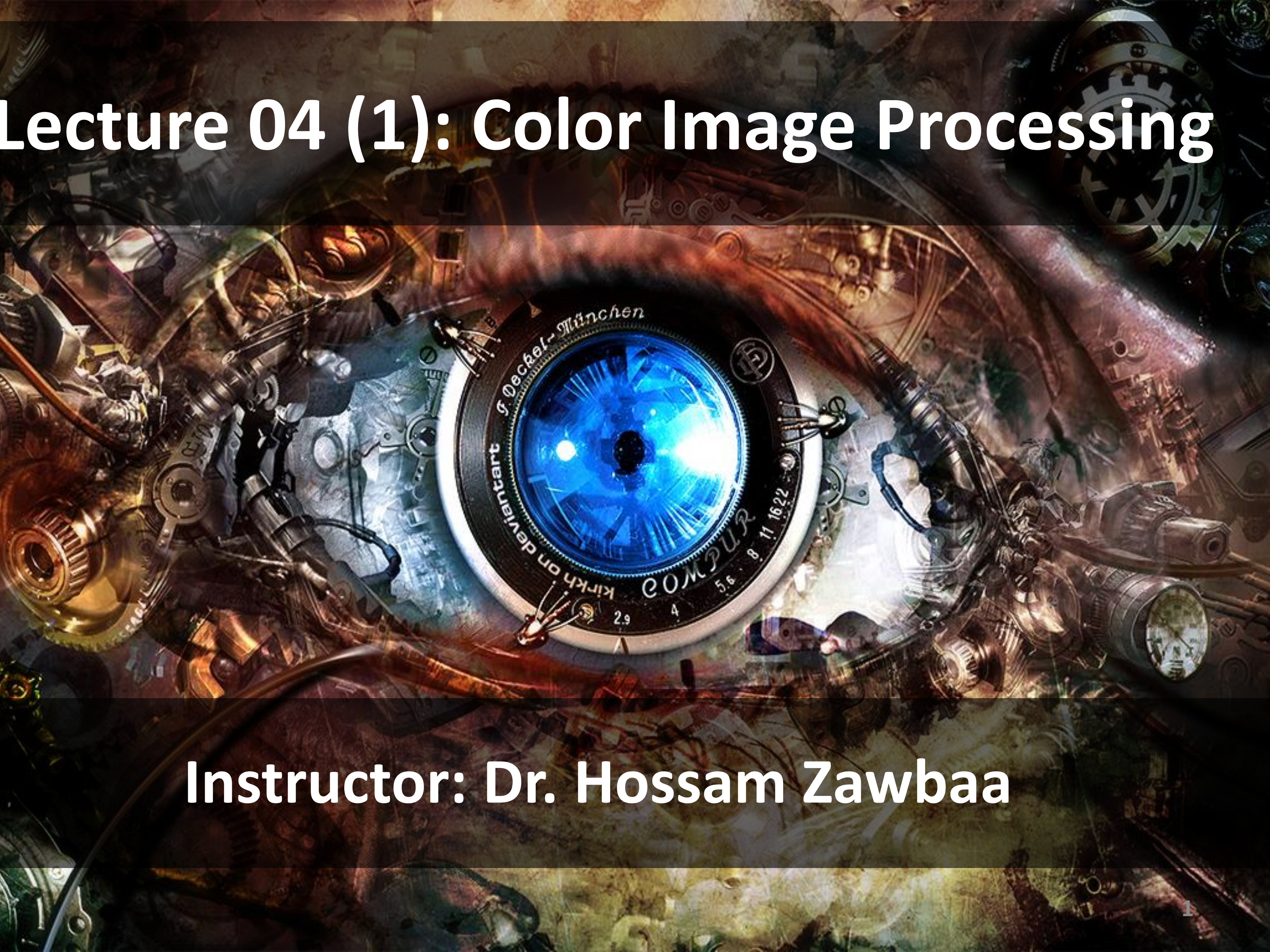
Lecture 04 (1): Color Image Processing



Instructor: Dr. Hossam Zawbaa

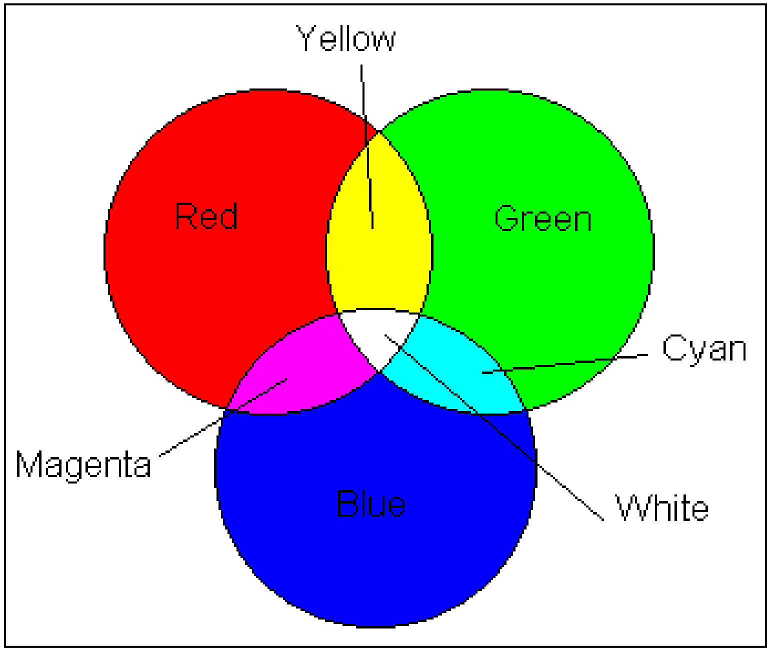
Introduction

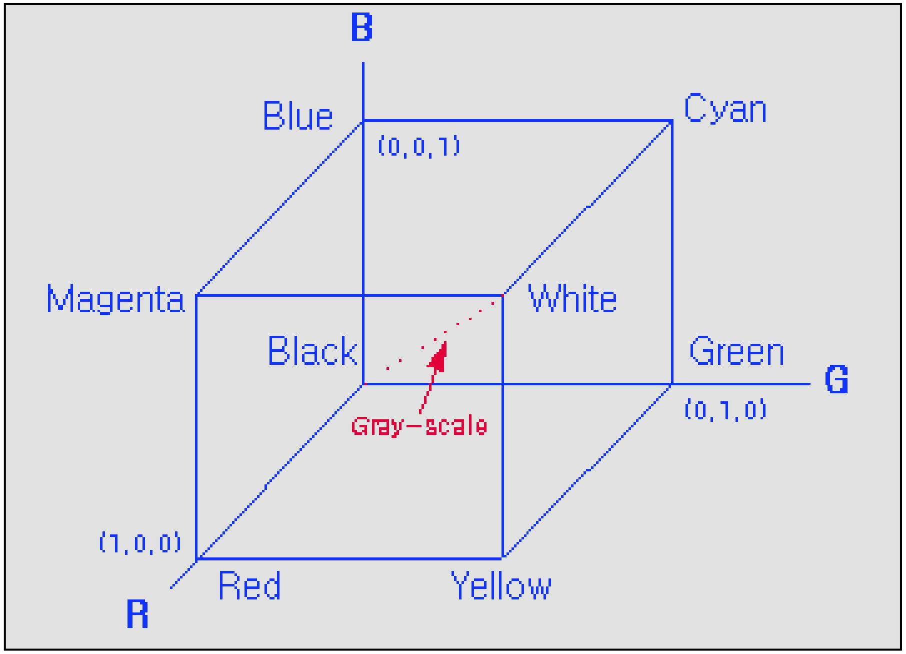
- Color
  - simplifies object extraction and identification
  - human vision : thousands of colors vs max-24 gray levels
- Color Spectrum
  - white light with a prism (1966, Newton)

Introduction

- **RGB (Red, Green, Blue):** Color Monitor, Color Camera, Color Scanner
- **CMY (Cyan, Magenta, Yellow):** Color Printer, Color Copier
- **YIQ :** Color TV **Y(luminance), I(Inphase), Q(quadrature)**
- **HSI : Hue, Saturation, Intensity**
- **HSV : Hue, Saturation, Value**

RGB Model to CMY Model





CMY Model

– Color Printer, Color Copier
– RGB conversion to CMY

$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

YIQ Model

- **Color TV**
- **RGB conversion to YIQ**

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 & 0.956 & 0.620 \\ 1 & -0.272 & -0.647 \\ 1 & -1.108 & 1.705 \end{bmatrix} \times \begin{bmatrix} Y \\ I \\ Q \end{bmatrix}$$

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.596 & -0.275 & -0.321 \\ 0.212 & -0.523 & 0.311 \end{bmatrix} \times \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

HSV Model

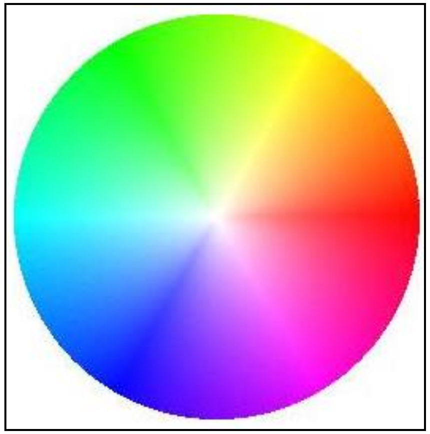
- **RGB to HSI Conversion**

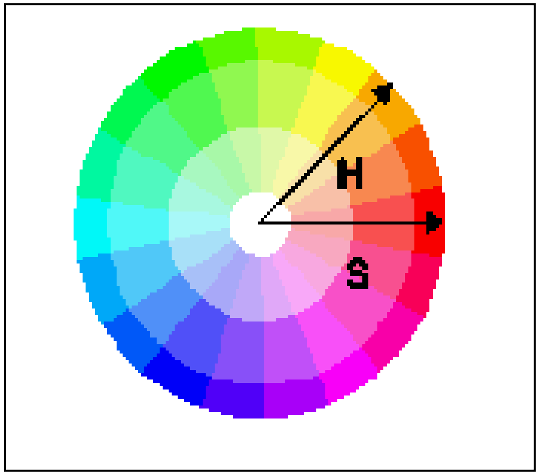
$$I = \frac{1}{3}(R+G+B), \quad \text{where } 0 \le I, R, G, B \le 1$$

$$H = \cos^{-1}\{\frac{\frac{1}{2}[(R-G)+(R-B)]}{\sqrt{(R-G)^2+(R-B)(G-B)}}\}, \quad \text{if } g_0 > b_0$$

$$H = 360^\circ - H, \quad \text{if } g_0 < b_0 \qquad \text{where } g_0 = G/I, \;\; b_0 = B/I$$

$$S = 1 - \frac{3}{R+G+B} \times (\min\{R, G, B\})$$





HSI Model

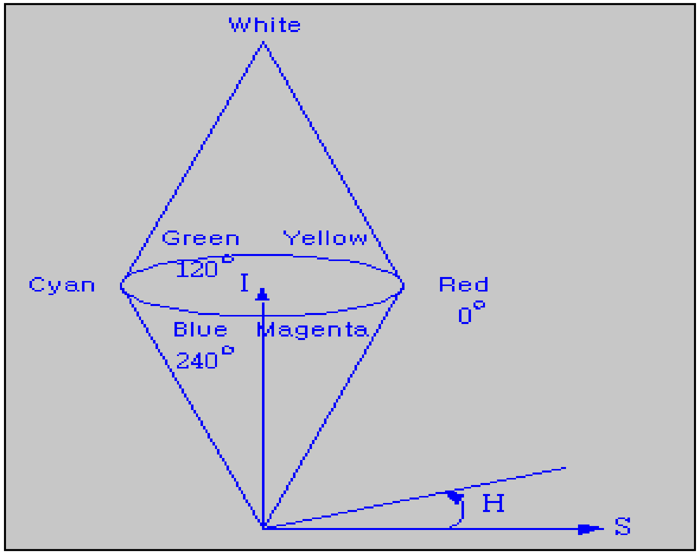
- HSI to RGB Conversion

$$B = \frac{1}{3}(1 - S)$$

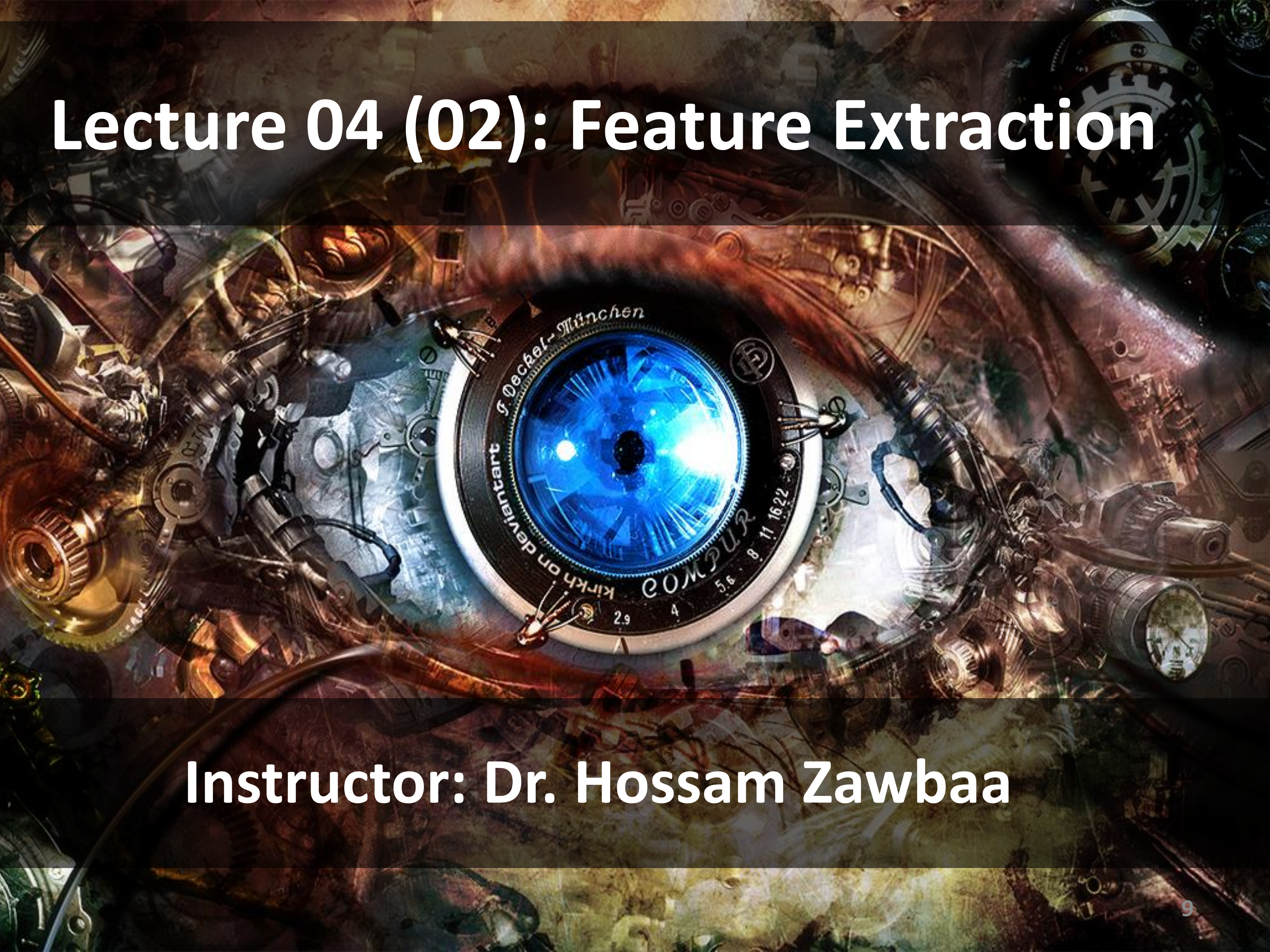
$$R = \frac{1}{3}[1 + \frac{S \cos H}{\cos(60^\circ - H)}]$$

assume $0^\circ \le H \le 120^\circ$

$$G = 1 - R - B$$



Lecture 04 (02): Feature Extraction



Instructor: Dr. Hossam Zawbaa

Feature extraction

- Goal is to extract important features from image data, from which a description, interpretation, or understanding of the scene can be provided.
- Feature extraction involves finding features of the segmented image.
- Usually performed on a binary image produced from a thresholding operation.

Region Features

- Many features can be use to characterize a region and its properties
- Supports many tasks, including object recognition.
- Example features:
  - area, height, and width
  - perimeter, bounding box, area of bounding box
  - centroid
  - orientation
  - compactness
  - moments
  - .....etc.

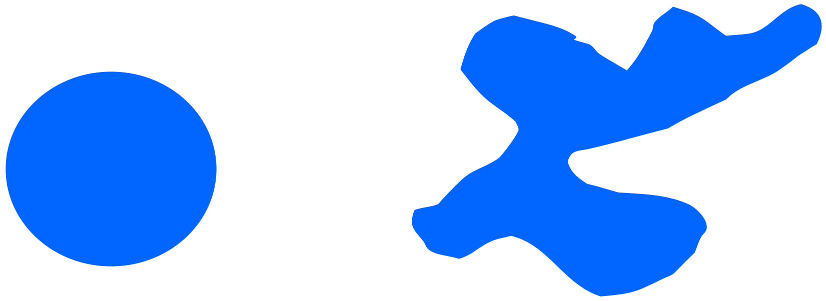
Area (or size): $$A = \sum_{r} \sum_{c} I(r,c)$$

Width: $$w = 1 + \max_{I(r,c)=1}(c) - \min_{I(r,c)=1}(c)$$

Height $$h = 1 + \max_{I(r,c)=1}(r) - \min_{I(r,c)=1}(r)$$

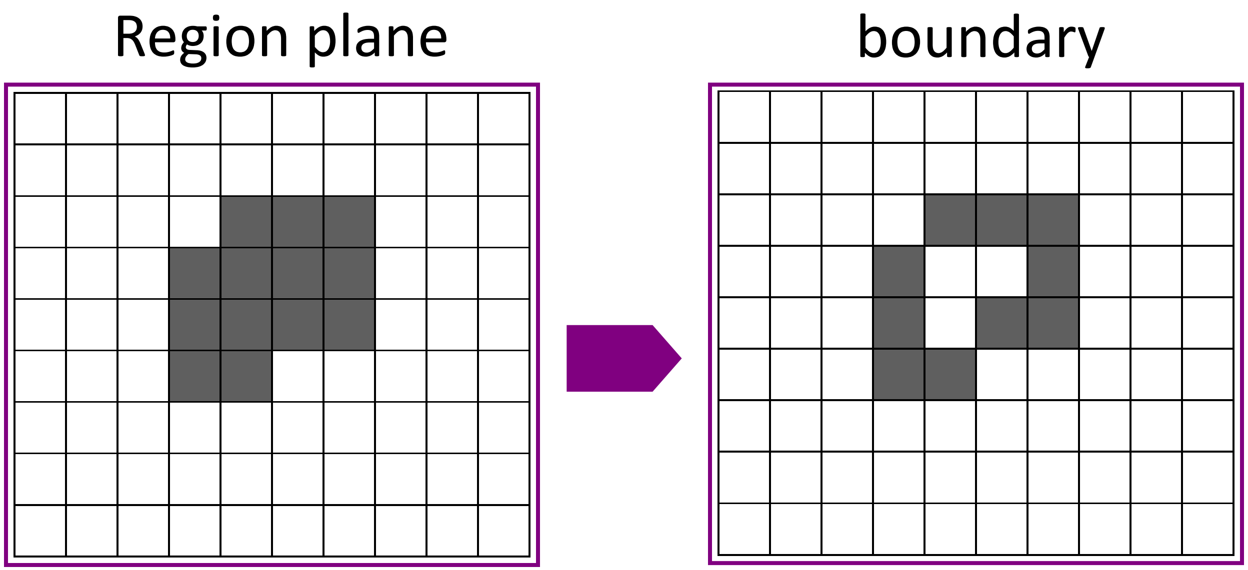
Region Features

- Basics
  - Perimeter
  - Area
  - Orientation



- Rotation/translation/scale invariant
  - Compactness = $perimeter^2/area$
  - Rectangularity = AreaRect/AreaObject
  - Euler number = #regions - #holes
  - Convexity = AreaConvexHull/AreaObject

Perimeter



P = number of pixels on boundary

or

P = number of horizontal steps + number of vertical steps + $\sqrt{2}$ x number of diagonal steps

Other features

Table 1. Representative shape descriptors

$$Formfactor = \frac{4\pi \cdot Area}{Perimeter^2}$$

$$Roundness = \frac{4 \cdot Area}{\pi \cdot Max\ Diameter^2}$$

$$Aspect\ Ratio = \frac{Max\ Diameter}{Min\ Diameter}$$

$$Elongation = \frac{Fiber\ Length}{Fiber\ Width}$$

$$Curl = \frac{Length}{Fiber\ Length}$$

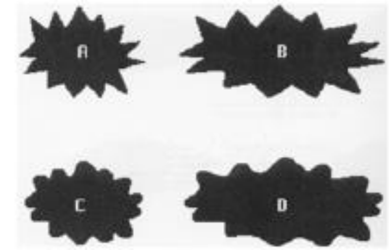
$$Convexity = \frac{Convex\ Perimeter}{Perimeter}$$

$$Solidity = \frac{Area}{Convex\ Area}$$

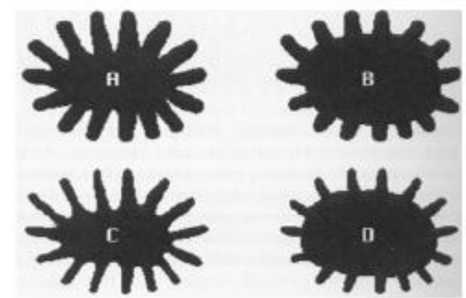
$$Compactness = \frac{\sqrt{\left(\frac{4}{\pi}\right) Area}}{Max\ Diameter}$$

$$Modification\ Ratio = \frac{Inscribed\ Diameter}{Maximum\ Diameter}$$

$$Extent = \frac{Net\ Area}{Bounding\ Rectangle}$$



	Formfactor	Aspect Ratio
A	0.257	1.339
B	0.256	2.005
C	0.459	1.294
D	0.457	2.017



	Roundness	Convexity	Solidity	Compactness
A	0.587	0.351	0.731	0.766
B	0.584	0.483	0.782	0.764
C	0.447	0.349	0.592	0.668
D	0.589	0.497	0.714	0.768

OCR Feature Extraction

- In feature extraction stage **each character is represented as a feature vector**, which becomes its identity.
- **The major goal of feature extraction is to extract a set of features, which maximizes the recognition rate with the least amount of elements.**
- Due to the nature of handwriting with its high degree of variability and imprecision obtaining these features, is a difficult task. Feature extraction methods are based on 2 types of features:
    - Statistical
    - Structural

Statistical Features

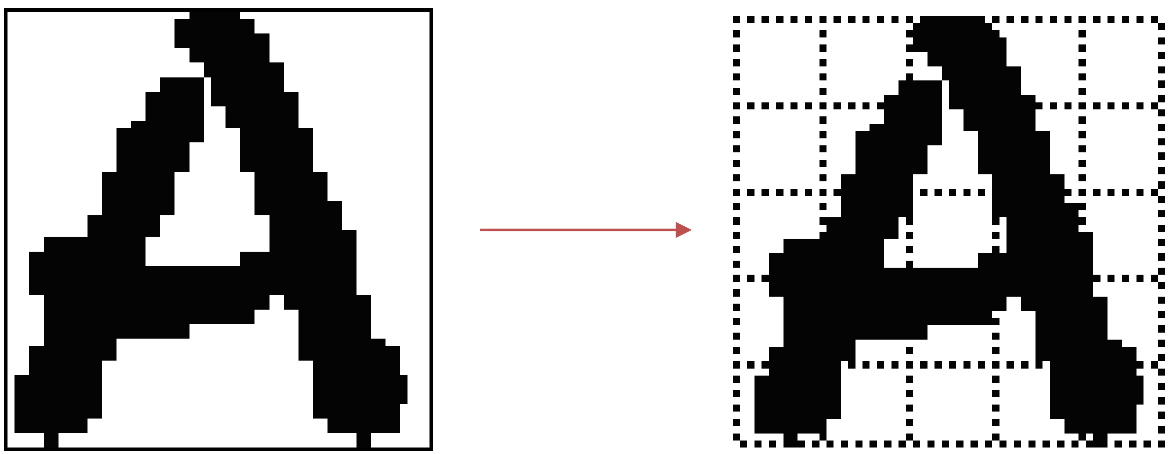
- **Representation of a character image by statistical distribution** of points takes care of style variations to some extent.

- The major statistical features used for character representation are:
  - Zoning
  - Projections

Zoning

□ **The character image is divided into NxM zones**. From each zone features are extracted to form the feature vector.

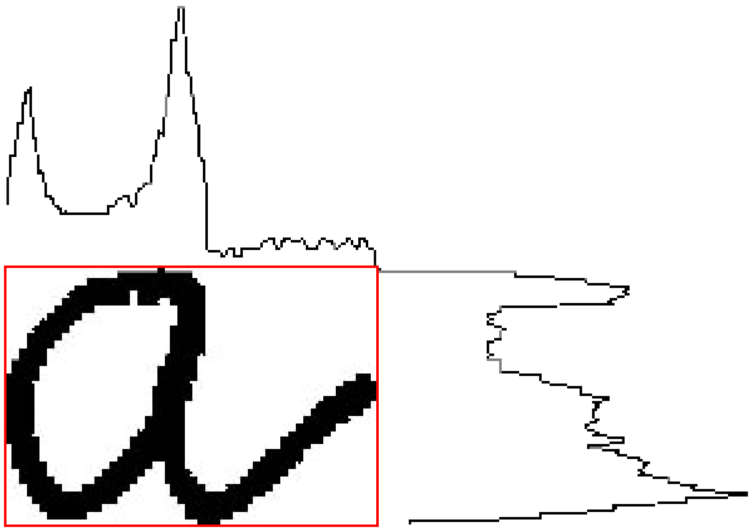
□***The goal of zoning is to obtain the local characteristics instead of global characteristics***.



Projection Histograms

- The basic idea behind using projections is that character images, which are 2-D signals, **can be represented as 1-D signal**. These features, although independent to noise and deformation, depend on rotation.

- **Projection histograms count the number of pixels in each column and row of a character image**. Projection histograms can separate characters such as “m” and “n” .



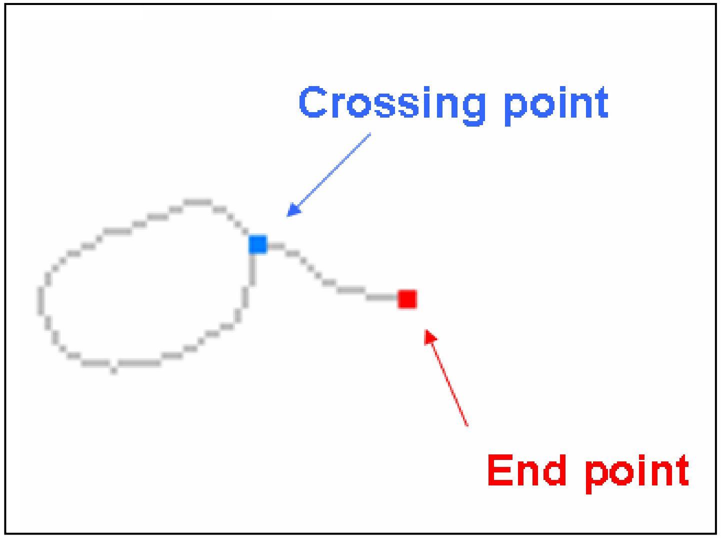
Structural Features

- Characters can be represented by structural features **with high tolerance to distortions and style variations**.

- This type of representation may also encode **some knowledge about the structure of the object** or may provide some knowledge as to what sort of components make up that object.

- **Structural features are based on topological and geometrical properties of the character**, such as *cross points, loops, branch points, strokes and their directions, inflection between two points, horizontal curves at top or bottom*, etc.

Structural Features



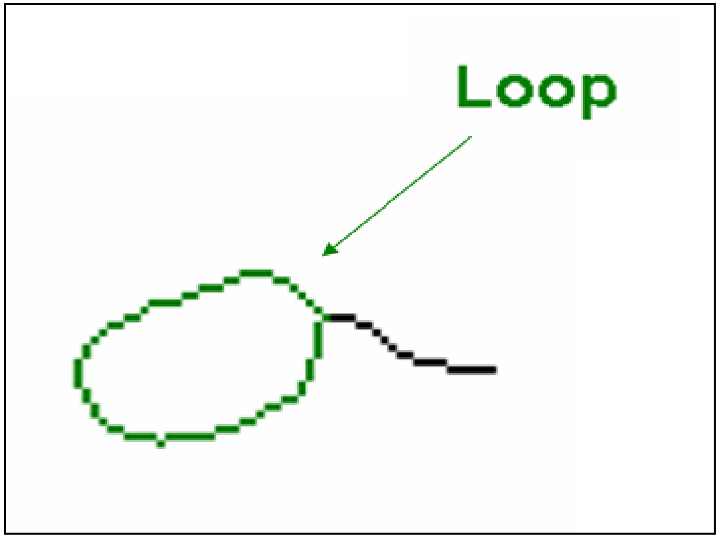


Image Retrieval Application

- Content-Based Image Retrieval System

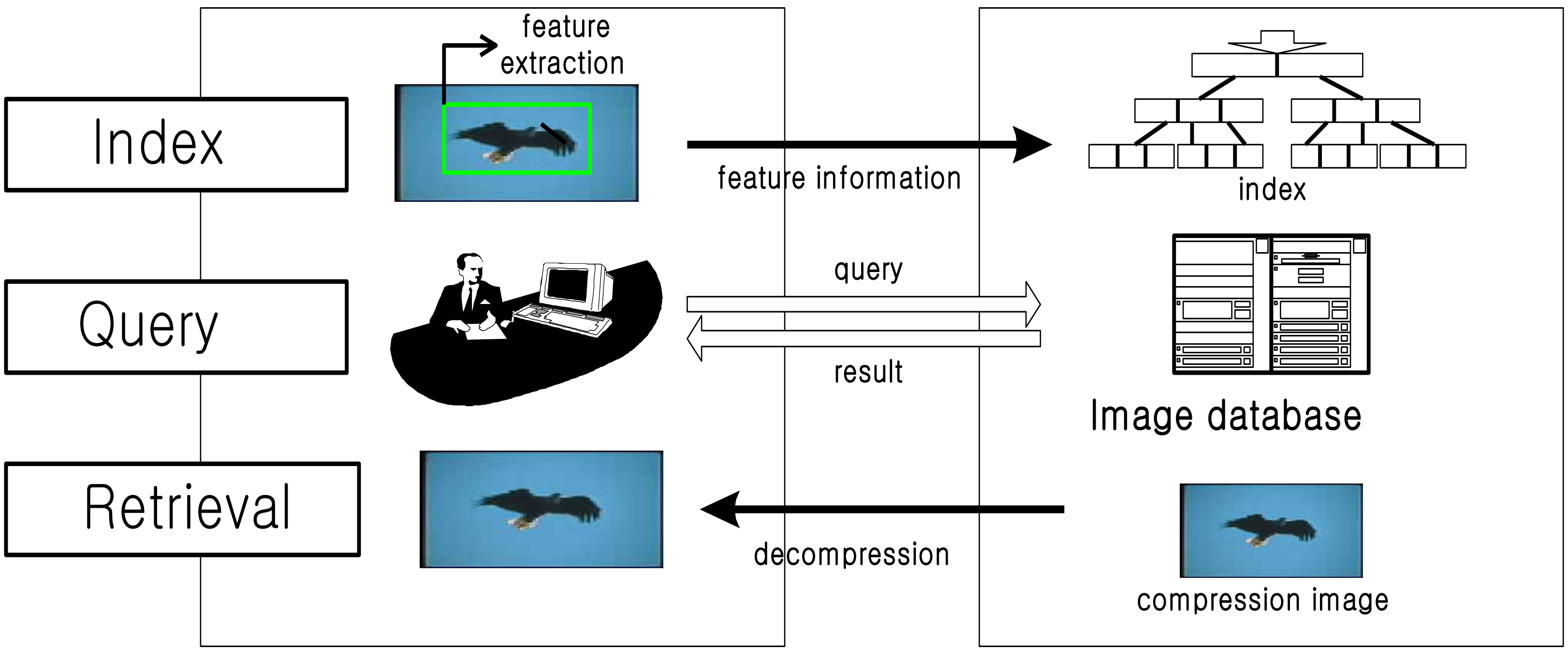
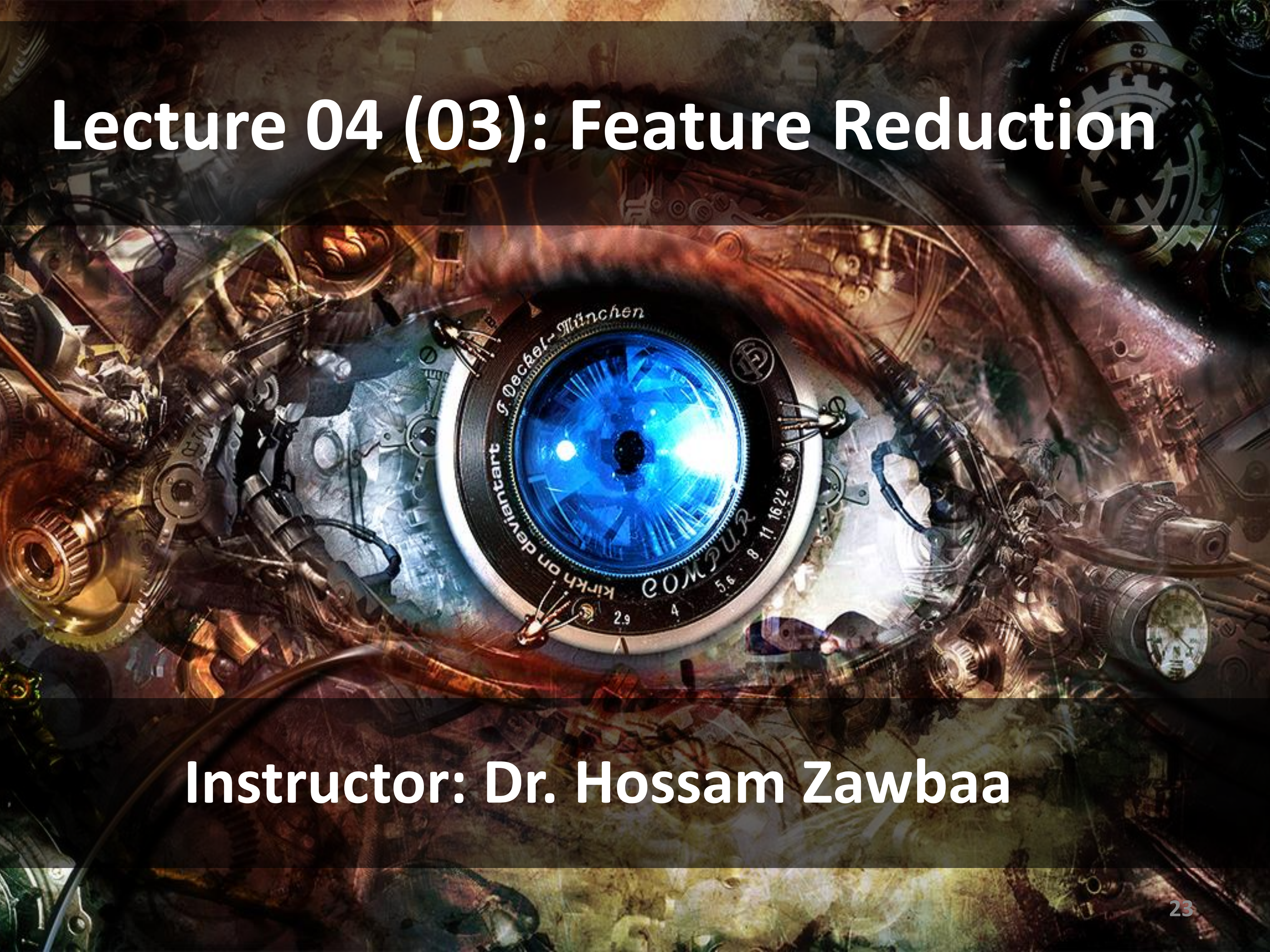


Image Retrieval Application

- Color Features for Image Indexing
  - Color Histogram
    - an estimate of the probability of occurrence of color intensities
    - simple and geometric invariance(translation, rotation, and scaling)
    - lack of spatial information of objects
  - Dominant Colors
  - Color Moments
    - moment invariants for color distribution

Lecture 04 (03): Feature Reduction

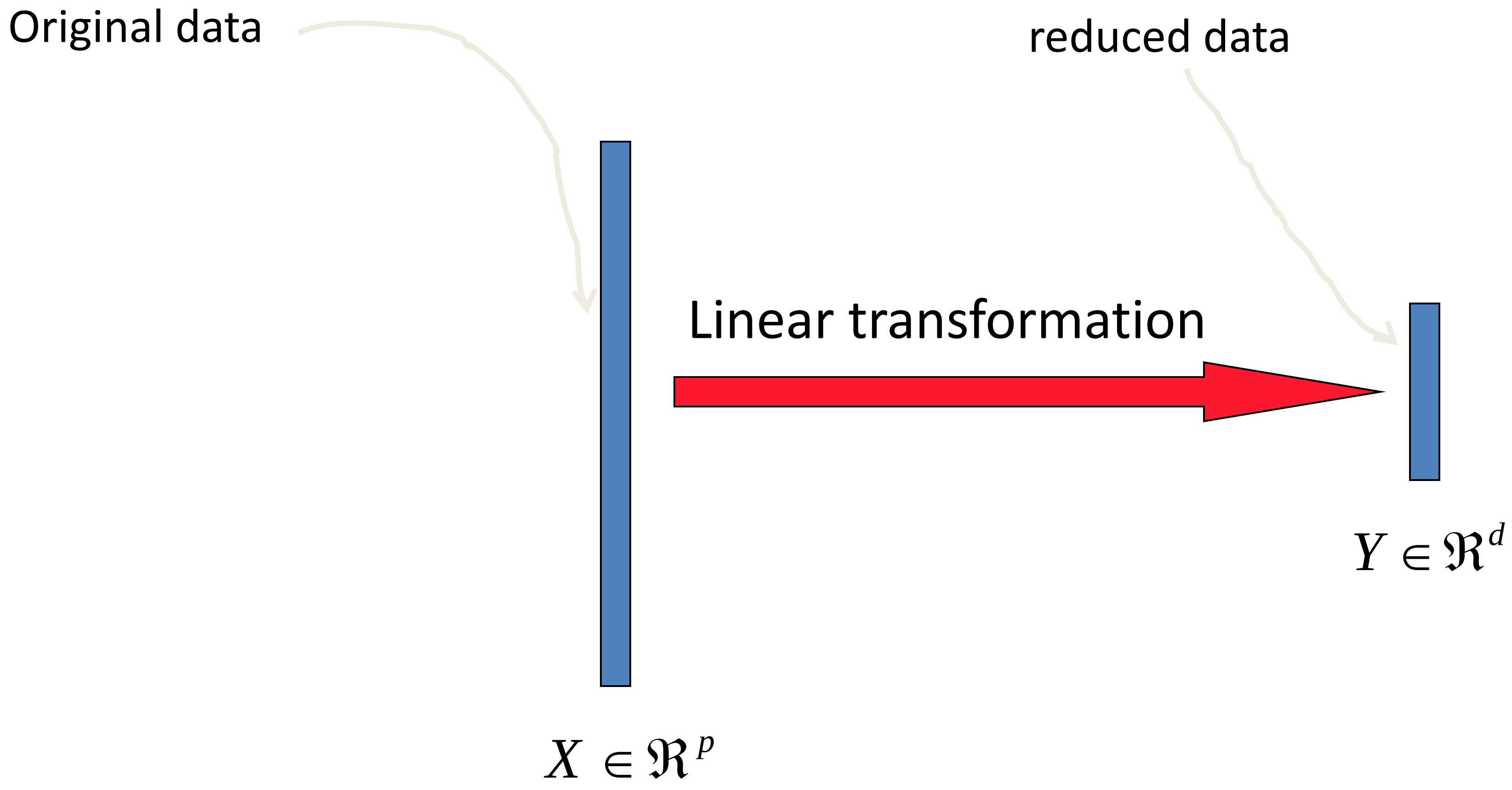


Instructor: Dr. Hossam Zawbaa

Feature reduction

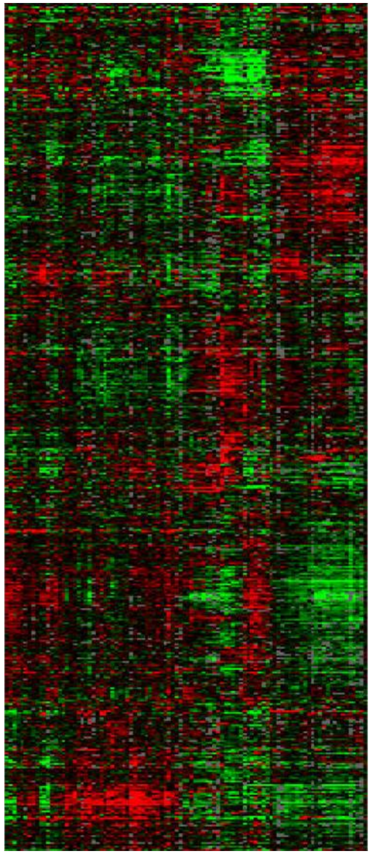
- **Feature reduction refers to the mapping of the original high-dimensional data onto a lower-dimensional space**.
  - Criterion for feature reduction can be different based on different problem settings.
    - **Unsupervised setting**: minimize the information loss
    - **Supervised setting**: maximize the class discrimination

Feature reduction



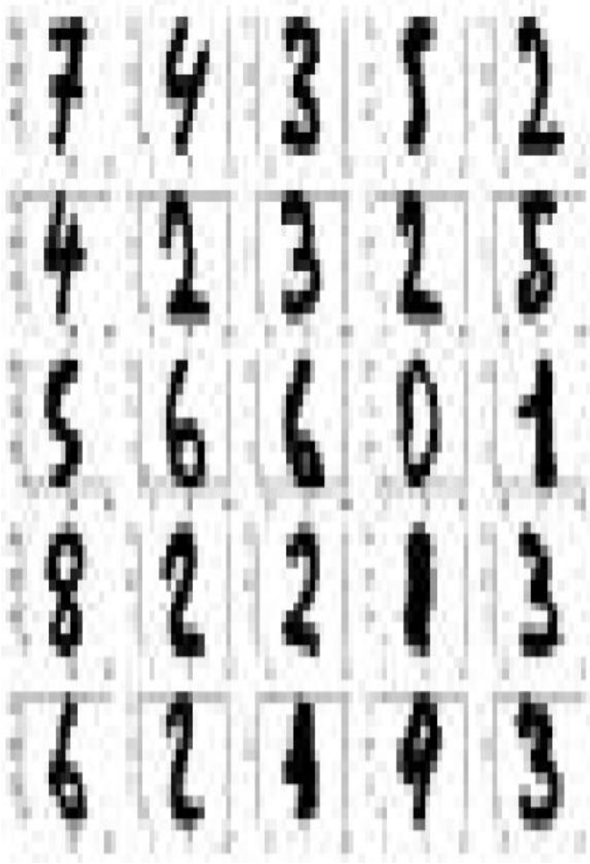
Feature reduction

High-dimensional data



Gene expression

Face images



Handwritten digits

Feature reduction

- Most machine learning and data mining techniques may not be effective for high-dimensional data
  - Curse of Dimensionality
  - Query accuracy and efficiency degrade rapidly as the dimension increases.

- The essential dimension may be small.
  - For example, the number of genes responsible for a certain type of disease may be small.

Feature reduction

- Visualization: projection of high-dimensional data onto 2D or 3D.
- Data compression: efficient storage and retrieval.
- Noise removal: positive effect on query accuracy.

Feature reduction
Applications

- Face recognition
- Handwritten digit recognition
- Text mining
- Image retrieval
- Microarray data analysis
- Protein classification

Feature reduction algorithms

- Unsupervised
  - Latent Semantic Indexing (LSI): truncated SVD
  - Independent Component Analysis (ICA)
  - **Principal Component Analysis (PCA)**
  - Canonical Correlation Analysis (CCA)
- Supervised
  - Linear Discriminant Analysis (LDA)
- Semi-supervised
  - Research topic

Principal Component Analysis

- Principal component analysis (PCA)
  - Reduce the dimensionality of a data set by finding a new set of variables, smaller than the original set of variables
  - Retains most of the sample's information.
  - Useful for the compression and classification of data.

- By information, we mean the variation present in the sample, given by the correlations between the original variables.
- The new variables, called principal components (PCs), are uncorrelated, and are ordered by the fraction of the total information each retains.

Geometric Rationale of PCA

- degree to which the variables are linearly correlated is represented by their **covariances**.

$$C_{ij} = \frac{1}{n-1}\sum_{m=1}^{n}\left(X_{im} - \bar{X}_i\right)\left(X_{jm} - \bar{X}_j\right)$$

- C_{ij}: Covariance of variables *i* and *j*
- $\sum_{m=1}^{n}$: Sum over all *n* objects
- X_{im}: Value of variable *i* in object *m*
- $\bar{X}_i$: Mean of variable *i*
- X_{jm}: Value of variable *j* in object *m*
- $\bar{X}_j$: Mean of variable *j*

If covariance is <u>positive</u>, both dimensions increase together. If <u>negative</u>, as one increases, the other decreases. <u>Zero</u>: independent of each other.

	Hours(H)	*Mark(M)*
Data	9	39
	15	56
	25	93
	14	61
	10	50
	18	75
	0	32
	16	85
	5	42
	19	70
	16	66
	20	80
Totals	167	749
Averages	13.92	62.42

$$\begin{pmatrix} \text{cov}(H,H) & \text{cov}(H,M) \\ \text{cov}(M,H) & \text{cov}(M,M) \end{pmatrix}$$

$$= \begin{pmatrix} \text{var}(H) & 104.5 \\ 104.5 & \text{var}(M) \end{pmatrix}$$

$$= \begin{pmatrix} 47.7 & 104.5 \\ 104.5 & 370 \end{pmatrix}$$

Covariance matrix

Covariance:

H	M	$(H_i - \bar{H})$	$(M_i - \bar{M})$	$(H_i - \bar{H})(M_i - \bar{M})$
9	39	-4.92	-23.42	115.23
15	56	1.08	-6.42	-6.93
25	93	11.08	30.58	338.83
14	61	0.08	-1.42	-0.11
10	50	-3.92	-12.42	48.69
18	75	4.08	12.58	51.33
0	32	-13.92	-30.42	423.45
16	85	2.08	22.58	46.97
5	42	-8.92	-20.42	182.15
19	70	5.08	7.58	38.51
16	66	2.08	3.58	7.45
20	80	6.08	17.58	106.89
Total				1149.89
Average				104.54

Table 2.2: 2-dimensional data set and covariance calculation

- Eigenvectors:
  - C * V = λ*V
  - Let **M be** an $n \times n$ matrix.
    - **v** is an *eigenvector* of M if M × **v** = λ**v**
    - λ is called the *eigenvalue* associated with **v**
  - For any eigenvector **v** of **M** and scalar *a*,

$$\mathbf{M} \times a\mathbf{v} = \lambda a\mathbf{v}$$

  - Thus, you can always choose eigenvectors of length 1:

$$\sqrt{v_1^{\ 2} + \ldots + v_n^{\ 2}} = 1$$

  - Thus eigenvectors can be used as a new basis for a n-dimensional vector space.

PCA example

1. Given original data set S = {$\mathbf{x}^1$, ..., $\mathbf{x}^k$}, **produce new set** by subtracting the mean

Data =

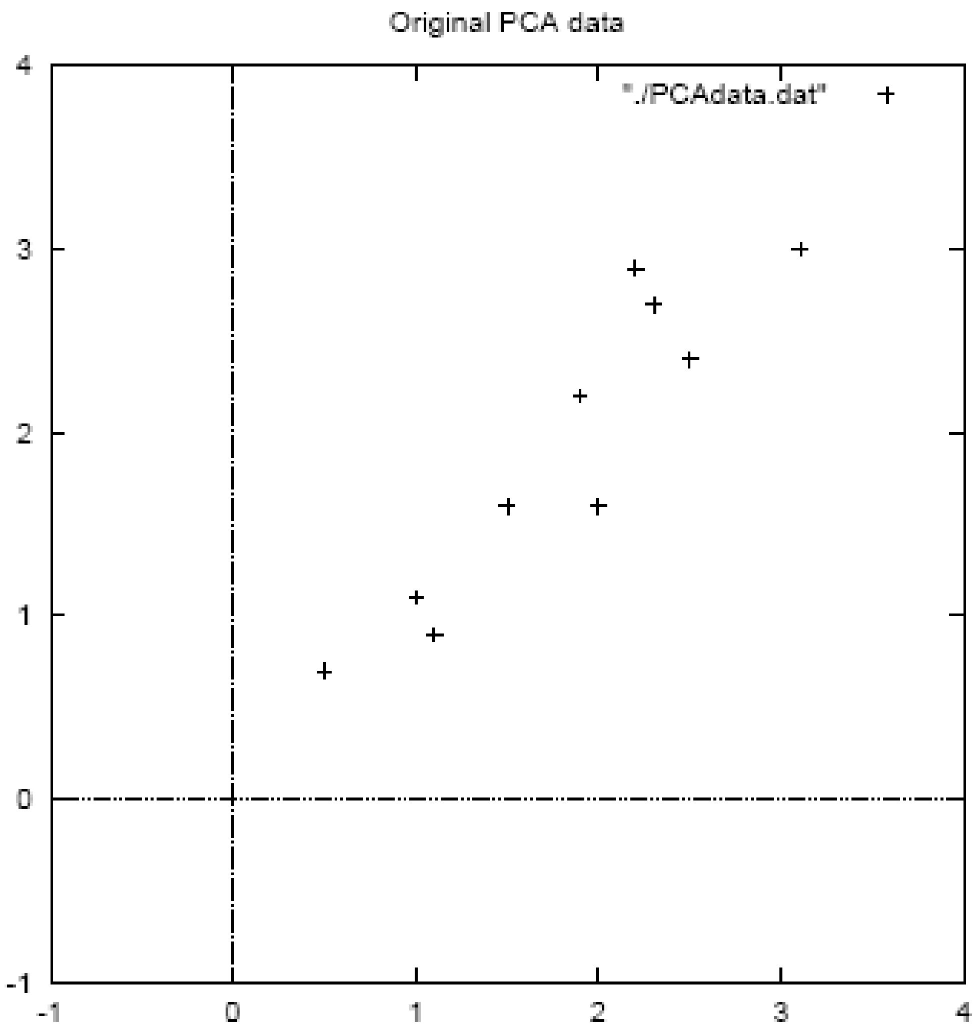
x	y
2.5	2.4
0.5	0.7
2.2	2.9
1.9	2.2
3.1	3.0
2.3	2.7
2	1.6
1	1.1
1.5	1.6
1.1	0.9

Mean: 1.81 1.91

DataAdjust =

x	y
.69	.49
-1.31	-1.21
.39	.99
.09	.29
1.29	1.09
.49	.79
.19	-.31
-.81	-.81
-.31	-.31
-.71	-1.01

Mean: 0 0



Original PCA data
"./PCAdata.dat"
4
3
2
1
0
-1
-1
0
1
2
3
4

2. Calculate the covariance matrix:

$$cov = \begin{array}{c} \\ \mathbf{x} \\ \mathbf{y} \end{array} \begin{array}{c} \begin{array}{cc} \mathbf{x} & \mathbf{y} \end{array} \\ \begin{pmatrix} .616555556 & .615444444 \\ .615444444 & .716555556 \end{pmatrix} \end{array}$$

3. Calculate the (unit) eigenvectors and eigenvalues of the covariance matrix:

$$eigenvalues = \begin{pmatrix} .0490833989 \\ 1.28402771 \end{pmatrix}$$

$$eigenvectors = \begin{pmatrix} -.735178656 & -.677873399 \\ .677873399 & -.735178656 \end{pmatrix}$$

4. Order eigenvectors by eigenvalue, highest to lowest.

$$\mathbf{v}_1 = \begin{pmatrix} -.677873399 \\ -.735178956 \end{pmatrix} \quad \lambda = 1.28402771$$

$$\mathbf{v}_2 = \begin{pmatrix} -.735178956 \\ .677873399 \end{pmatrix} \quad \lambda = .0490833989$$

In general, you get *n* components. To reduce dimensionality to *p*, ignore $n-p$ components at the bottom of the list.

Construct new feature vector.

Feature vector = ($\mathbf{v}_1$, $\mathbf{v}_2$, ...$\mathbf{v}_p$)

$$FeatureVector1 = \begin{pmatrix} -.677873399 & -.735178956 \\ -.735178956 & .677873399 \end{pmatrix}$$

or reduced dimension feature vector :

$$FeatureVector2 = \begin{pmatrix} -.677873399 \\ -.735178956 \end{pmatrix}$$

5. Derive the new data set.

$$TransformedData = RowFeatureVector \times RowDataAdjust$$

$$RowFeatureVector1 = \begin{pmatrix} -.677873399 & -.735178956 \\ -.735178956 & .677873399 \end{pmatrix}$$

$$RowFeatureVector2 = \begin{pmatrix} -.677873399 & -.735178956 \end{pmatrix}$$

This gives original data in terms of chosen components (eigenvectors)—that is, along these axes.

$$RowDataAdjust = \begin{pmatrix} .69 & -1.31 & .39 & .09 & 1.29 & .49 & .19 & -.81 & -.31 & -.71 \\ .49 & -1.21 & .99 & .29 & 1.09 & .79 & -.31 & -.81 & -.31 & -1.01 \end{pmatrix}$$

Transformed Data=

x	y
-.827970186	-.175115307
1.77758033	.142857227
-.992197494	.384374989
-.274210416	.130417207
-1.67580142	-.209498461
-.912949103	.175282444
.0991094375	-.349824698
1.14457216	.0464172582
.438046137	.0177646297
1.22382056	-.162675287

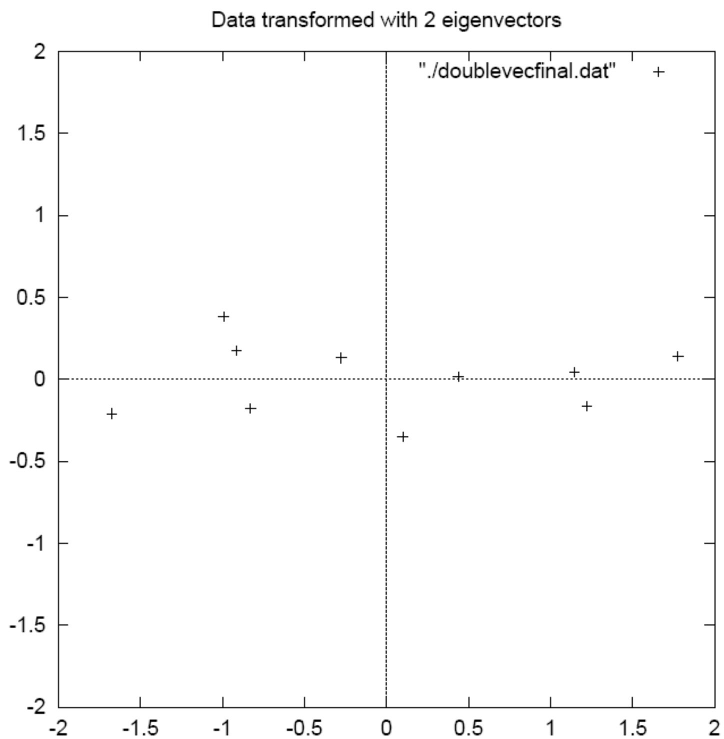
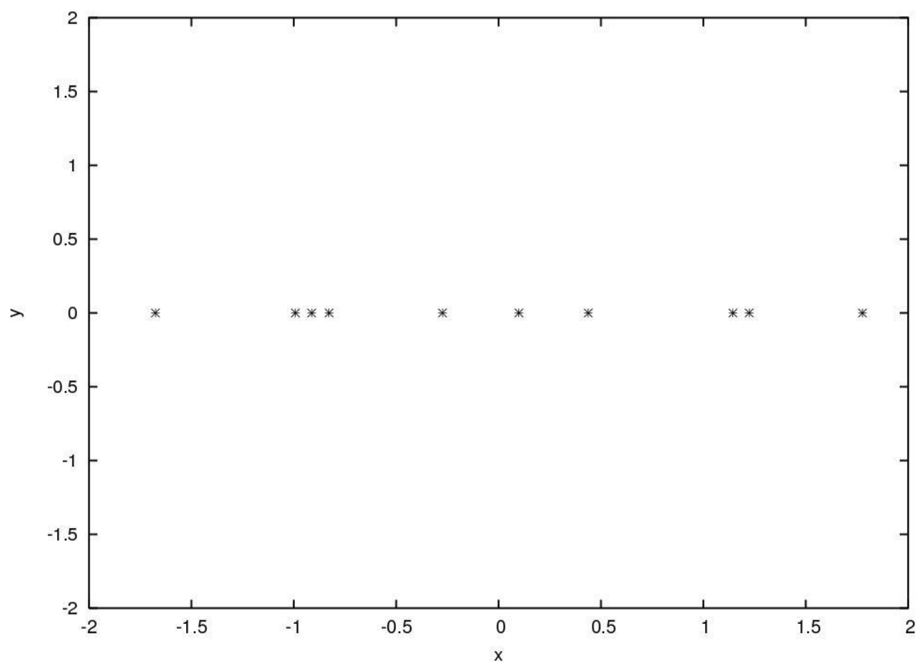
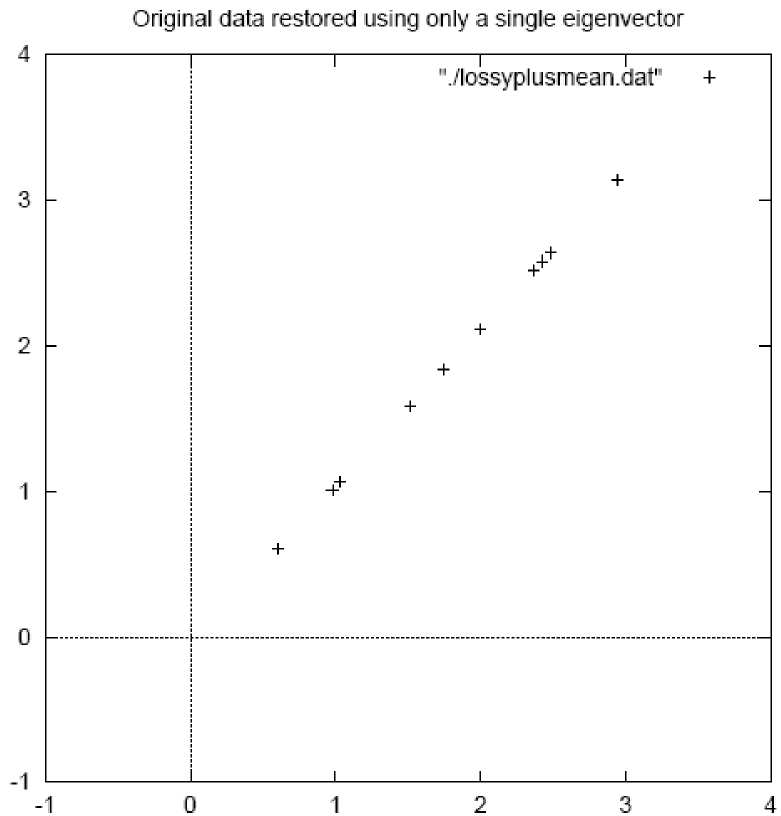
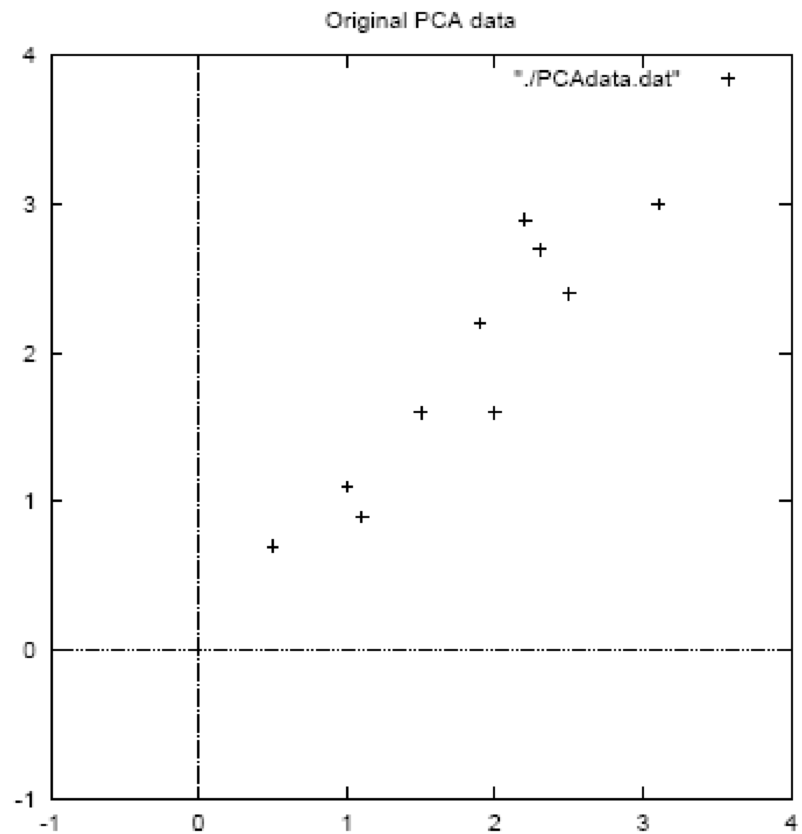


Figure 3.3: The table of data by applying the PCA analysis using both eigenvectors, and a plot of the new data points.

Transformed Data (Single eigenvector)
x
-.827970186
1.77758033
-.992197494
-.274210416
-1.67580142
-.912949103
.0991094375
1.14457216
.438046137
1.22382056




Original data restored using only a single eigenvector
"./lossyplusmean.dat"
4
3
2
1
0
-1
-1
0
1
2
3
4


Original PCA data
"./PCAdata.dat"
4
3
2
1
0
-1
-1
0
1
2
3
4

Applications of PCA

- *Eigenfaces for recognition*. Turk and Pentland. 1991.
- *Principal Component Analysis for clustering gene expression data*. Yeung and Ruzzo. 2001.
- *Probabilistic Disease Classification of Expression-Dependent Proteomic Data from Mass Spectrometry of Human Serum.* Lilien. 2003.

PCA for image compression




d=1 d=2 d=4 d=8

d=16 d=32 d=64 d=100 Original Image