Scientific and Large Data Visualization
29 November 2017
High Dimensional Data – Part II

Massimiliano Corsini

Visual Computing Lab, ISTI - CNR - Italy

Overview

- Graphs Extensions
- Glyphs
  - Chernoff Faces
  - Multi-dimensional Icons
- Parallel Coordinates
- Star Plots
- **Dimensionality Reduction**
  - **Principal Component Analysis (PCA)**
  - **Locally Linear Embedding (LLE)**
  - **IsoMap**
  - **Summon Mapping**
  - **t-SNE**

Dimensionality Reduction

- *N*-dimensional data are projected to 2 or 3 dimensions for better visualization/ understanding.
- Widely used strategy.
- In general, it is a mapping not a geometric transformation.
- Different mappings have different properties.

Principal Component Analysis (PCA)

- A classic multi-dimensional reduction technique is Principal Component Analysis (PCA).
- It is a linear non-parametric technique.
- The core idea to find a basis formed by the directions that maximize the variance of the data.

PCA as a change of basis

- The idea is to express the data in a new basis, that *best* express our dataset.

$$\mathbf{PX} = \mathbf{Y}$$

- The new basis is a linear combination of the original basis.

PCA as a change of basis

$$\mathbf{PX} = \left[\begin{array}{c} \mathbf{p}_1 \\ \vdots \\ \mathbf{p}_m \end{array} \right] \left[\mathbf{x}_1 \ldots \mathbf{x}_n \right]$$

$$\mathbf{y}_i = \left[\begin{array}{c} \mathbf{p}_1 \cdot \mathbf{x}_i \\ \vdots \\ \mathbf{p}_m \cdot \mathbf{x}_i \end{array} \right]$$

Signal-to-noise Ratio (SNR)

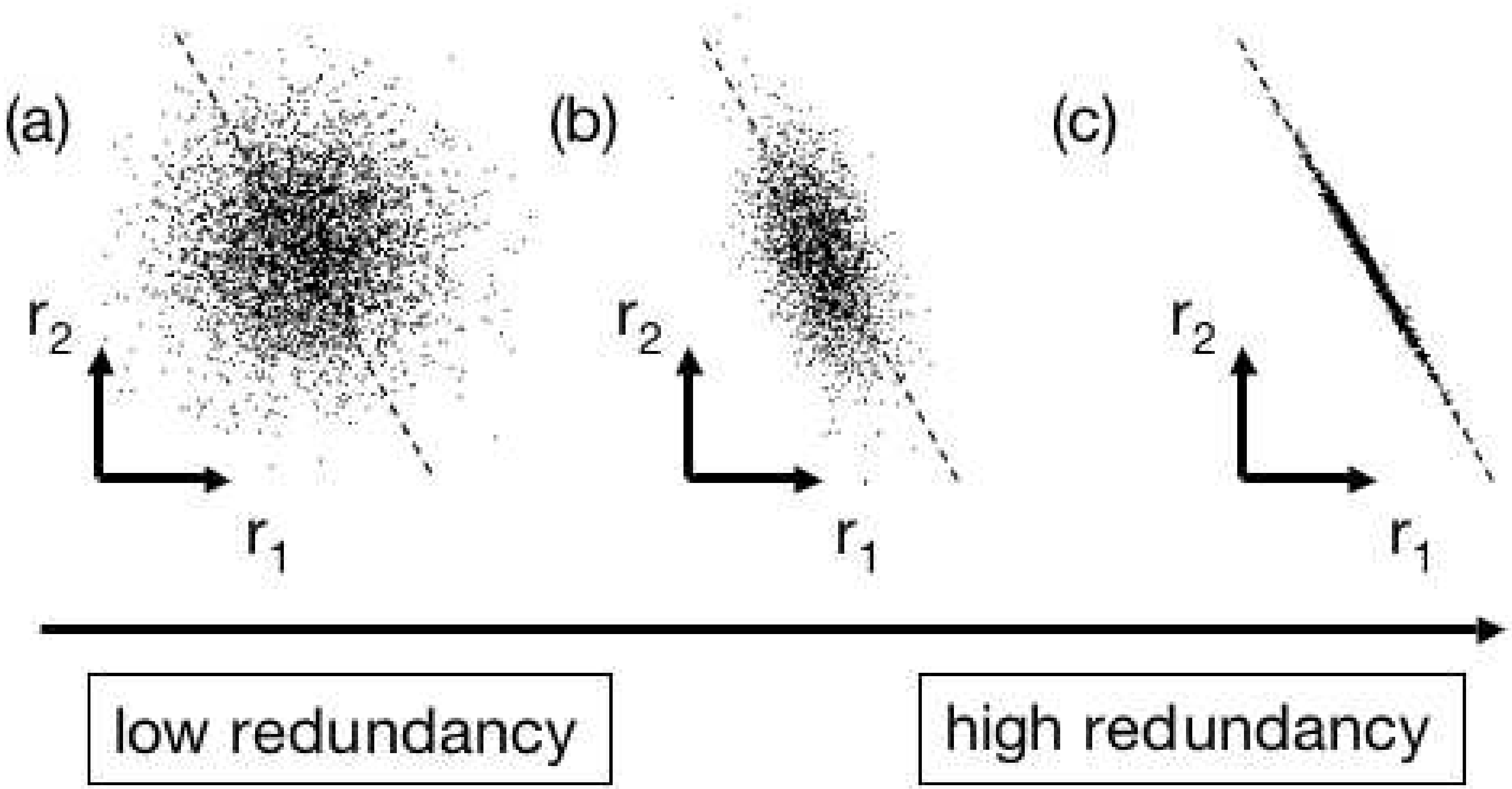
- Given a signal with noise:

$$SNR = \frac{P_{signal}}{P_{noise}}$$

- It can be expressed as:

$$SNR = \frac{\sigma^2_{signal}}{\sigma^2_{noise}}$$

Redundancy



Redundant variables convey no relevant information!

Figure From Jonathon Shlens, *"A Tutorial on Principal Component Analysis"*, arXiv preprint arXiv:1404.1100, 2015.

Covariance Matrix

$$Cov(\mathbf{X}) = \mathbf{C_X} = \frac{\mathbf{1}}{\mathbf{n-1}}\mathbf{X}\mathbf{X}^{\mathbf{T}}$$

- Square symmetric matrix.
- The diagonal terms are the variance of a particular variable.
- The off-diagonal terms are the covariance between the different variables.

Goals

- How to select the best **P** ?
  - Minimize redundancy
  - Maximize the variance
- Goal: to diagonalize the covariance matrix of **Y**
  - High values of the diagonal terms means that the dynamics of the single variables has been maximized.
  - Low values of the off-diagonal terms means that the redundancy between variables is minimized.

Solving PCA

$$
\begin{aligned}
\mathbf{C_Y} &= \frac{1}{n-1}\mathbf{Y}\mathbf{Y}^T \\
&= \frac{1}{n-1}(\mathbf{P}\mathbf{X})(\mathbf{P}\mathbf{X})^T \\
&= \frac{1}{n-1}\mathbf{P}\mathbf{X}\mathbf{X}^T\mathbf{P}^T \\
&= \frac{1}{n-1}\mathbf{P}(\mathbf{X}\mathbf{X}^T)\mathbf{P}^T \\
\mathbf{C_Y} &= \frac{1}{n-1}\mathbf{P}\mathbf{A}\mathbf{P}^T
\end{aligned}
$$

Remember that

$$\mathbf{Y} = \mathbf{P}\mathbf{X}$$

Solving PCA

- Theorem: a symmetric matrix ***A*** can be diagonalized by a matrix formed by its eigenvectors as $\boldsymbol{A = EDE^T}$.
- The column of ***E*** are the eigenvectors of ***A***.

PCA Computation

- Organize the data as an *m x n* matrix.
- Subtract the corresponding mean to each row.
- Calculate the eigenvalues and eigenvectors of $\boldsymbol{XX^T}$.
- Organize them to form the matrix ***P***.

PCA for Dimensionality Reduction

- The idea is to find the *k*-th principal components ($k < m$).
- Project the data on these directions and use such data instead of the original ones.
- This data are the best approximation w.r.t the sum of the squared differences.

PCA as the Projection that Minimizes the Reconstruction Error

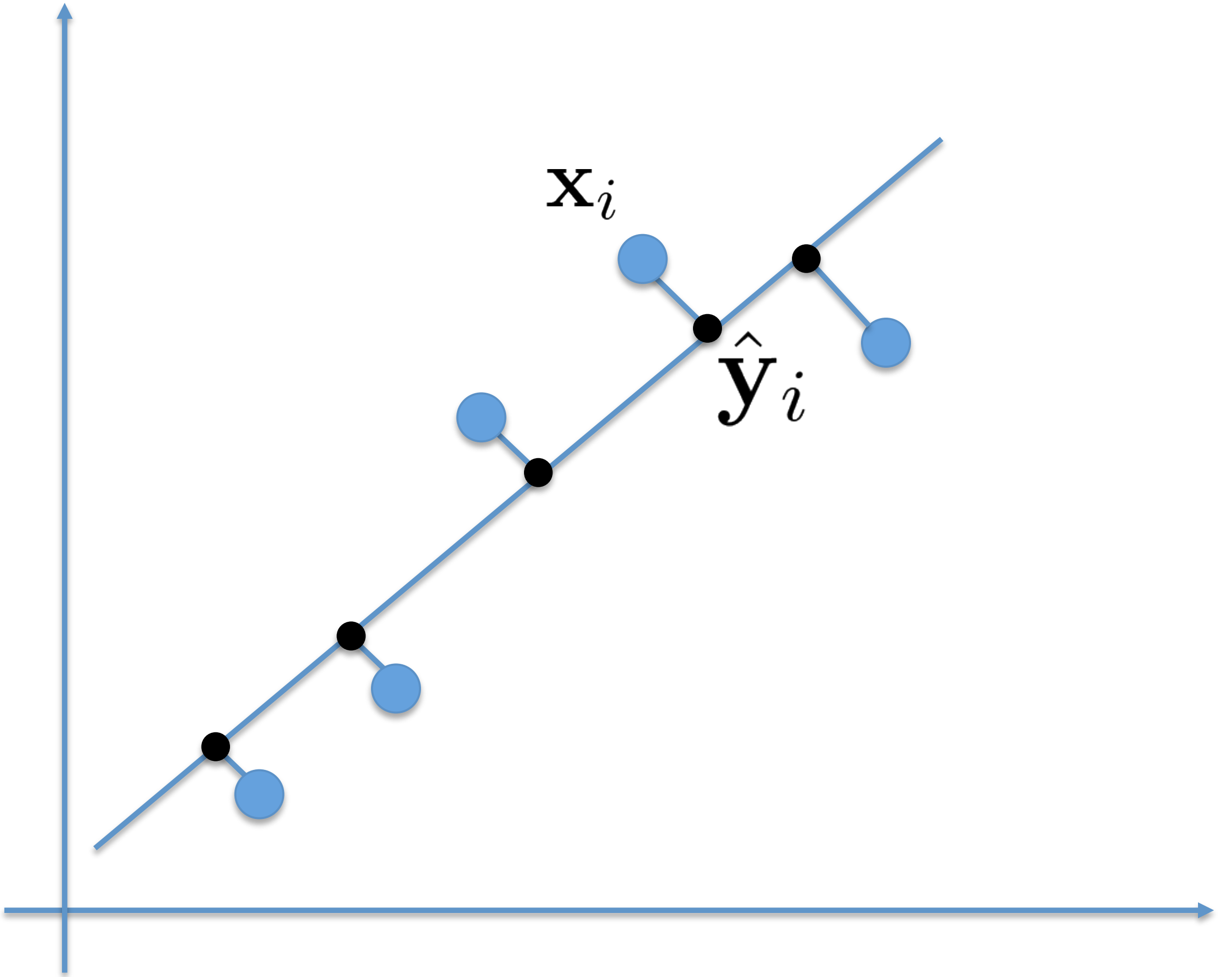
- If we use only the first *k* < *m* components we obtain the best reconstruction in terms of squared error.

$$e = \sum_i (\hat{\mathbf{y}}_i - \mathbf{y}_i)^2$$

Data point projected on the first *k* components.

Data point projected on all the components.

PCA as the Projection that Minimizes the Reconstruction Error



Example

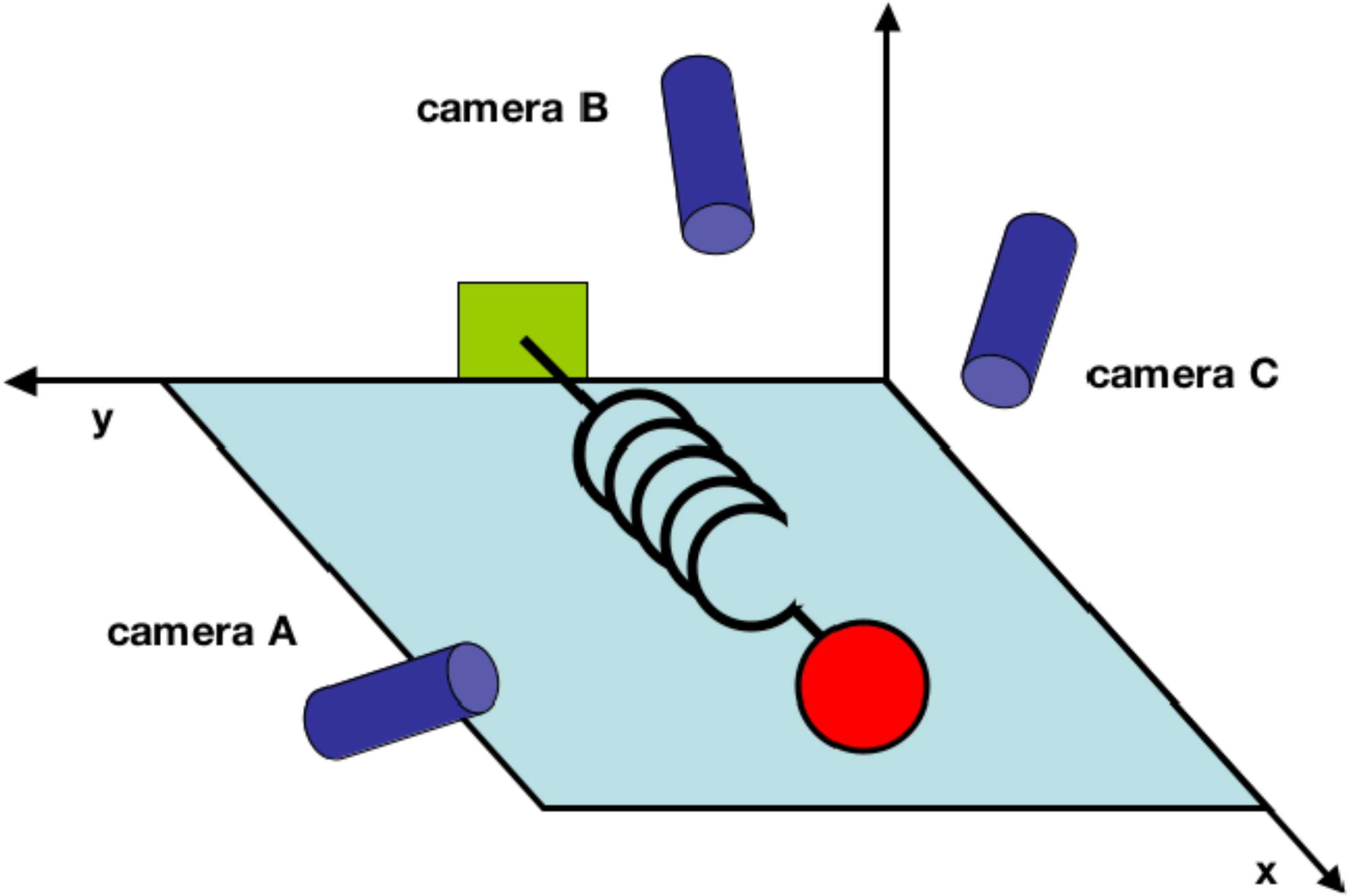


Figure From Jonathon Shlens, *"A Tutorial on Principal Component Analysis",* arXiv preprint arXiv:1404.1100, 2015.

PCA – Example

$$m = \begin{bmatrix} x_A \\ y_A \\ x_B \\ y_B \\ x_C \\ y_C \end{bmatrix}$$

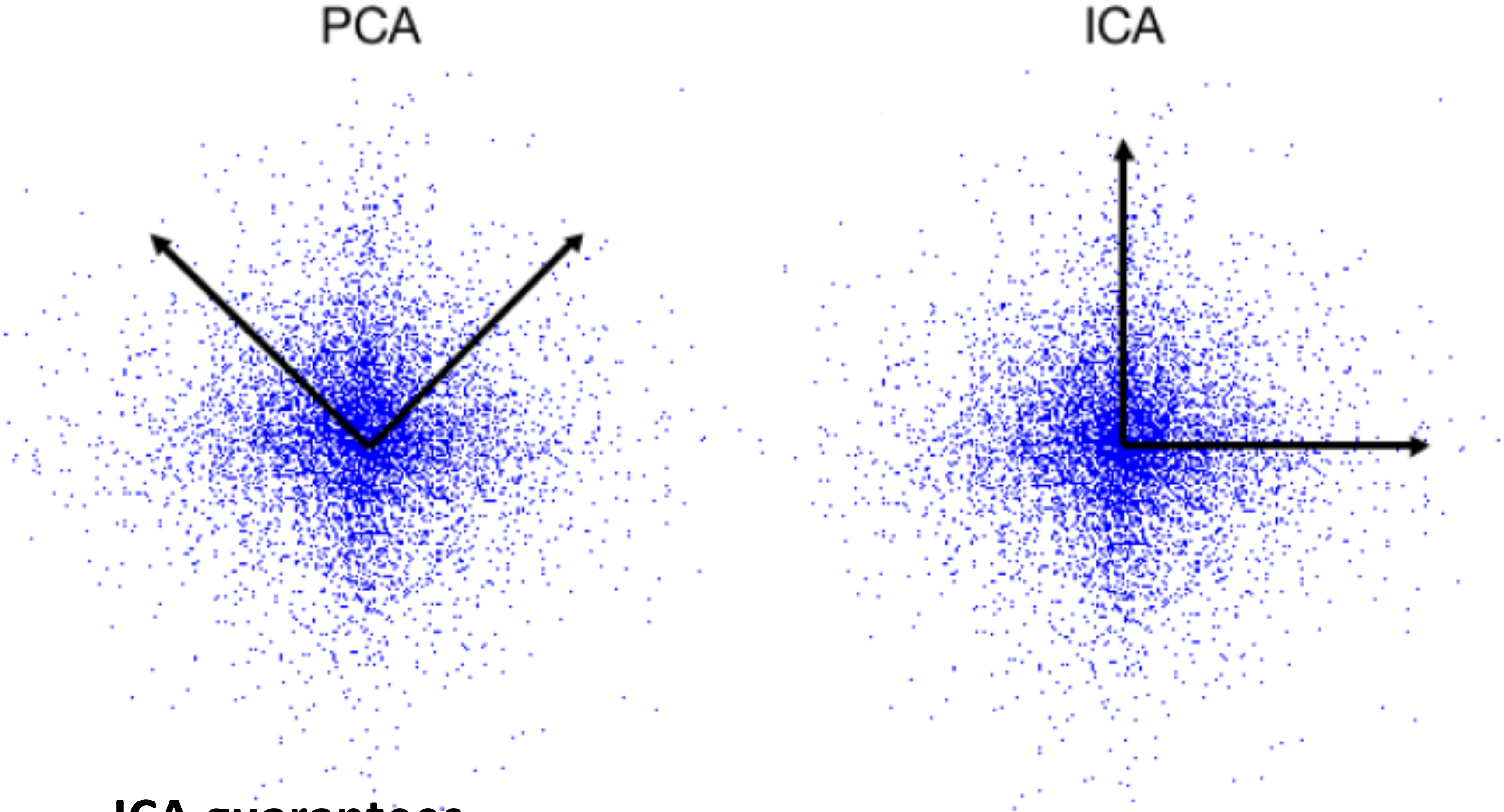
Each measure has 6 dimensions (!)

But the ball moves along the X-axis only..

Limits of PCA

- It is non-parametric → this is a strength point but it can be also a weak point.
- It fails for non-Gaussian distributed data.
- It can be extended to account for non-linear transformation → *kernel PCA*.

Limits of PCA



ICA guarantees statistical independence → $p(x, y) = p(x)p(y)$

Classic MDS

- Find the linear mapping $\mathbf{y}_i = \mathbf{M}\mathbf{x}_i$ which minimizes:

$$\phi(\mathbf{Y}) = \sum_{i,j} \overbrace{d_{ij}^2}^{\textbf{Euclidean distance in high dimensional space}} - \underbrace{\|\mathbf{y}_i - \mathbf{y}_j\|^2}_{\textbf{Euclidean distance in low dimensional space}}$$

PCA and MDS

- We want to minimize $\phi(\mathbf{Y})$, this corresponds to maximize:

$$\sum_{i,j} \|\mathbf{M}\mathbf{x}_i - \mathbf{M}\mathbf{x}_j\|^2$$

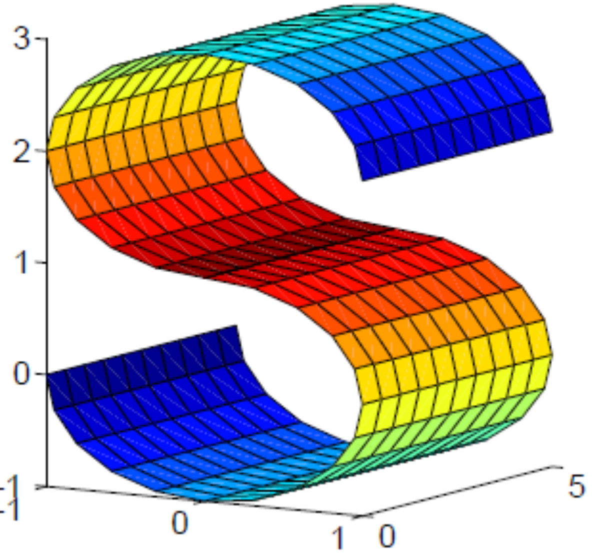
That is the variance of the low-dimensional points (same goal of the PCA).

PCA and MDS

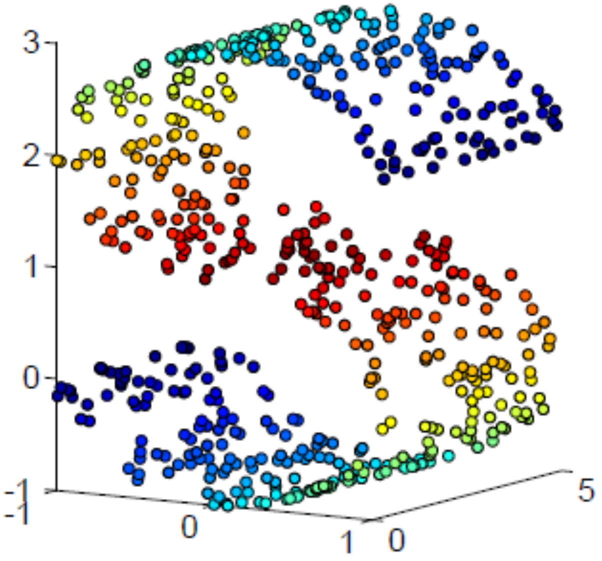
- The size of the covariance matrix is proportional to the dimension of the data.
- MDS scales with the number of data points instead of the dimensions of the data.
- Both PCA and MDS preserve better large pairwise distances.

Locally Linear Embedding (LLE)

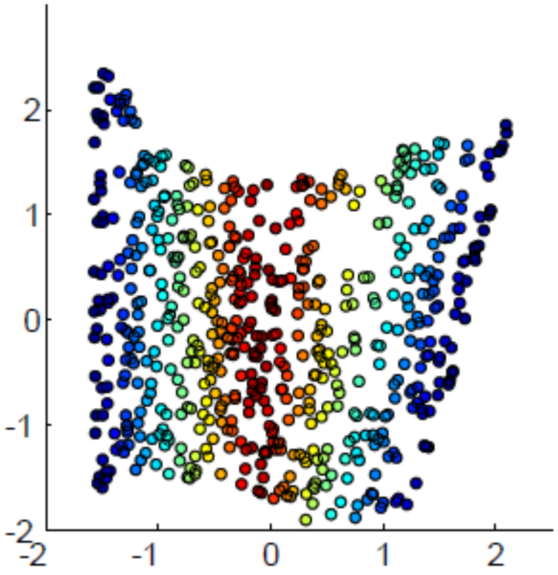
- LLE attempts to discover *nonlinear* structure in high dimension by exploiting local linear approximation.



Nonlinear Manifold *M*



Samples on *M*



Mapping Discovered

Locally Linear Embedding (LLE)

- *INTUITION* → assuming that there is sufficient data (well-sampled manifold) we expect each data point and its neighbors can be approximated by a local linear patch.
- The patch is represented by a weighted sum of the local data points.

Compute Local Patch

- Choose a set of data points close to a given one (ball-radius or K-nearest neighbours).
- Solve for W_{ij} :

$$\mathcal{E}(W) = \sum_i \left| \vec{X}_i - \sum\nolimits_j W_{ij} \vec{X}_j \right|^2$$

LLE Mapping

- Find $\vec{Y}_i$ which minimizes the embedding cost function:

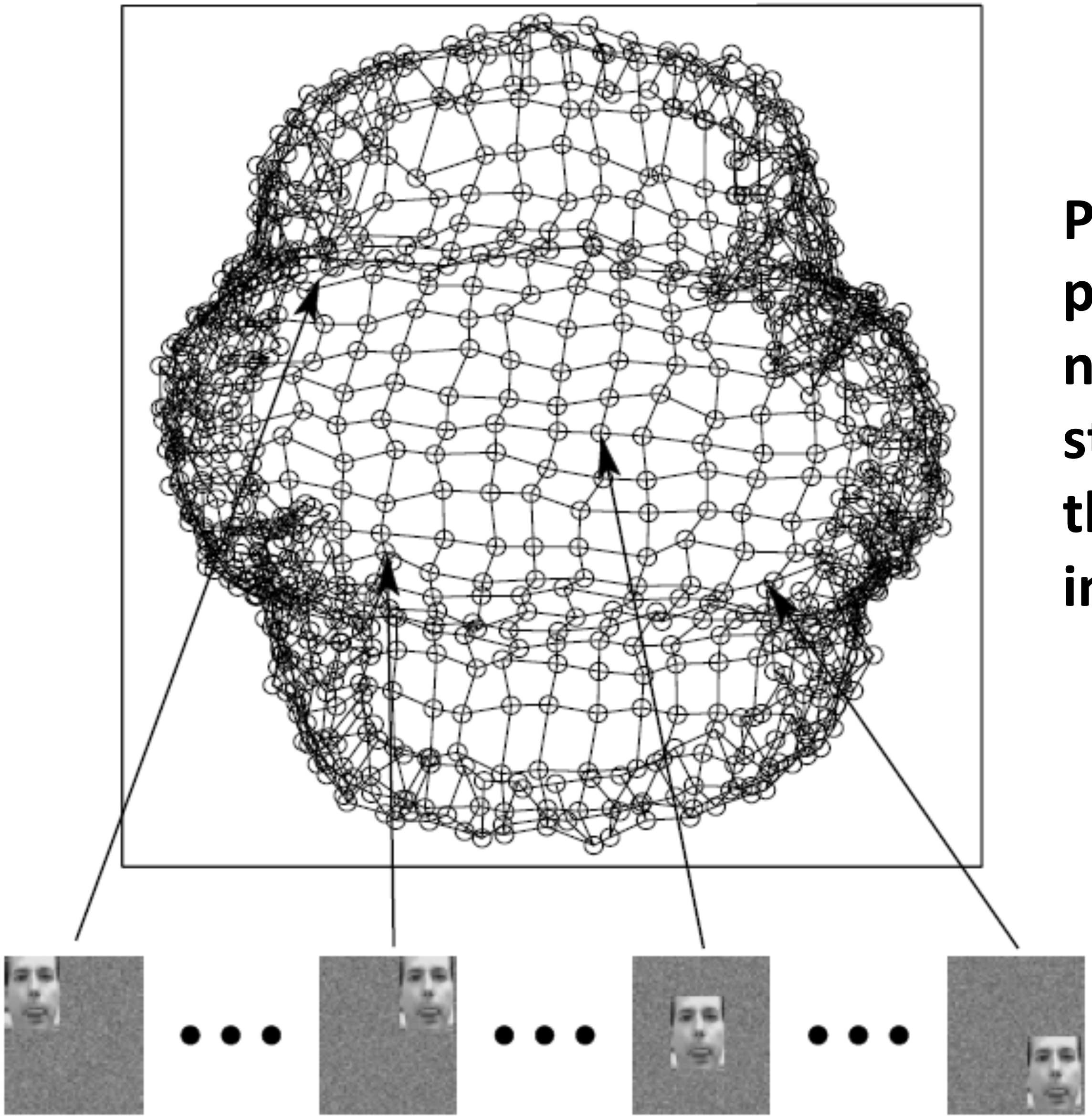
$$\Phi(Y) = \sum_i \left| \vec{Y}_i - \sum_j W_{ij} \vec{Y}_j \right|^2$$

Note that weights are fixed in this case!

LLE Algorithm

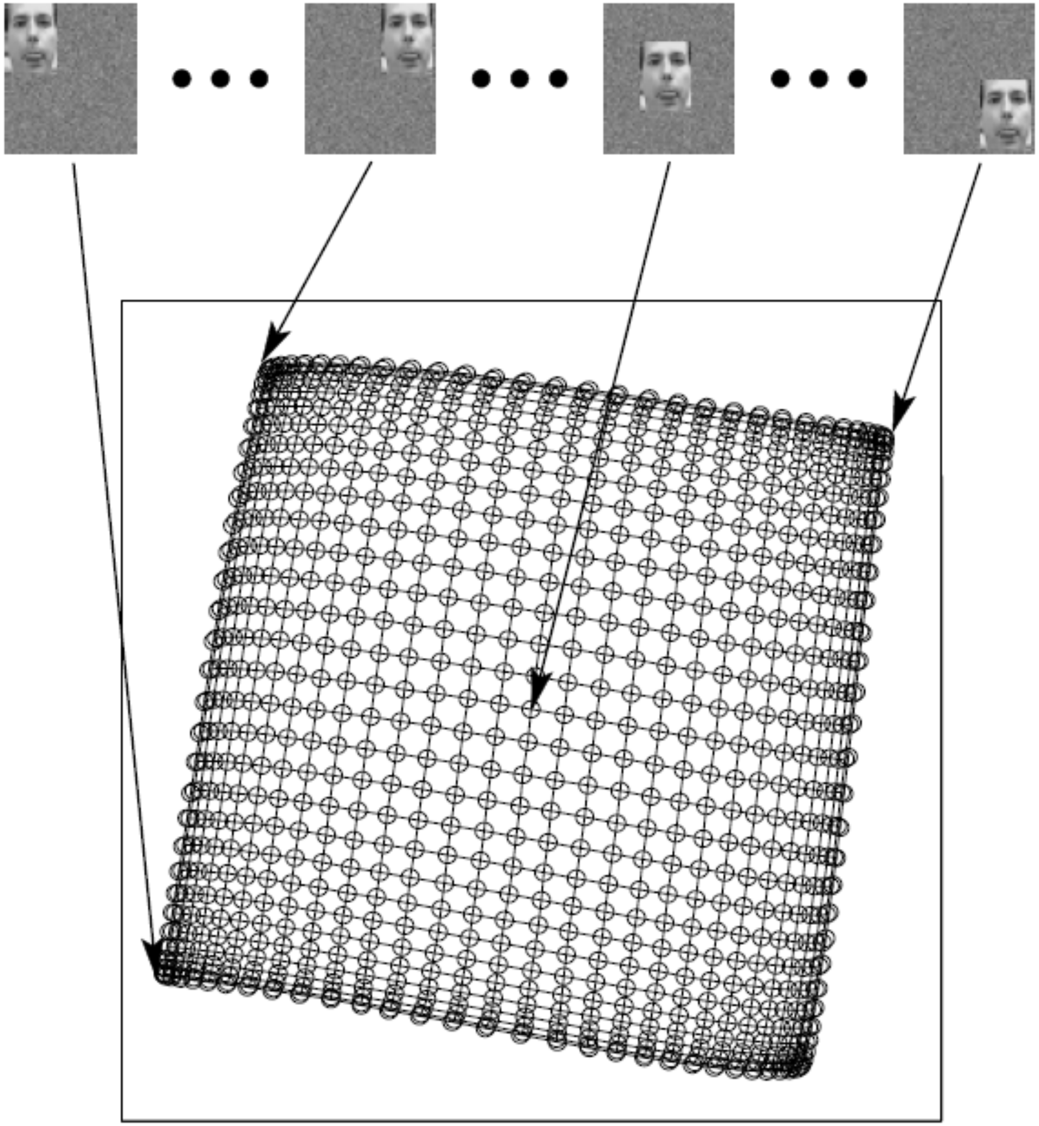
1. Compute the neighbors of each data point, $\vec{X}_i$.
2. Compute the weights W_{ij} that best reconstruct $\vec{X}_i$.
3. Compute the vectors $\vec{Y}_i$ that minimizes the cost function.

LLE – Example



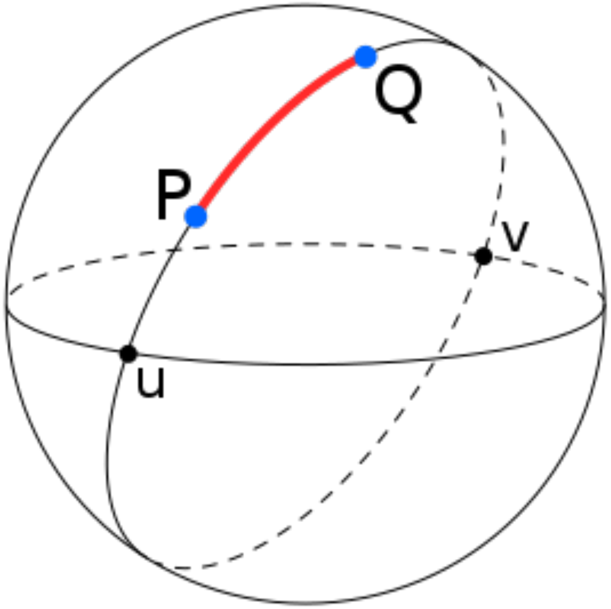
PCA fails to preserve the neighborhood structure of the nearby images.

LLE – Example

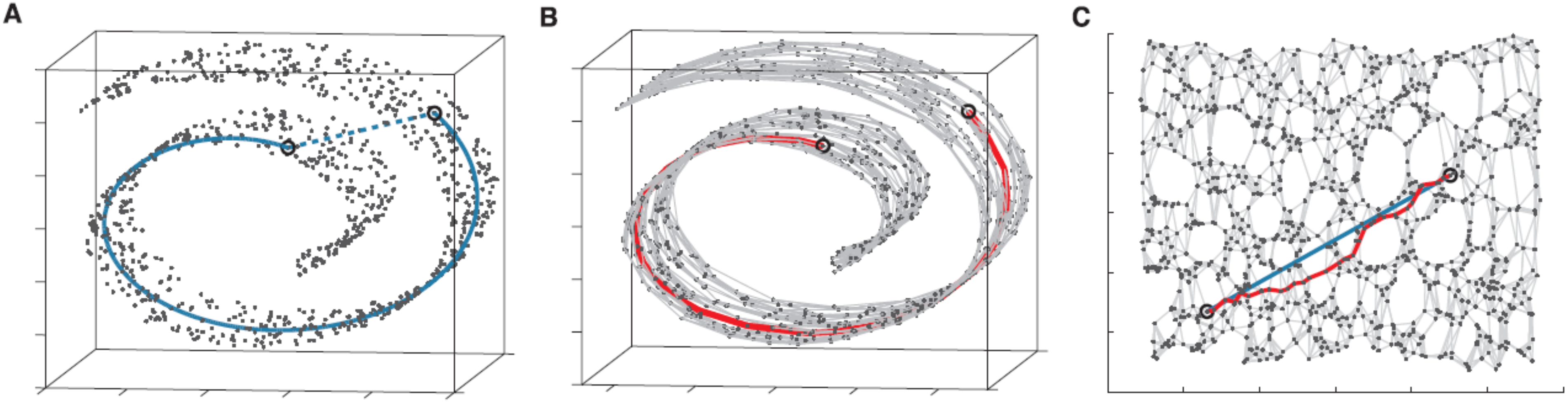


ISOMAP

- The core idea is to preserve the geodesic distance between data points.
- Geodesic is the shortest path between two points on a curved space.



ISOMAP



Euclidean distance
vs
Geodesic distance

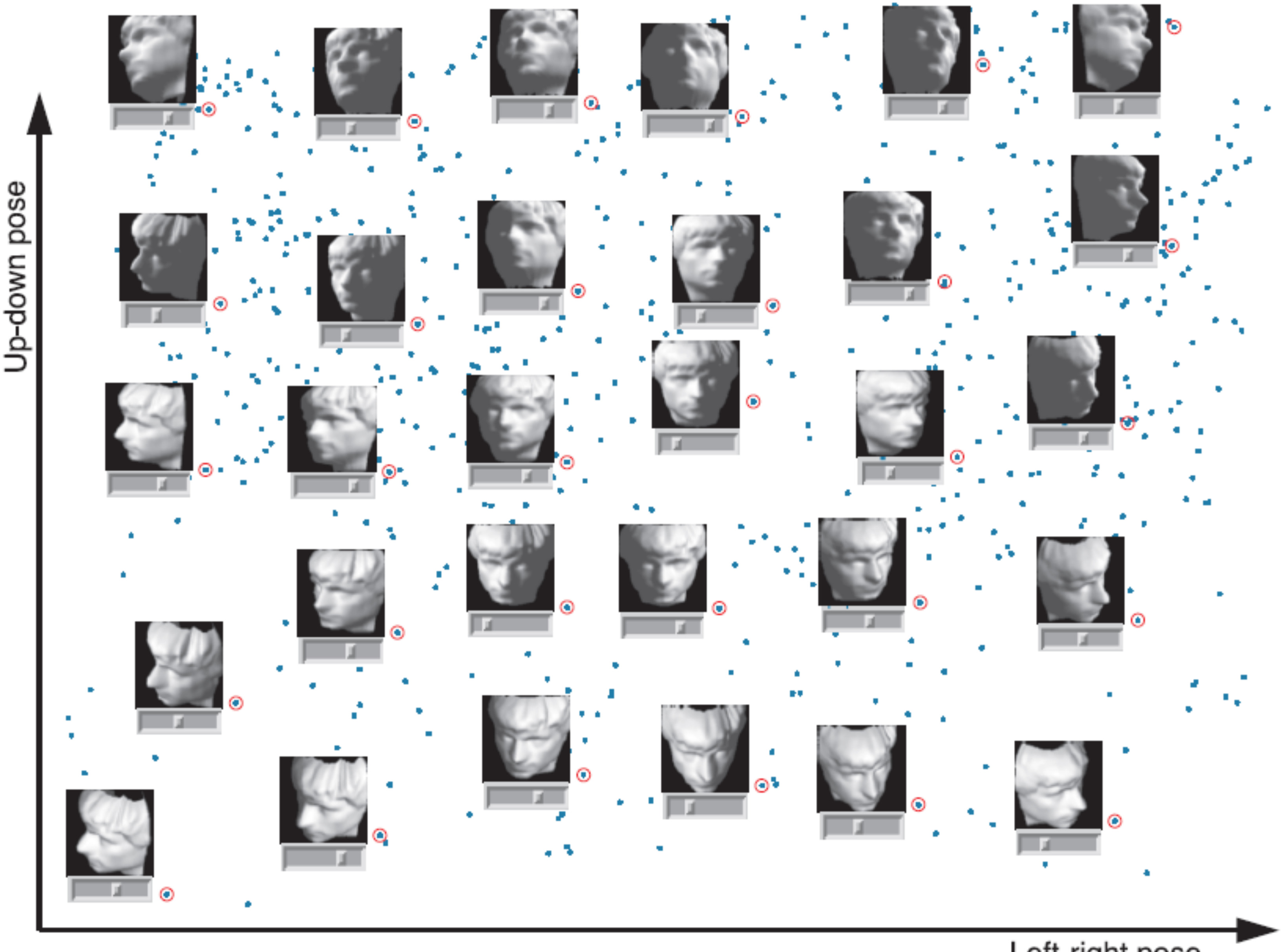
Graph build
and
Geodesic distance
Approximation

Geodesic distance
vs
Approximated Geodesic

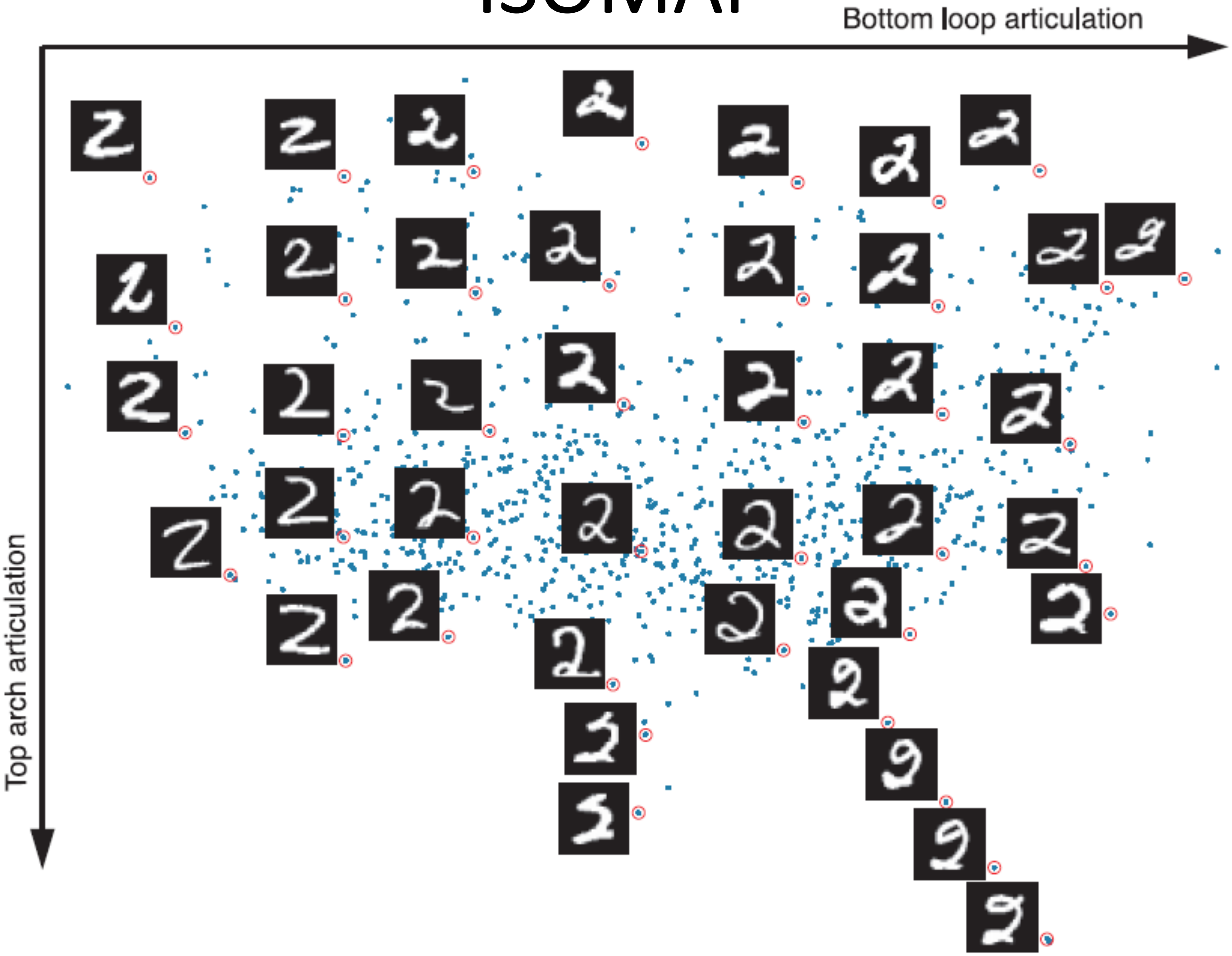
ISOMAP

- Construct neighborhood graph
  - Define graph *G* over all data points by connecting points *(i,j)* if and only if the point *i* is a K neareast neighbor of point *j*
- Compute the shortest path
  - Using the Floyd's algorithm
- Construct the *d*-dimensional embedding

ISOMAP



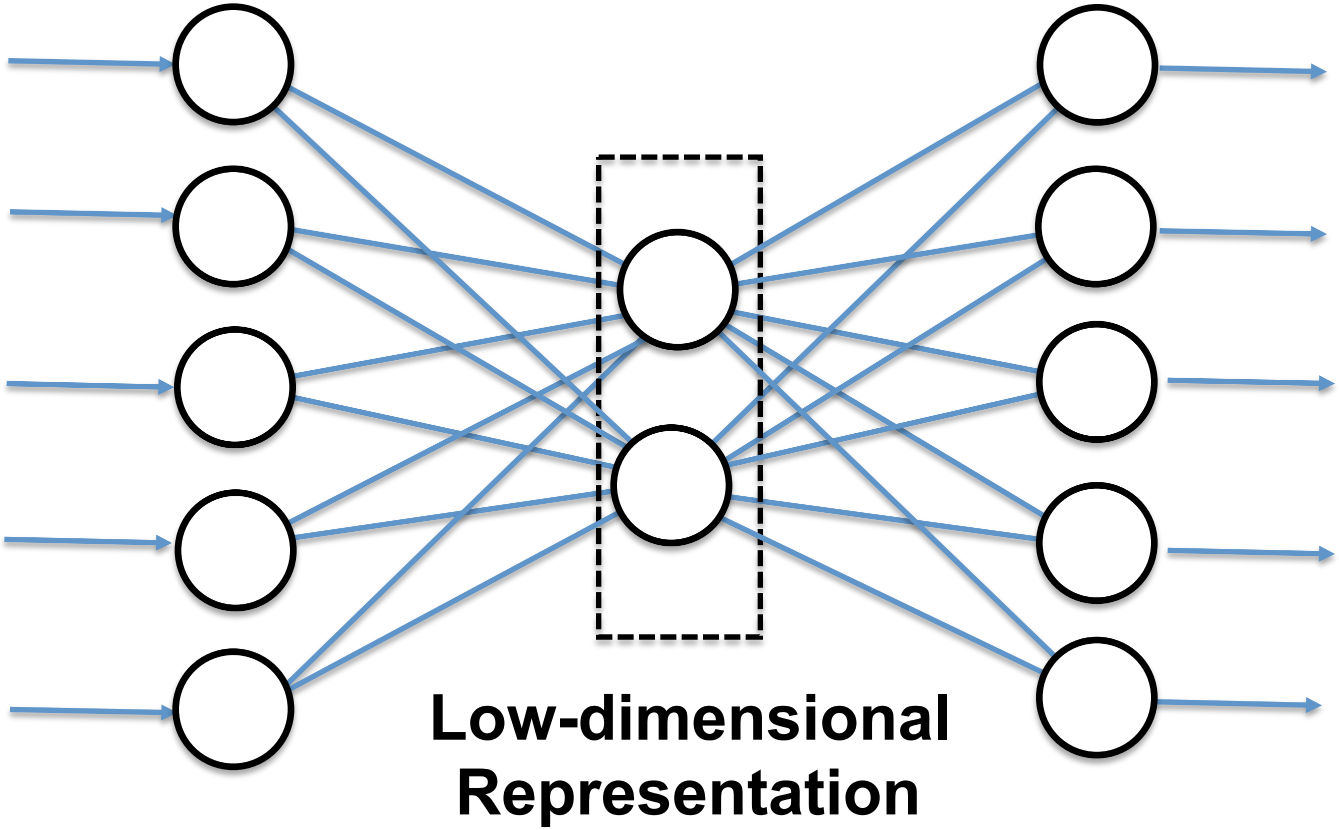
ISOMAP



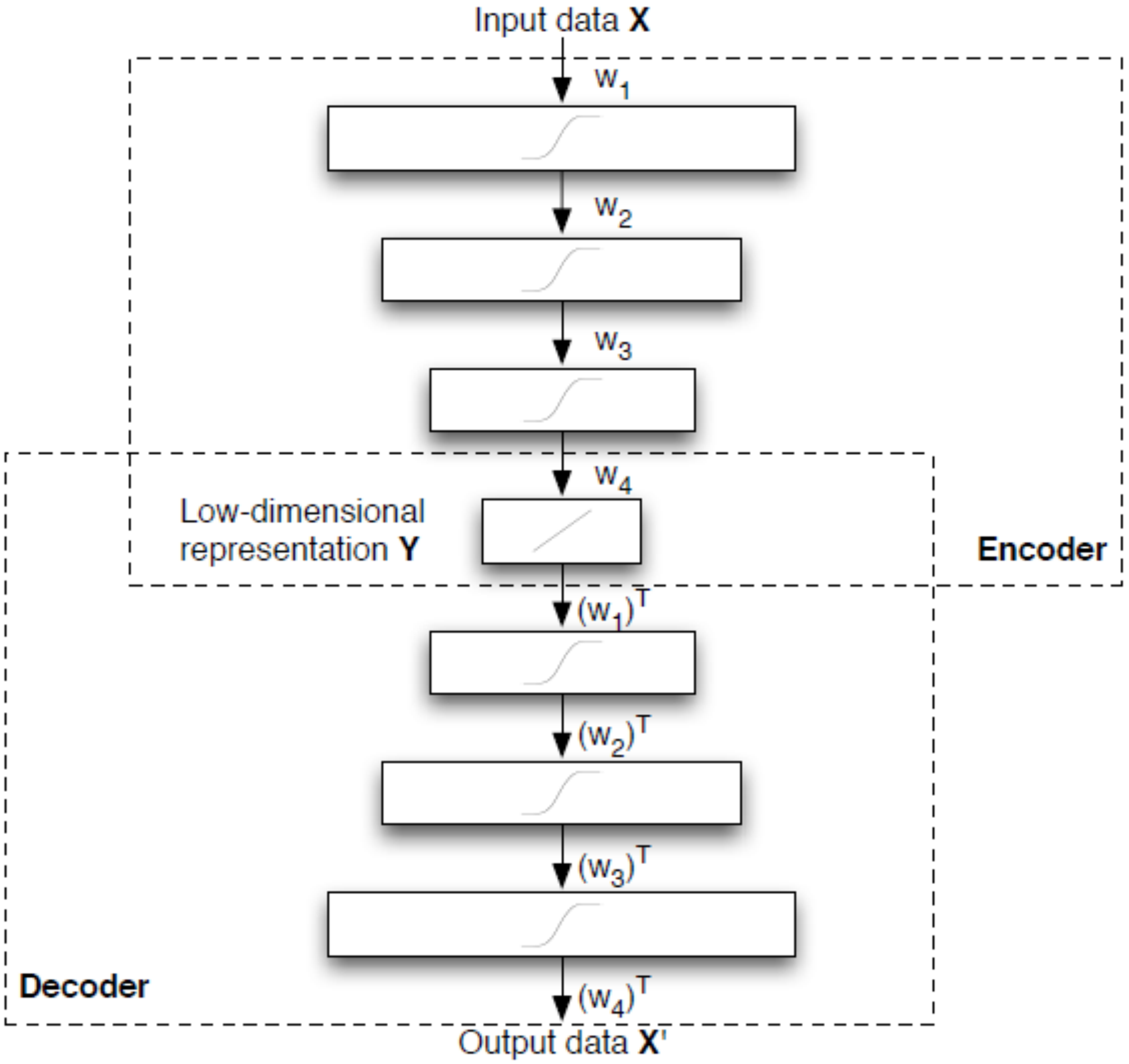
Autoencoders

- Machine learning is becoming ubiquitous in Computer Science.
- A special type of neural network is called *autoencoder*.
- An autoencoder can be used to perform dimensionality reduction.
- First, let me say something about neural network..

Autoencoder



Multi-layer Autoencoder



Summon Mapping

- Adaptation of MDS by weighting the contribution of each *(i,j)* pair:

$$\phi(\mathbf{Y}) = \frac{1}{\sum_{i,j} d_{ij}} \sum_{i \neq j} \frac{\left(d_{ij} - \|\mathbf{y}_i - \mathbf{y}_j\|\right)^2}{d_{ij}}$$

- This allows to retain the local structure of the data better than classical scaling (the retain of high distances is not privileged).

t-SNE

- Most techniques for dimensionality reduction are not able to retain both the local and the global structure of the data in a single map.
- Simple tests on handwritten digits demonstrate this (Song et al. 2007).

L. Song, A. J. Smola, K. Borgwardt and A. Gretton, *"Colored Maximum Variance Unfolding",* in Advances in Neural Information Processing Systems. Vol. 21, 2007.

Stochastic Neighbor Embedding (SNE)

- Similarities between high- and low-dimensional data points is modeled with conditional probabilities.
- Conditional probability that the point x_i would peak x_j as its neighbor:

$$p_{j|i} = \frac{\exp\left(-\|x_i - x_j\|^2 / 2\sigma_i^2\right)}{\sum_{k \neq i} \exp\left(-\|x_i - x_k\|^2 / 2\sigma_i^2\right)}$$

Stochastic Neighbor Embedding (SNE)

- We are interested only in pairwise distance

$$p_{i|i} = 0$$

- For the low-dimensional points an analogous conditional probability is used:

$$q_{j|i} = \frac{\exp\left(-\|y_i - y_j\|^2\right)}{\sum_{k \neq i} \exp\left(-\|y_i - y_k\|^2\right)}$$

Kullback-Leibler Divergence

- Coding theory: expected number of extra bits required to code samples from the distribution *P* if the current code is optimize for the distribution *Q*.
- Bayesian view: a measure of the information gained when one revises one's beliefs from the prior distribution *Q* to the posterior distribution *P.*
- It is also called *relative entropy*.

Kullback-Leibler Divergence

- Definition for discrete distributions:

$$D_{KL}(P||Q) = \sum_i P_i \log \frac{P_i}{Q_i}$$

- Definition for continuos distributions:

$$D_{KL}(P||Q) = \int_{-\infty}^{+\infty} p(x) \log \frac{p(x)}{q(x)} dx$$

Stochastic Neighbor Embedding (SNE)

- The goal is to minimizes the mismatch between $p_{j|i}$ and $q_{j|i}$.
- Using the Kullback-Leibler divergence this goal can be achieved by minimizing the function:

$$C = \sum_i KL(P_i || Q_i) = \sum_i \sum_j p_{j|i} \log \frac{p_{j|i}}{q_{j|i}}$$

Note that KL(P||Q) is not symmetric !

Problems of SNE

- The cost function is difficult to optimize.
- SNE suffers, as other dimensionality reduction techniques, of the *crowding problem*.

t-SNE

- SNE is made symmetric:

$$C = KL(P||Q) = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}}$$

- It employs a Student-t distribution instead of a Gaussian distribution to evaluate the similarity between points in low dimension.

$$q_{ij} = \frac{\left(1 + \|y_i - y_j\|^2\right)^{-1}}{\sum_{k \neq l} \left(1 + \|y_k - y_l\|^2\right)^{-1}}$$

t-SNE Advantages

- The crowding problem is alleviated.
- Optimization is made simpler.

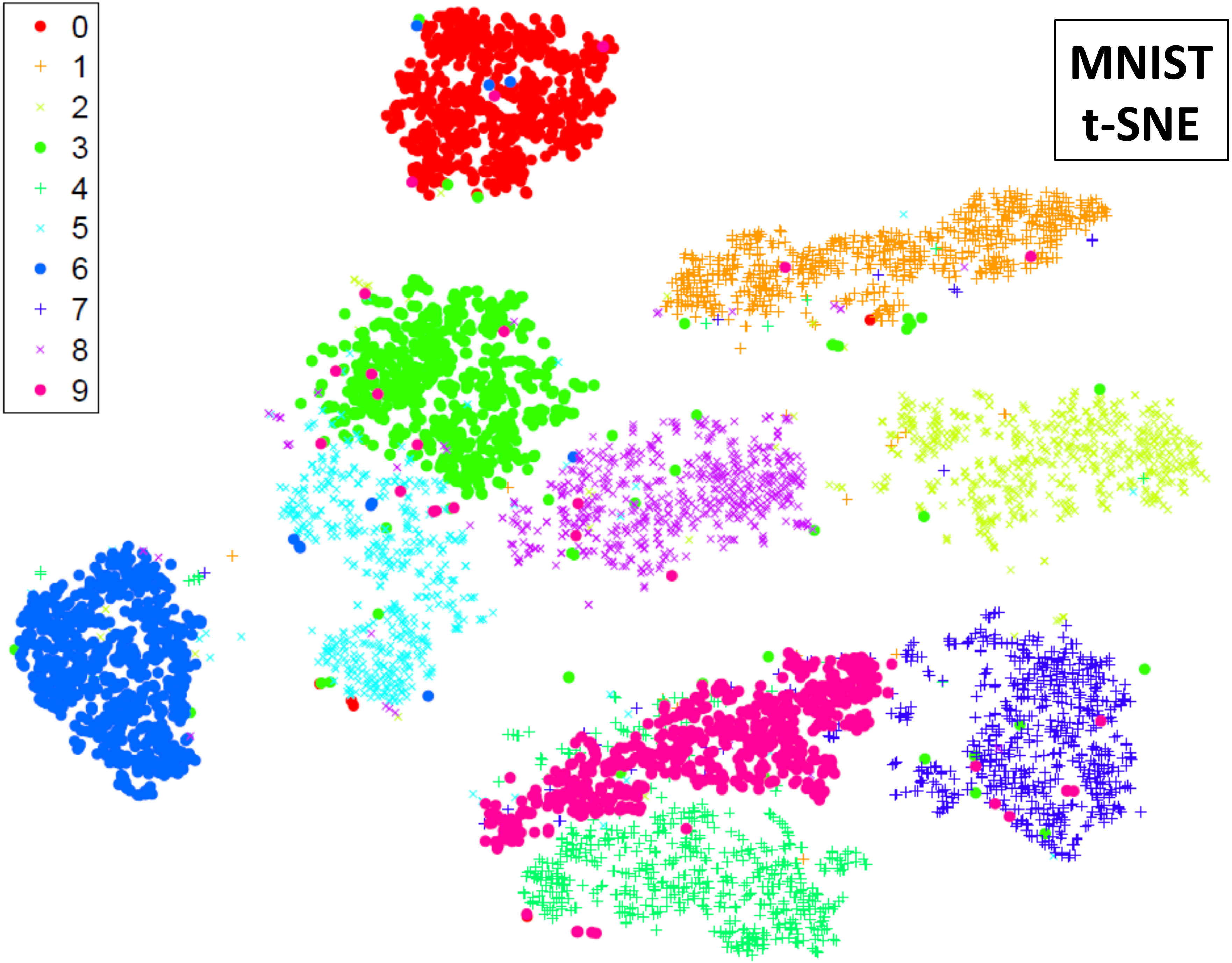
Experiments

- Comparison with LLE, Isomap and Summon Mapping.
- Datasets:
  - MNIST dataset
  - Olivetti face dataset
  - COIL-20 dataset

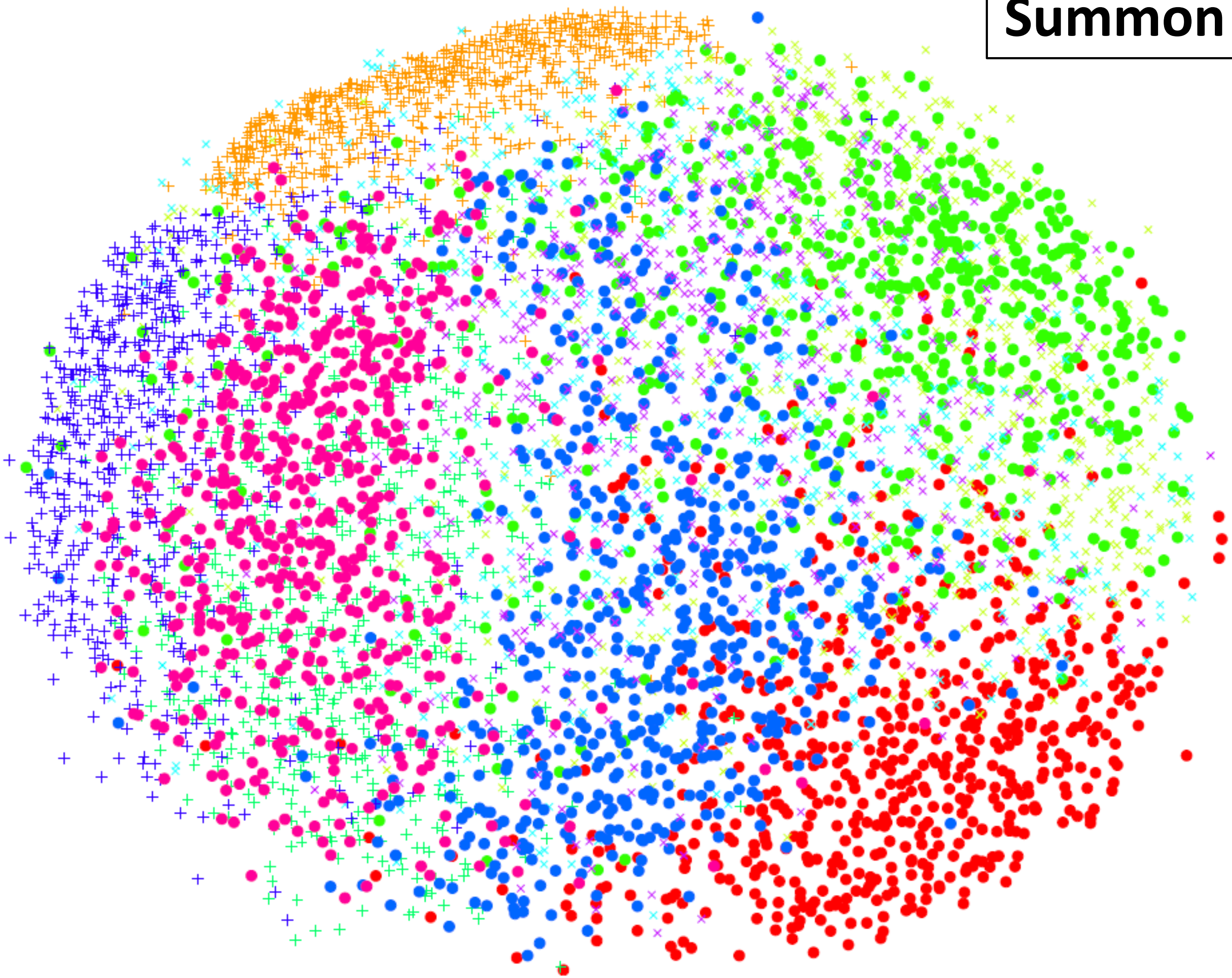
Comparison figures are from the paper L.J.P. van der Maaten and G.E. Hinton, *"Visualizing High-Dimensional Data Using t-SNE"*, Journal of Machine Learning Research, *Vol.* 9, pp. 2579-2605, 2008.

MNIST Dataset

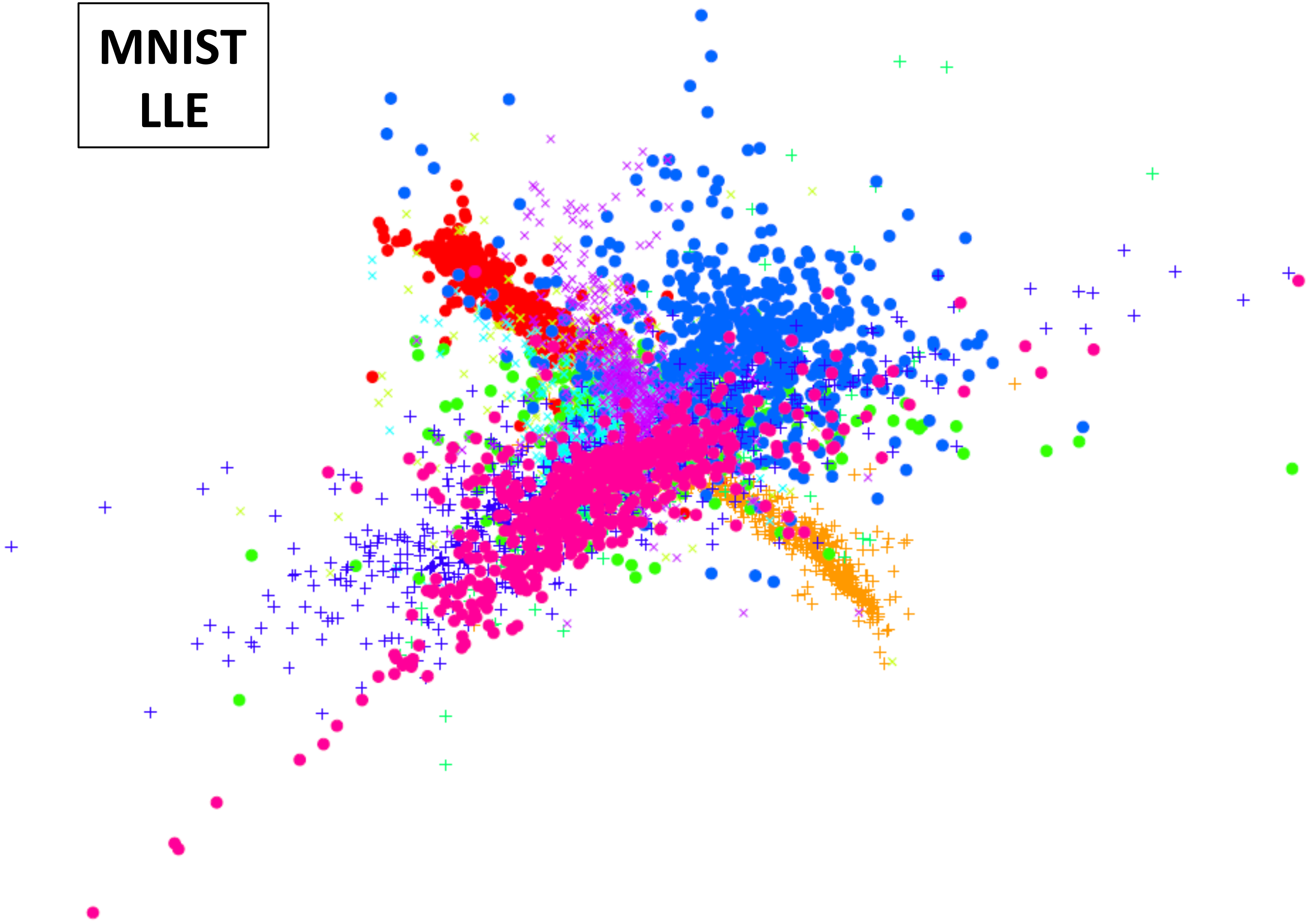
- 60,000 images of handwritten digits.
- Image resolution: 28 x 28 (784 dimensions).
- A subset of 6,000 images randomly selected has been used.



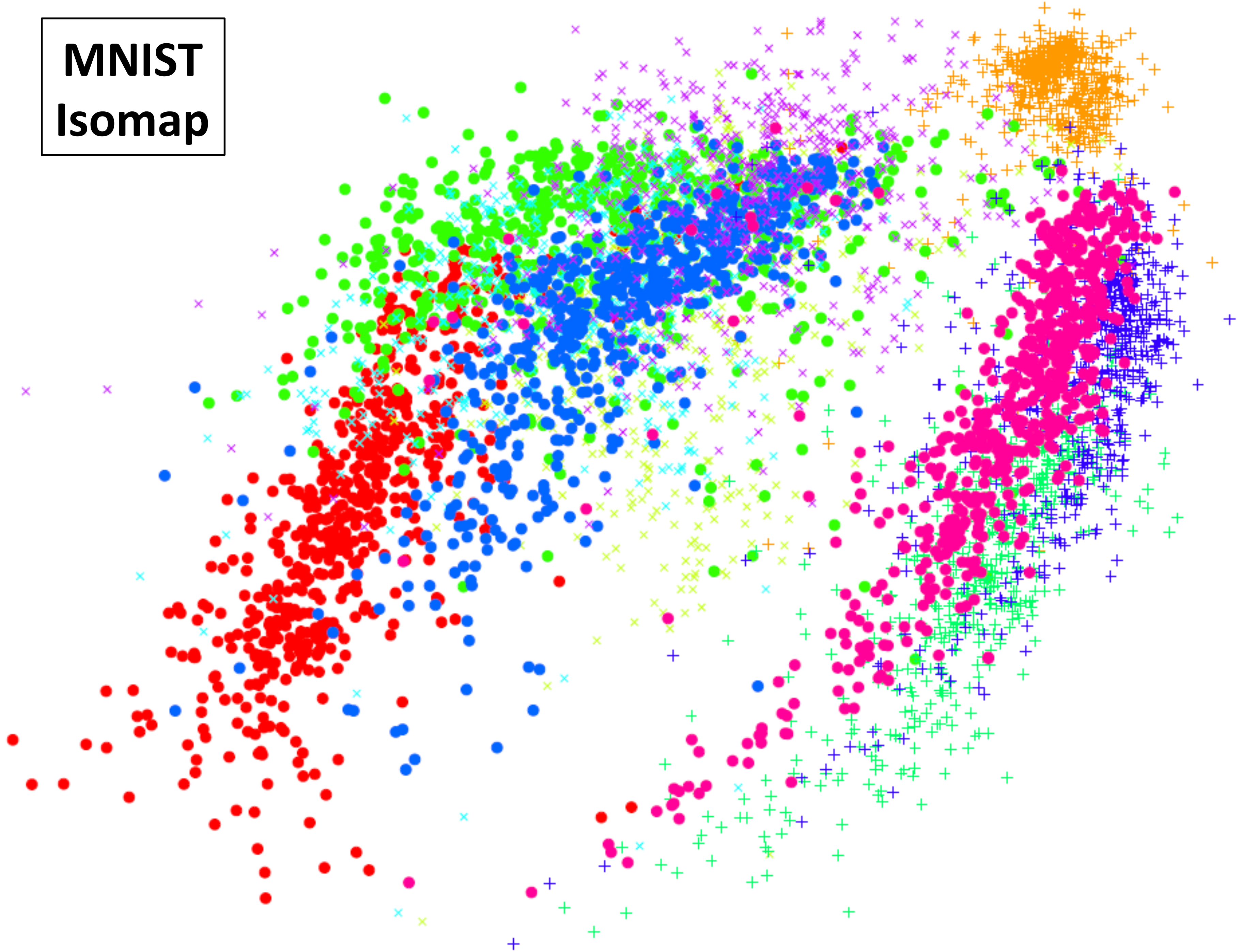
MNIST t-SNE



MNIST
Summon Mapping



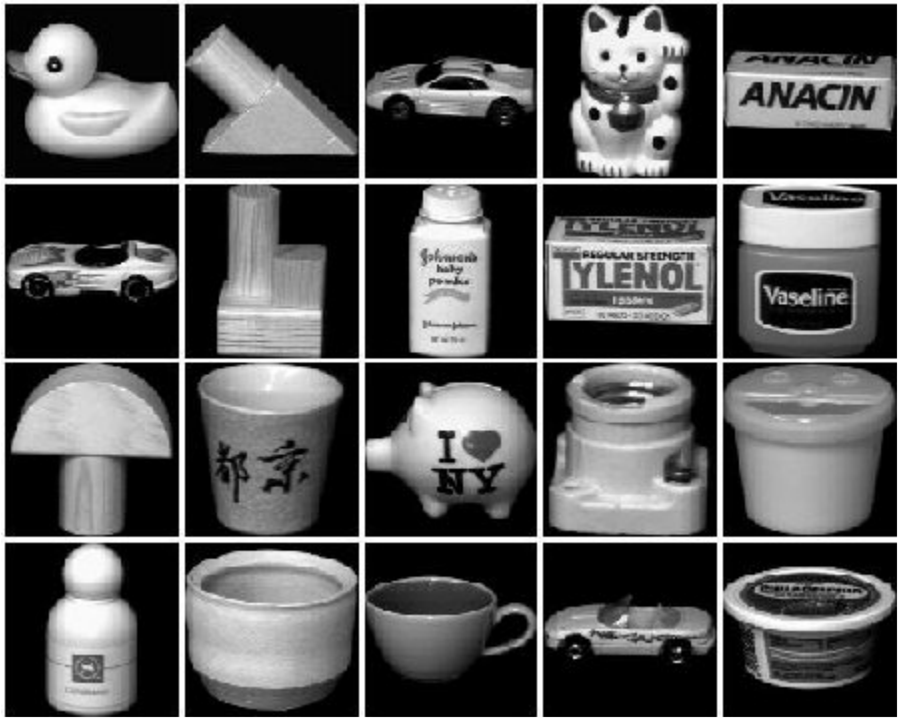
MNIST
LLE



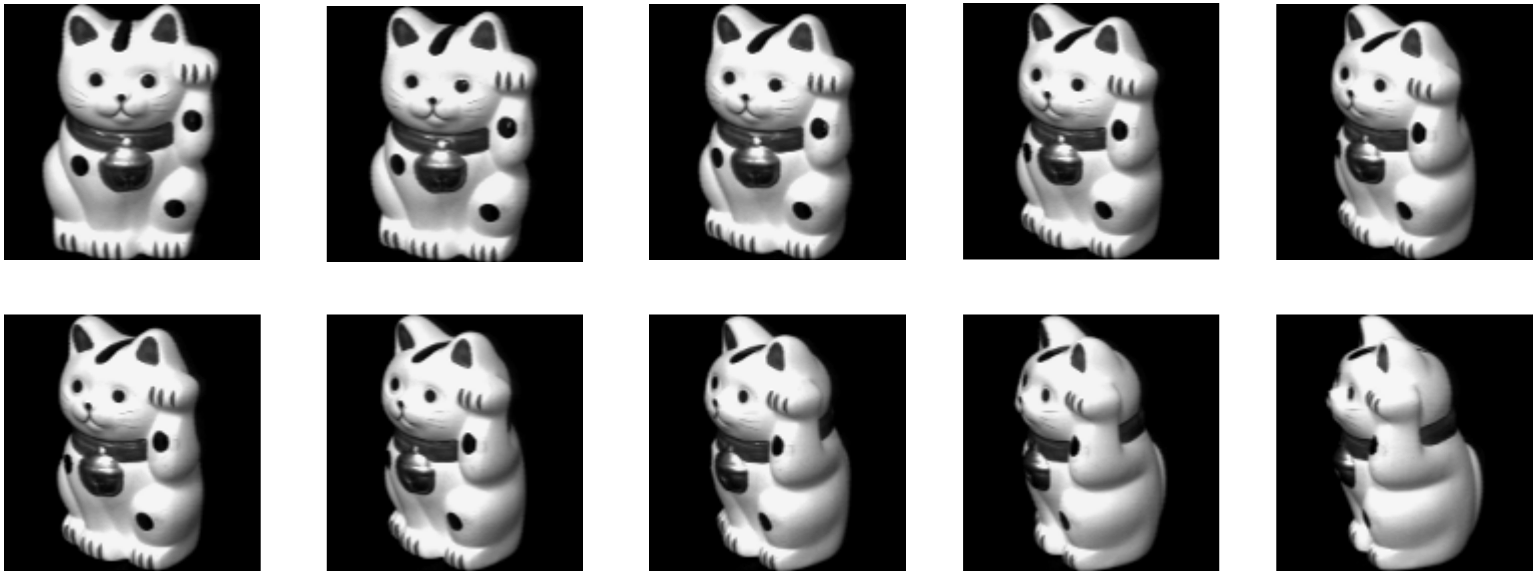
MNIST
Isomap

COIL-20 Dataset

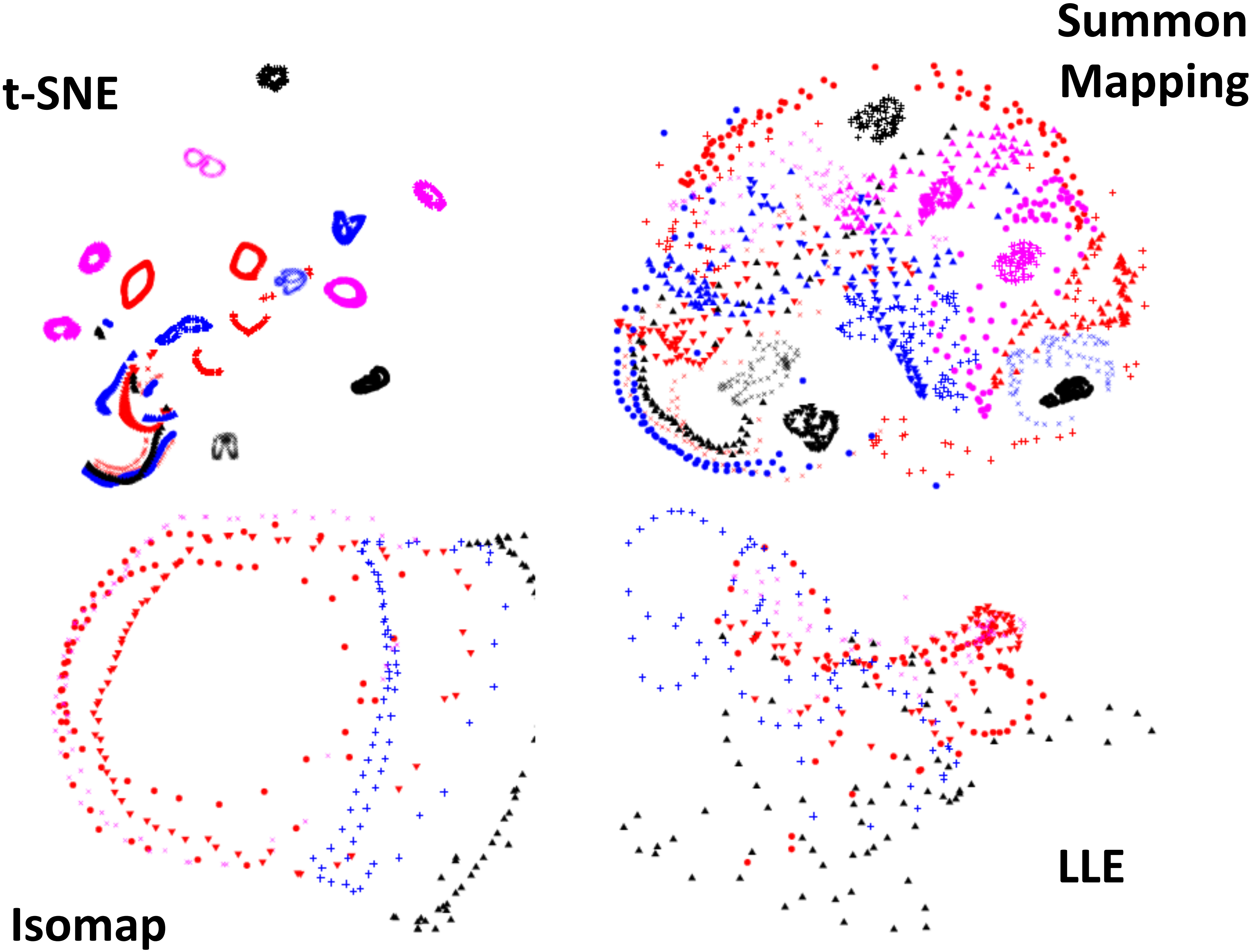
- Images of 20 objects viewed from 72 different viewpoints (1440 images).
- Image size: 32 x 32 (1024 dimensions).



COIL-20 Dataset



…


t-SNE
Summon Mapping
Isomap
LLE

Objects Arrangement

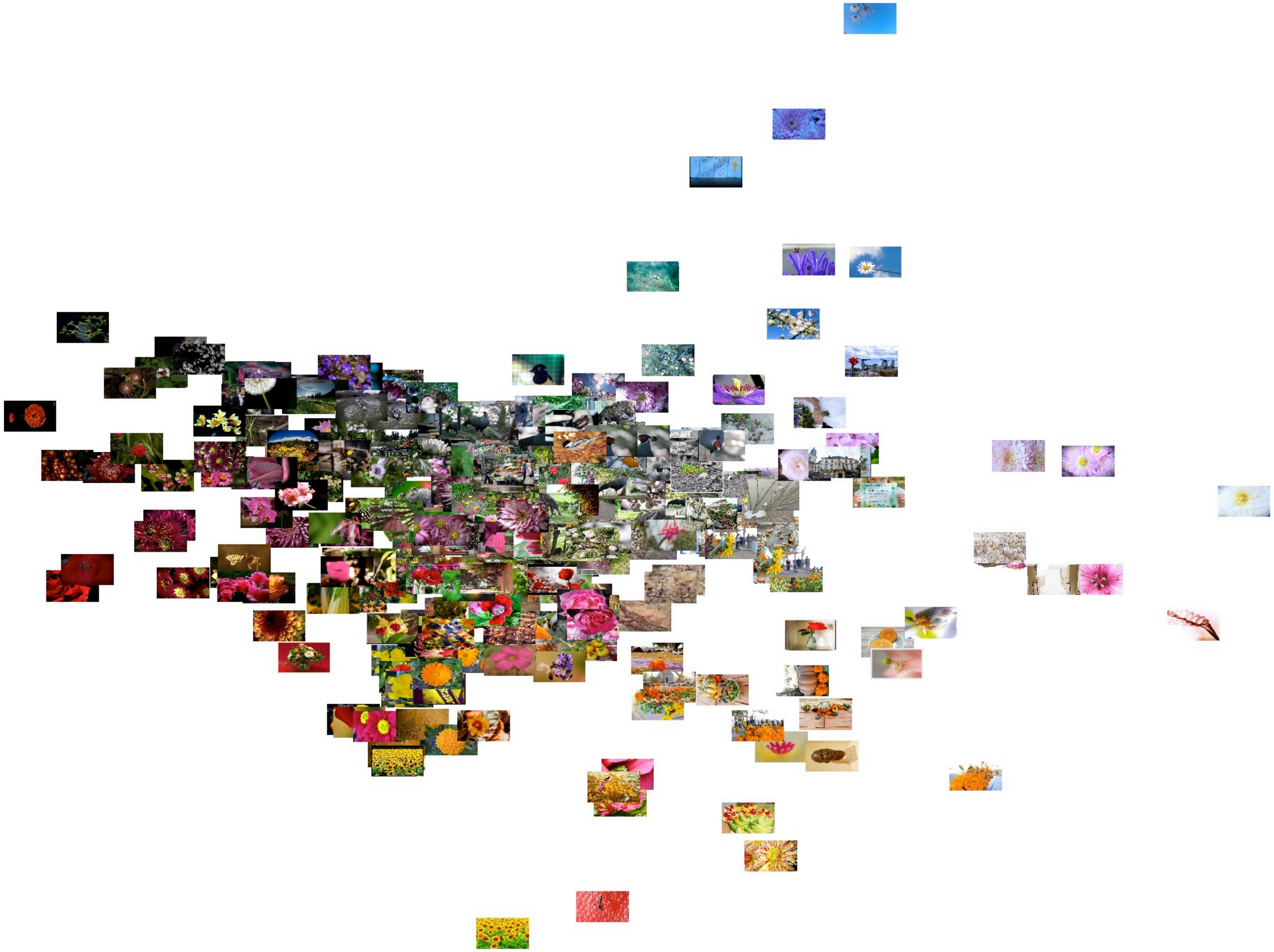
Motivations

- Multidimensional reduction can be used to arrange objects in 2D or 3D preserving pairwise distances (but the final placement is arbitrary).
- Many applications require to place the objects in a set of pre-defined, discrete, positions (e.g. on a grid).

Example – Images of Flowers

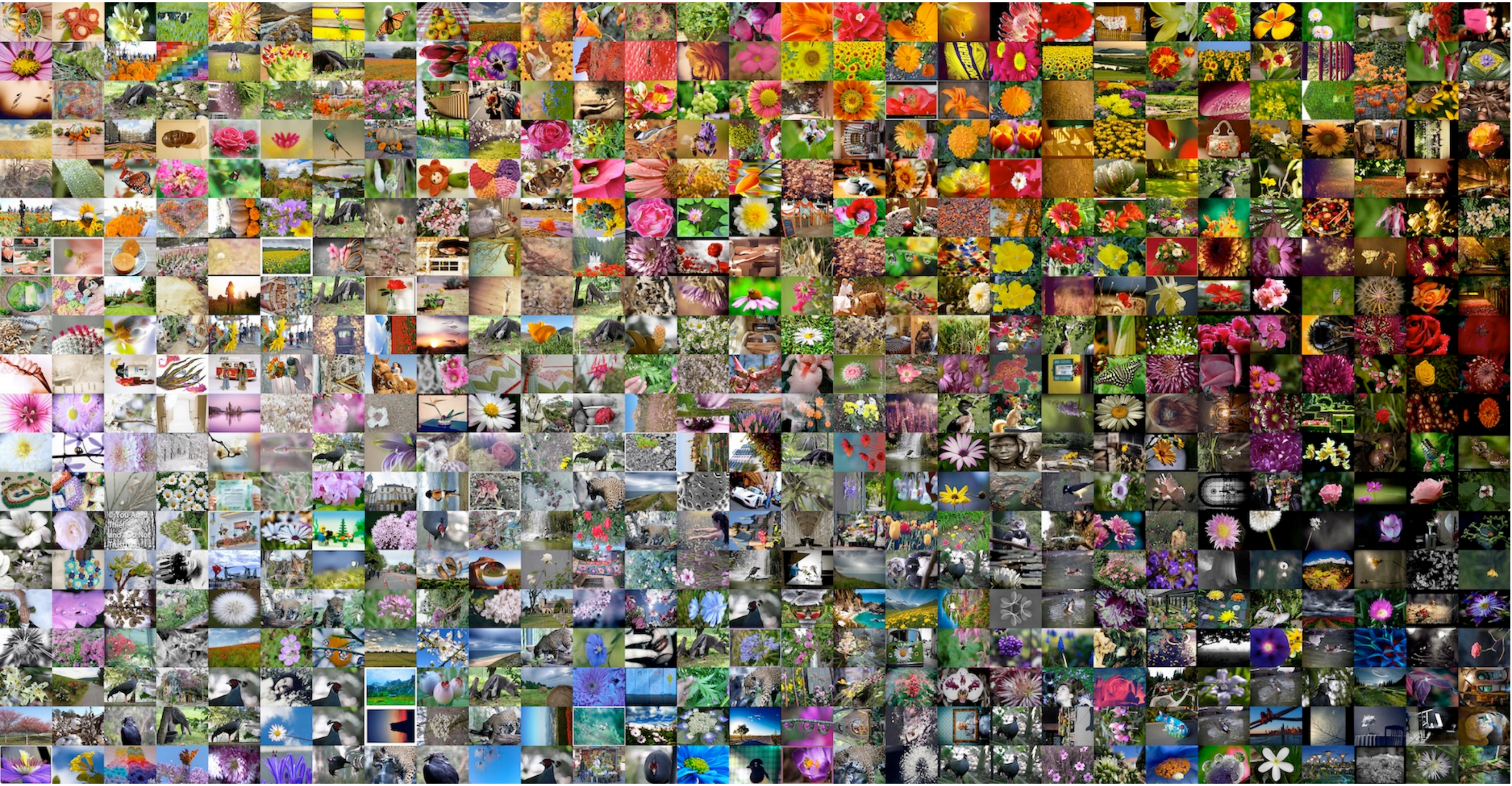
Random Order

Example – Images of Flowers



Isomap

Example – Images of Flowers



IsoMatch (computed on colors)

Problem Statement

The goal is to find the permutation *π* that minimizes the following energy:

$$E_p(\pi) = \min_{c} \left(\sum_{i,j} cd(i,j) - d(\pi(i),\pi(j)) \right)^{\frac{1}{p}}$$

Permutation

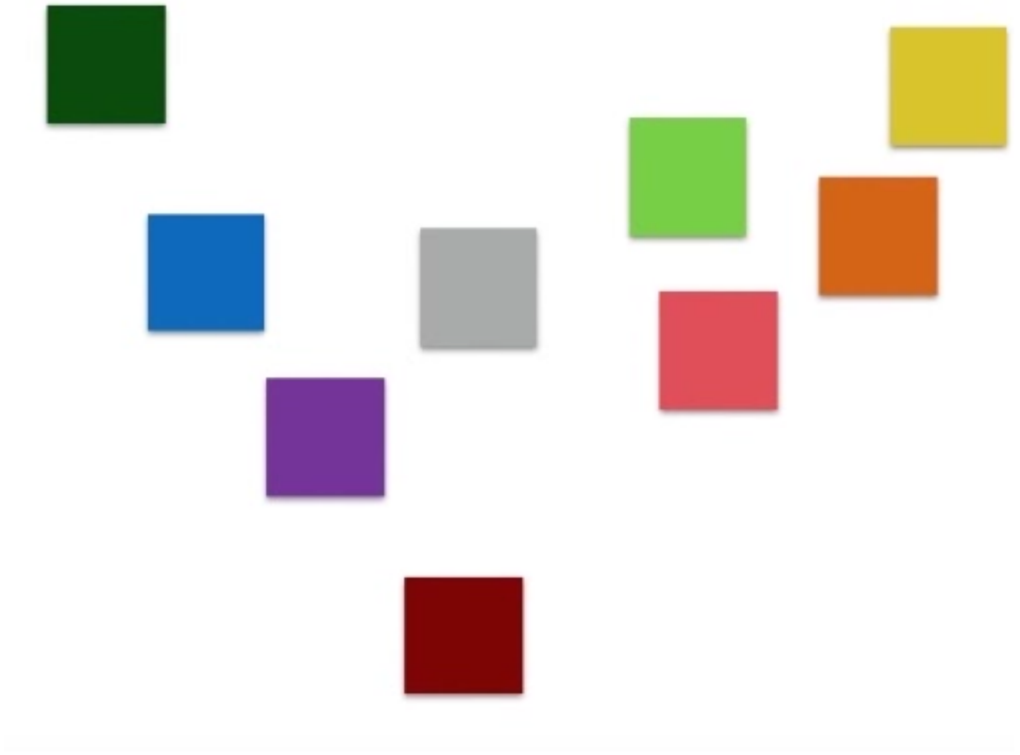
Original pairwise distance

Euclidean distance in the grid

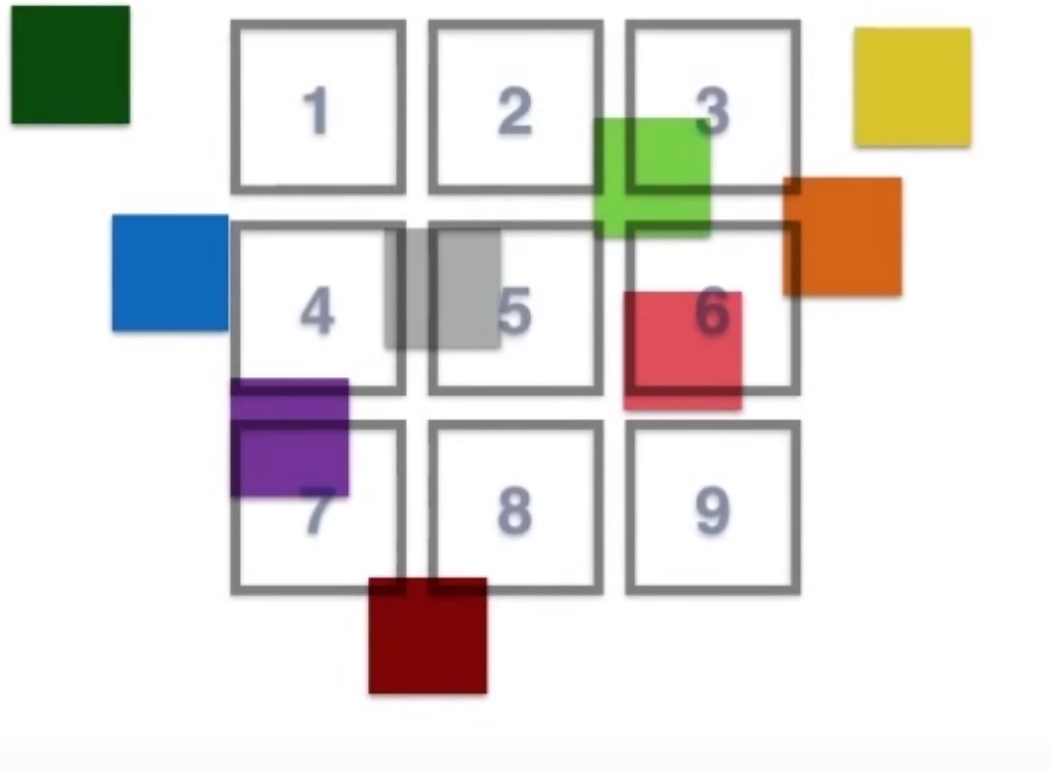
IsoMatch – Algorithm

- Step I : Dimensionality Reduction (using Isomap)
- Step II : Coarse Alignment (bounding box)
- Step III : Bipartite Matching
- Step IV (optional) : Random Refinement (elements swap)

Algorithm – Step I Dimensionality Reduction



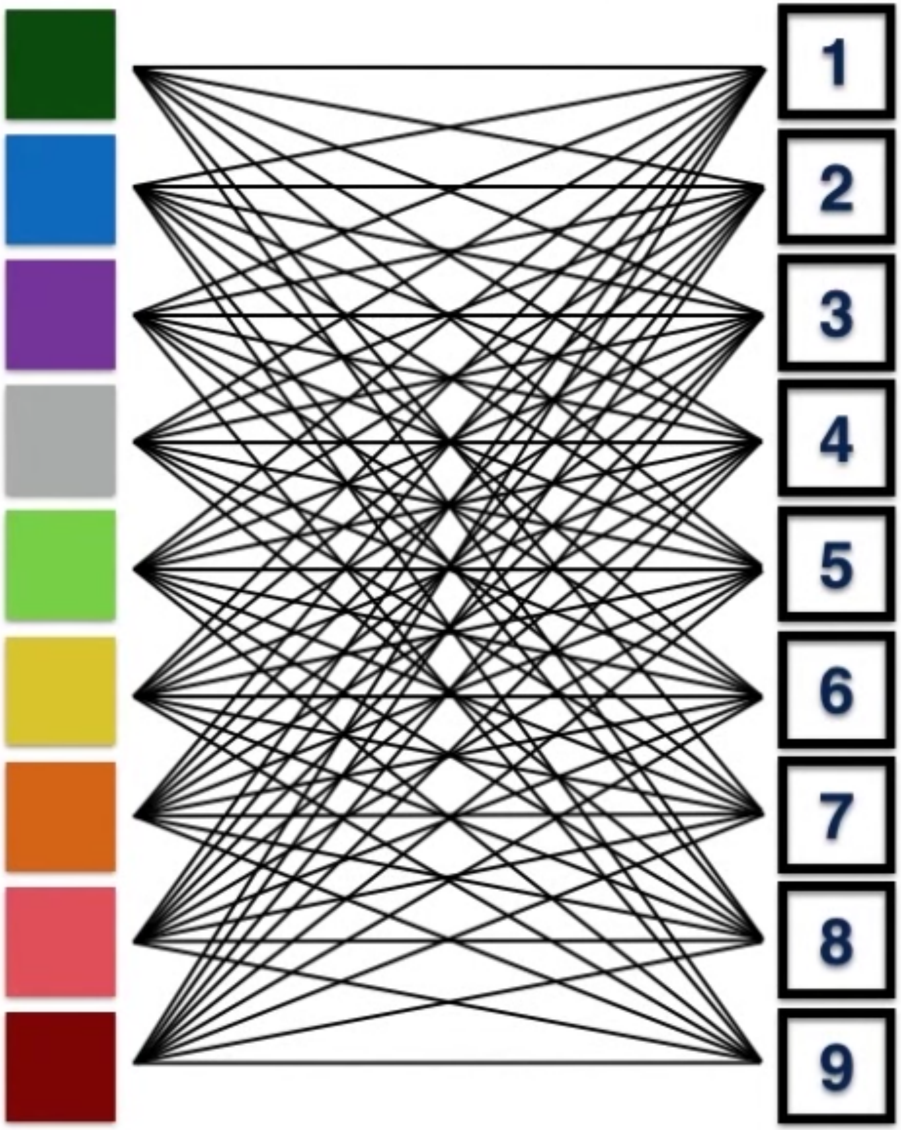
Algorithm – Step II
Coarse Alignment



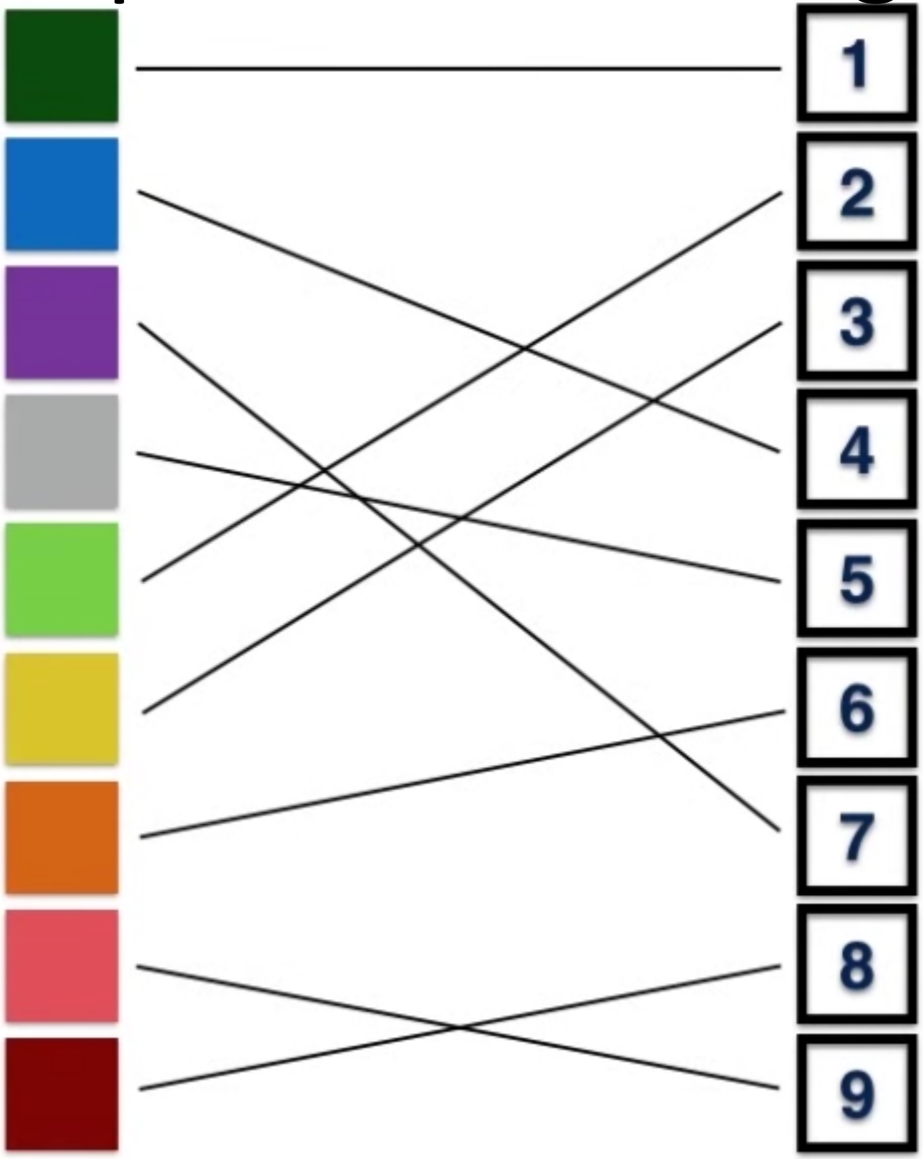
Bipartite Matching

- A complete bipartite graph is built (one with the starting locations, one with the target locations)
- The arc *(i,j)* is weighted according to the corresponding pairwise distance.
- A minimal bipartite matching is calculated using the Hungarian algorithm.

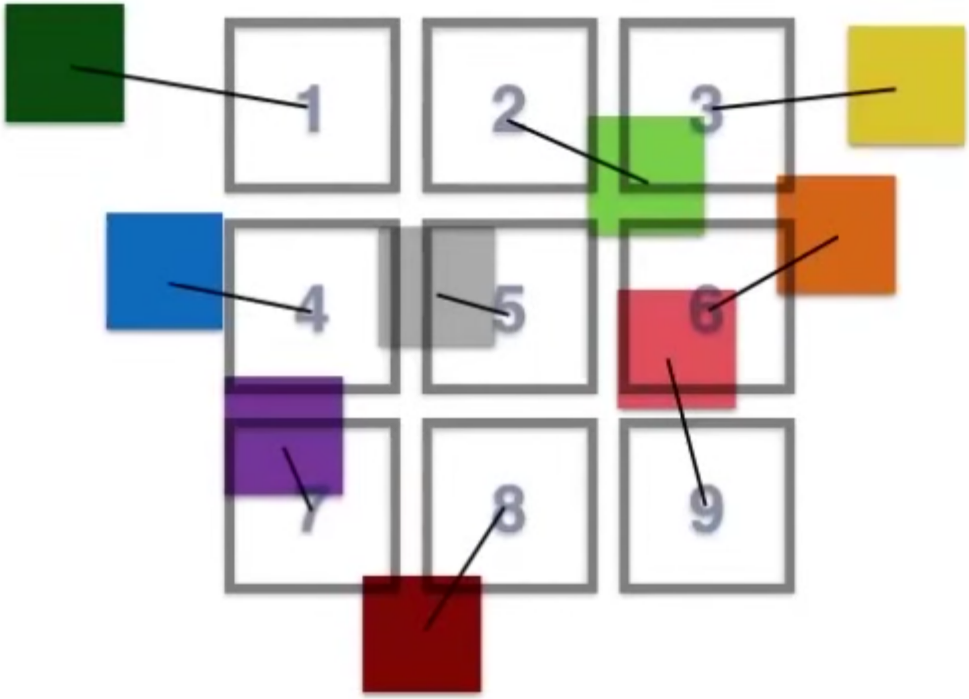
Algorithm – Step III
Bipartite Matching (graph built)



Algorithm – Step III
Bipartite Matching



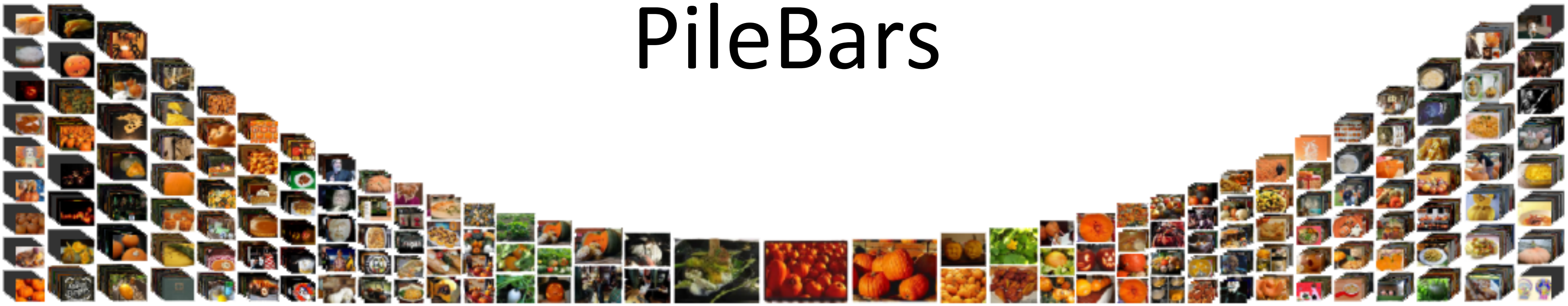
Algorithm – Step III
Final Assignment



Average Colors

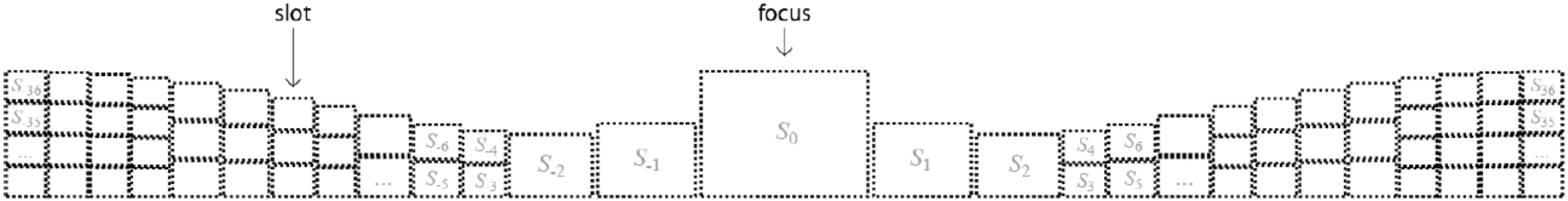
Word Similarity

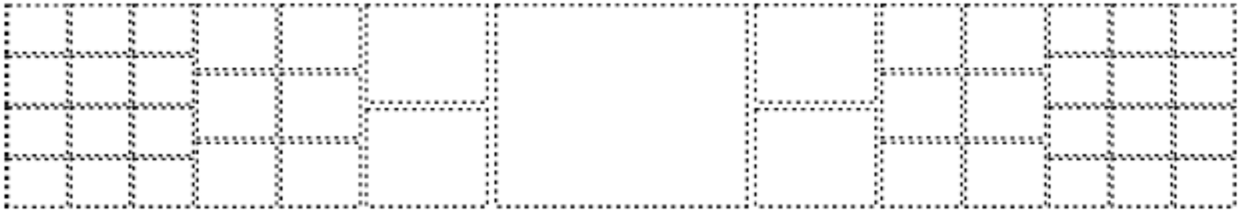
PileBars



- A new type of thumbnail bar.
- Paradigm: *focus + context*.
- Objects are arranged in a small space (images are subdivided into clusters to save space).
- Support any image-image distance.
- PileBars are *dynamic* !

PileBars – Layouts





Slots

1 image

2 images

3 images

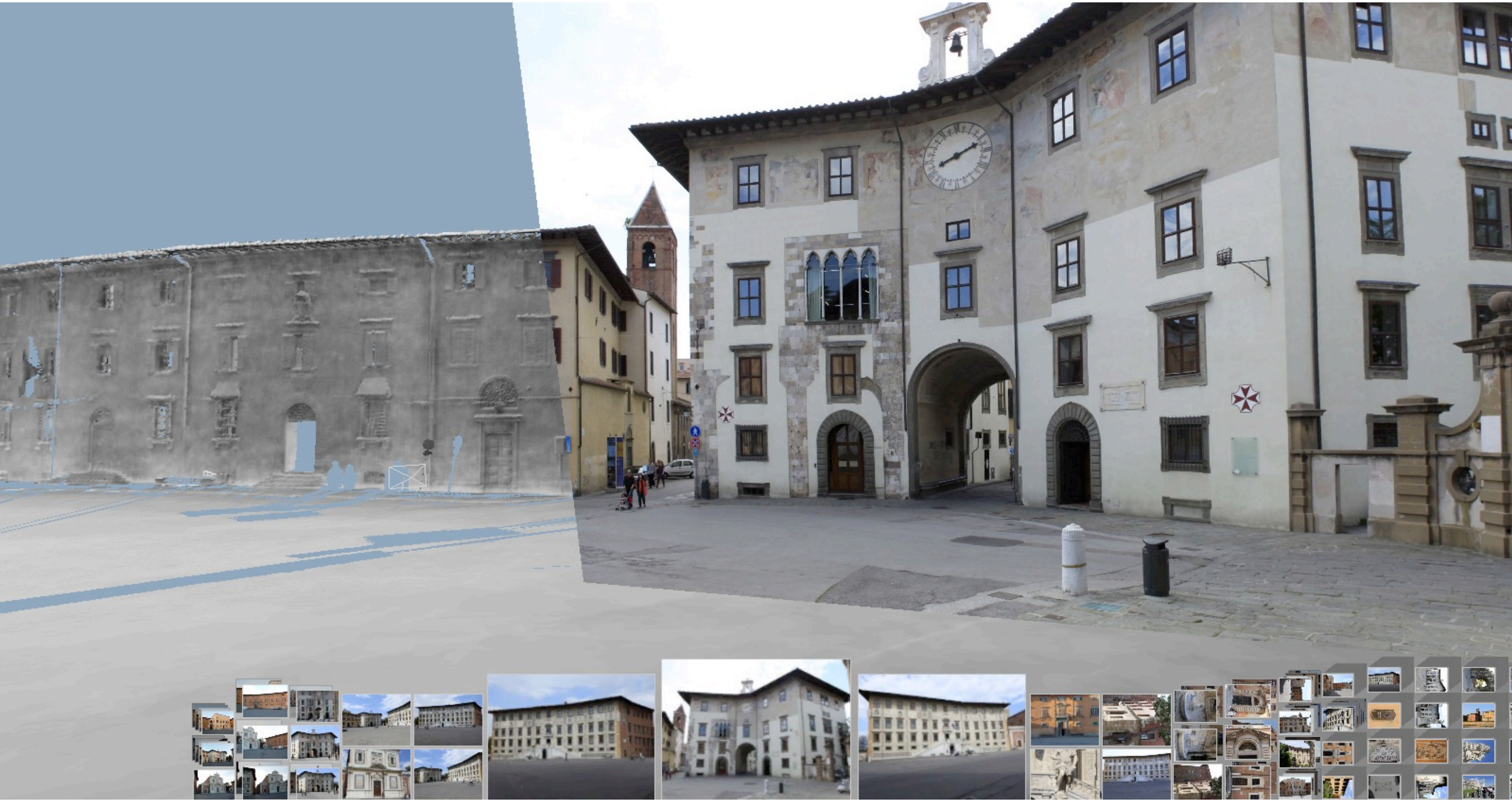
4 images

12 images

PileBars

- Thumbnails are dynamically rearranged, resized and reclustered adaptively during the browsing.
- This is done in a way to ensure *smooth transitions*.

PileBars - Application Example
Navigation of Registered Photographs



Take a look at http://vcg.isti.cnr.it/photocloud .

Questions ?