Manifold Learning

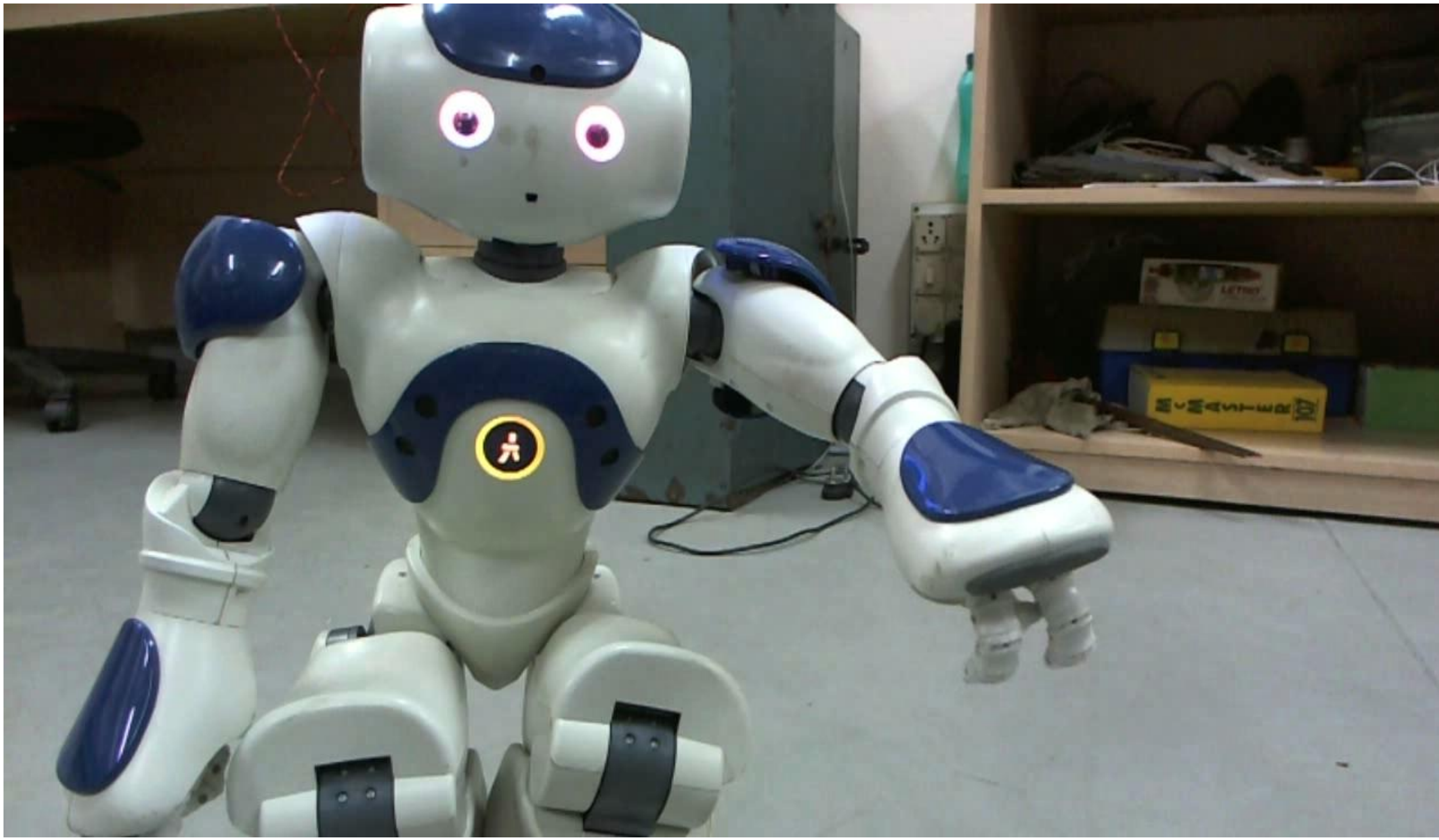
Generative modeling of high-dimensional data

Manifolds as Representation

images: 100 x 100 pixels

Ack: A. Efros, original images from hormel corp.

Learning to represent



Representations in AI

A representation for an object is a “frame” or collection of parameters associated with the object, the relations between them, and also a set of rules and functions for solving problems on the object.

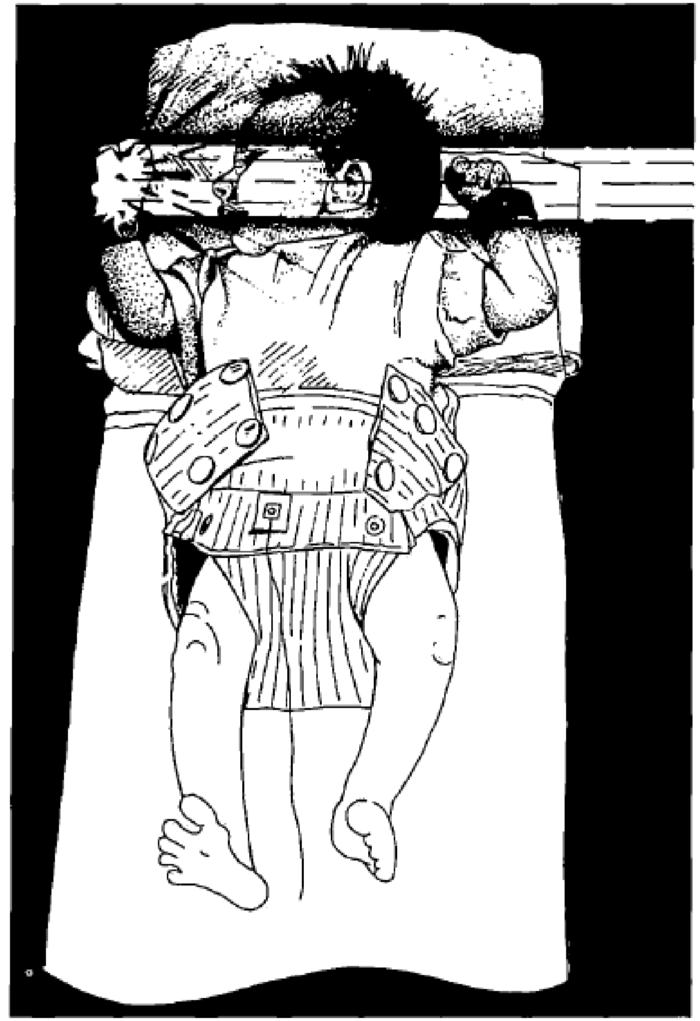
The set of variables and the predicates defined on them are determined by a knowledge engineer.

Q. Can we learn representations?

Role of Perception?

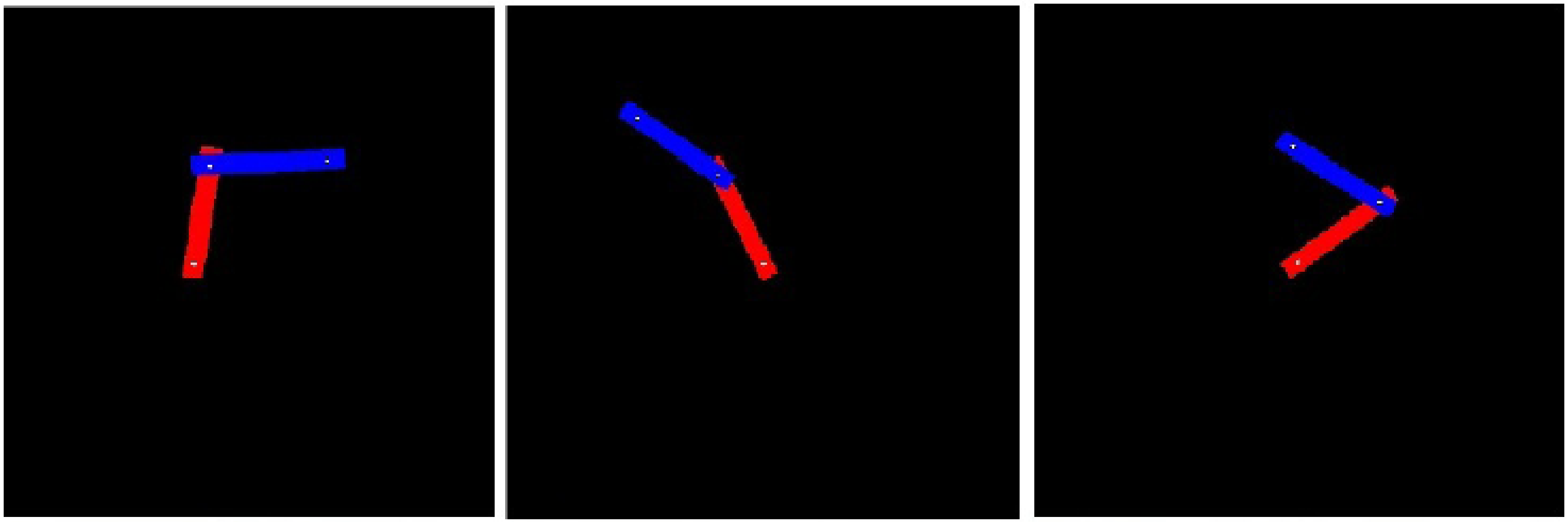
Newborns (10-24 day old) in dark room work hard to position hand so it is visbile in a narrow beam of light. …

Q. Can perception help in learning a representation?



[A. van der Meer, 1997: Keeping the arm in the limelight]

Learning to represent: robot motions



Representations in AI

A representation for an object is a "frame" or collection of parameters and function associated with the object.

How to represent a "robot"?

Must include: degrees of freedom (2)

parameters (θ_1, θ_2)

\+ rules / functions

Manifolds

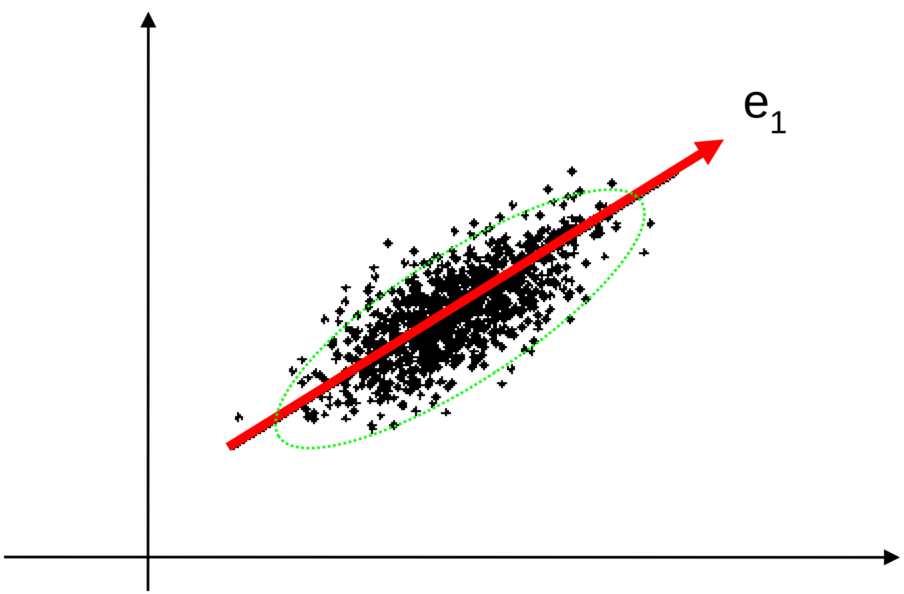
Linear dimensionality reduction

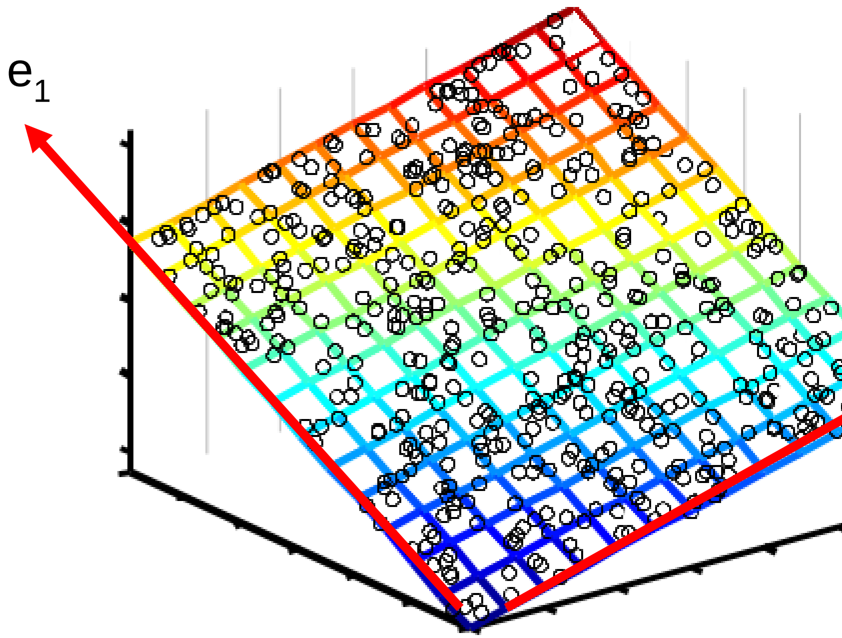
project data onto subspace of maximum variance

PCA: principal components analysis

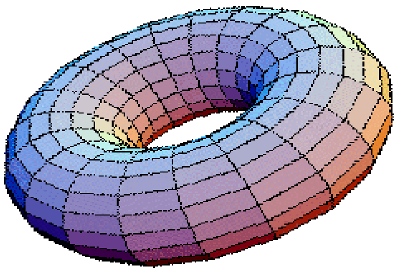
[A] = top eigenvectors of covariance matrix $[XX^T]$

Y = [A] X



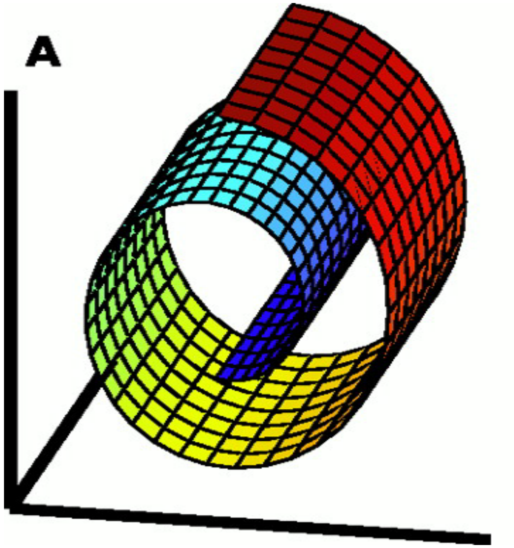


Manifolds



A manifold is a topological space which is locally Euclidean.

nbrhood N in $R^n \leftrightarrow$ ball B in R^d
(homeomorphic)



Homeomorphic: Every x in N has a map to a y in B

Dimensionality of manifold = d

Embedding dimension = n

Manifolds

A manifold is a topological space which is locally Euclidean.

nbrhood in $R^n \leftrightarrow$ ball in R^d
(homeomorphic)

Dimensionality of manifold = d

Embedding dimension = n

Real life data (e.g. images) : $D = 10^5$
motions = smooth variation
of just a few parameters

DOFs = pose of faces $\rightarrow$ d = 1

Ideally, d = number of varying parameters

Manifolds in video

Dimensionality of Actions

bend jack jump pjump run

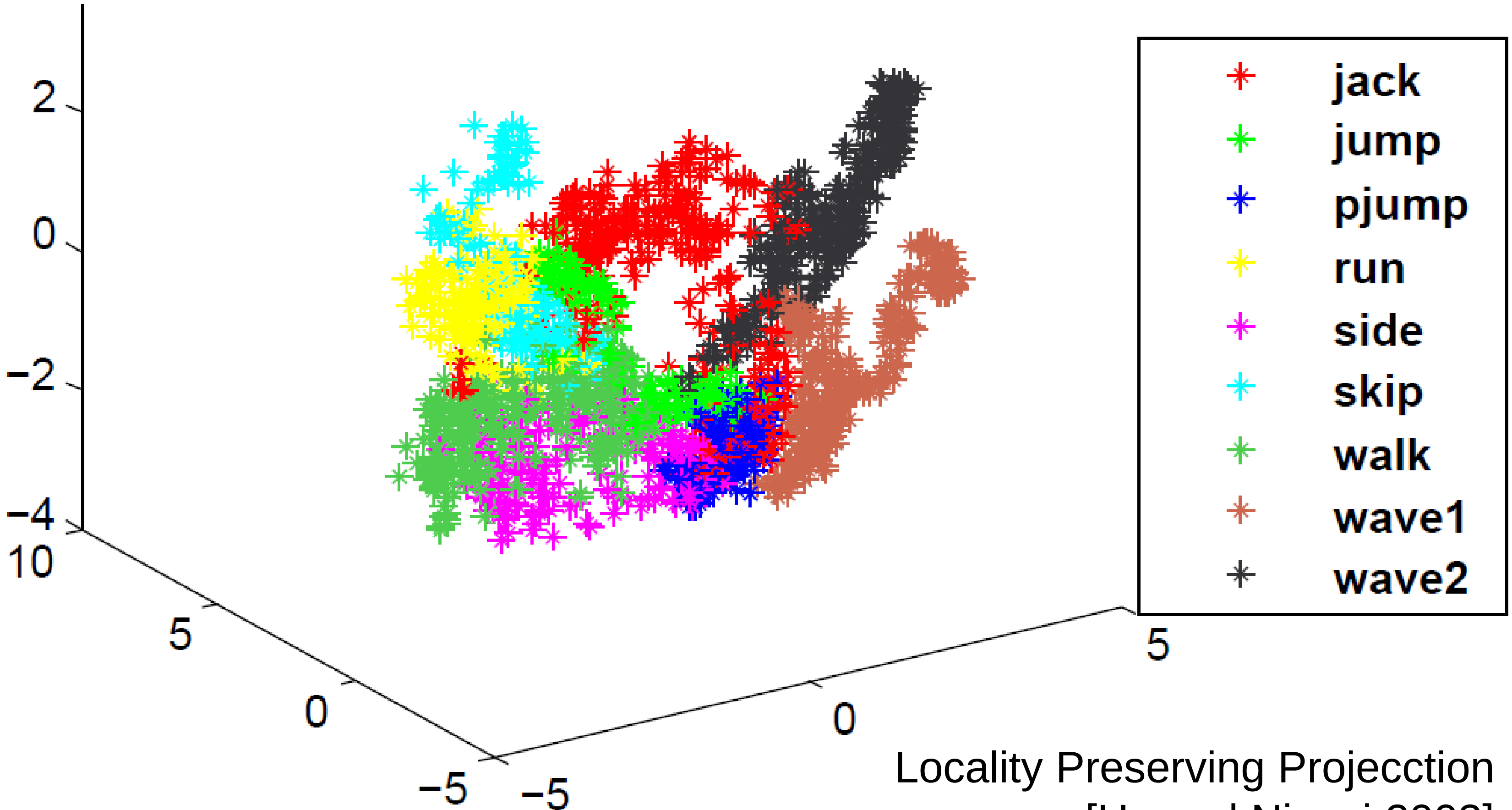
side skip walk wave1 wave2

Weizmann activity dataset:

videos of 10 actions by 12 actors

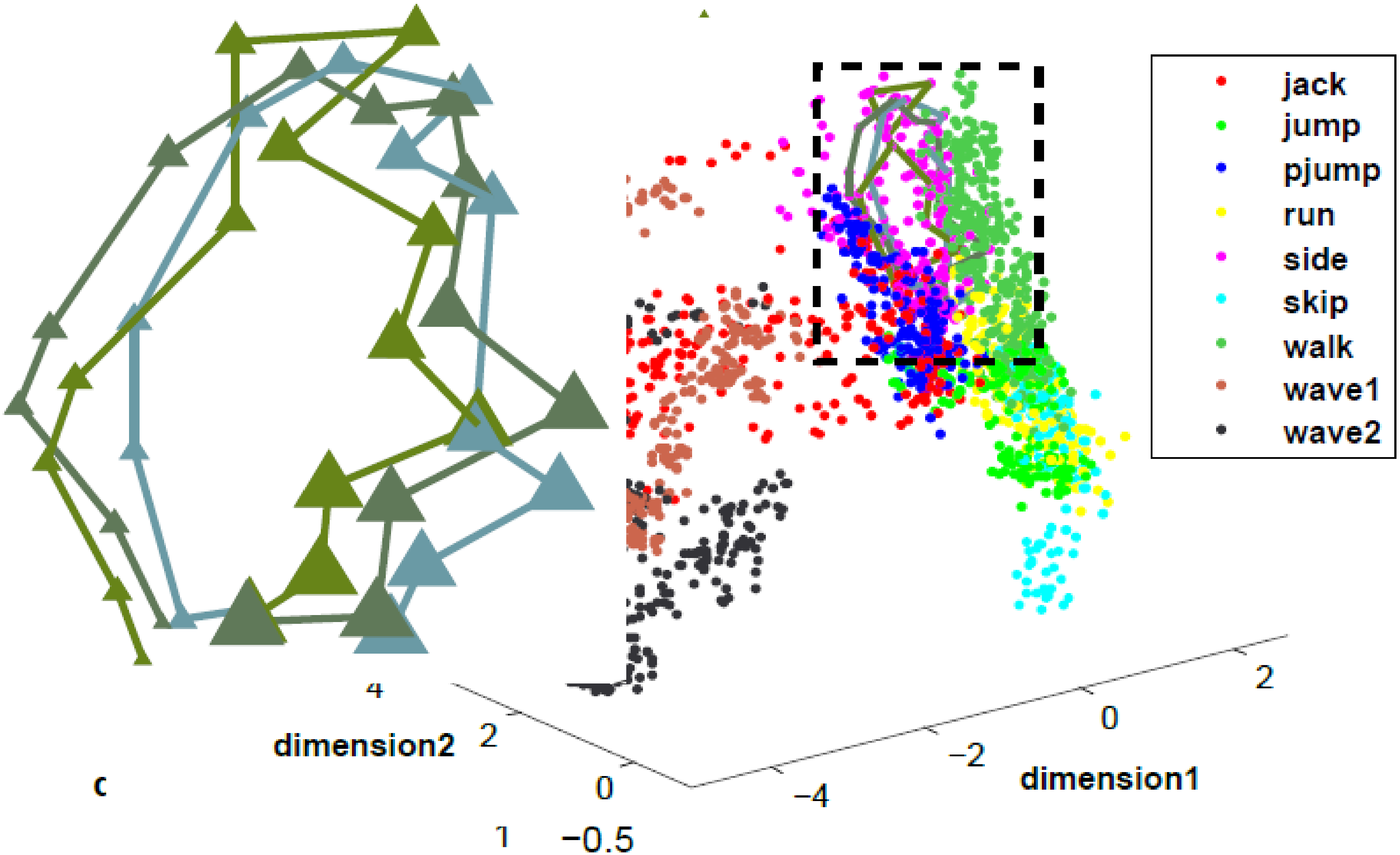
[Gorelick / Blank / Irani : 2005 / 07]

Reduced dimensionality

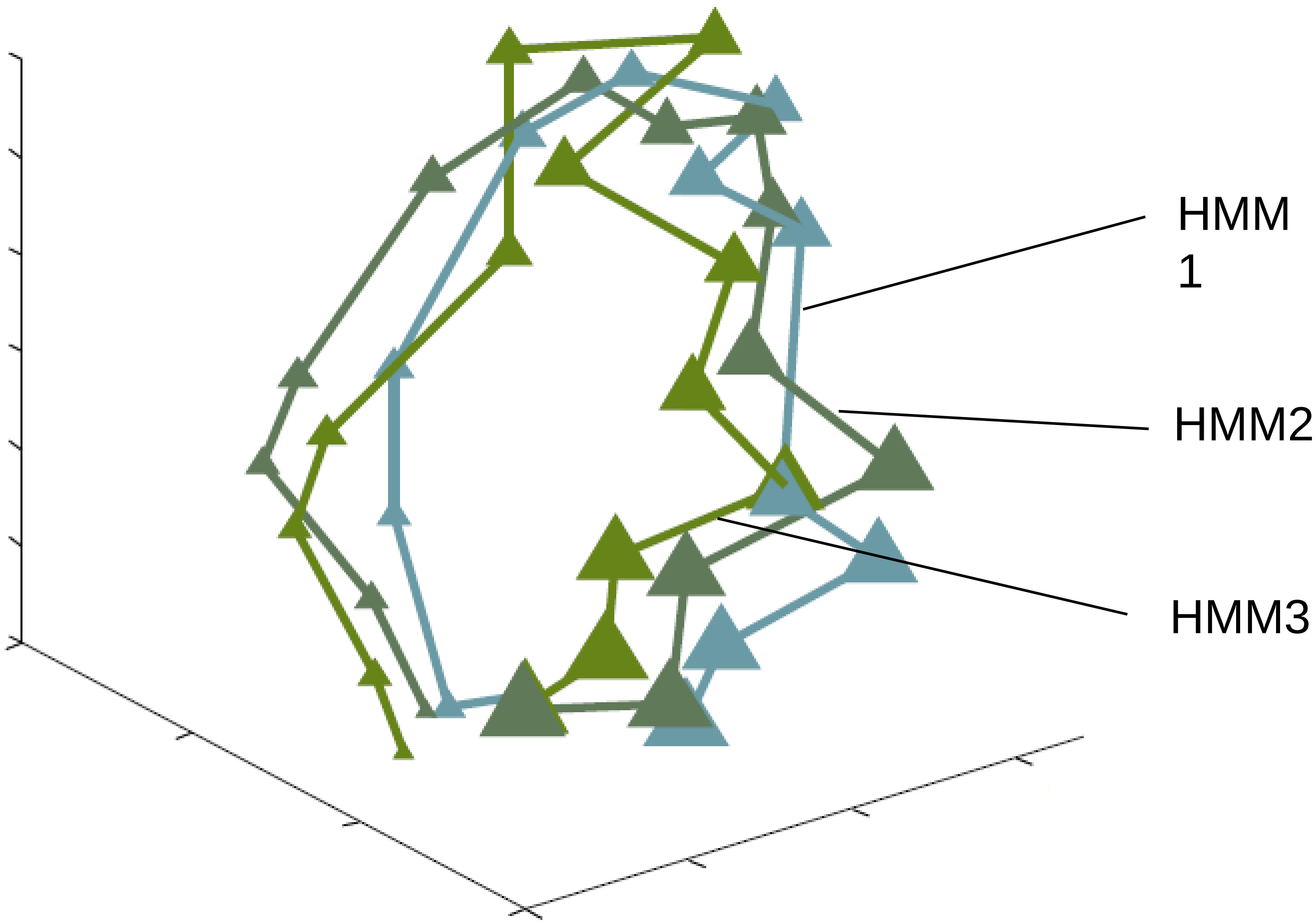


Locality Preserving Projecction
[He and Niyogi 2003]

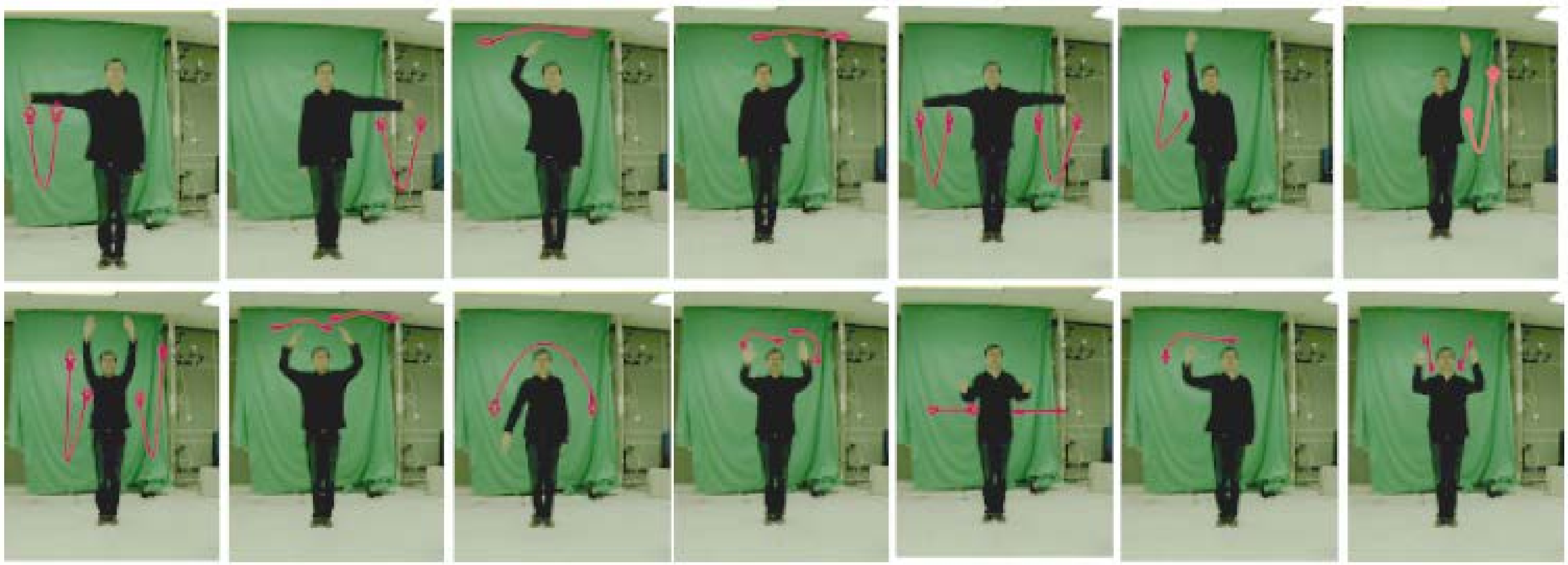
Gestures in low dimensions



Recognizing gestures



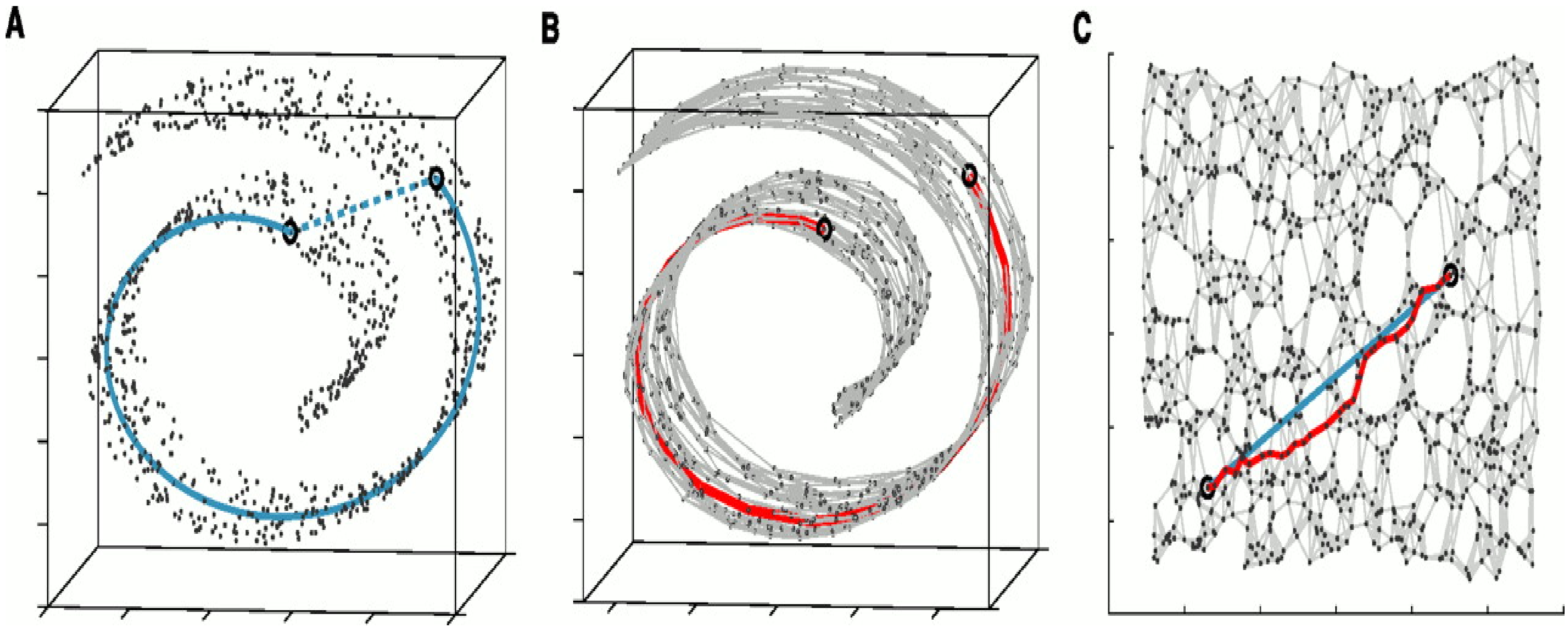
Recognizing gestures



Keck gesture dataset

Non-Linear Dimensionality Reduction (NLDR) algorithms: ISOMAP

Euclidean or Geodesic distance?



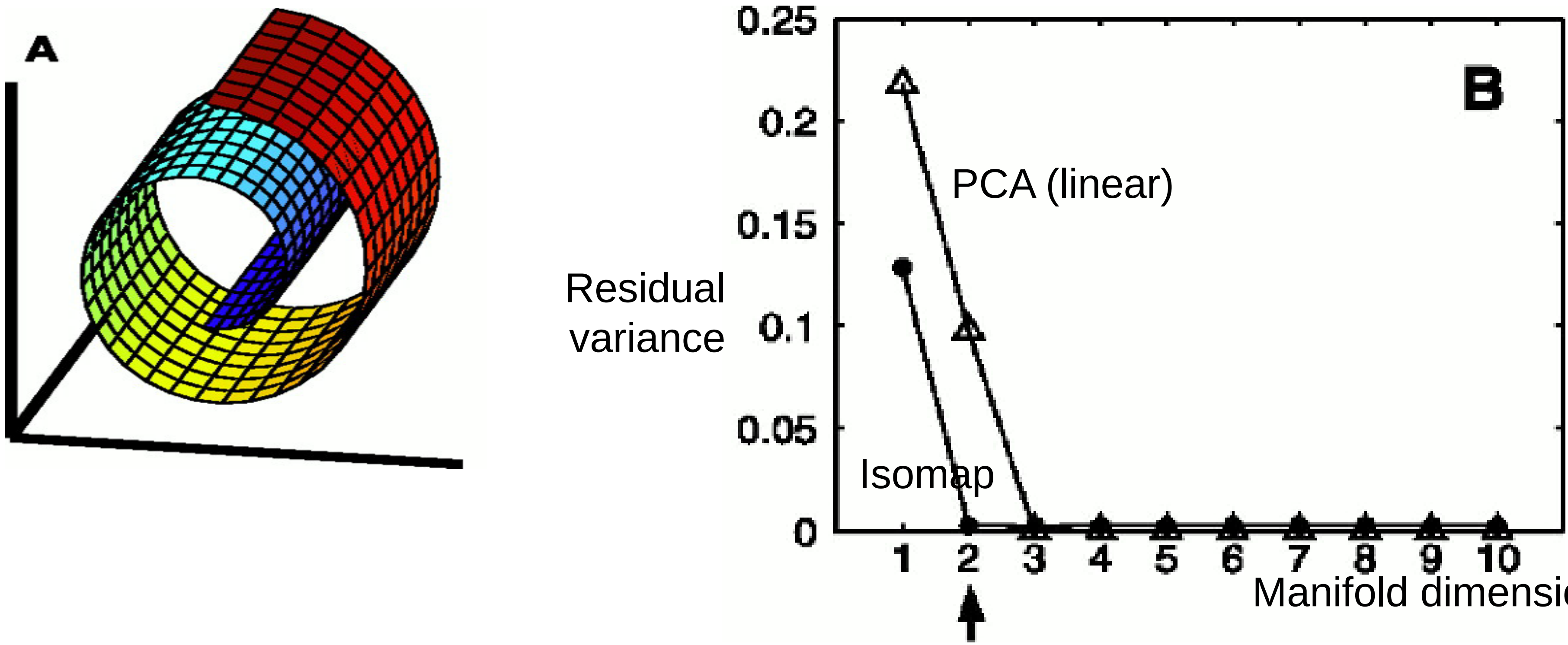
Geodesic = shortest path along manifold

Isomap Algorithm

- Identify neighbors.
  - points within epsilon-ball (ε-ball)
  - k nearest neighbors (k-NN)
- Construct neighborhood graph.
  - -- x connected to y if *neighbor(x,y)*.
  - -- edge length = distance(x,y)
- Compute shortest path between nodes
  - Djkastra / Floyd-Warshall algorithm
- Construct a lower dimensional embedding.
  - Multi-Dimensional Scaling (MDS)

[Tenenbaum, de Silva and Langford 2001]

Residual Variance and Dimensionality



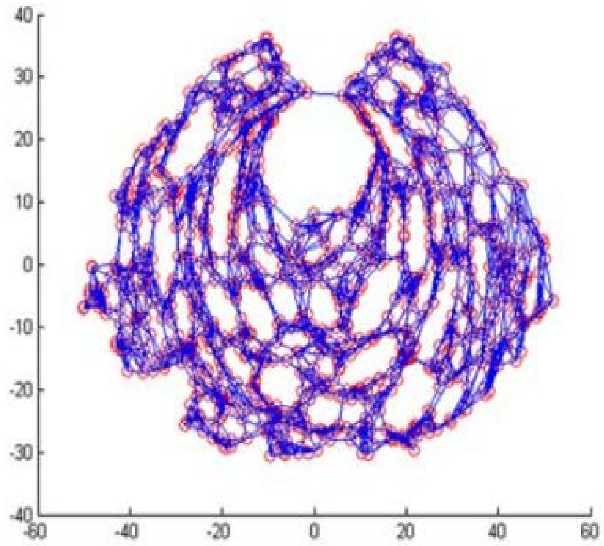
residual variance = 1 – $r^2(D_g, D_y)$; r = linear correlation coefficient
D_g = geodesic distance matrix; D_y = manifold distance

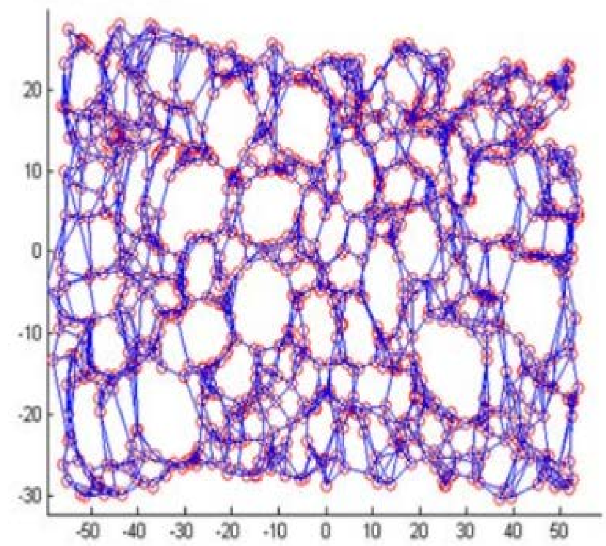
Short Circuits & Neighbourhood selection

neighbourhood size

too big: short-circuit errors

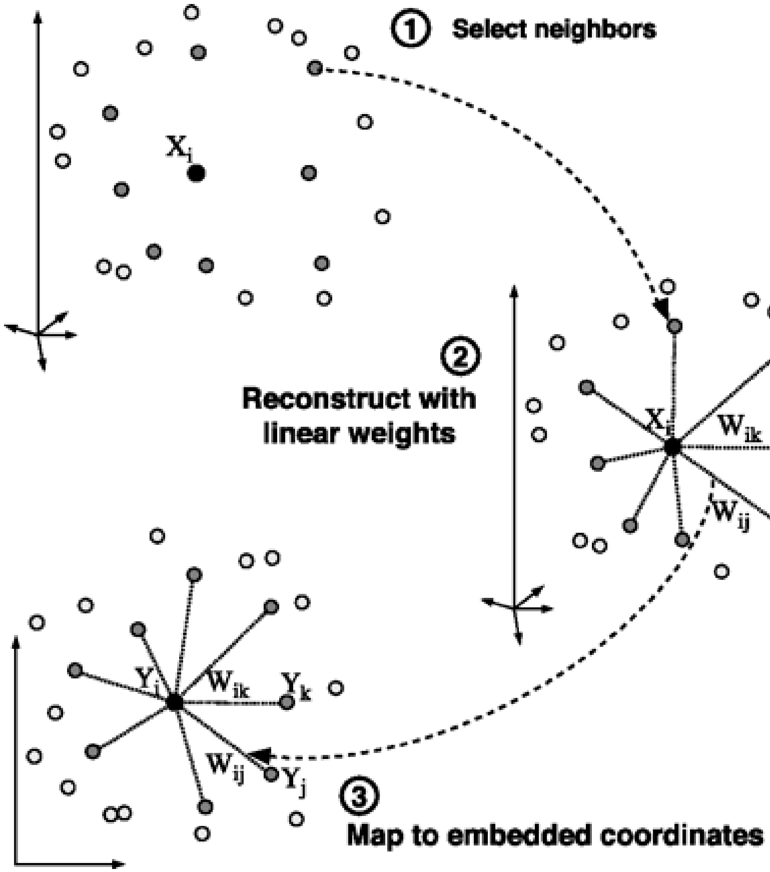
too small: isolated patches





[saxena, gupta mukerjee 04]

Locally-Linear Embedding



$$J_1(\mathbf{W}) = \sum_{i=1}^{N} \left\| \mathbf{x}_i - \sum_{j=1}^{K} W_{ij} \mathbf{x}_j \right\|^2$$

$$J_2(\mathbf{Y}) = \sum_{i=1}^{N} \left\| \mathbf{y}_i - \sum_{j=1}^{N} W_{ij} \mathbf{y}_j \right\|^2$$

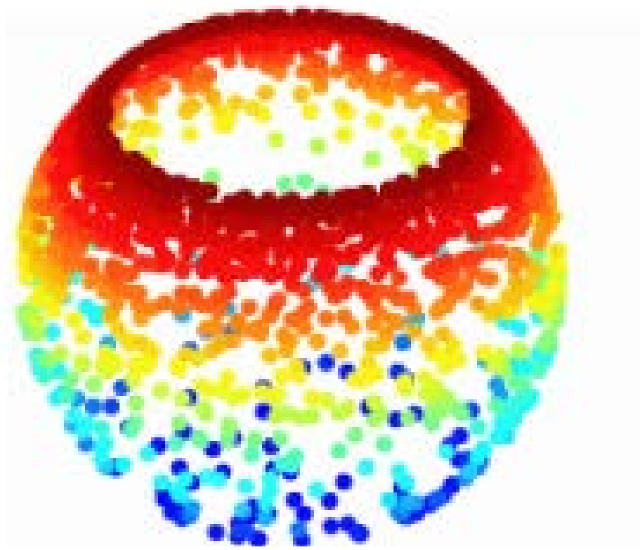
$$\mathbf{Y} = \begin{bmatrix} | & | & & | \\ \mathbf{y}_1 & \mathbf{y}_2 & \cdots & \mathbf{y}_N \\ | & | & & | \end{bmatrix}_{d\ x\ N} = \begin{bmatrix} - & \mathbf{u}_1 & - \\ - & \mathbf{u}_2 & - \\ & \vdots & \\ - & \mathbf{u}_d & - \end{bmatrix}_{d\ x}$$

Non-isometric maps

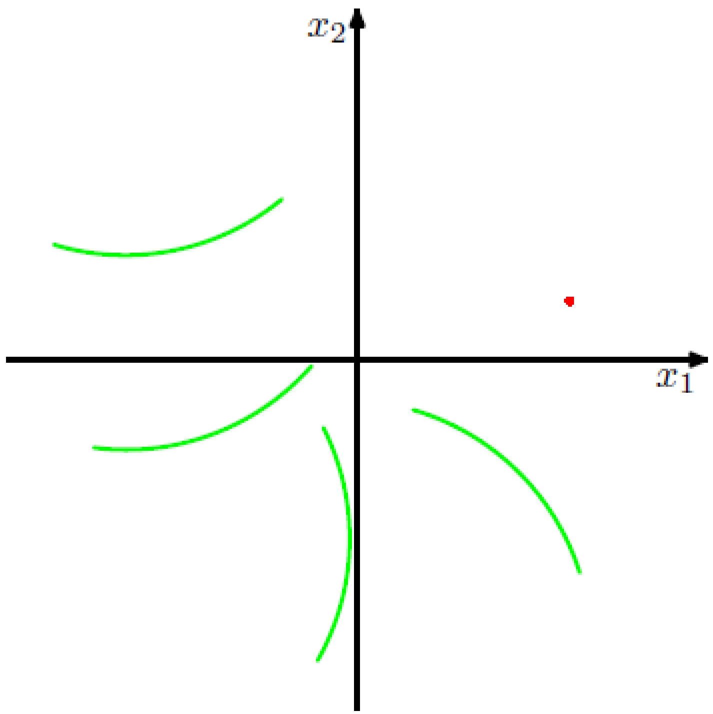
Fishbowl dataset : no isomorphic map to plane

- Conformal mappings: preserve angles,
  not distances

- Assume data is uniformly distributed in low dim



Kernel PCA



Kernel PCA

PCA: top eigenvectors of covariance matrix [XX^T]

Kernel PCA: replace X by $\phi(x)$

$$\mathbf{C} = \frac{1}{N}\sum_{n=1}^{N}\phi(\mathbf{x}_n)\phi(\mathbf{x}_n)^{\mathrm{T}}$$

Eigenvalue expression $\boldsymbol{C}\boldsymbol{v}_i = \lambda_i \boldsymbol{v}_i$

$$\frac{1}{N}\sum_{n=1}^{N}\phi(\mathbf{x}_n)\left\{\phi(\mathbf{x}_n)^{\mathrm{T}}\mathbf{v}_i\right\} = \lambda_i\mathbf{v}_i$$

To express in terms of kernel fn $k(x_n, x_m) = \varphi(x_n)^T\varphi(x_m)$, substitute

$$\mathbf{v}_i = \sum_{n=1}^{N} a_{in}\phi(\mathbf{x}_n)$$

Bishop section 12.5

Kernel PCA

$$\frac{1}{N}\sum_{n=1}^{N}\phi(\mathbf{x}_n)\phi(\mathbf{x}_n)^{\mathrm{T}}\sum_{m=1}^{N}a_{im}\phi(\mathbf{x}_m) = \lambda_i\sum_{n=1}^{N}a_{in}\phi(\mathbf{x}_n).$$

Multiply both sides by $\varphi(x_n)^T$

$$\frac{1}{N}\sum_{n=1}^{N}k(\mathbf{x}_l,\mathbf{x}_n)\sum_{m=1}^{m}a_{im}k(\mathbf{x}_n,\mathbf{x}_m) = \lambda_i\sum_{n=1}^{N}a_{in}k(\mathbf{x}_l,\mathbf{x}_n).$$

which reduces to

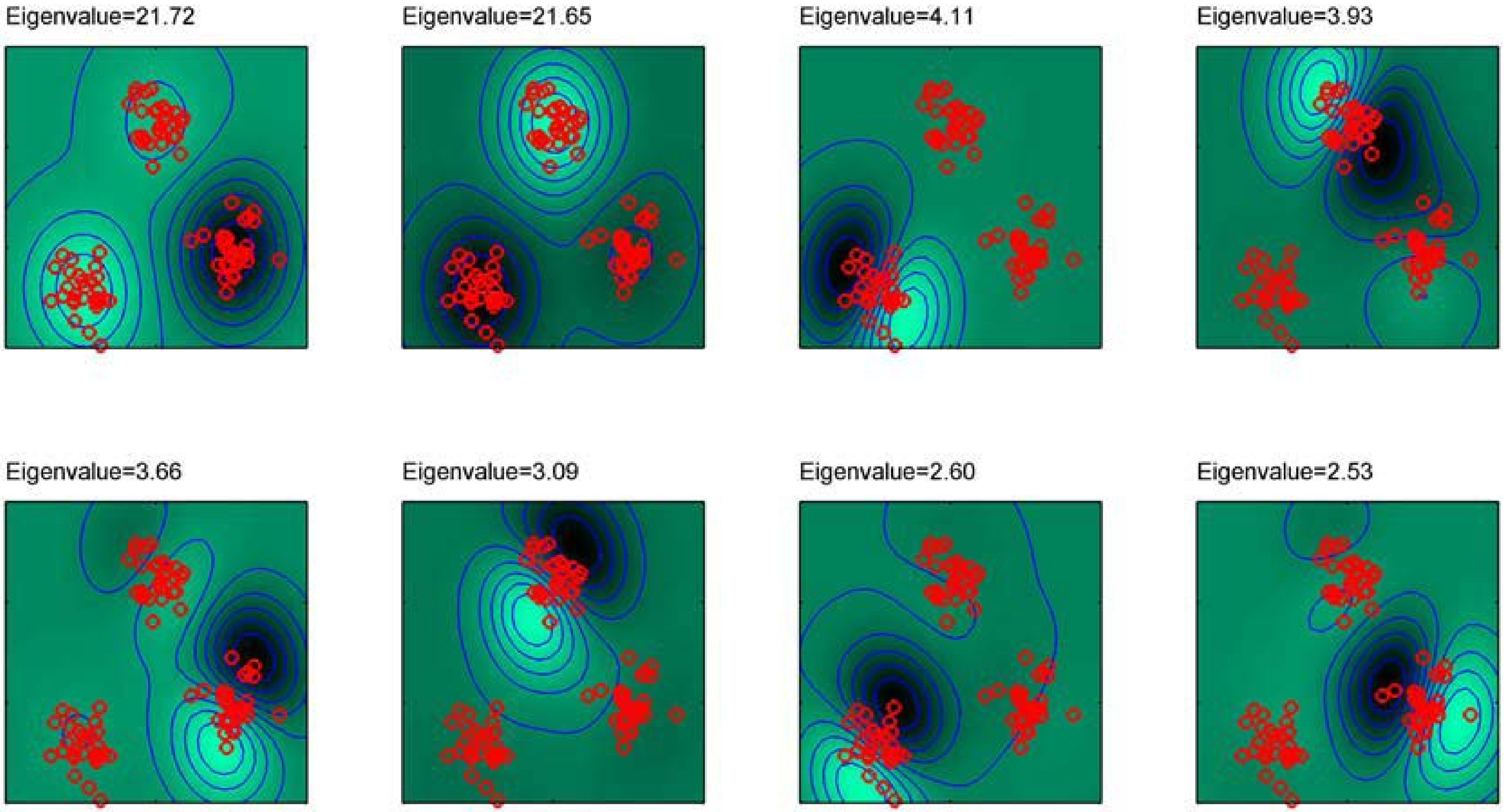
$$\boldsymbol{K}\boldsymbol{a}_i = \lambda_i N \boldsymbol{a}_i$$

(a K is removed from both sides – affects only zero λ_i).

Projections yi = $\sum_{n=1}^{N}a_{in}k(\mathbf{x},\mathbf{x}_n)$

What happens when we use a linear kernel k(x, x') = $x^T x'$?

Kernel PCA : Demonstration



Kernel: $k(x, x') = \exp(-|x - x'|^2 / 0.1)$

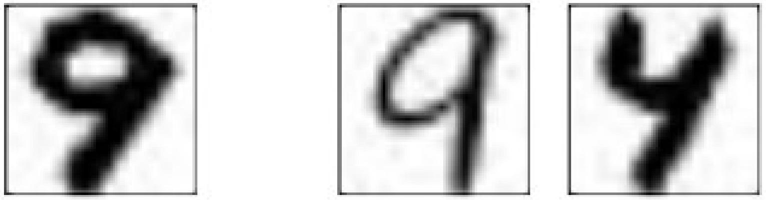
[Scholkopf 98]

Learning representations: Handwritten Digits

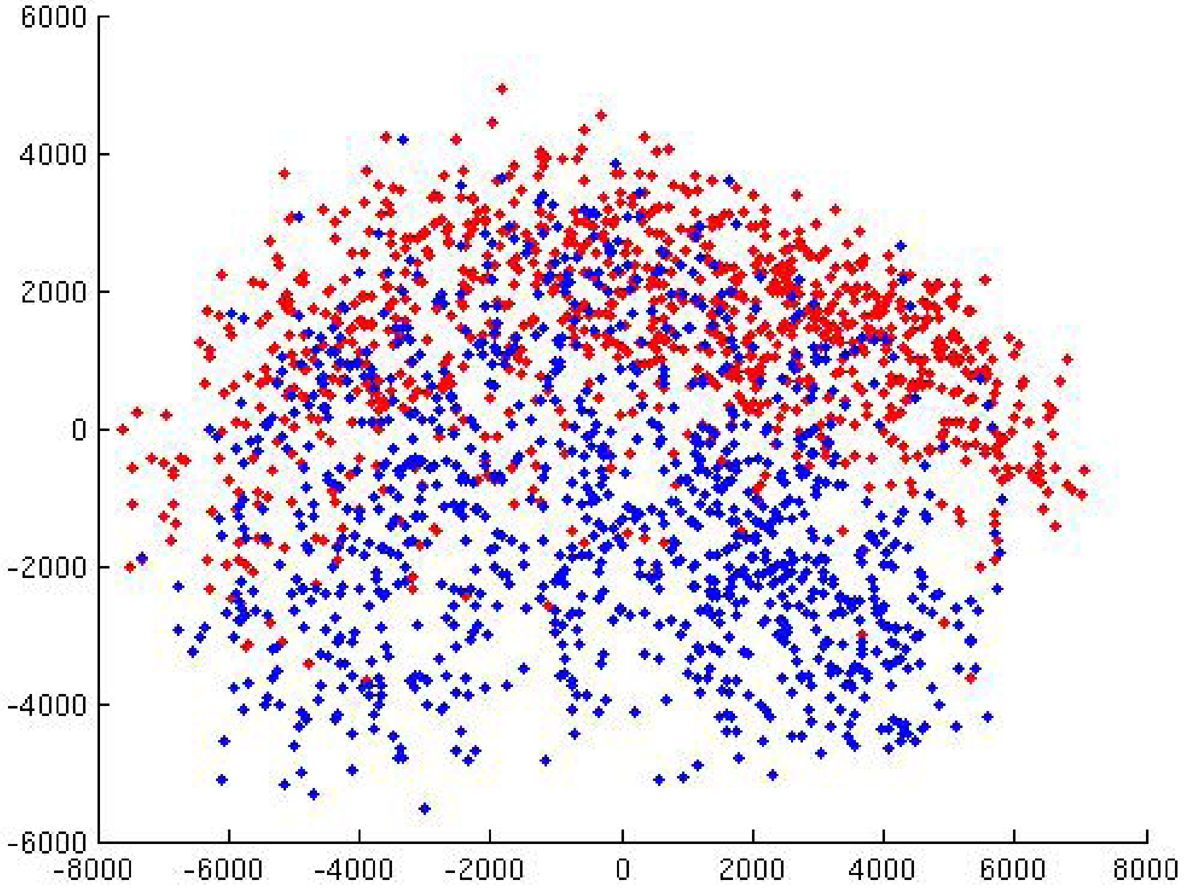
handwritttten numerals (MNIST)

Modified NIST digits database: 60K + 10K 28x28 images

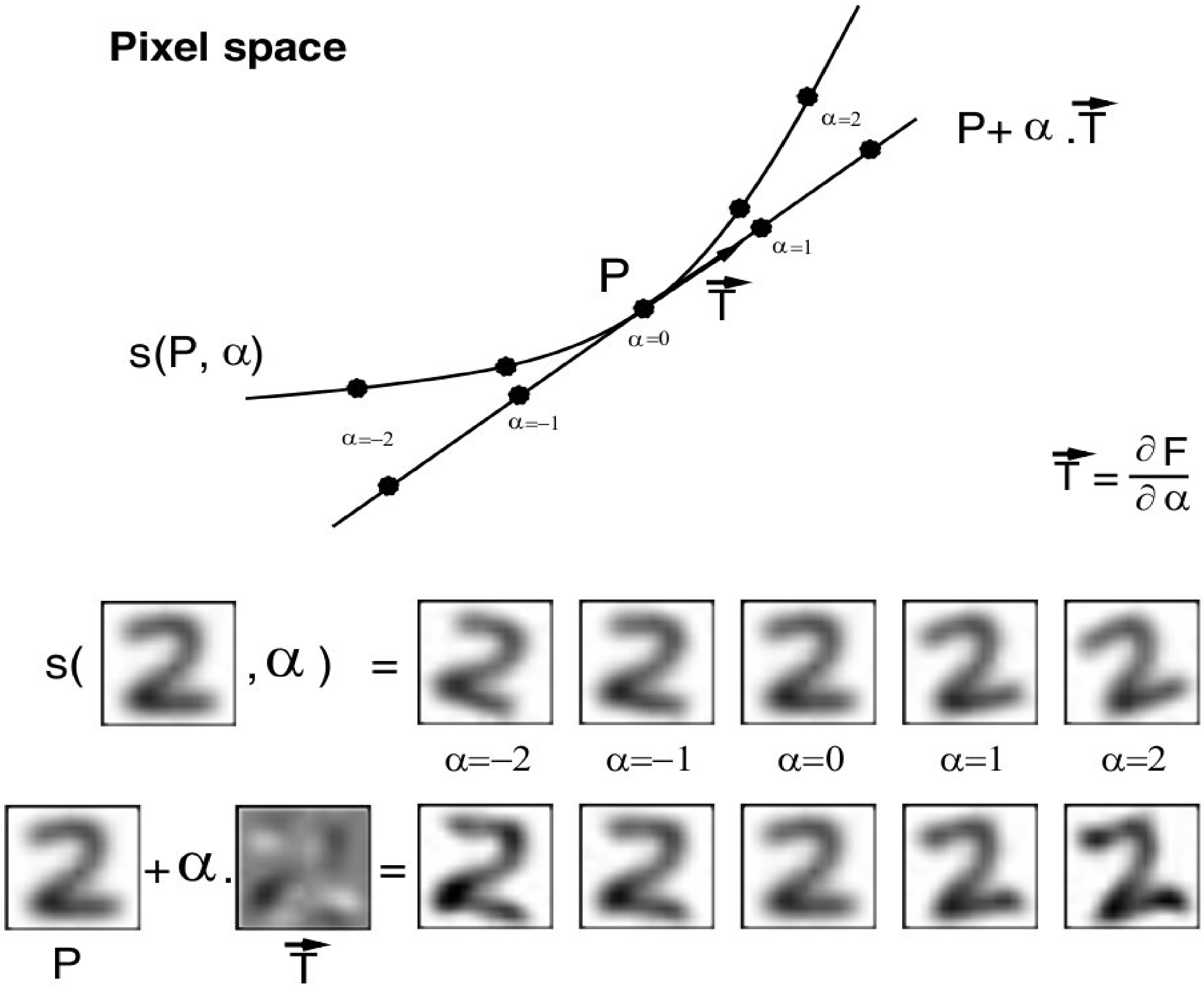
Importance of choosing a metric



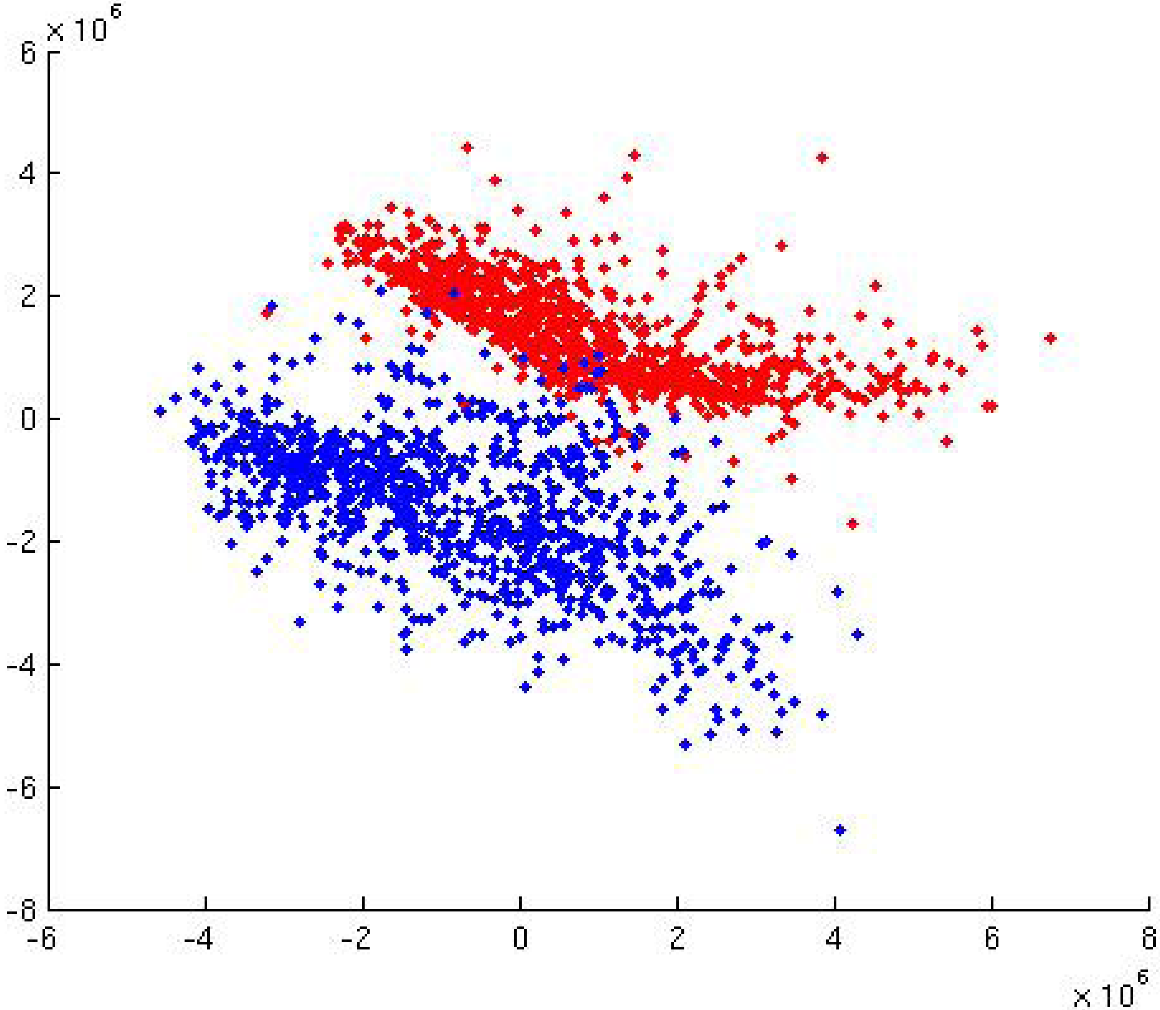
Manifold mapping with Euclidean Distance

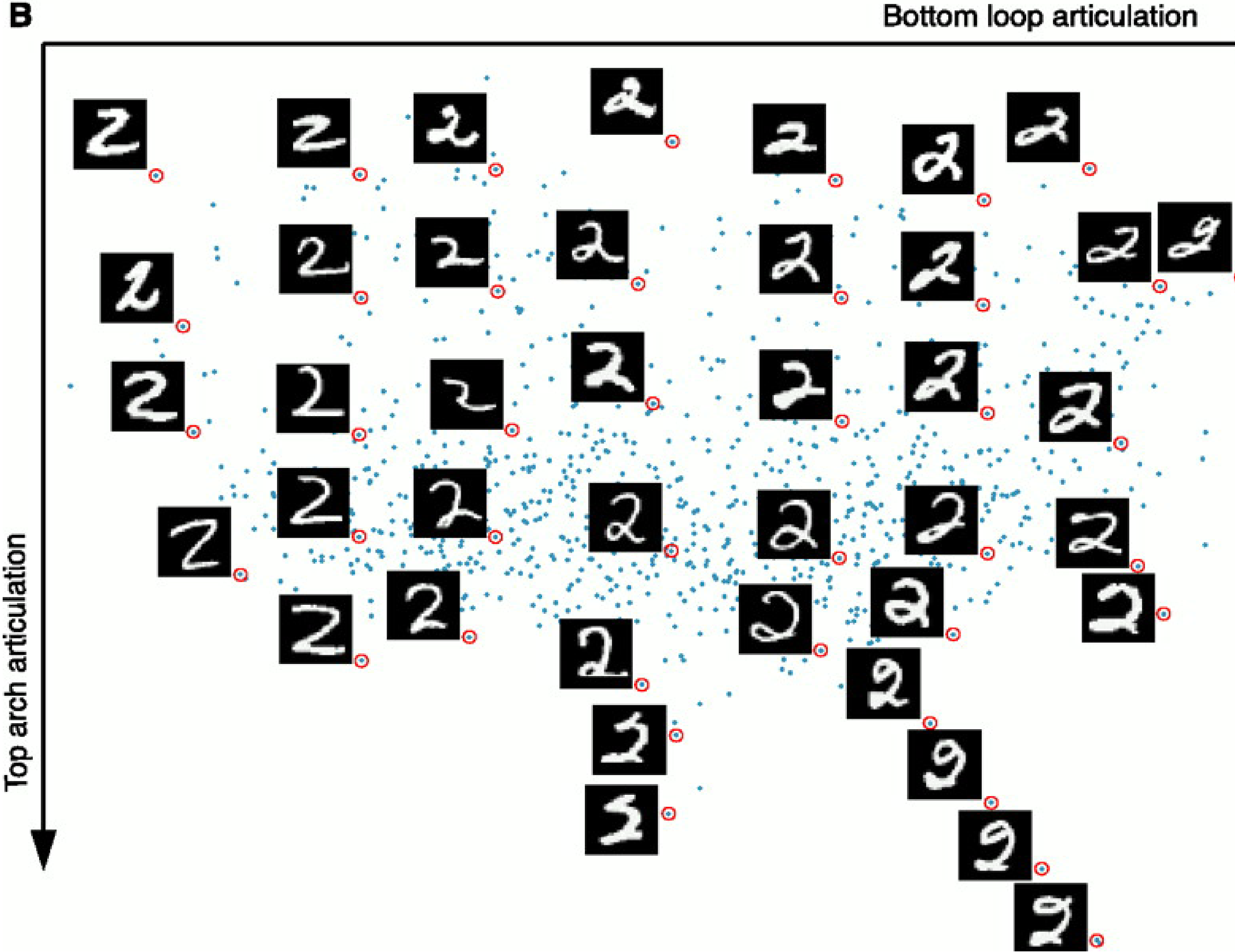


"tangent distance"

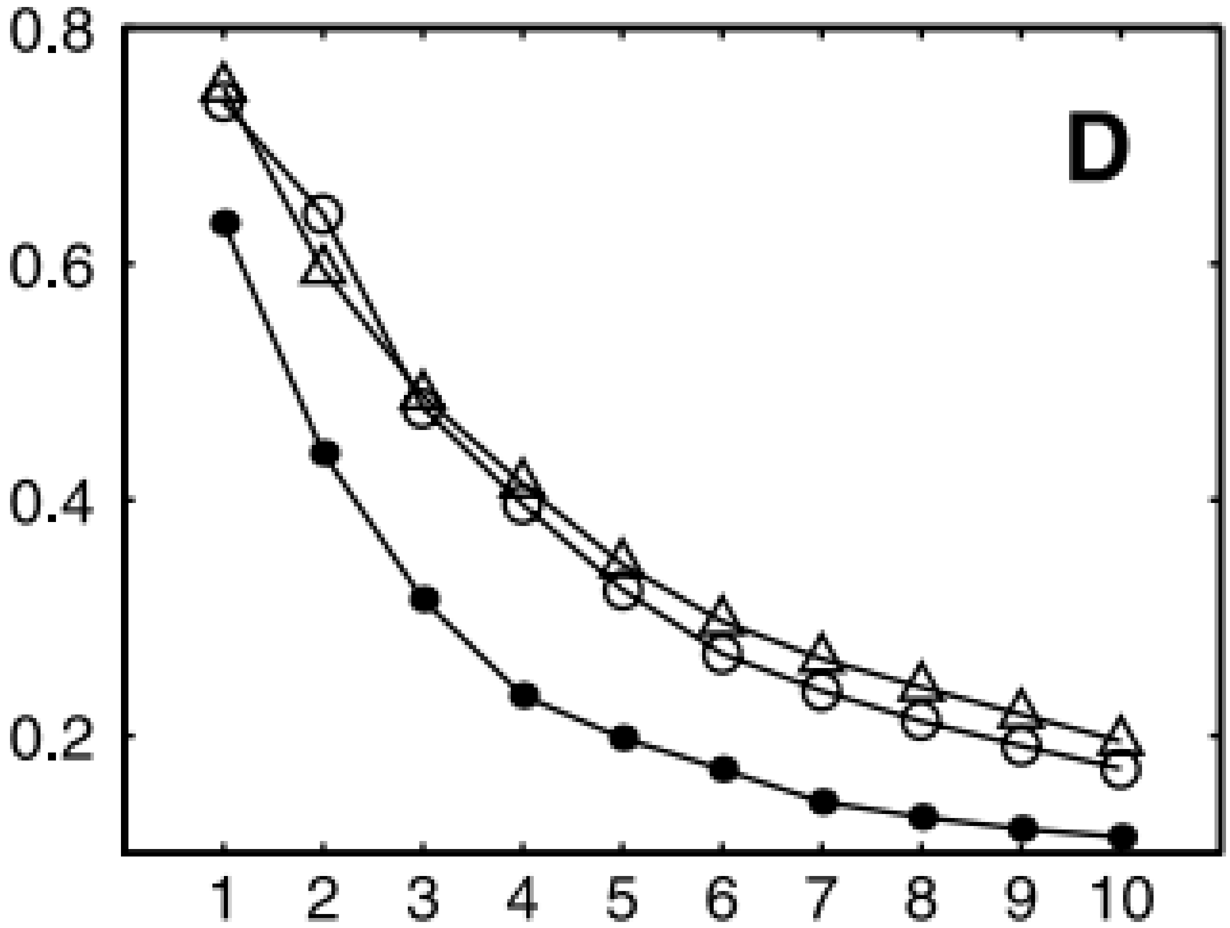


Manifold mapping with




B
Bottom loop articulation
Top arch articulation

Dimensionality: handwritten digits

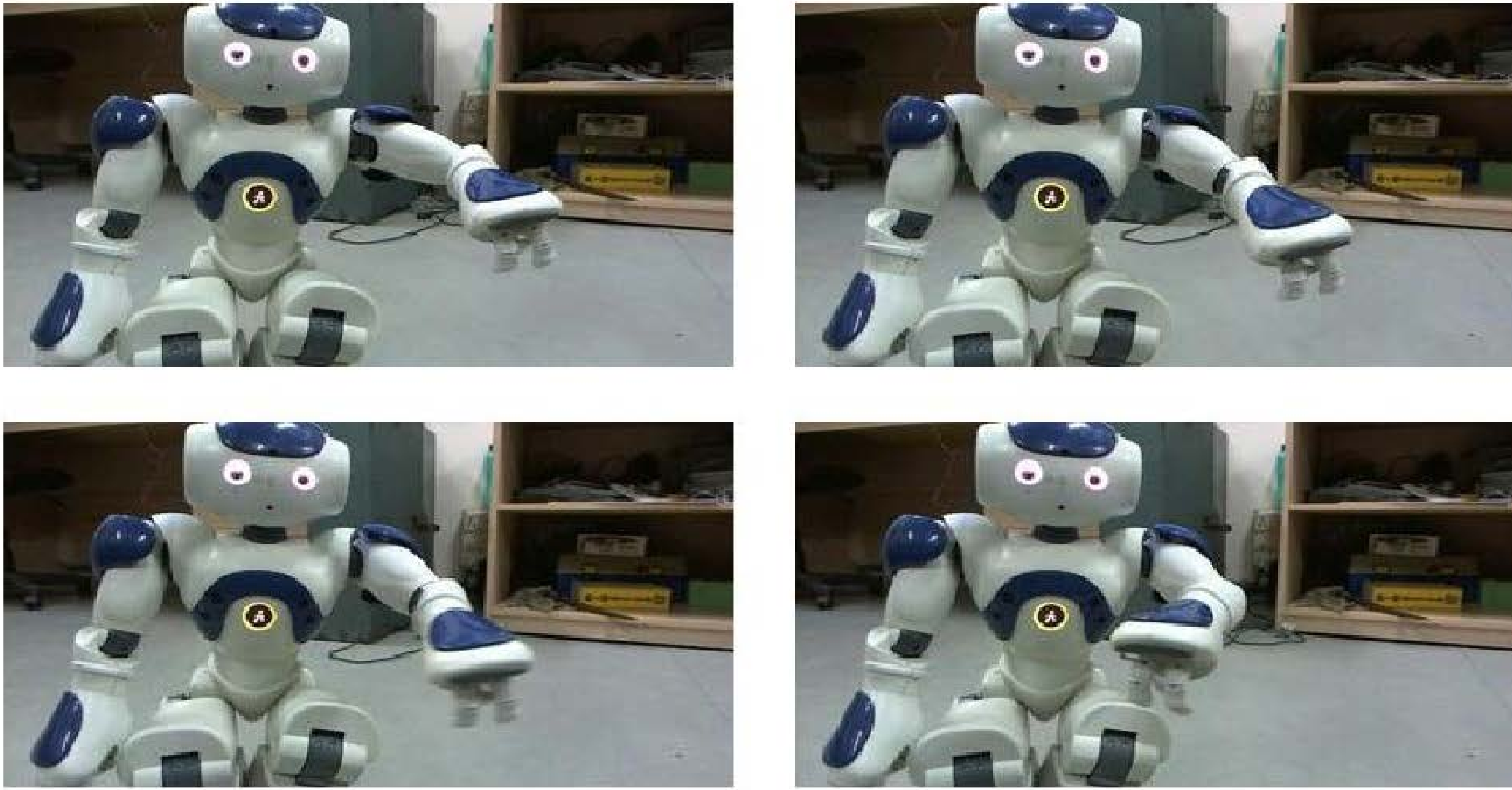


Residual variance

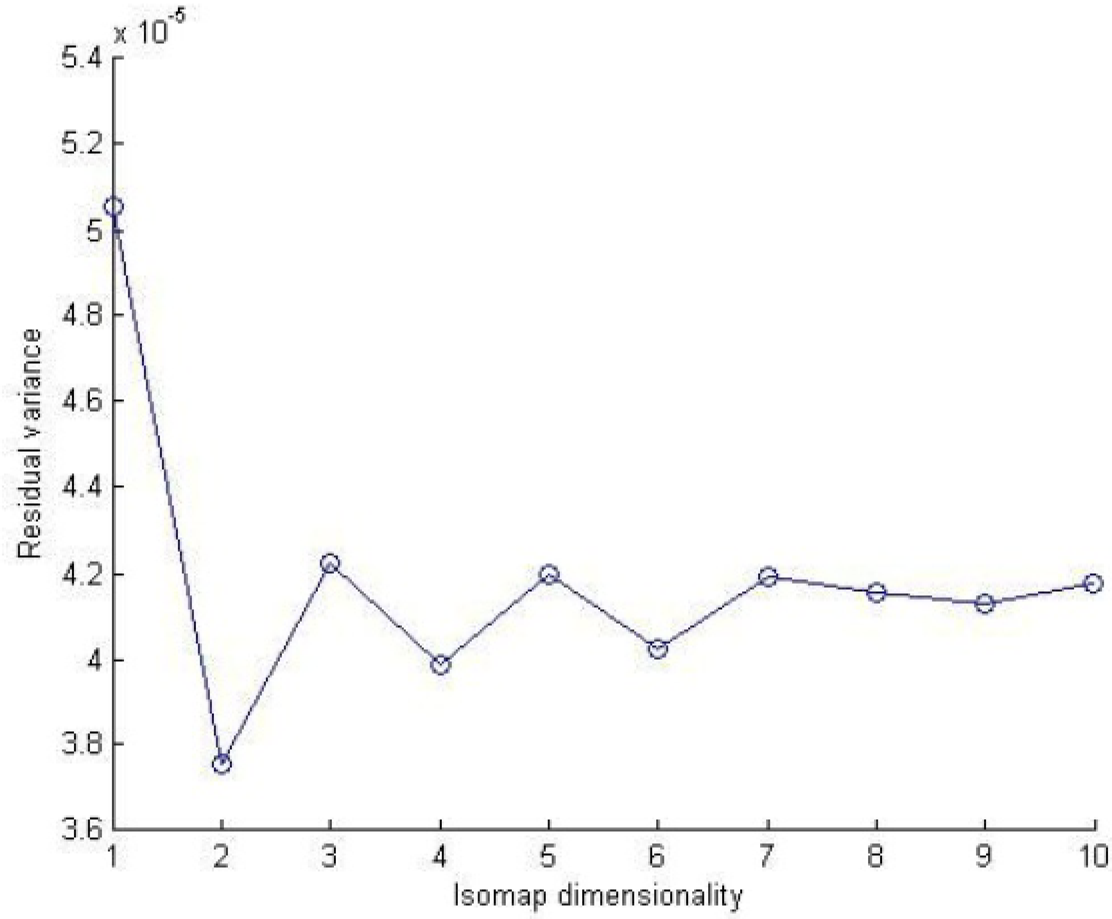
Manifold dimension

NLDR algorithms: Representing a robot

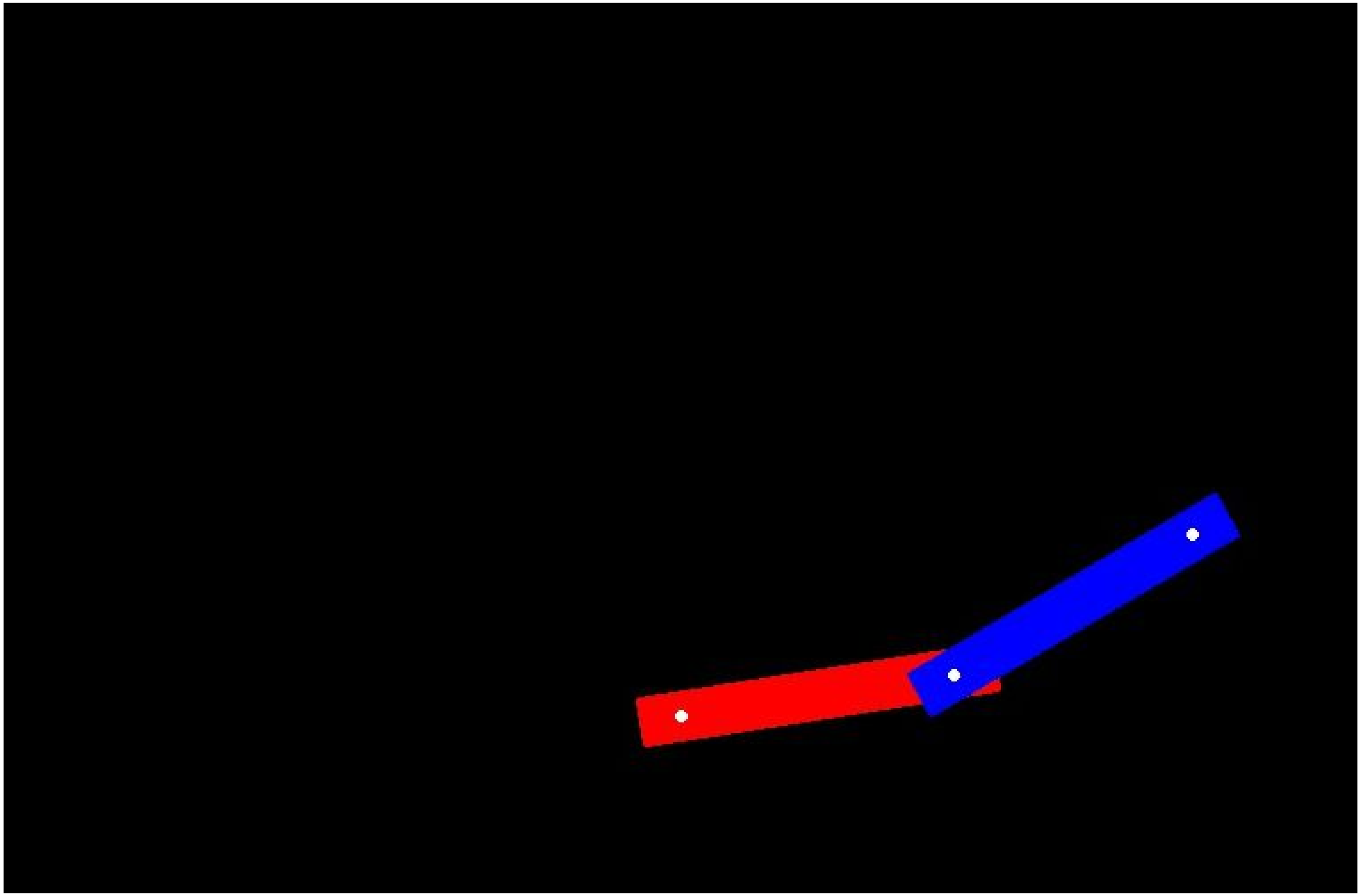
Input = images



Manifold dimension



2-DOF motion

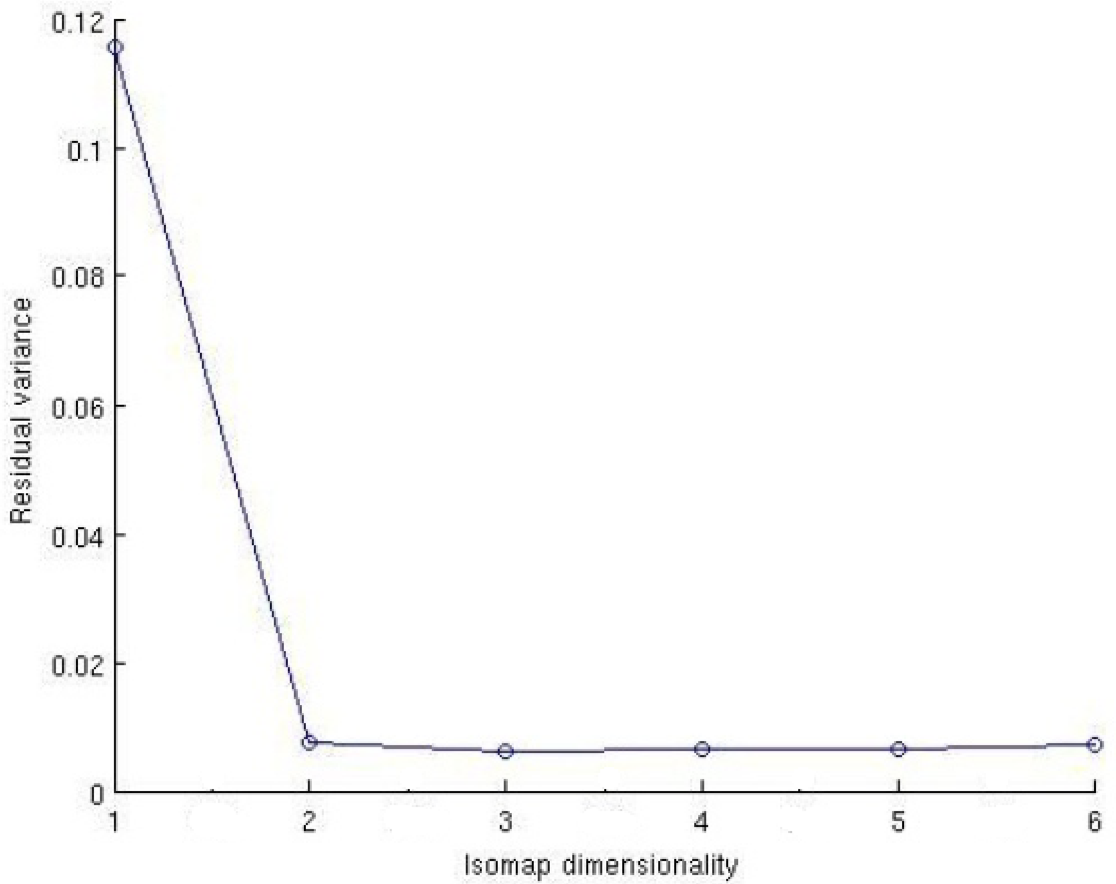


Dimensionality reduction

- Identify neighbors.
  - “neighbours” may have link1 in same pose, but link 2 varying
  - Alternately, link2 similar, but link1 varying

    → variation is along 2 dimensions in image space,
- Construct neighborhood graph.
- Compute geodesics = shortest path between nodes
- Find low-dimensional embedding that preserves geodesics
  - Target dimension for low-D space not known. Just try 1,2,3, … n

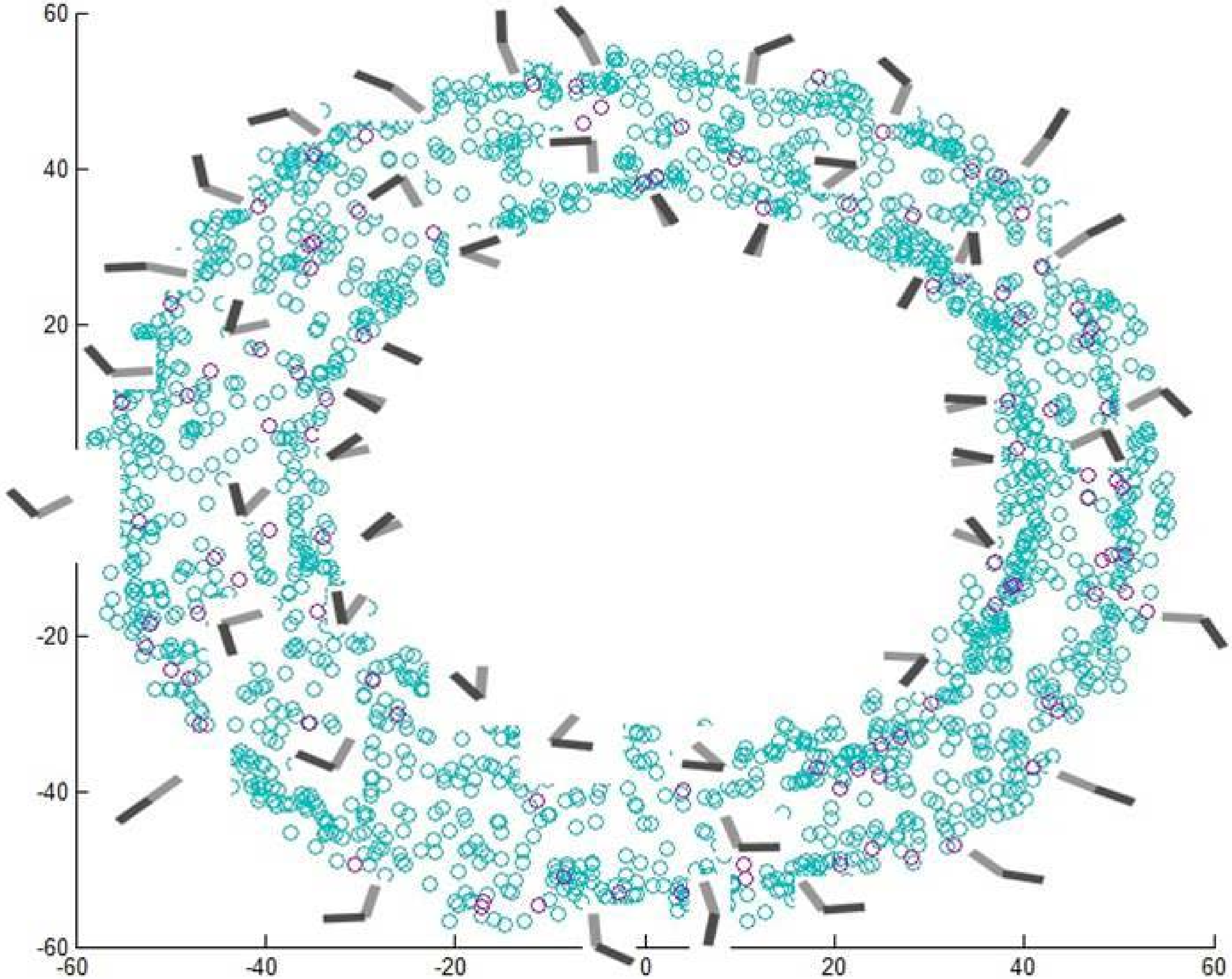
Residual variance vs dimension



dofs : 2

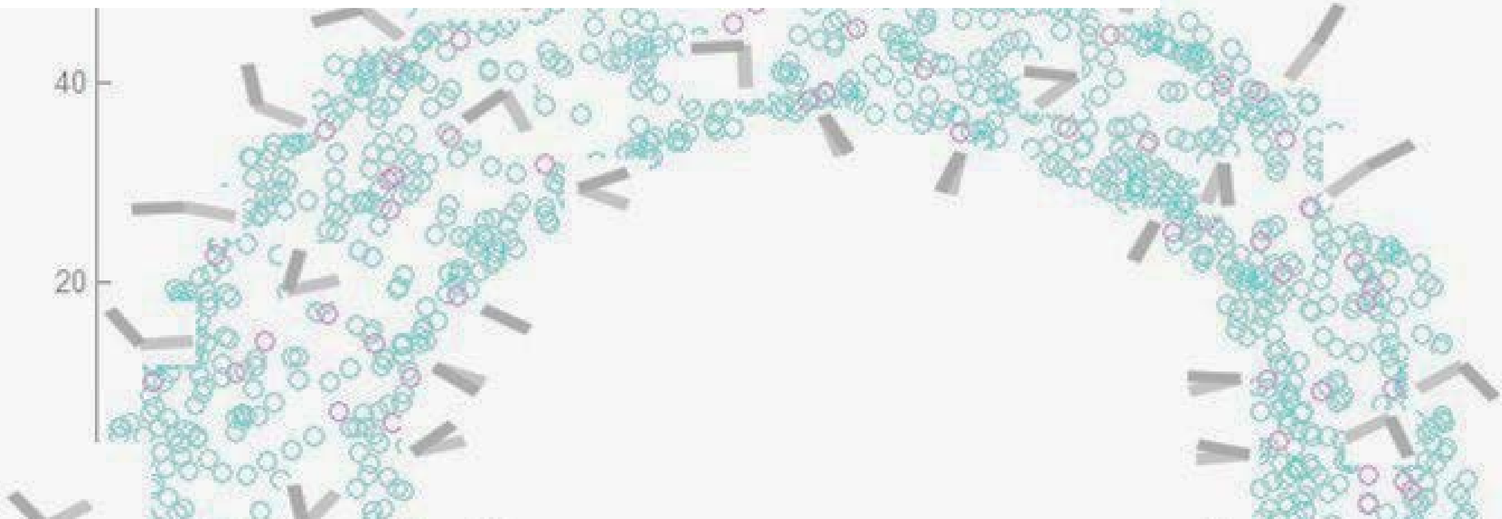
Robot Structure Learning

- Consider many images of robot configurations
  - Construct manifold on images
  - Dimensionality that explains variance
- Resultant graph
  - -- *neighbor(x,y)* → neighbouring configurations.
  - – Topology (for unbounded theta) = torus



60
40
20
-20
-40
-60
-60
-40
-20
0
20
40
60

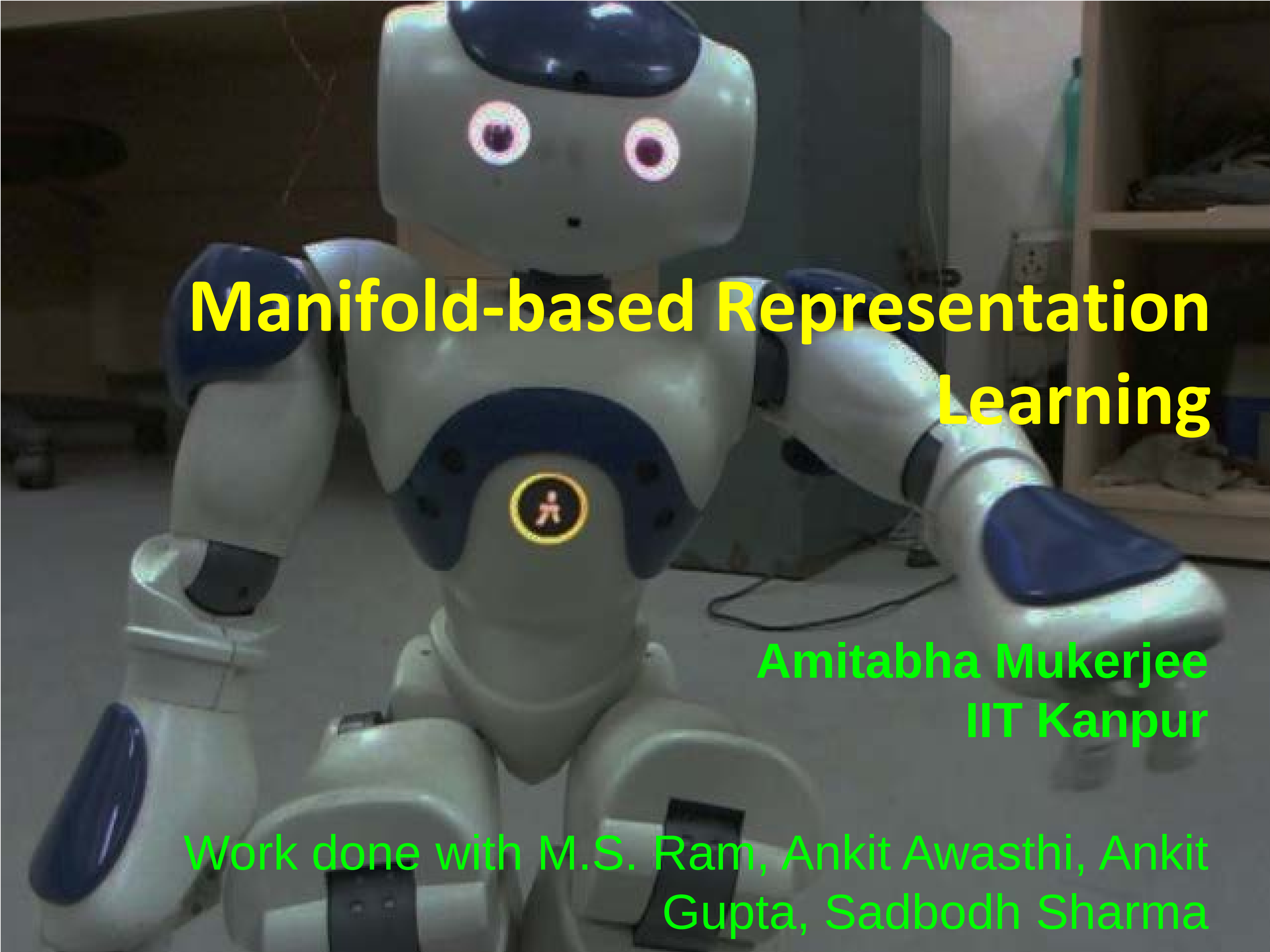
Robot Structure Learning



- Latent variables:
  - Distributed on $S^1 \times S^1$ topology
  - Along circumferential path: θ_1; along radial: θ_2
  - Naïve, non-metric representation of θ_1,θ_2
- Manifold transformation = mapping between input images (workspace) ↔ naive θ_1,θ_2 (C-space)
  - manifold → image ≈ naïve forward kinematics
  - image → manifold ≈ naïve inverse kinematics

Differences with AI representation

- Grounding
  - AI models are defined only in terms of other logical structures $\rightarrow$ circularity of definitions
  - Manifold-based naïve representations : grounded on sensory data:
- Physics like formulation (θ_1,θ_2) may not be needed
  - Topologically consistent representation of (θ_1,θ_2)
  - Non-uniform sampling $\rightarrow$ higher resolution for functionally relevant regions



Manifold-based Representation Learning

Amitabha Mukerjee
IIT Kanpur

Work done with M.S. Ram, Ankit Awasthi, Ankit Gupta, Sadbodh Sharma