Computer Vision (CSE3010)

Dr. Susant Kumar Panigrahi
Assistant Professor
School of Electrical & Electronics Engineering
VIT BHOPAL

Module-1 Syllabus

Digital Image Formation And Low Level Processing:

- Overview and State-of-the-art, Fundamentals of Image Formation, Transformation: Orthogonal, Euclidean, Affine, Projective, Fourier Transform,
- Convolution and Filtering, Image Enhancement, Restoration, Histogram Processing.

Module-2 Syllabus

Depth Estimation And Multi-Camera Views:

Depth Estimation and Multi-Camera Views: Perspective, Binocular Stereopsis: Camera and Epipolar Geometry; Homography, Rectification, DLT, RANSAC, 3-D reconstruction framework; Auto-calibration. apparel.

Module-3 Syllabus

Feature Extraction And Image Segmentation:

- **Feature Extraction**: Edges - Canny, LOG, DOG; Line detectors (Hough Transform), Corners - Harris and Hessian Affine, Orientation Histogram, SIFT, SURF, HOG, GLOH, Scale-Space Analysis- Image Pyramids and Gaussian derivative filters, Gabor Filters and DWT.
- **Image Segmentation:** Region Growing, Edge Based approaches to segmentation, Graph-Cut, Mean-Shift, MRFs, Texture Segmentation; Object detection.

Module-4 Syllabus

Pattern Analysis And Motion Analysis:

- **Pattern Analysis**: Clustering: K-Means, K-Medoids, Mixture of Gaussians, Classification: Discriminant Function, Supervised, Un-supervised, Semi-supervised; Classifiers: Bayes, KNN, ANN models;
- **Dimensionality Reduction**: PCA, LDA, ICA; Non-parametric methods. Motion Analysis: Background Subtraction and Modelling, Optical Flow, KLT, Spatio-Temporal Analysis, Dynamic Stereo; Motion parameter estimation.

Module-5 Syllabus

Shape From X:

Light at Surfaces; Phong Model; Reflectance Map;

Albedo estimation; Photometric Stereo; Use of Surface Smoothness

Constraint; Shape from Texture, color, motion and edges.

Guest Lecture on Contemporary Topics

Text Books

1. Richard Szeliski, Computer Vision: Algorithms and Applications, Springer-Verlag London Limited 2011.
2. Computer Vision: A Modern Approach, D. A. Forsyth, J. Ponce, Pearson Education, 2003.

Reference Book(s):

1. R.C. Gonzalez and R.E. Woods, Digital Image Processing, Addison- Wesley, 1992.
2. Richard Hartley and Andrew Zisserman, Multiple View Geometry in Computer Vision, Second Edition, Cambridge University Press, March 2004.
3. K. Fukunaga; Introduction to Statistical Pattern Recognition, Second Edition, Academic Press, Morgan Kaufmann, 1990.

Required Tools/Software/IDLE:

1. Python/jupyter-notebook/google-colab
2. OpenCV
3. MATLAB

Indicative List of Experiments:

1. Implement image preprocessing and Edge
2. Implement camera calibration methods
3. Implement Projection
4. Determine depth map from Stereo pair
5. Construct 3D model from Stereo pair
6. Implement Segmentation methods
7. Construct 3D model from defocus image
8. Construct 3D model from Images
9. Implement optical flow method
10. Implement object detection and tracking from video
11. Face detection and Recognition
12. Object detection from dynamic Background for Surveillance
13. Content based video retrieval
14. Construct 3D model from single image

Computer Vision

Unit – 04

Dimensionality Reduction: PCA

Standing on the shoulder of Giants: Ref: Few Slides borrowed from:

1. Prof. Shree Nayar, *First Principles of Computer Vision is a lecture series.*
2. Prof. Mubarak Shah, *Computer Vision Video Lectures.*

KL Transform or Hoteling Transform or PCA

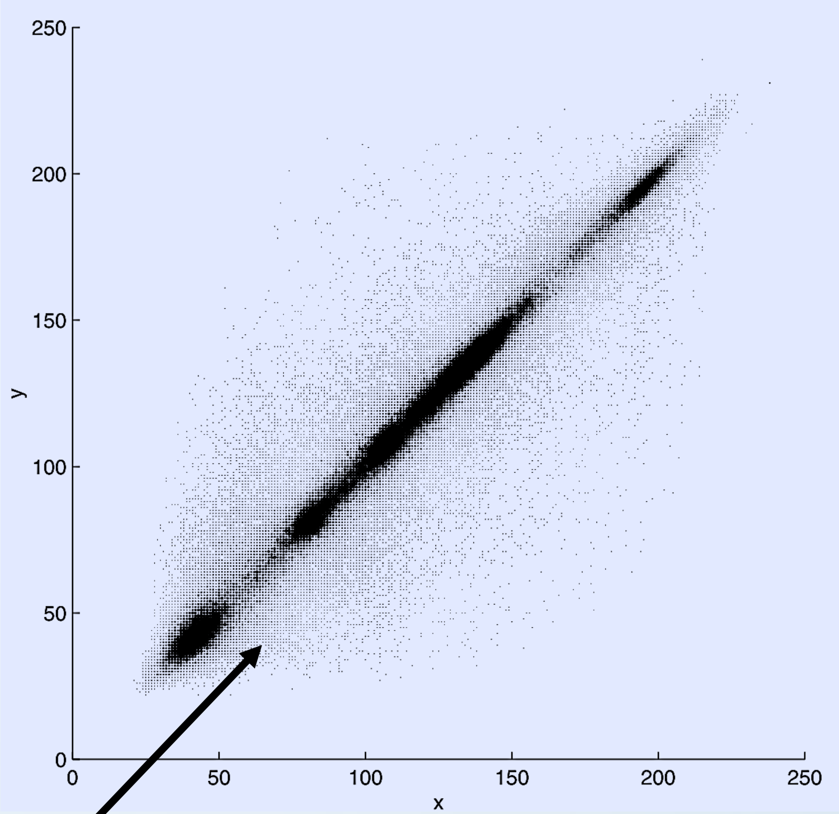
- ✓ Karhunen and Loève (KL) Transform named after Kari Karhunen and Michel Loève. It is also known as Hoteling transform or Eigen vector transform.

- ✓ The KLT analyzes a set of vectors or images, into basis functions or images where the choice of the basis set depends on the statistics of the image set - depends on image covariance matrix.

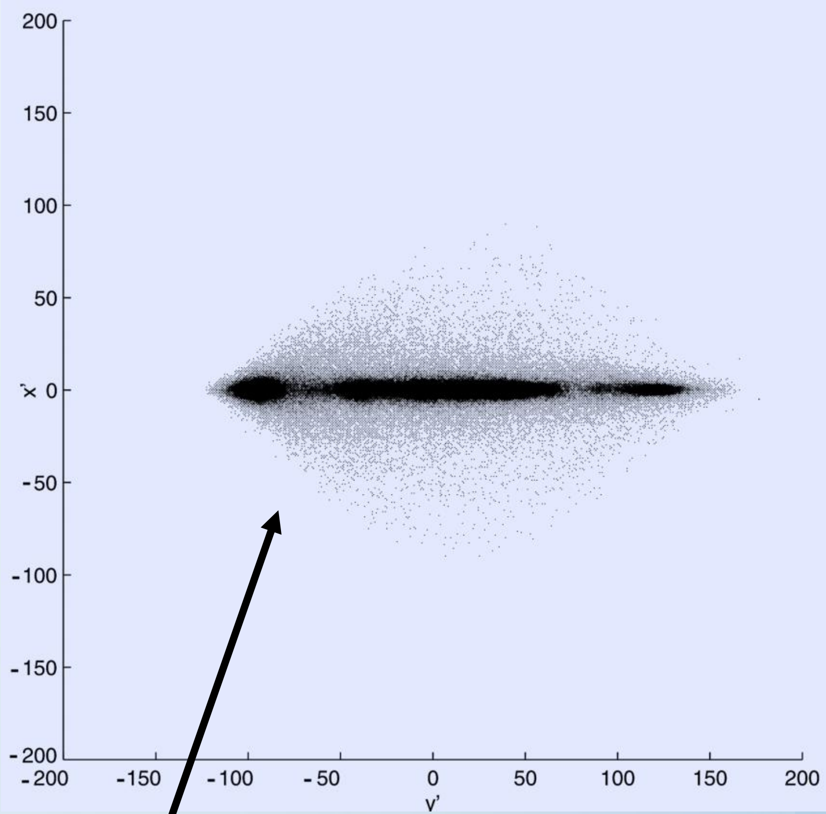
- ✓ Moreover, the importance of KL transform is that it yields the best such basis in the sense that it minimizes the total mean square error (MSE).

Data Decorrelation

Grayscale image of dimension 512×512. Pixel intensities between $[0, 255]$. Require 8-bits to represent each pixel intensity.



Pixels in the image with x-coordinate being its gray-level values and y-coordinate being the gray-level values of its neighbor to the right.



Same information represented in a rotated coordinate system. The new coordinate is rotated by 45^0 about the center.

Data correlation in this coordinate system is high

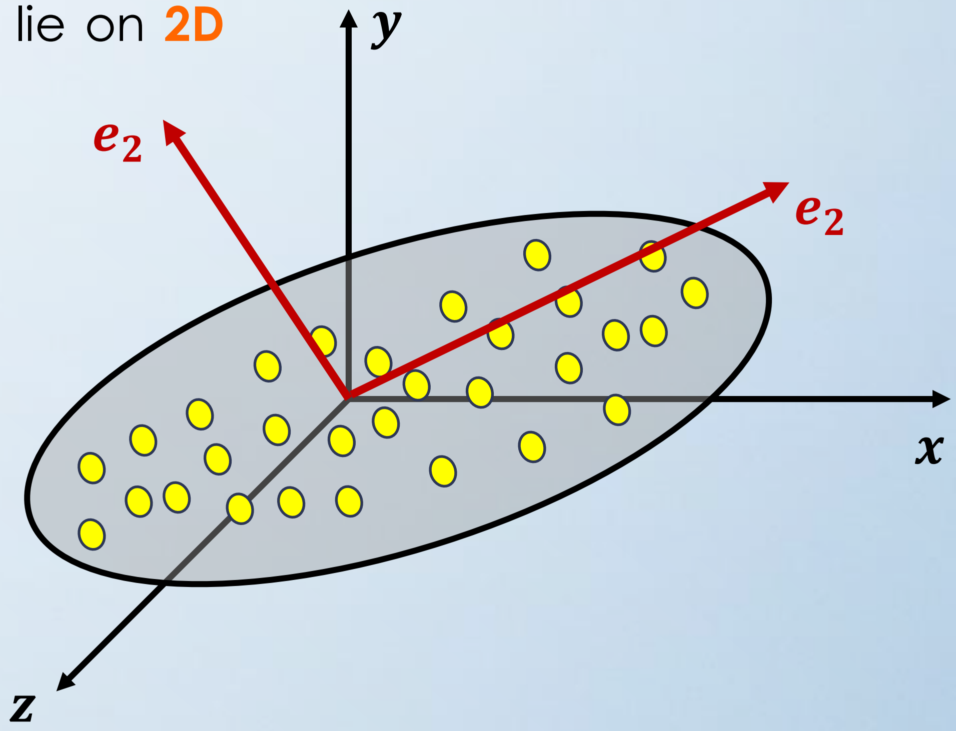
Data is relatively decorelated and requires less number of bits to represent. As the dynamic range along x' direction is less/.

Decorrelation & Dimensionality Reduction

- ✓ Karhunen and Loève (KL) Transform also Principal Component Analysis (PCA).

- ✓ Consider a distribution of points in **3D space** that actually lie on **2D plane**.

- ✓ It is redundant to represent each point in with 3 coordinate values.



- ✓ If we use a **new 2D coordinate system,** $\{\boldsymbol{e_1}, \boldsymbol{e_2}\}$, that lie on the plane of the distribution, each point can be represented with just 2 coordinates.

KL Transform Formulation

- ✓ First data needs to be represented in a vector as:

x_i may represent the gray-levels of the image i or it may be the location information of any image i.

$$\underline{X} = \begin{bmatrix} x_1 \\ x_2 \\ . \\ . \\ x_n \end{bmatrix}$$

- ✓ We find out the mean value as:

$$\mu_x = E\{\underline{X}\}$$

- ✓ The covariance of the above data is:

$$C_X = E\left\{(\underline{X} - \mu_x)(\underline{X} - \mu_x)^T\right\}$$

C_X is of dimension $n \times n$.

- ✓ The diagonal values of C_X: C_{ii} represents the variance of the element x_i.
- ✓ The c_{ij} represents the covariance between x_i and x_j.
- ✓ C_X: Real and Symmetry.

KL Transform Formulation

- ✓ As C_X is real and symmetry, we can find a set of orthogonal Eigen vectors of the above matrix.
- ✓ Assuming e_i = i^{th}Eigen vector of C_X corresponds to:

  λ_i = Eigen value of C_X.

  Also assume, n-Eigen values are arranged:

$$\lambda_j \geq \lambda_{j+1};\ for\ j = 1, 2, 3\ \ldots n$$

- ✓ Let us form a matrix A that consisting of all the Eigen vectors derived from the corresponding Eigen values arranged in a descending order as:

$$A = \begin{bmatrix} \uparrow & \uparrow & & \uparrow \\ e_1 & e_2 & \cdots & e_n \\ \downarrow & \downarrow & \downarrow & \downarrow \end{bmatrix}$$

Eigen vector corresponds to largest Eigen value

Eigen vector corresponds to smallest Eigen value

KL Transform Formulation

- ✓ A is the transform matrix for KL Transform.
- ✓ Now KL transform is:

$$Y = A\,(X - \mu_X)$$

The inverse transform is very straight forward:

$$X = A^T Y + \mu_X$$

Properties of Y

- ✓ $\mu_y = 0$ (Mean of transformed matrix y is 0.)
- ✓ C_y = Covariance matrix of y = AC_XA^T: it is a diagonal matrix and the elements of diagonal matrix (C_Y) are Eigen values of C_X

$$C_Y = \begin{bmatrix} \lambda_1 & \cdots & \\ \vdots & \ddots & \vdots \\ & \cdots & \lambda_n \end{bmatrix}$$

- ✓ Elements of off-diagonal values of C_Y is $\mathbf{0}$. It indicates that the elements of Y vectors are uncorrelated.

KL Transform Implication & Example

✓ **The given image is represented in a correlated space. Apply KL Transform to represent the object in the image in decorelated space:**

Step – 01: $X = \left\{\begin{pmatrix}3\\4\end{pmatrix}, \begin{pmatrix}4\\3\end{pmatrix}, \begin{pmatrix}4\\4\end{pmatrix}, \begin{pmatrix}4\\5\end{pmatrix}, \begin{pmatrix}5\\4\end{pmatrix}, \begin{pmatrix}5\\5\end{pmatrix}, \begin{pmatrix}5\\6\end{pmatrix}, \begin{pmatrix}6\\5\end{pmatrix}\right\}$

Representing in vector:

$$X = \begin{bmatrix}3 & 4 & 4 & 4 & 5 & 5 & 5 & 6\\ 4 & 3 & 4 & 5 & 4 & 5 & 6 & 5\end{bmatrix}$$

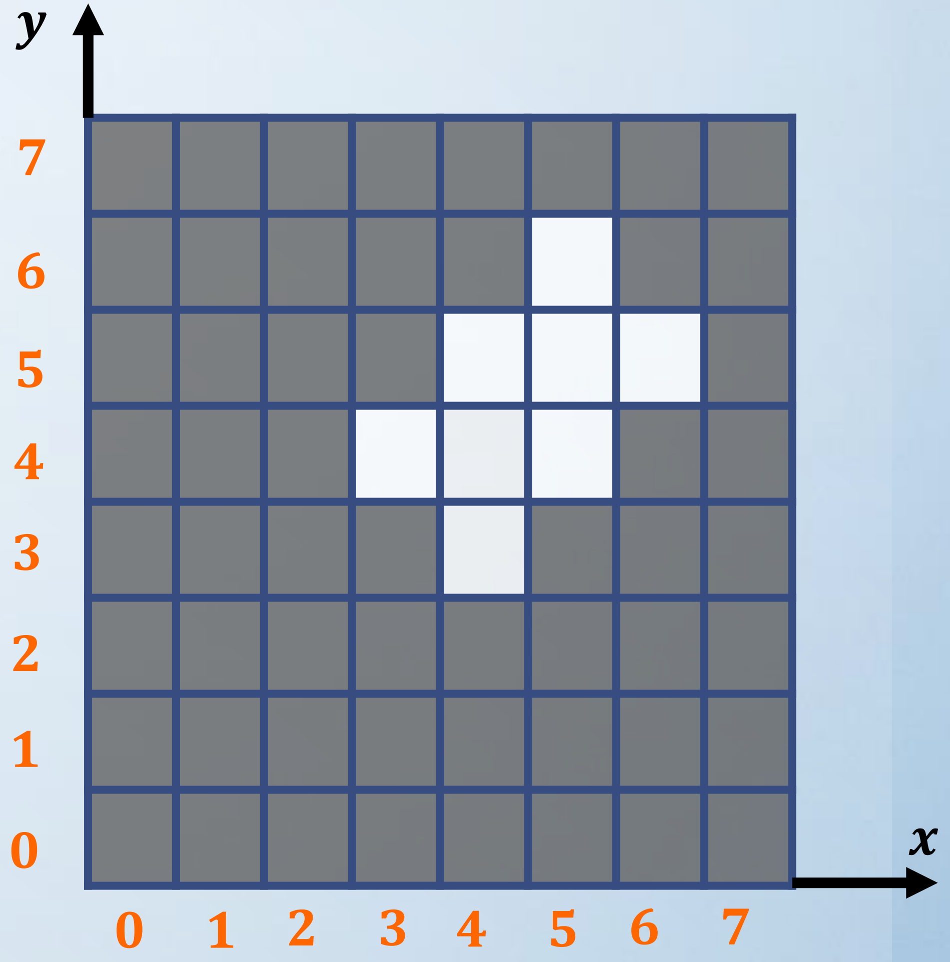
Step – 02: Finding mean along row-wise:

$$\mu_X = \begin{bmatrix}4.5\\ 4.5\end{bmatrix}$$

Step – 03: Covariance matrix:

$$C_X = E\{(X - \mu_X)(X - \mu_X)^T\}$$

$$x_1 = \begin{pmatrix}3\\4\end{pmatrix};\ x_1 - \mu_X = \begin{pmatrix}-1.5\\-0.5\end{pmatrix};\ (x_1 - \mu_X)(x_1 - \mu_X)^T = \begin{bmatrix}2.25 & 0.75\\ 0.75 & 0.25\end{bmatrix}$$



KL Transform Implication & Example

Step – 03: $(x_2 - \mu_X)(x_2 - \mu_X)^T = \begin{bmatrix} 0.25 & 0.75 \\ 0.75 & 2.25 \end{bmatrix}$

$$(x_3 - \mu_X)(x_3 - \mu_X)^T = \begin{bmatrix} 0.25 & 0.25 \\ 0.25 & 0.25 \end{bmatrix}$$

$$(x_4 - \mu_X)(x_4 - \mu_X)^T = \begin{bmatrix} 0.25 & -0.25 \\ -0.25 & 0.25 \end{bmatrix}$$

$$(x_5 - \mu_X)(x_5 - \mu_X)^T = \begin{bmatrix} 0.25 & -0.25 \\ -0.25 & 0.25 \end{bmatrix}$$

$$(x_6 - \mu_X)(x_6 - \mu_X)^T = \begin{bmatrix} 0.25 & 0.25 \\ 0.25 & 0.25 \end{bmatrix}$$

$$(x_7 - \mu_X)(x_7 - \mu_X)^T = \begin{bmatrix} 0.25 & 0.75 \\ 0.75 & 2.25 \end{bmatrix}$$

$$(x_8 - \mu_X)(x_8 - \mu_X)^T = \begin{bmatrix} 2.25 & 0.75 \\ 0.75 & 0.25 \end{bmatrix}$$

Covariance matrix:

$$C_X = E\{(X - \mu_X)(X - \mu_X)^T\}$$

$$= \begin{bmatrix} 0.75 & 0.375 \\ 0.375 & 0.75 \end{bmatrix}$$

KL Transform Implication & Example

Step – 04: Finding Eigen values:

$$C_X - \lambda I = 0$$

$$\begin{vmatrix} 0.75 - \lambda & 0.375 \\ 0.375 & 0.75 - \lambda \end{vmatrix} = 0$$

$\lambda_1 = 1.125$ and $\lambda_2 = 0.375$

Step – 05: Eigen vectors of corresponding Eigen value:

Assuming Z Eigen vector corresponding to Eigen value.

$$C_X Z = \lambda Z$$

Putting $\lambda_1 = 1.125$ $\quad e_1 = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ 1 \end{pmatrix}$ (Eigen vector corresponds to maximum Eigen value)

$$\lambda_2 = 0.375 \qquad e_1 = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ -1 \end{pmatrix}$$

Step – 06: The transform matrix:

$$\boldsymbol{A} = \frac{\mathbf{1}}{\sqrt{\mathbf{2}}}\begin{bmatrix} \mathbf{1} & \mathbf{1} \\ \mathbf{1} & \mathbf{-1} \end{bmatrix}$$

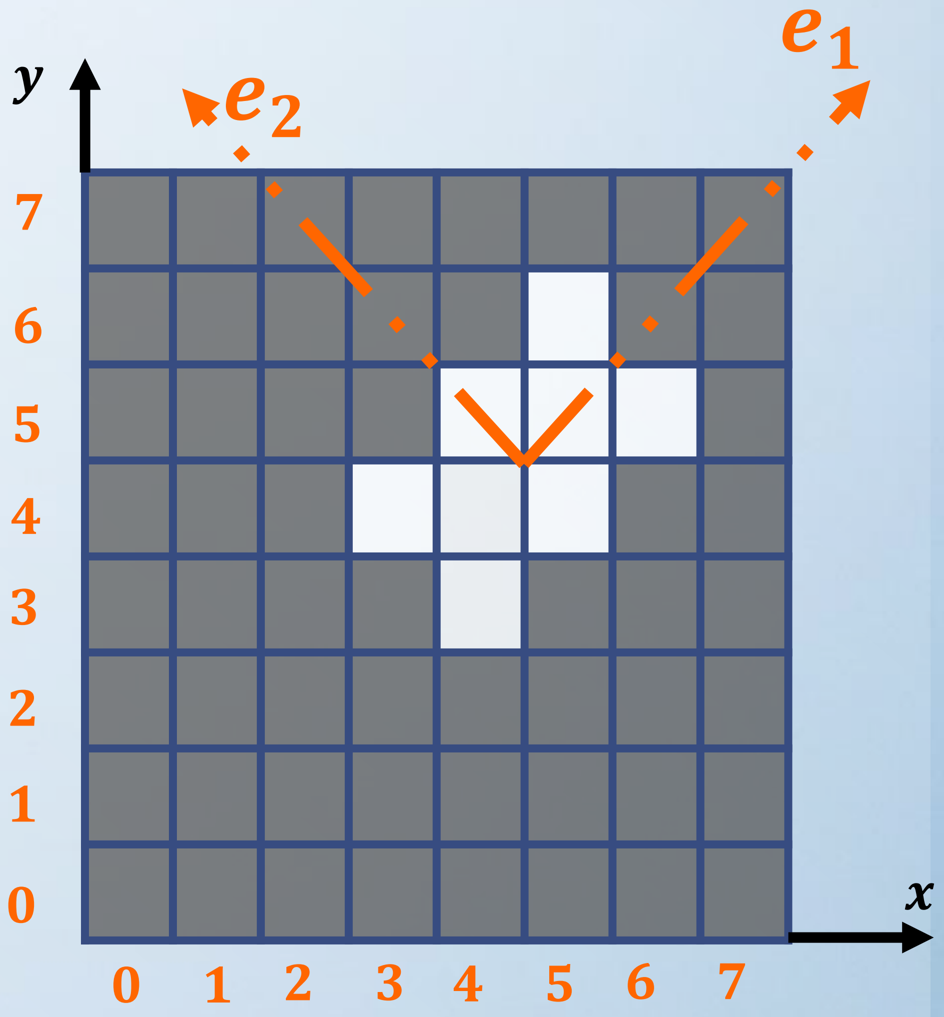
Rotated and Sifted by 45^0 about the axis.

Now, the KL-transform is:

$$\boldsymbol{Y} = \boldsymbol{A}(\boldsymbol{X} - \boldsymbol{\mu}_X)$$

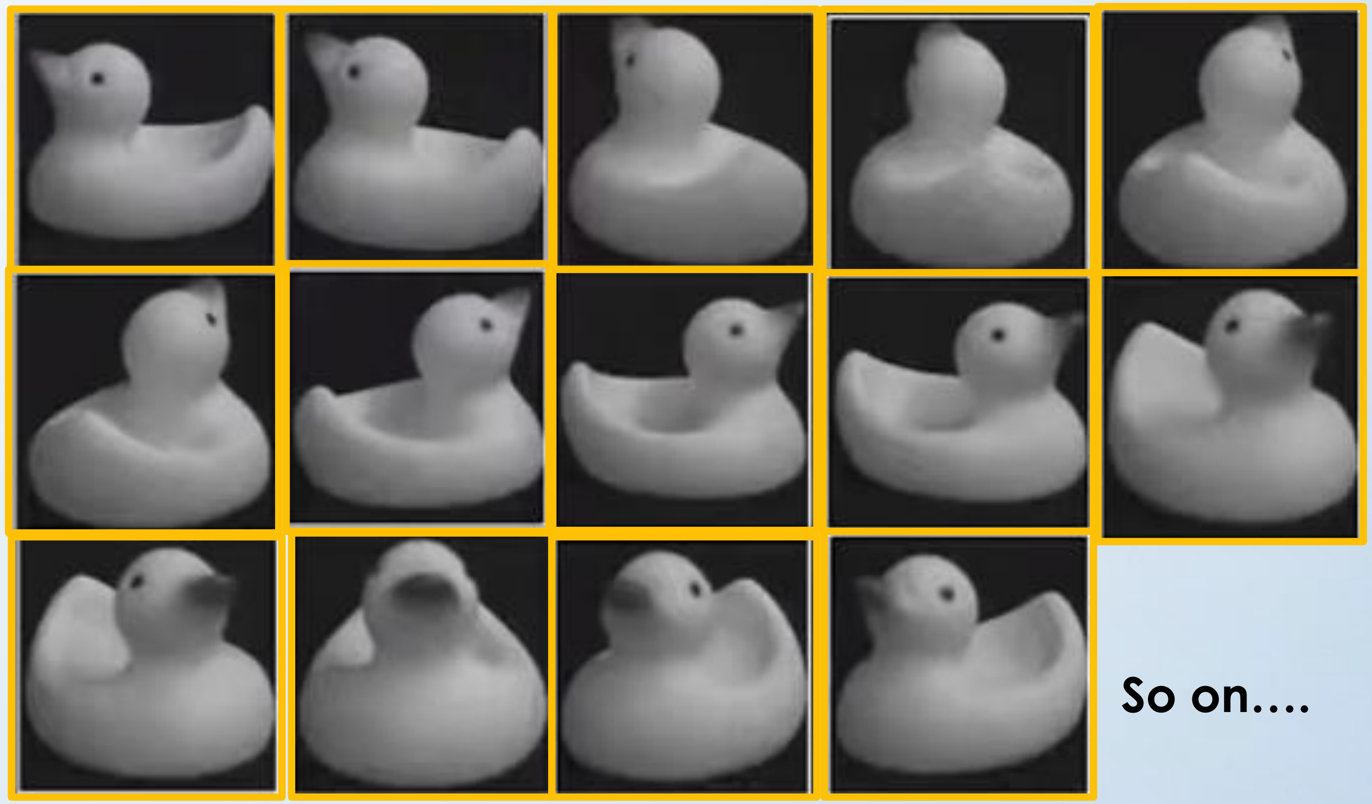
KL Transform Implication

- ✓ A creates a new coordinate system with origin is at center.
- ✓ Axis of this new coordinate system is parallel to the direction of change (Eigen vector). It also shows the direction of maximum change.
- ✓ It aligns the data along the Eigen vector direction because of which data become uncorellated.



Appearance Matching and PCA

Input Image.



So on....

Object Image Sets (Templates)...

We want to transfer the image into a different "Space" or "Domain", where matching one image with other will be more efficient.

Vector Representation of Image

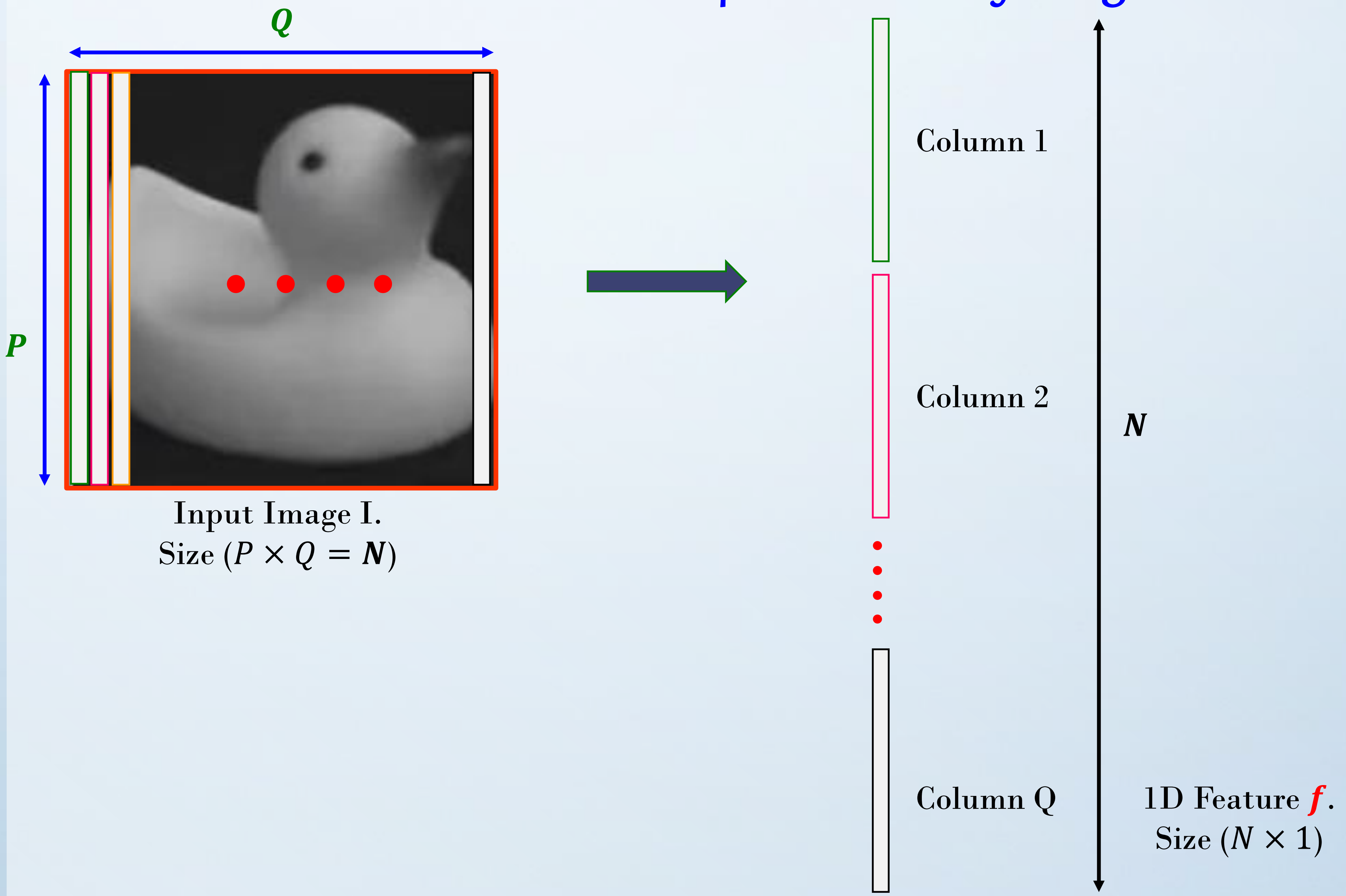
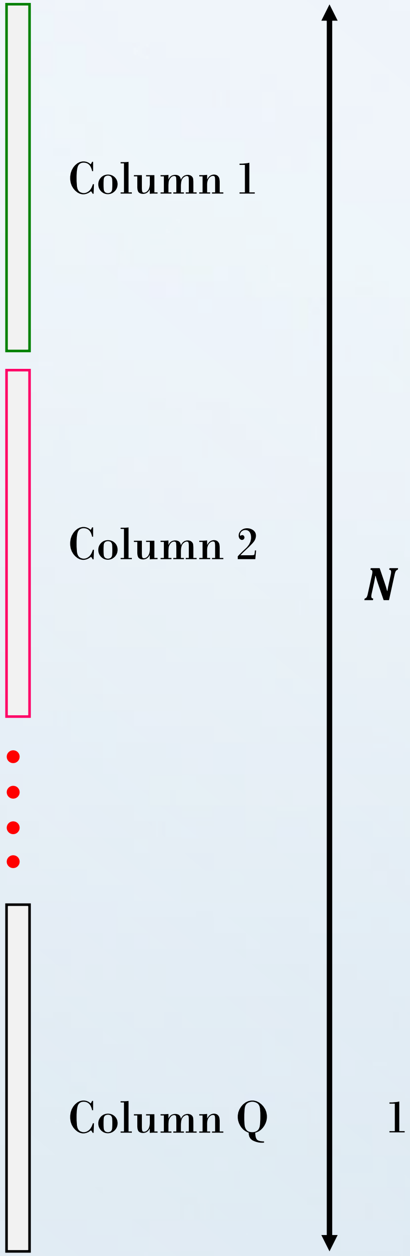
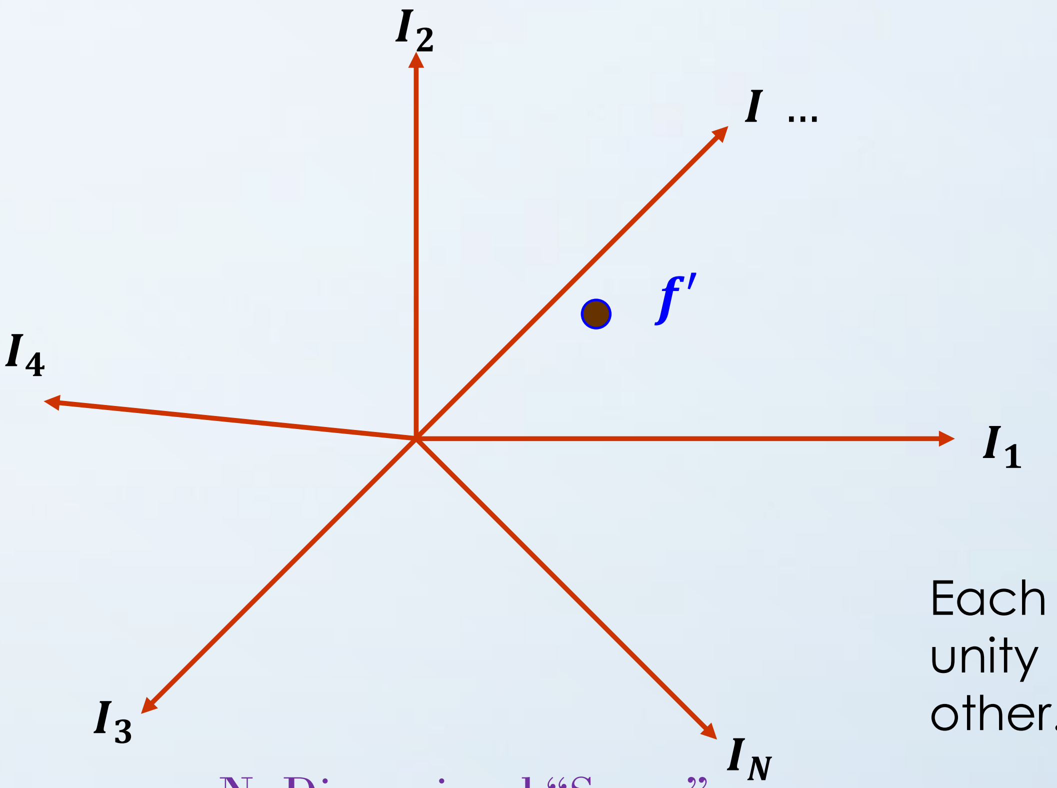


Image: A point in N-D Space





N- Dimensional "Space"

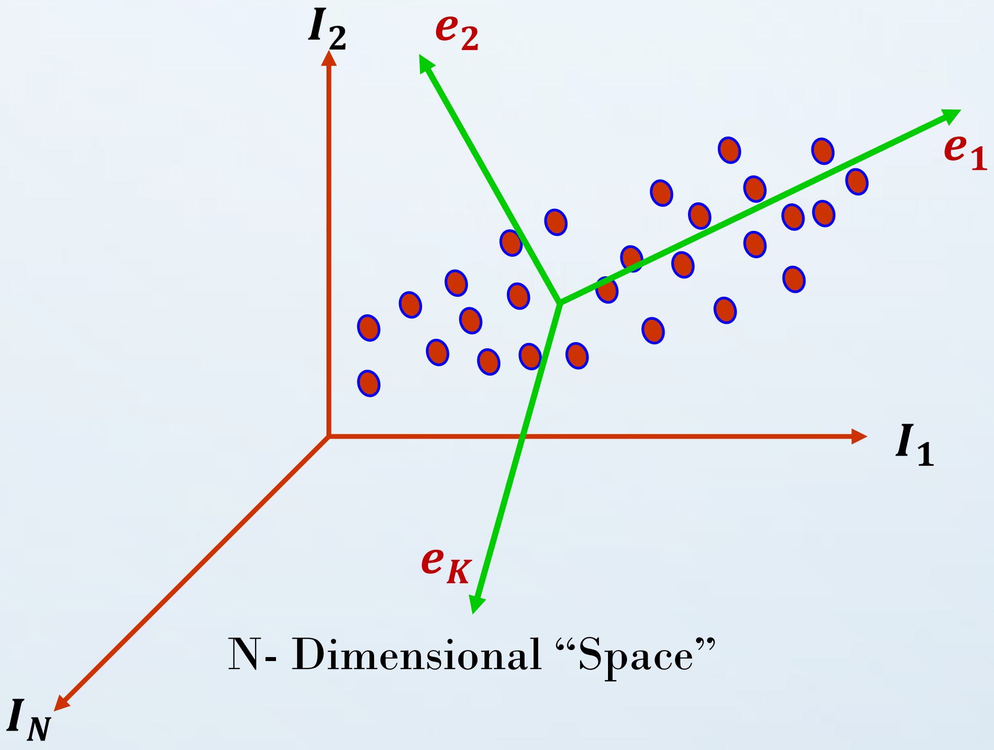
Each dimension $\{I_1, I_2, I_3, \ldots I_N\}$ are unity vectors orthogonal to each other.

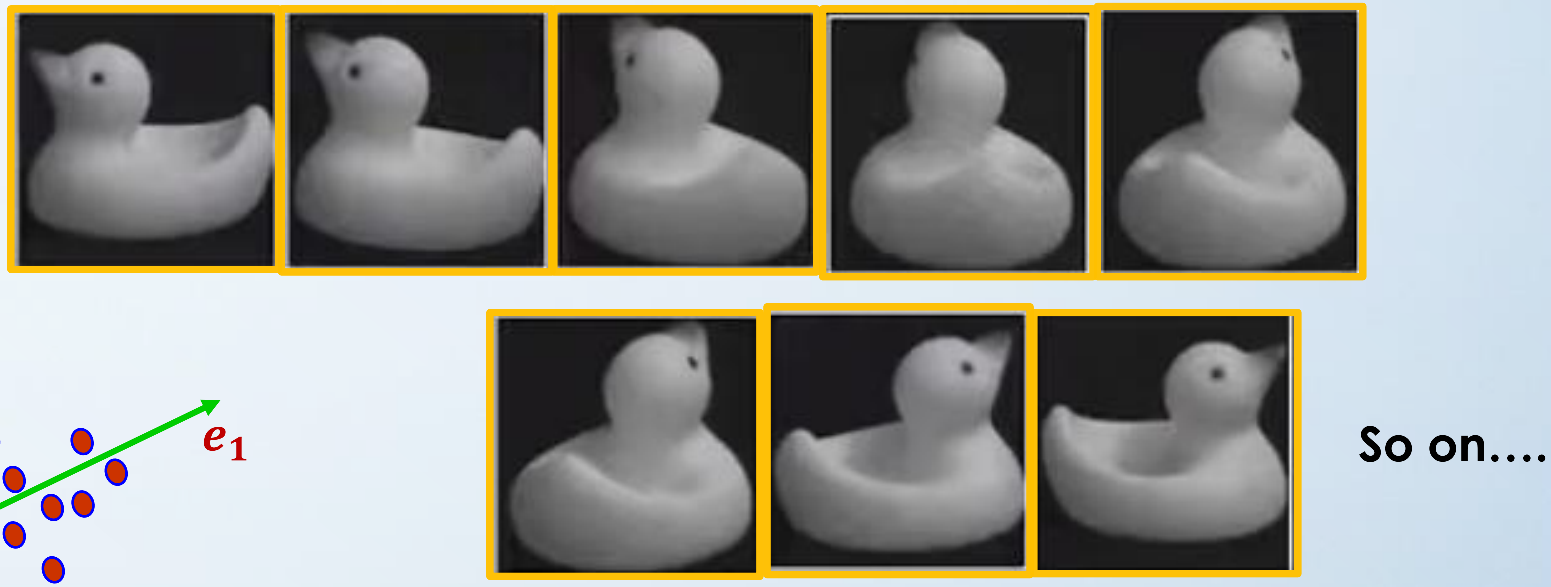
Ex: $I_2 = [0, 1, 0, 0, \ldots, 0]^T$

1D Feature f'.
Size $(N \times 1)$

Each dimension represents image intensity value corresponds to pixel and are perpendicular to each other.

Appearance Distribution in N-D Space



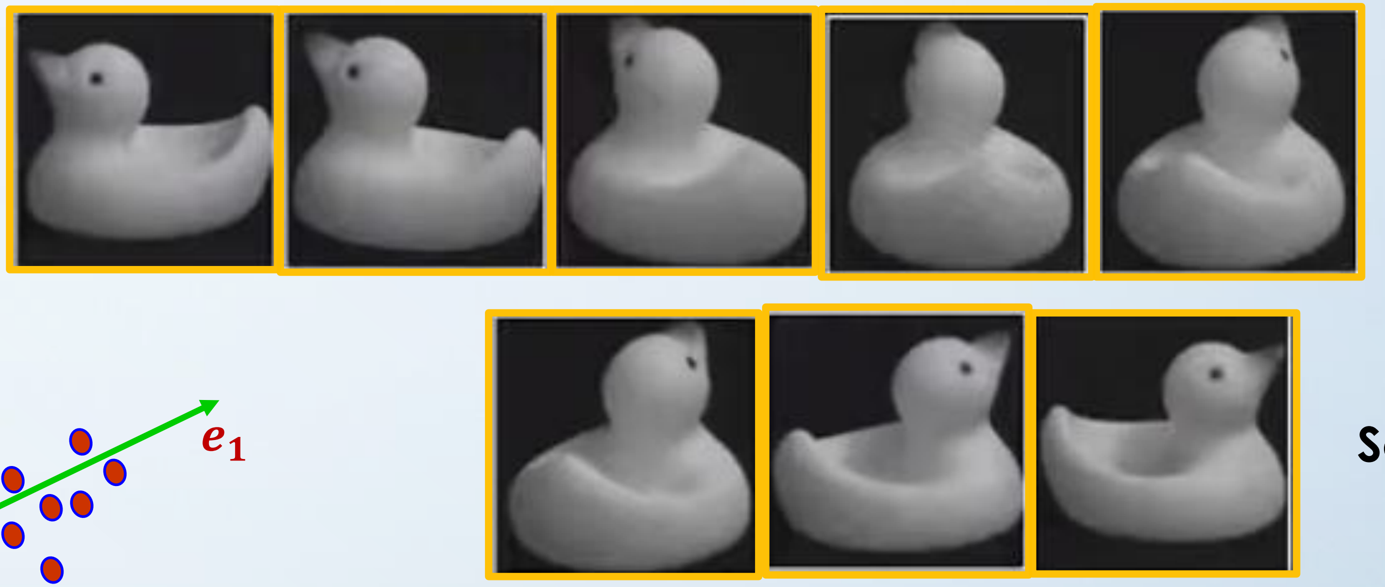


Object Image Set (M Different Objects)

It is possible to express the feature point distribution in a lower dimension space ($K < N$) with good accuracy.

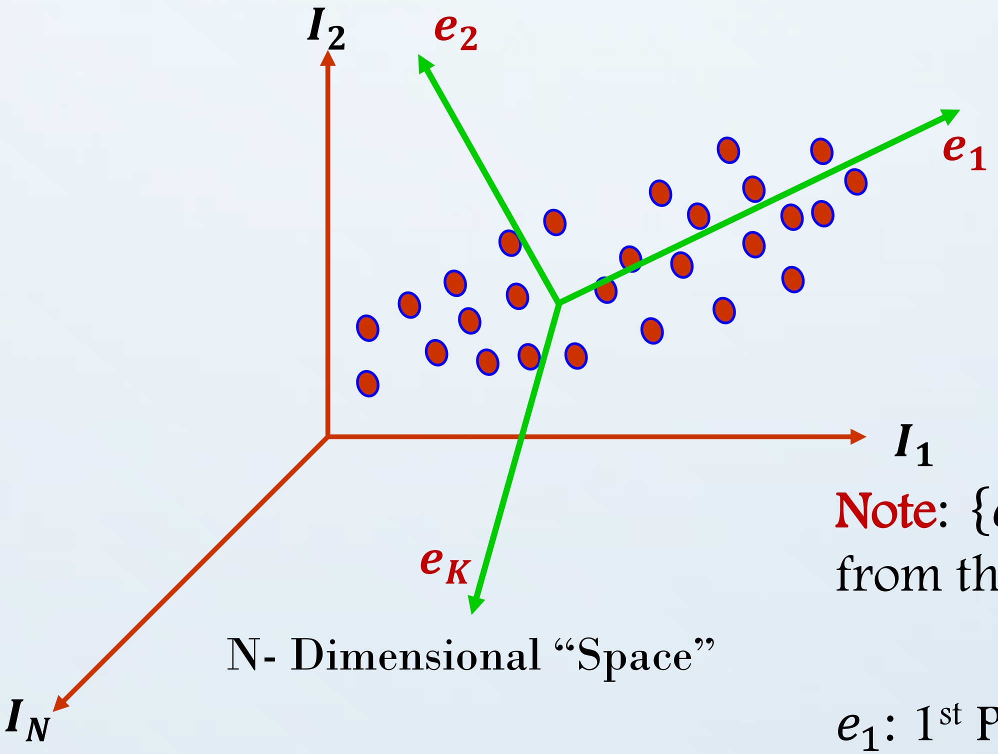
Distribution of feature points in the higher dimension is highly structures and usually resides in smaller dimensions.

Appearance Distribution in N-D Space



So on….

Object Image Set (M Different Objects)



Note: $\{e_1, e_2, e_3 \ldots . e_K\}$ are Eigenvectors that creates orthogonal basis from the covariance matrix of the M different images … $(K < N)$

e_1: 1st Principal component (Direction of maximum change)

e_2: 2nd Principal Component corresponds to second highest Eigenvalue

So on and So forth with e_K: K^{th} Principal component.

Image Representation using Principal Components

- ✓ Principal components (Eigenvectors with maximum Eigenvalues) are orthogonal to each other.

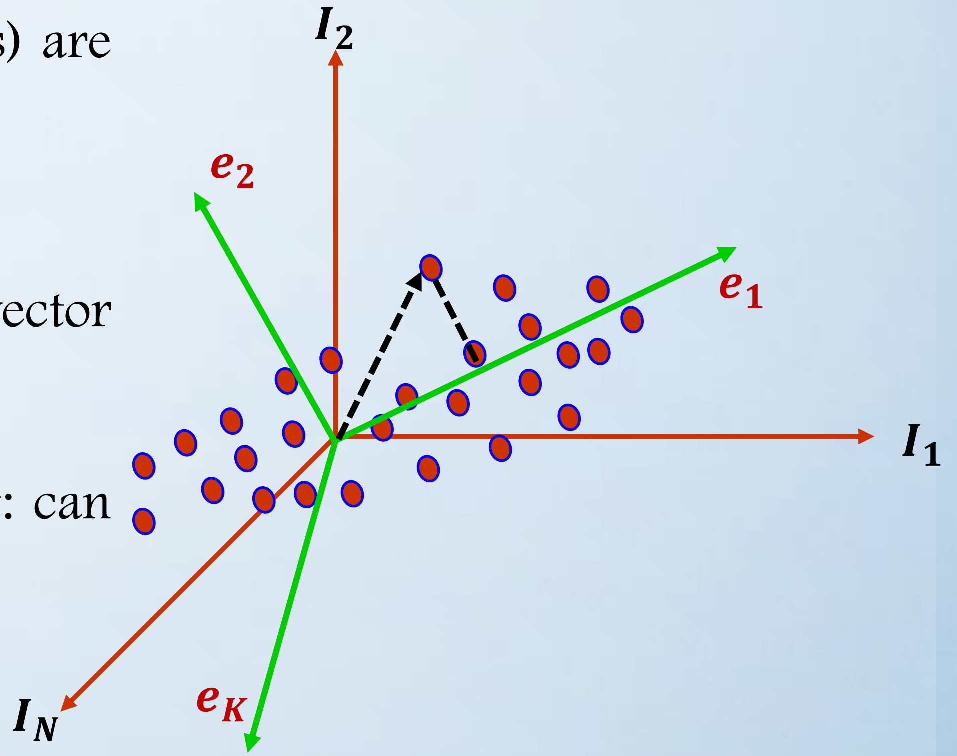
$$\boldsymbol{e_1} \perp \boldsymbol{e_2} \perp \boldsymbol{e_3} \ldots \perp \boldsymbol{e_K}$$

- ✓ Remember, $\boldsymbol{e_1}$ is the first principal component with eigenvector corresponds to maximum Eigenvalue.

- ✓ Representing any image f only using one principal component: can be formulated as projection of image onto the eigenvector e_1

$$\boldsymbol{p} = \boldsymbol{e_1}.\boldsymbol{f} \quad \text{(Dot Product)}$$

A single value p.



K –Principal Components: $\boldsymbol{e_1} \perp \boldsymbol{e_2} \perp \boldsymbol{e_3} \ldots \perp \boldsymbol{e_K}$

Two Principal Components: $\boldsymbol{e_1} \perp \boldsymbol{e_2}$

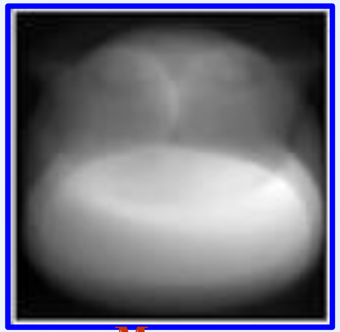
$$P = \begin{bmatrix} p_1 \\ p_2 \end{bmatrix} = [e_1 \quad e_2]^T f$$

$$P = \begin{bmatrix} p_1 \\ p_2 \\ \cdots \\ p_K \end{bmatrix} = [e_1 \quad e_2 \quad \cdots \quad e_K]^T f$$

Note: Image is now represented in K-dimensions. ($K \ll N$)

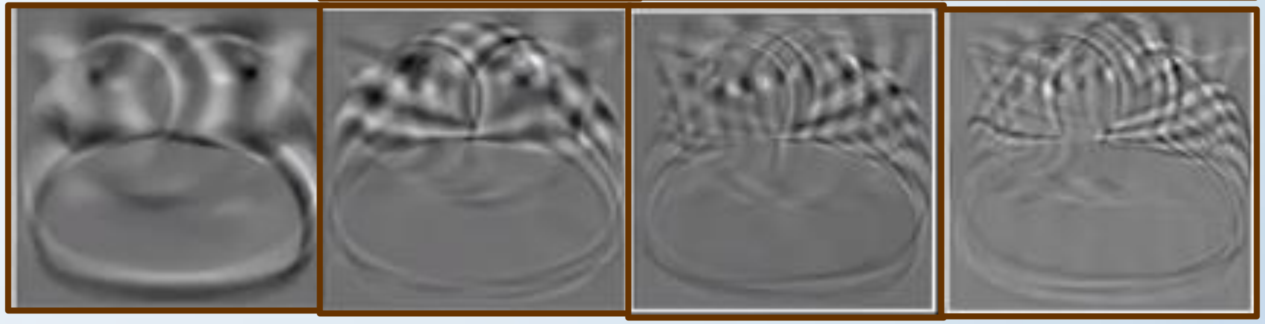
Dimensionality Reduction

Object Image Set



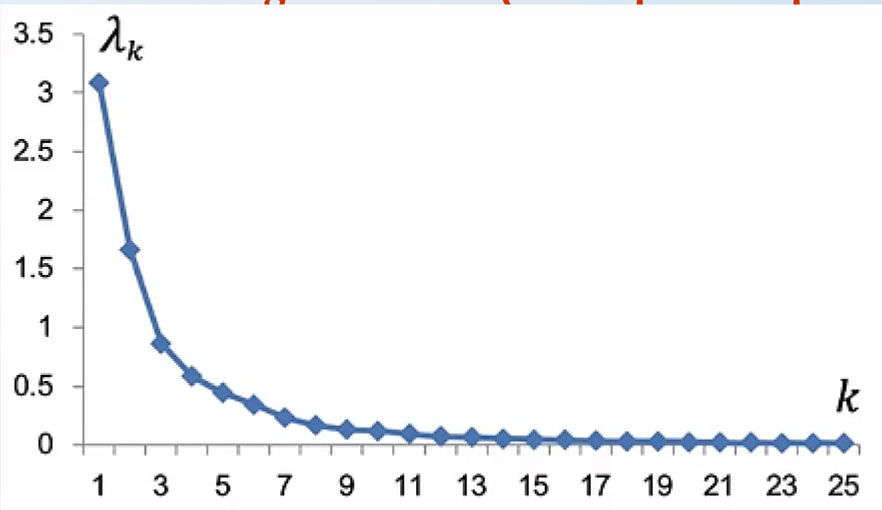
Mean

1st Eigenvector   2nd Eigenvector   3rd Eigenvector



10th Eigenvector   20th Eigenvector   30th Eigenvector   50th Eigenvector

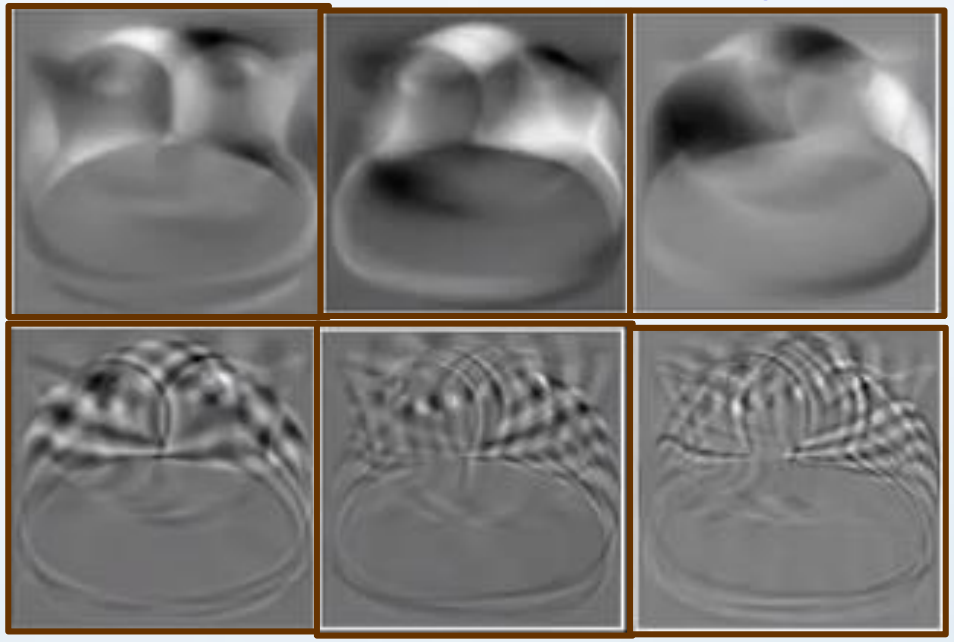
Eigen Vectors (Principal Components)



Eigenvalues

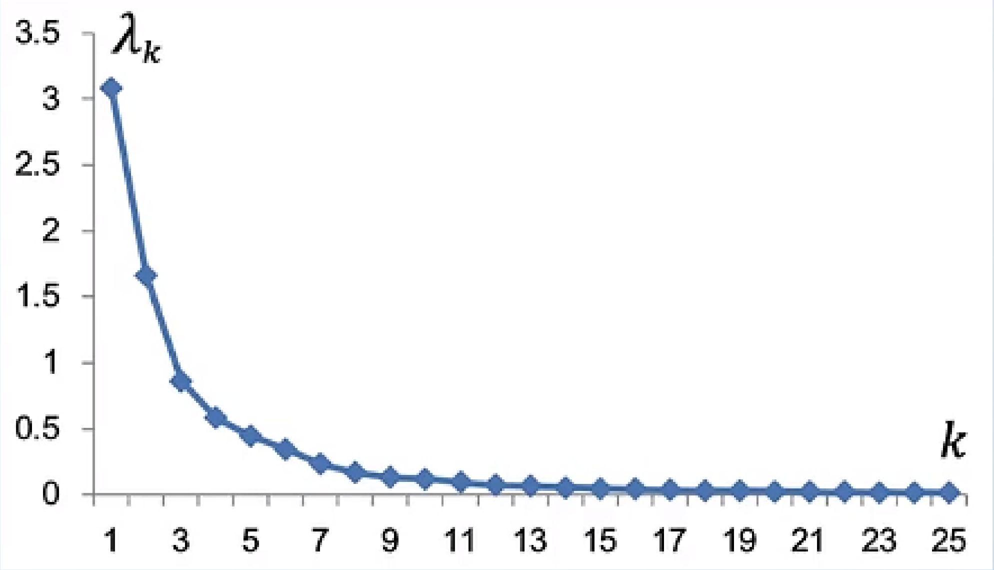
Dimensionality Reduction: Choosing the Value of K

1st Eigenvector 2nd Eigenvector 3rd Eigenvector



20th Eigenvector 30th Eigenvector 50th Eigenvector

Eigen Vectors (Principal Components)



Eigenvalues

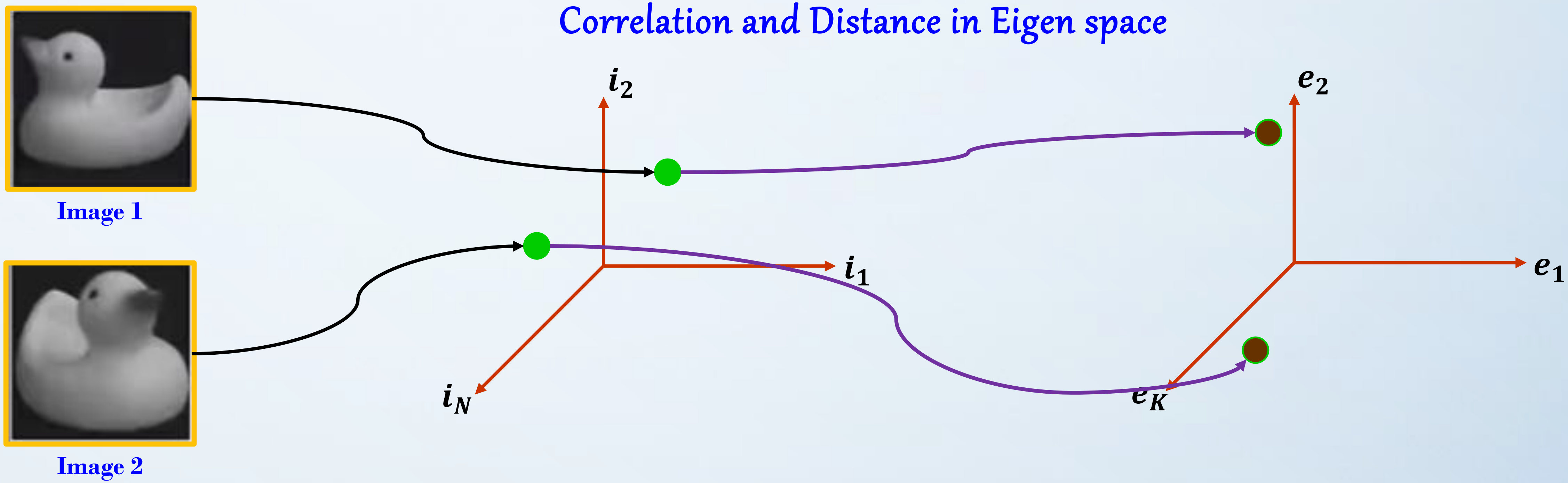
How many Principal Components are sufficient?

- ✓ Say we want to capture 95% of the variations of the total data set.
- ✓ Then find the smallest K such that:

$$\frac{Sum\ of\ K\ Largest\ Eigenvalues}{Sum\ of\ all\ the\ Eigenvalues} = \frac{\sum_1^k \lambda_i}{\sum_1^N \lambda_i} \geq 0.95$$

Appearance Matching in Reduced Dimension:

Correlation and Distance in Eigen space



Correlation in image-space:

$$SSD = \sum_p \sum_q \left(I_1(p,q) - I_2(p,q)\right)^2$$
$$= \|f_1' - f_2'\|^2$$

L^2 Distance in K-D Space:

$$d = \|p_1 - p_2\|^2$$

$$= \left\| \sum_{k=1}^{K} p_k^{(1)} e_k - p_k^{(2)} e_k \right\|^2$$

$$\approx \|f_1' - f_2'\|^2$$

Drawbacks of the KL Transform

Despite its favorable theoretical properties, the KLT is not used in practice (for many applications) for the following reasons.

- ✓ Its basis functions depend on the covariance matrix of the image, and hence they have to recomputed and transmitted for every image.
- ✓ Perfect decorrelation is not possible, since images can rarely be modelled as realizations of ergodic fields.
- ✓ There are no fast computational algorithms for its implementation.