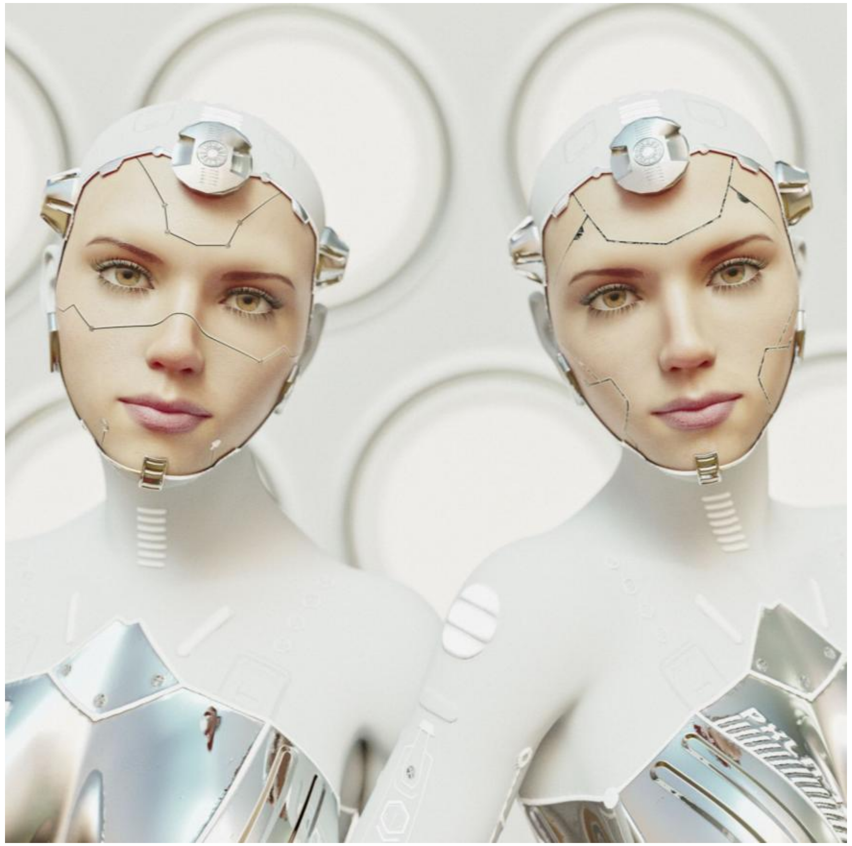
APPEARANCE MATCHING



Principal Components

Given: Mean-Subtracted image set $\{f_1, f_2, \ldots, f_M\}$ (f_m is N*1 vector)

Find: Orthonormal basis $\{e_1, e_2, \ldots, e_K\}$ (e_k is N*1 vector)

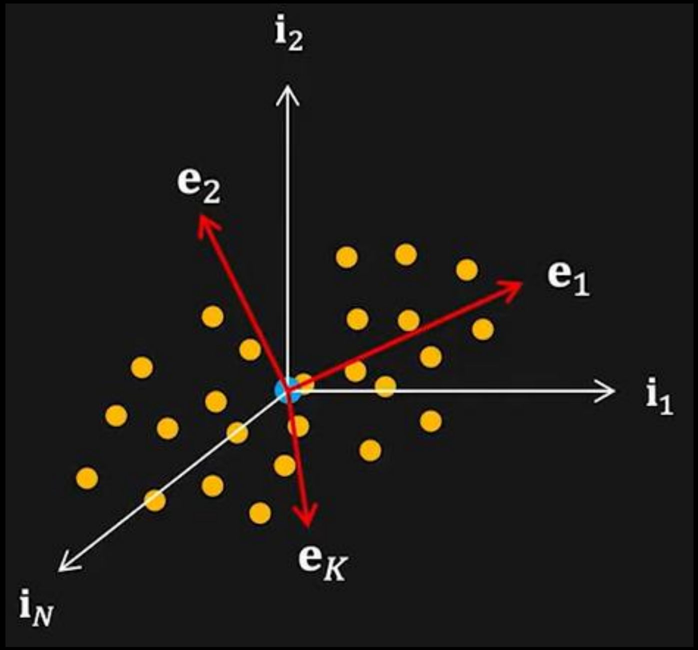
Such that

$$\boldsymbol{f}_m \approx \sum_{k=1}^{K} p_k^{(m)} \mathbf{e}_k$$

Where:

$$p_k^{(m)} = \mathbf{e}_k{}^T \boldsymbol{f}_m \qquad \textit{(linear)}$$

$$\mathbf{e}_i{}^T \mathbf{e}_j = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases} \qquad \textit{(Orthonormal)}$$



Principal Components

Projection of the Random variable (image) f along the 1^{st} Principal Component e is (f * e)

Find e that maximizes Variance of (f * e)

$\mathrm{Var}\,[f * e] = E[\{e * (f - E[f])\}^2]$

Mean of f : E[f] = 0)

$\mathrm{Var}\,[f * e] = E[\{e * f\}^2] = E[e^T f f^T e] = e^T E[f f^T]\, e = e^T R\, e$

Where $R_{N*N} = E[f f^T]$ is Covariance matrix

Principal Components

Projection of the Random variable (image) f along the 1st Principal Component e is (f * e)

Find e that maximizes Variance of (f * e)

$$\max_{\mathbf{e}}(\mathbf{e}^T R\mathbf{e}) \quad \text{such that} \quad \mathbf{e}^T\mathbf{e} = 1$$

Find eigenvalues and eigen vectors of R (Covariance matrix)

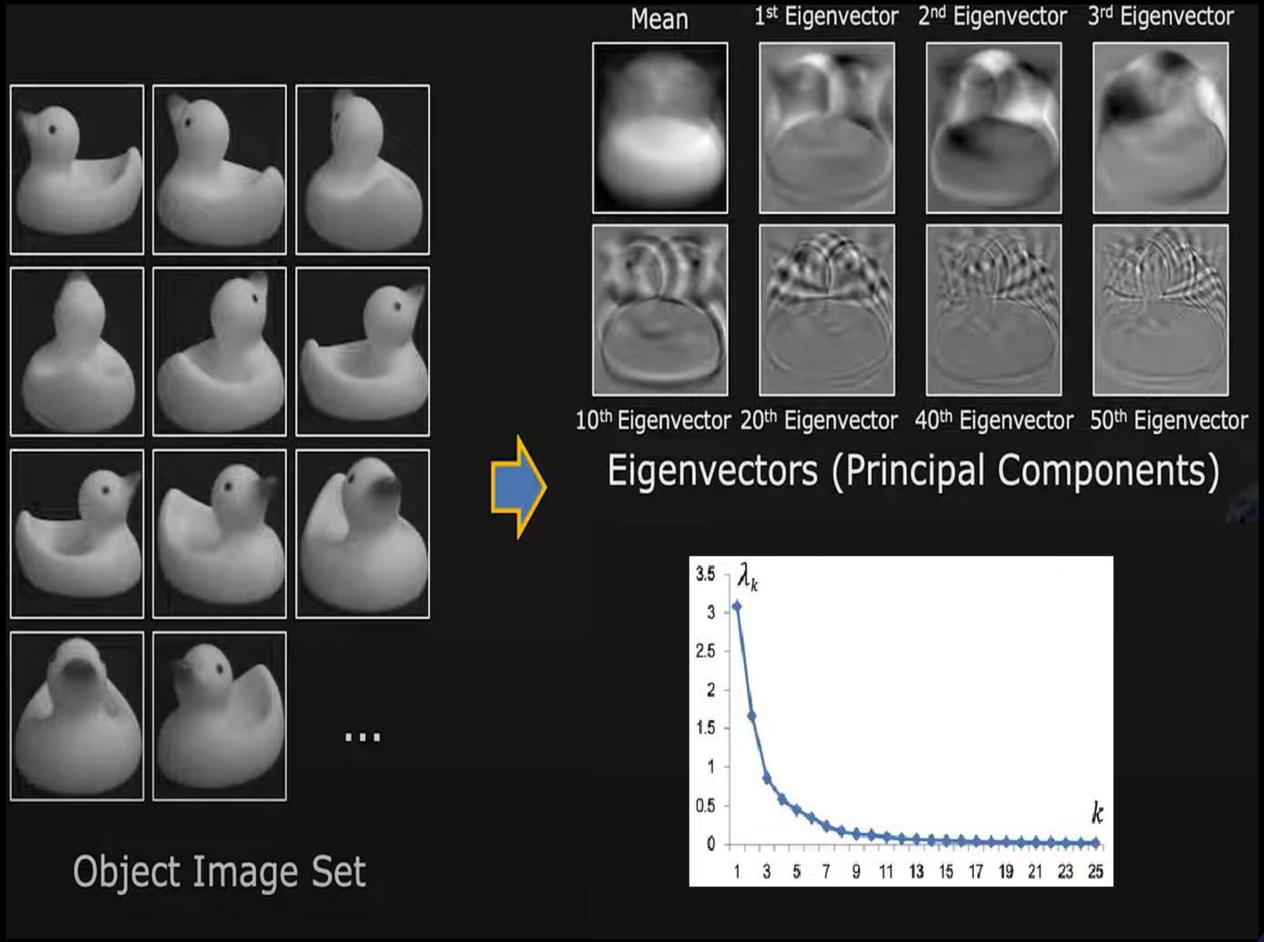
1st eigenvector corresponds to maximum eigenvalue

PCA algorithm

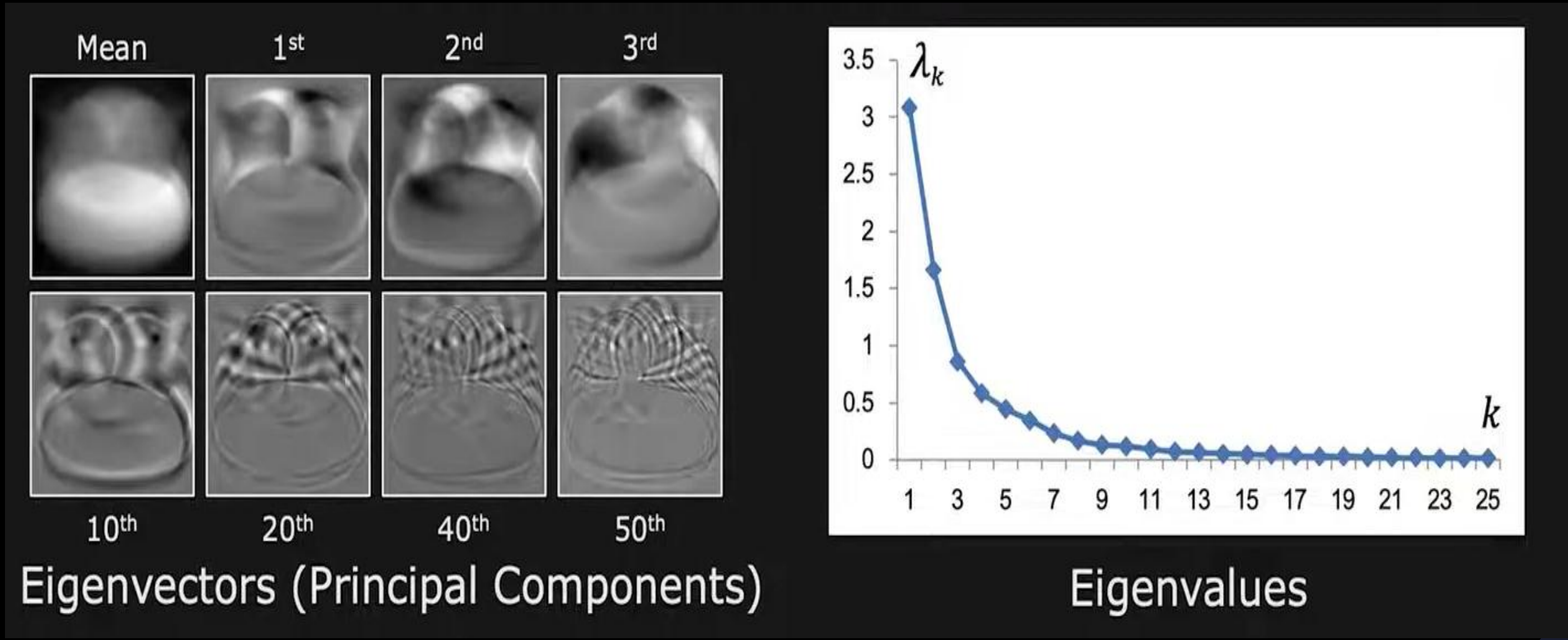
1. Data Matrix: $F = [f_1\ f_2\ \dots\ f_M]$ (N * M)
2. Covariance Matrix: $R = FF^T$ (N*N)
3. Solve for eigenvalues and eigenvectors
4. Eigenvalues: $\{\lambda_1 , \lambda_2 , \dots , \lambda_K\}$
5. Eigenvectors: $\{e_1 , e_2 , \dots , e_K\}$

Eigenvectors are calculated on Orthonormal basis and referred as Linear Subspace aka Eigenspace

Dimensionality Reduction



Dimensionality Reduction



If we want to capture 95% of variations in the dataset

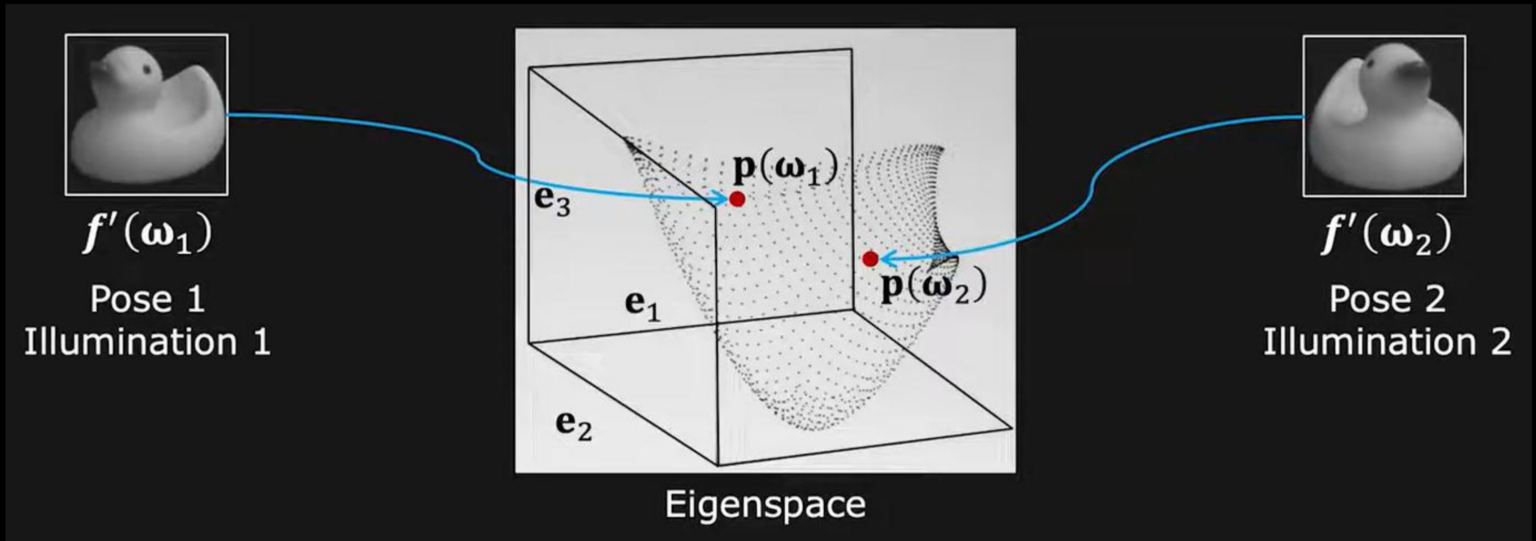
$$\frac{\textit{Sum of K largest Eigenvalues}}{\textit{Sum of all Eigenvalues}} = \frac{\sum_i^K \lambda_i}{\sum_j^N \lambda_j} \geq 0.95$$

Visual Appearance

$$\text{Visual Appearance} = \mathcal{F}\left\{\begin{array}{l} \left.\begin{array}{l}\text{Shape}\\ \text{Reflectance}\end{array}\right\} \quad \text{Intrinsic Parameters} \\ \\ \left.\begin{array}{l}\text{Pose}\\ \text{Illumination}\end{array}\right\} \quad \text{Extrinsic Parameters}\end{array}\right.$$

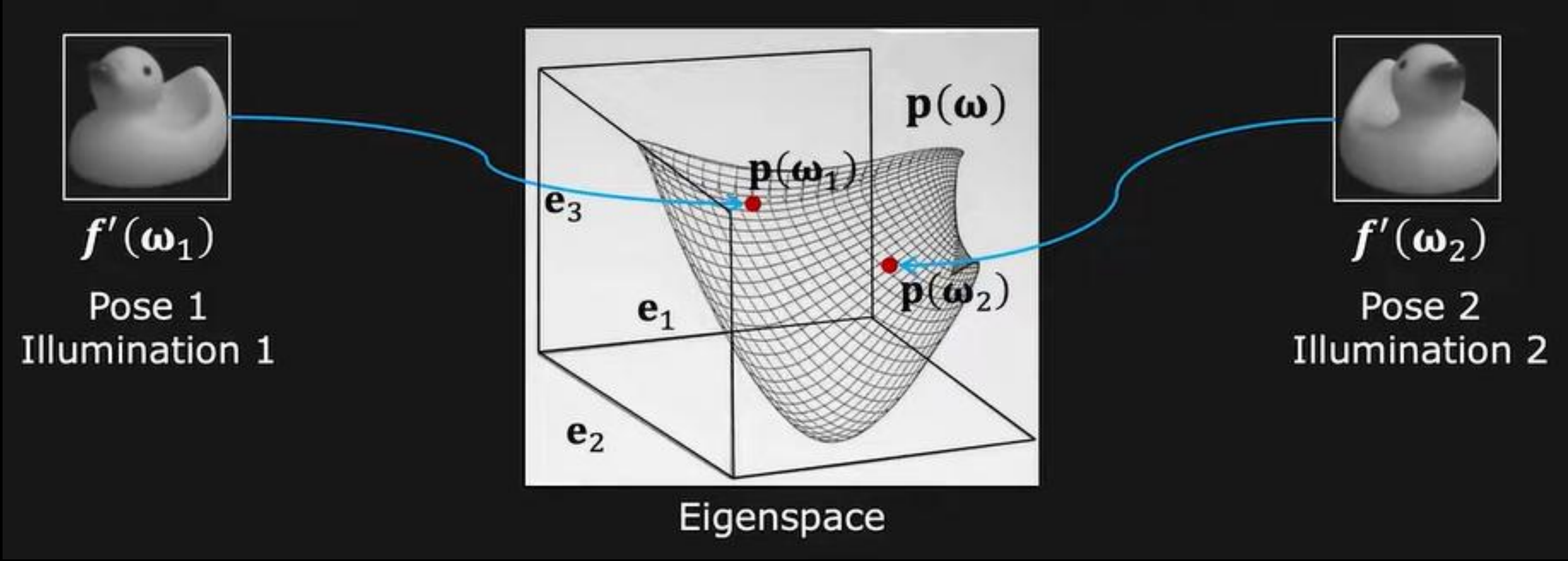
Let Extrinsic Parameters be: $\boldsymbol{\omega} = \begin{bmatrix}\omega_1\\ \omega_2\\ \vdots\\ \omega_T\end{bmatrix} = \begin{pmatrix}\text{Pose}\\ \vdots \\ \text{Illumination}\\ \vdots\end{pmatrix}$

Projecting learning images to Eigenspace



$$\boxed{\mathbf{p}(\omega) = [\mathbf{e}_1 \quad \mathbf{e}_2 \quad \dots \quad \mathbf{e}_K]^T \times (f'(\omega) - \mathrm{c})}$$

N-dimensional images are Points in K-dimensional Eigenspace where K << N



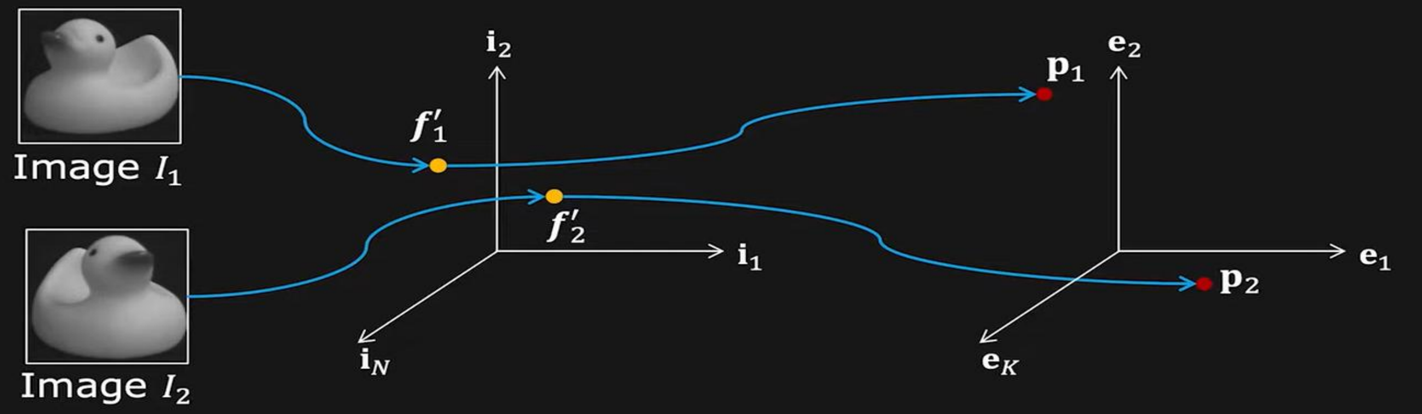
Parametric Appearance Representation

Fit a parameter model (Interpolation) p(ω) as a function of $\omega = [\omega_1, \omega_2, \ldots, \omega_T]$ to the distribution.

Object appearance representation is reduced to:

Mean image + Eigenvectors + Manifold

Correlation and distance in Eigenspace



Correlation in Image Space

$$SSD = \sum_p \sum_q (I_1[p,q] - I_2[p,q])^2$$

$$= \|\boldsymbol{f}'_1 - \boldsymbol{f}'_2\|^2$$

L^2 Distance in K-D Space

$$d = \|\mathbf{p}_1 - \mathbf{p}_2\|^2$$

$$= \left\| \textstyle\sum_{k=1}^{K} p_k^{(1)} \mathbf{e}_k - \sum_{k=1}^{K} p_k^{(2)} \mathbf{e}_k \right\|$$

$$\approx \|\boldsymbol{f}'_1 - \boldsymbol{f}'_2\|^2$$

Appearance Learning (offline)

Given: M learning images $\left\{I_1^{(q)}, I_2^{(q)}, \dots, I_M^{(q)}\right\}$ for each of the Q training objects. $q = \{1,2,\dots,Q\}$

For each object q, perform steps 1-8:

Step 1: Normalize all images to remove brightness variations.

$$I_m^{\prime(q)} = I_m^{(q)} / \left\| I_m^{(q)} \right\|$$

Step 2: Convert image $I_m^{\prime(q)}$ to feature vector $\boldsymbol{f}_m^{\prime(q)}$.

Step 3: Compute the mean feature vector $\mathbf{c}^{(q)}$.

Step 4: Subtract from each feature vector the mean vector:

$$\boldsymbol{f}_m^{(q)} = \boldsymbol{f}_m^{\prime(q)} - \mathbf{c}^{(q)}$$

Appearance Learning (offline)

Step 5: Compute the data matrix and covariance matrix.

$$F^{(q)} = \left[\boldsymbol{f}_1^{(q)} \quad \boldsymbol{f}_2^{(q)} \quad \dots \quad \boldsymbol{f}_M^{(q)}\right]$$

$$R^{(q)} = F^{(q)} {F^{(q)}}^T$$

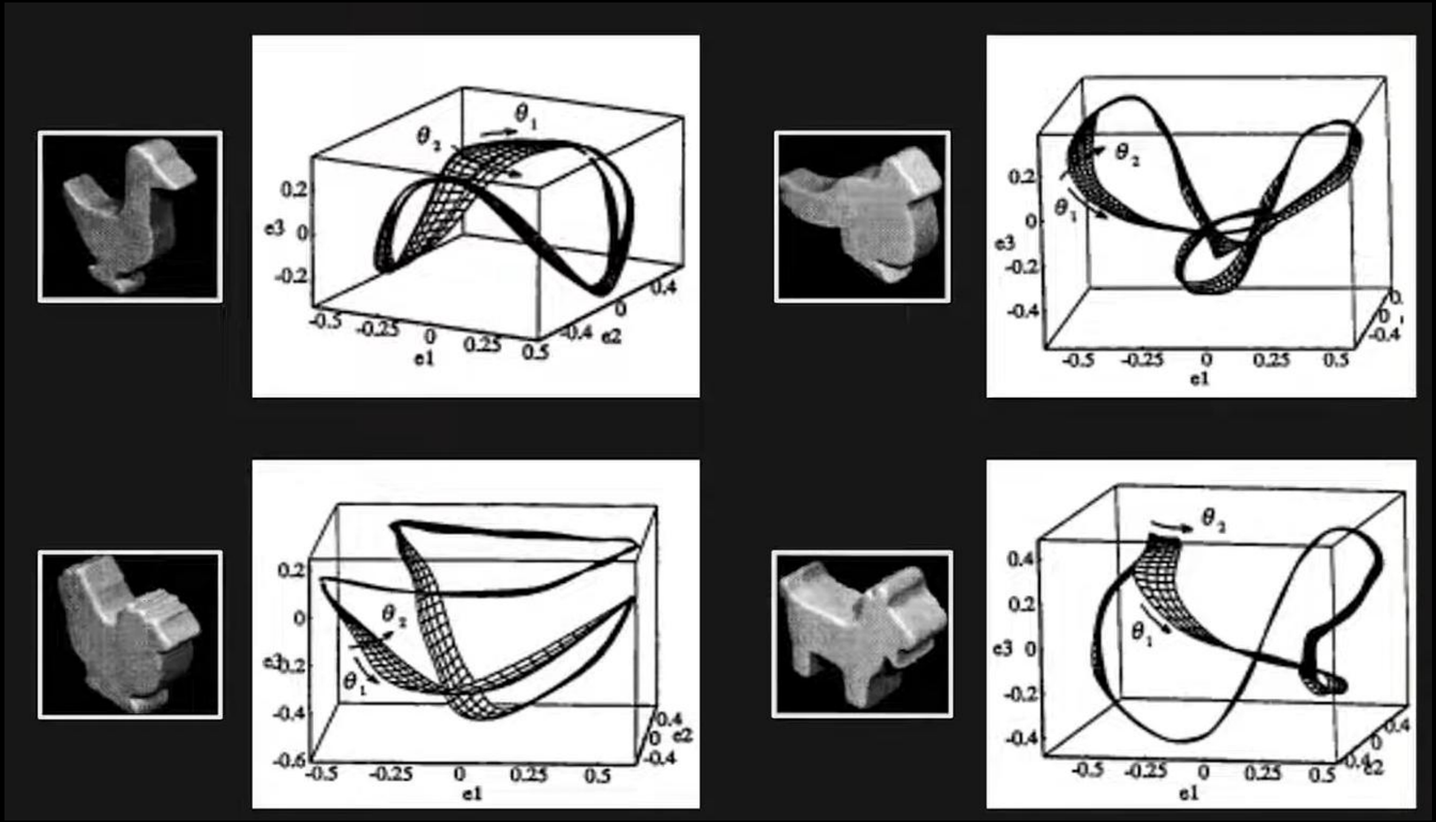
Step 6: Compute the K eigenvectors $\left\{\mathbf{e}_1^{(q)}, \mathbf{e}_2^{(q)}, \dots \mathbf{e}_K^{(q)}\right\}$ of $R^{(q)}$ that represent the new orthonormal basis ("eigenspace").

Step 7: Project the learning images to the eigenspace.

$$\mathbf{p}_m^{(q)} = \left[\mathbf{e}_1^{(q)} \quad \mathbf{e}_2^{(q)} \quad \dots \quad \mathbf{e}_K^{(q)}\right]^T \times \boldsymbol{f}_m^{(q)}$$

Step 8: Fit a parametric manifold to the projected image points as a function of extrinsic variables $\boldsymbol{\omega} = [\omega_1, \omega_2, \dots, \omega_T]$.

Example Object Manifolds in Eigenspace



Recognition (Online)

Given: Input image (I) for object recognition.

Step 1: Normalize the image to remove brightness variations:

$$I' = I/\|I\|$$

Step 2: Convert image I' to feature vector $\boldsymbol{f}'$.

For each object q in the database, perform steps 3-6:

Step 3: Subtract the mean feature vector for object q:

$$\boldsymbol{f}^{(q)} = \boldsymbol{f}' - \mathbf{c}^{(q)}$$

Step 4: Project feature vector to eigenspace for object q:

$$\mathbf{p}^{(q)} = \left[\mathbf{e}_1^{(q)} \quad \mathbf{e}_2^{(q)} \quad \dots \quad \mathbf{e}_K^{(q)}\right]^T \times \boldsymbol{f}^{(q)}$$

Recognition (Online)

Step 5: In the eigenspace for object q, find closest point on the manifold to projected point.

$$\boxed{\boldsymbol{\omega}^{(q)} = \arg\min_{\boldsymbol{\omega}} \left\| \mathbf{p}^{(q)} - \mathbf{p}^{(q)}(\boldsymbol{\omega}) \right\|}$$

Use a Nearest Neighbor Algorithm for finding closest point.

Step 6: Find the distance $d^{(q)}$ between the projected image point and the closest point on the manifold.

Step 7: Find the object q for which $d^{(q)}$ is minimum.