times 1}$

Class-specific Model

$X_i = [x_{i,1}, x_{i,2}, \ldots, x_{i,j}, \ldots, x_{i,N}]$

Precomputation

$H_{LRC,i} = X_i \left(X_i^T X_i \right)^{-1} X_i^T$

Linear Model

$y = X_i \beta_i$

Least Square Estimation

$\hat{\beta}_{LRC,i} = \left(X_i^T X_i \right)^{-1} X_i^T y$

Reconstructed Sample

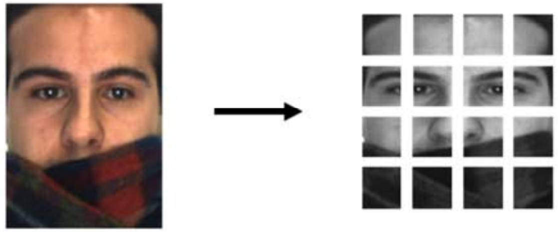
$\hat{y}_{LRC,i} = X_i \hat{\beta}_{LRC,i} = H_{LRC,i} y$

Distance Measure

$i^*_{LRC} = \arg\min_i \left\| y - \hat{y}_{LRC,i} \right\|, i = 1, 2, \ldots, C$

Module Linear Regression Classification (MLRC)

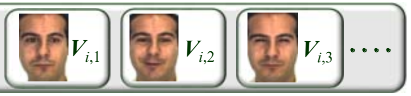
- For the module LRC (MLRC) approach, each training image $V_{i,j} \in R^{p\times q}$ is segmented into M non-overlapped partitions as $V^m_{i,j}$ Each partitioned image is formed as a column vector $x^m_{i,j}$, $i = 1,2,\ldots,C$, $j = 1,2,\ldots,N$, $m = 1,2,\ldots, M$.



Segmentation of a face image into M (M=16) modules.

Module Linear Regression Classification

Training Phase | **Testing Phase**



Segmentation and Cascading

$V_{i,j} \to V^m_{i,j} \to G^m_{i,j} \to x^m_{i,j} \in R^{d'\times 1}$

m^{th} module of Class-specific Model

$X^m_i = [x^m_{i,1}, x^m_{i,2}, \ldots, x^m_{i,j}, \ldots, x^m_{i,N}]$

Precomputation

$H^m_{MLRC,i} = X^m_i \left((X^m_i)^T X^m_i \right)^{-1} (X^m_i)^T$

Linear Model

$y^m = X^m_i \beta^m_i$

Least Square Estimation

$\hat{\beta}^m_{MLRC,i} = \left((X^m_i)^T X^m_i \right)^{-1} (X^m_i)^T y^m$

Reconstructed Sample

$\hat{y}^m_{MLRC,i} = X^m_i \hat{\beta}^m_{MLRC,i} = H^m_{MLRC,i} y^m$

Distance Measure

$i^*_{MLRC} = \arg\min_i (\min_m \left\| y^m - \hat{y}^m_{MLRC,i} \right\|)$

Conclusions

Features of Image:

- Histogram of Gradient (HOG)
- Scale Invariant Feature Transform (SIFT)
- Local Binary Pattern (LBP)
- Multi-Scale Orientated Patches (MSOP)
- Whole Image: Linear Regression (Image vector)

Image has been transformed to feature vector!

Intelligent Visualization: (Classification and Recognition)

- Linear Algebra: Vector and Matrix Operations
- Probability: Maximum Likelihood
- Machine Learning: Support Vector Machine
- Deep Learning: Convolution Neural Network

Smart Classification and Recognition by Using Data Vectors