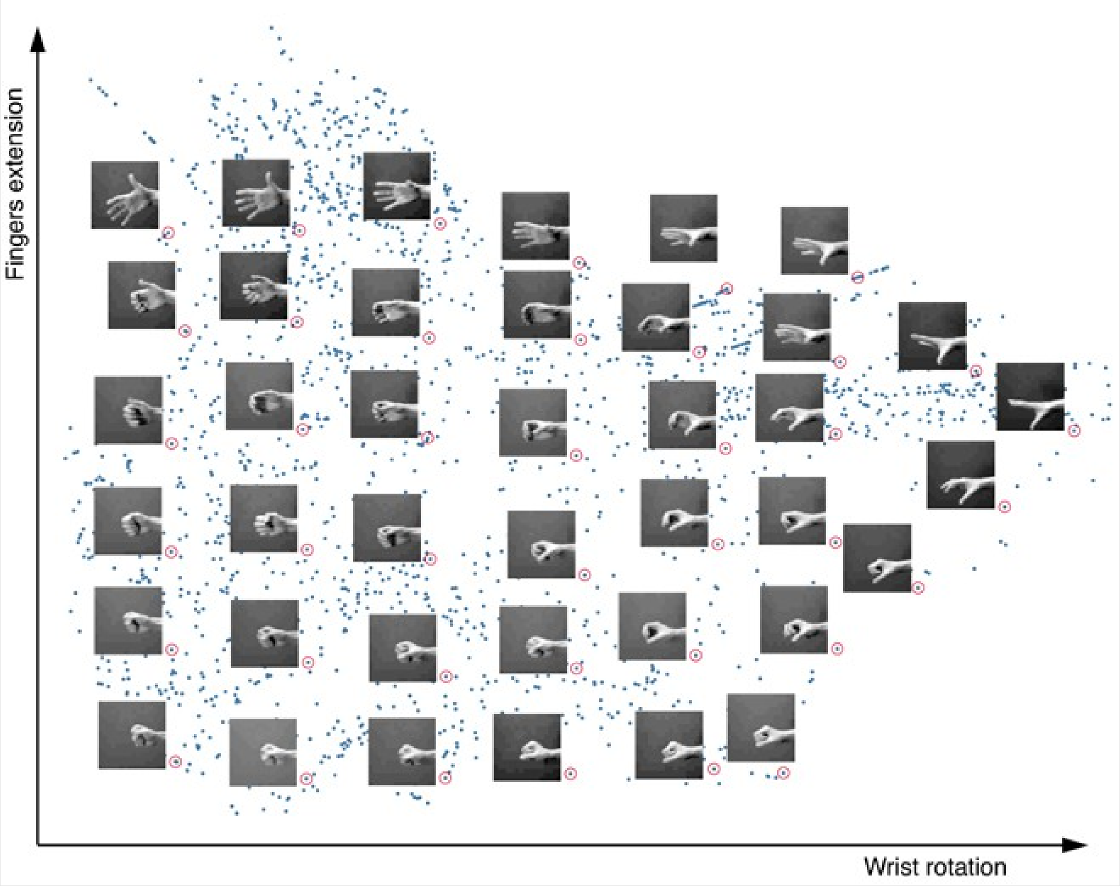
Manifold Learning: ISOMAP and LLE



Demonstration by Nik Melchior

Comparison of methods

- Shamelessly stolen from Todd Wittman
- http://www.math.umn.edu/~wittman/mani/

Hessian LLE *(Donoho & Grimes)*

- Summary
  - Build graph from K Nearest Neighbors.
  - Estimate tangent Hessians.
  - Compute embedding based on Hessians.

$$f: X \to \Re \qquad Basis\left(null\left(\int \| H_f(x) \| \right) dx \right) = Basis(X)$$

- Predictions
  - Specifically set up to handle non-convexity.
  - Slower than LLE & Laplacian.
  - Will perform poorly in sparse regions.
  - Only method with convergence guarantees.

Laplacian Eigenmap *(Belkin & Nyogi)*

- Summary
  - Build graph from K Nearest Neighbors.
  - Construct weighted adjacency matrix with Gaussian kernel.
  - Compute embedding from normalized Laplacian.
    - $$\text{minimize} \int \|\nabla f\|^2 dx \ \text{ subject to } \ \|f\| = 1$$
- Predictions
  - Assumes each point lies in the convex hull of its neighbors. So it might have trouble at the boundary.
  - Will have difficulty with non-uniform sampling.

Diffusion Map *(Coifman & Lafon)*

- Summary
  - Find Gaussian kernel $K(x,y)=\exp\left(-\frac{\|x-y\|^2}{\sigma}\right)$
  - Normalize kernel

$$K^{(\alpha)}(x,y)=\frac{K(x,y)}{p^{\alpha}(x)p^{\alpha}(y)} \quad \text{where} \quad p(x)=\int K(x,y)P(y)\,dy$$

  - Apply weighted graph Laplacian.

$$A(x,y)=\frac{K^{(\alpha)}(x,y)}{d^{(\alpha)}(x,y)} \quad \text{where} \quad d^{(\alpha)}(x,y)=\int K^{(\alpha)}(x,y)P(y)\,dy$$

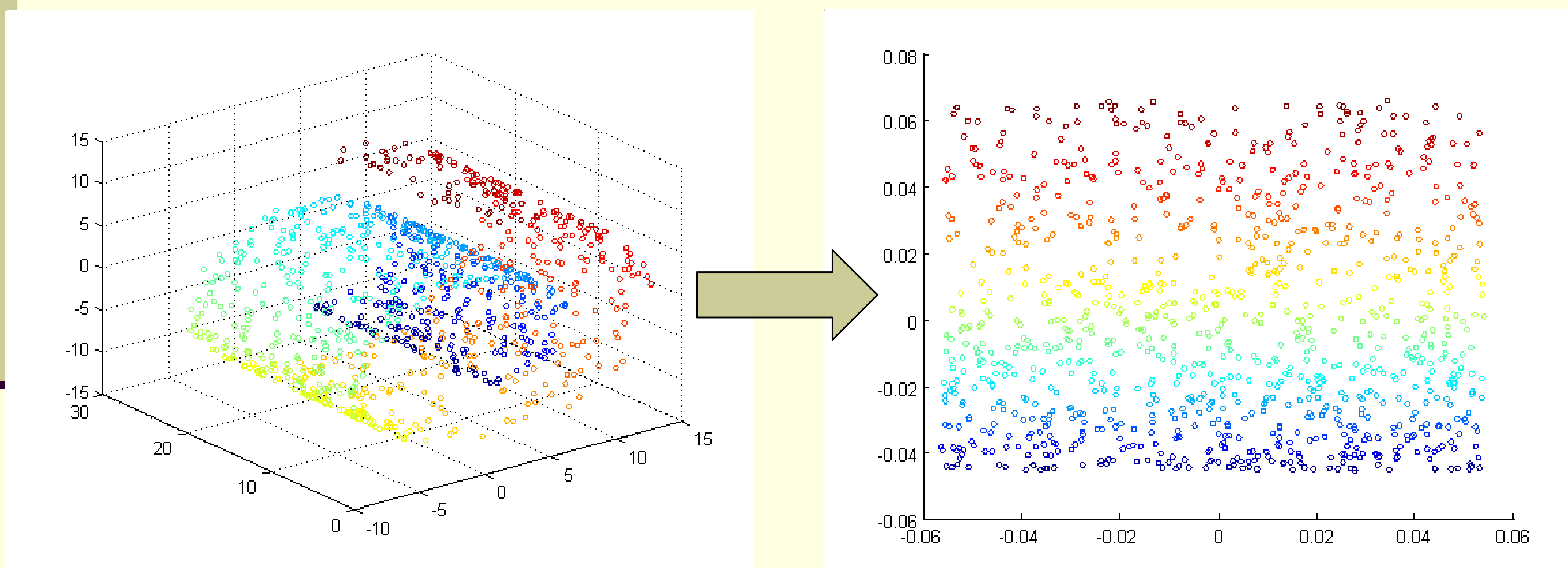
  - Compute SVD of A
- Predictions
  - Doesn't seem to infer geometry directly.
  - Need to set parameters alpha and sigma.

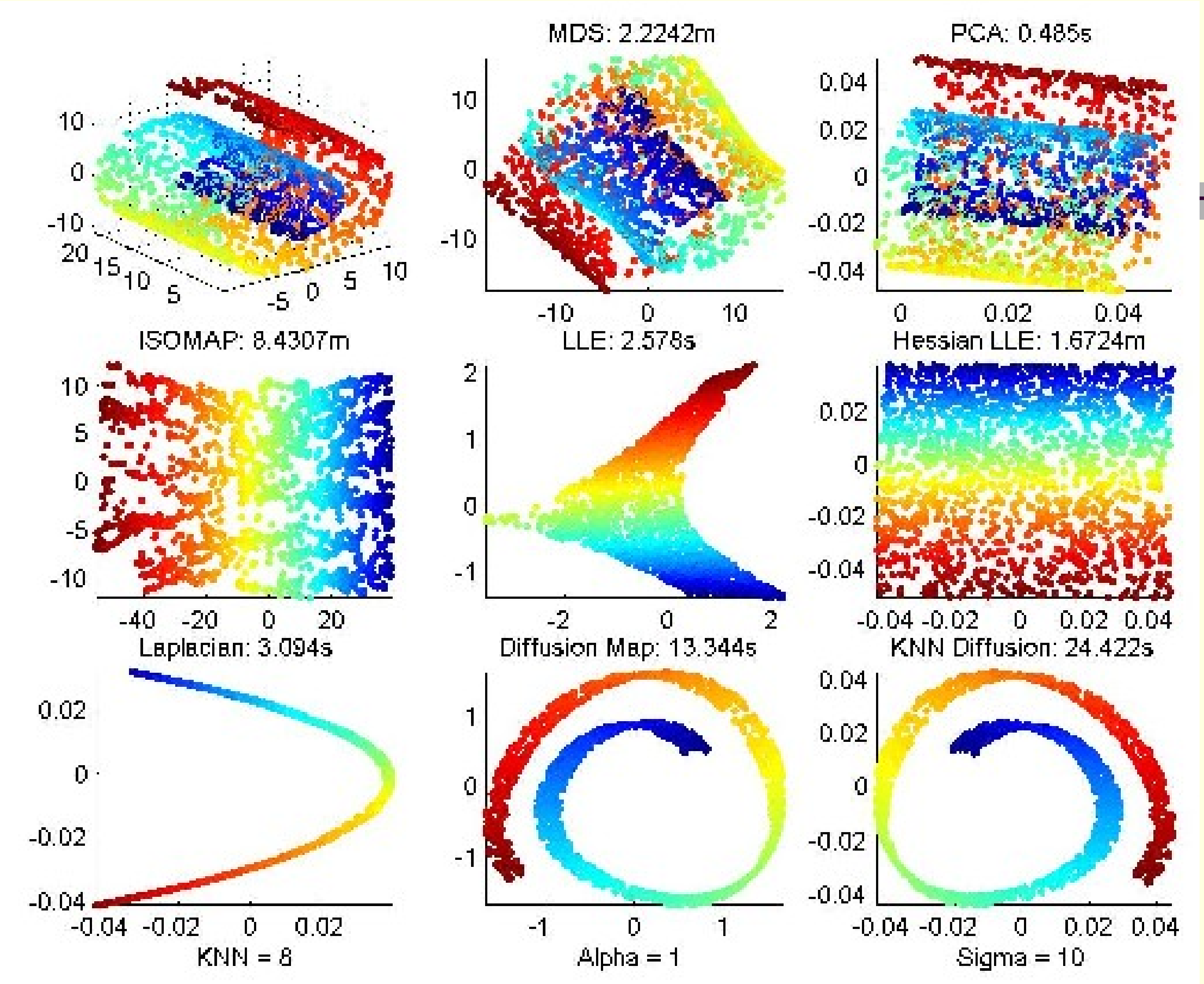
KNN Diffusion *(Mauro)*

- Summary
  - Build graph from K nearest neighbors.
  - Run Diffusion Map on graph.
- Predictions
  - Should infer geometry better than Diffusion Map.
  - Now we have to set the parameters alpha, sigma, and K.

Manifold Geometry

- First, let's try to unroll the Swiss Roll.
- We should see a plane.

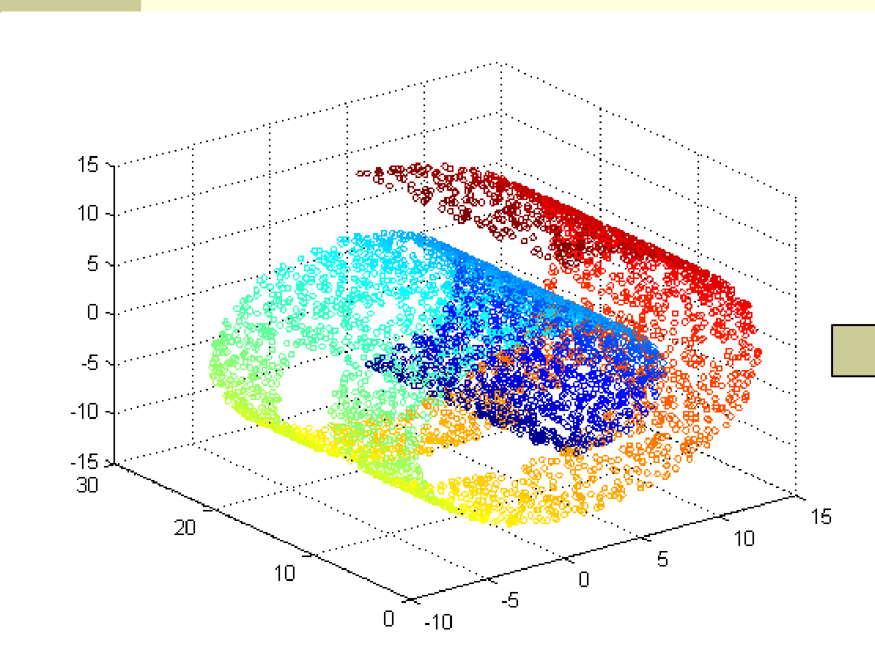


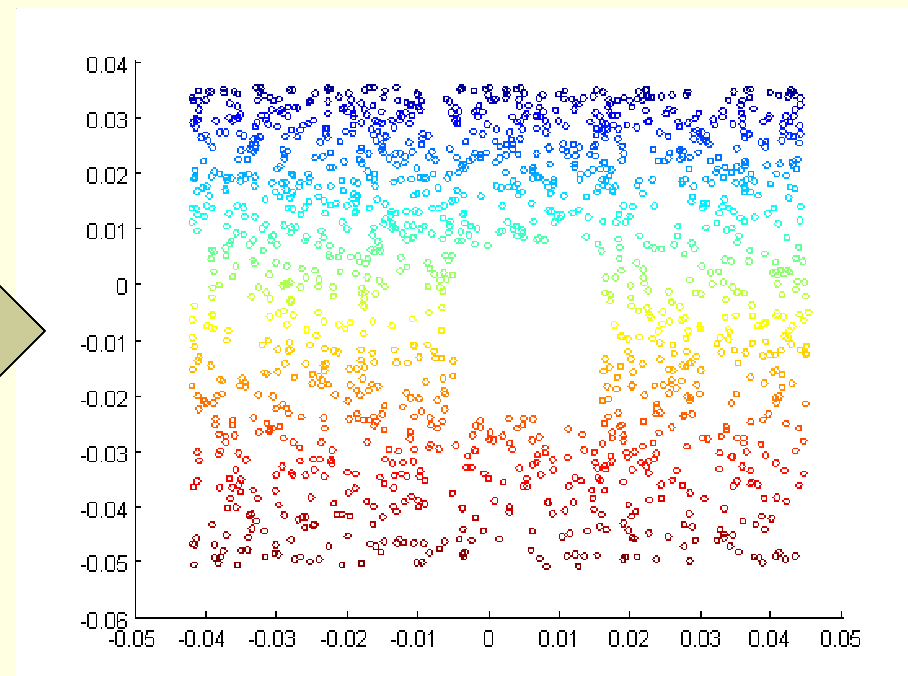


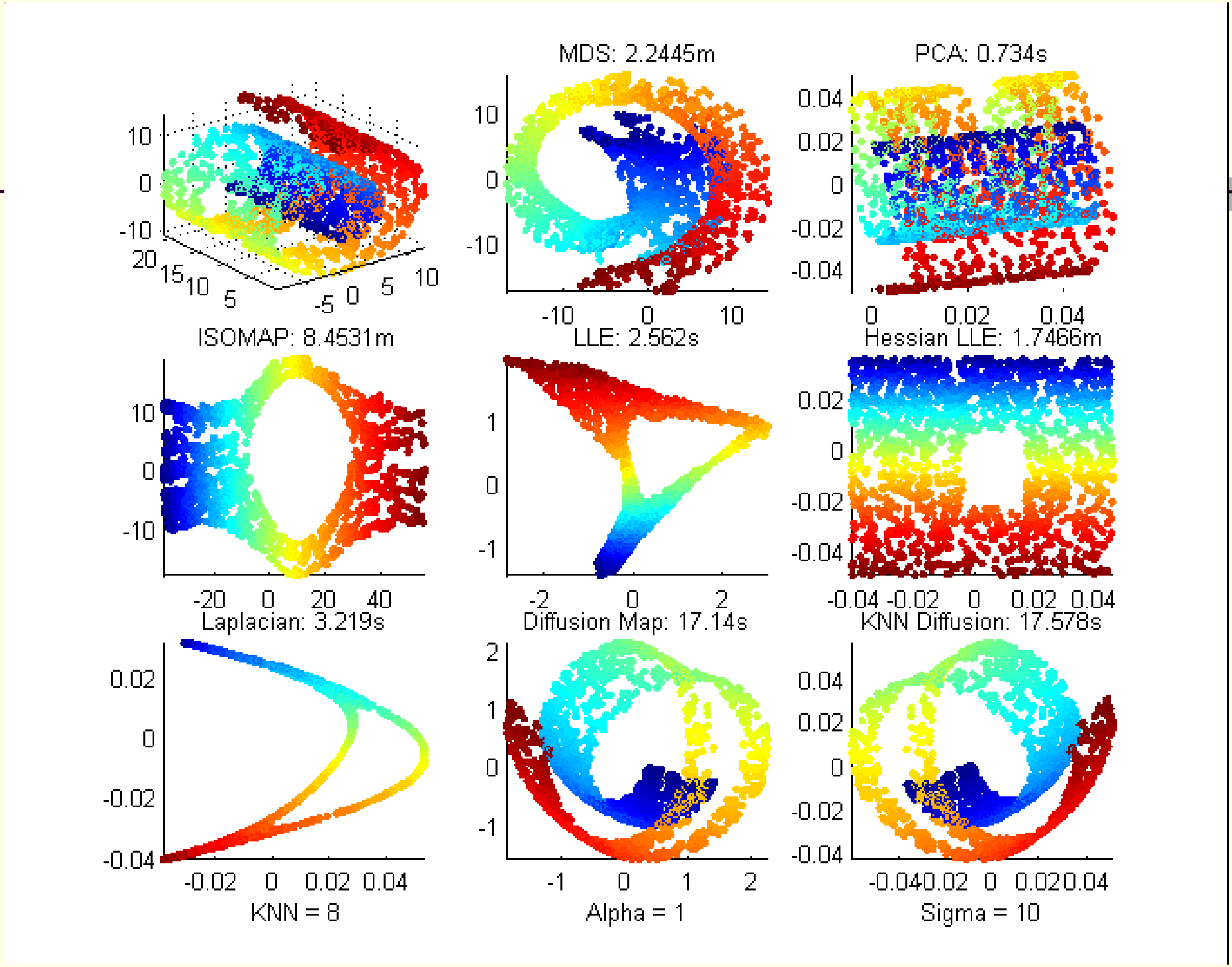
Hessian LLE is pretty slow, MDS is very slow, and ISOMAP is extremely slow.
MDS and PCA don't can't unroll Swiss Roll, use no manifold information.
LLE and Laplacian can't handle this data.
Diffusion Maps could not unroll Swiss Roll for any value of Sigma.

Non-Convexity

- Can we handle a data set with a hole?
- Swiss Hole: Can we still unroll the Swiss Roll when it has a hole in the middle?



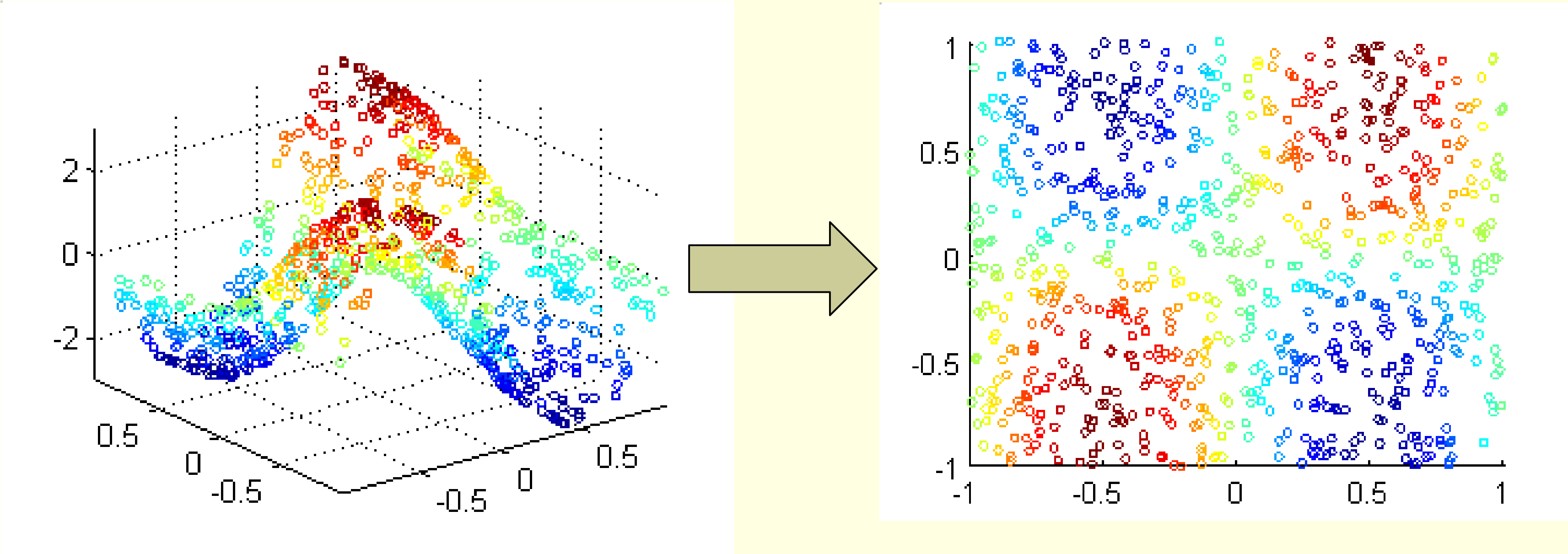


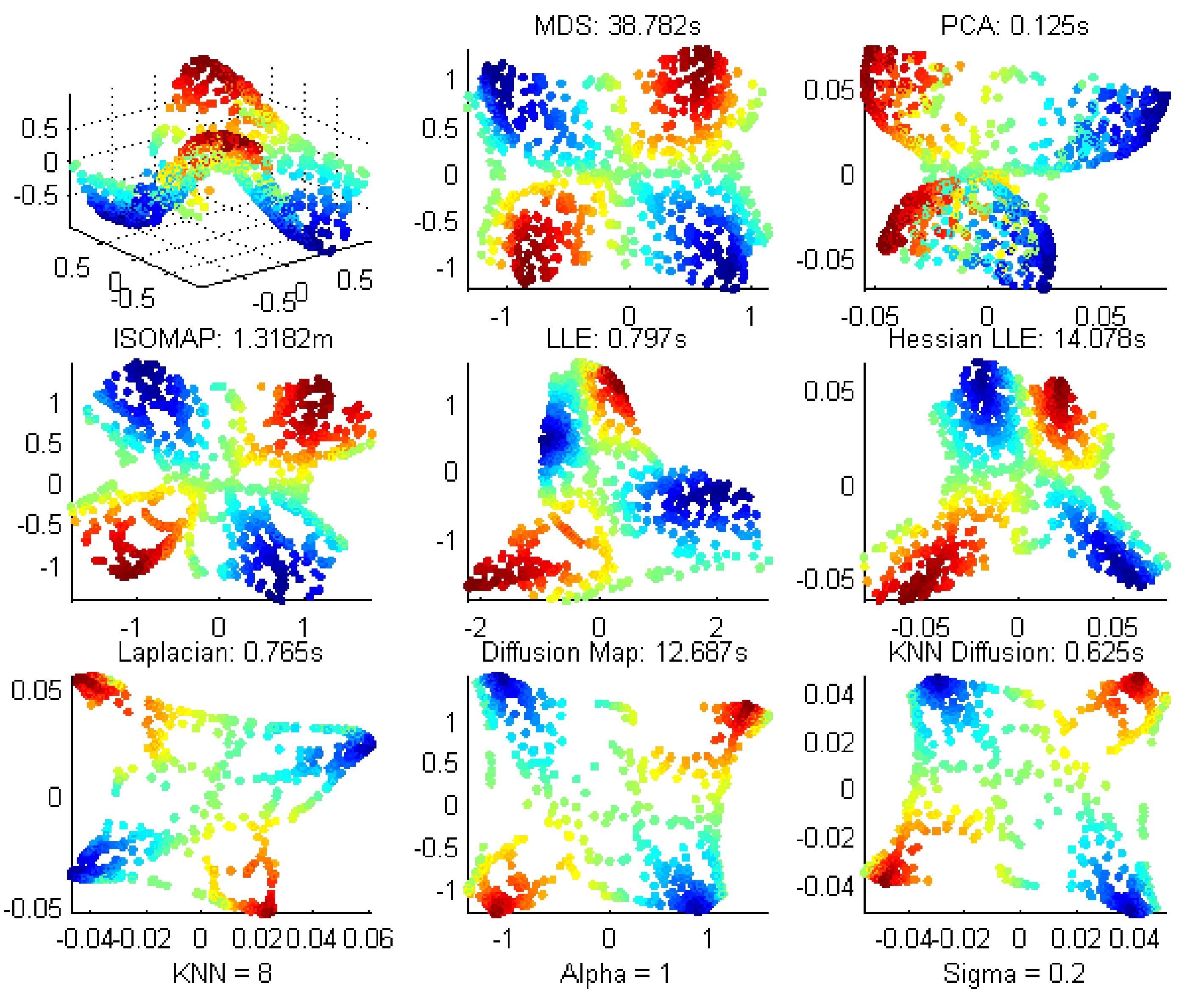


Only Hessian LLE can handle non-convexity.
ISOMAP, LLE, and Laplacian find the hole but the set is distorted.

Manifold Geometry

- Twin Peaks: fold up the corners of a plane.
- LLE will have trouble because it introduces curvature to plane.

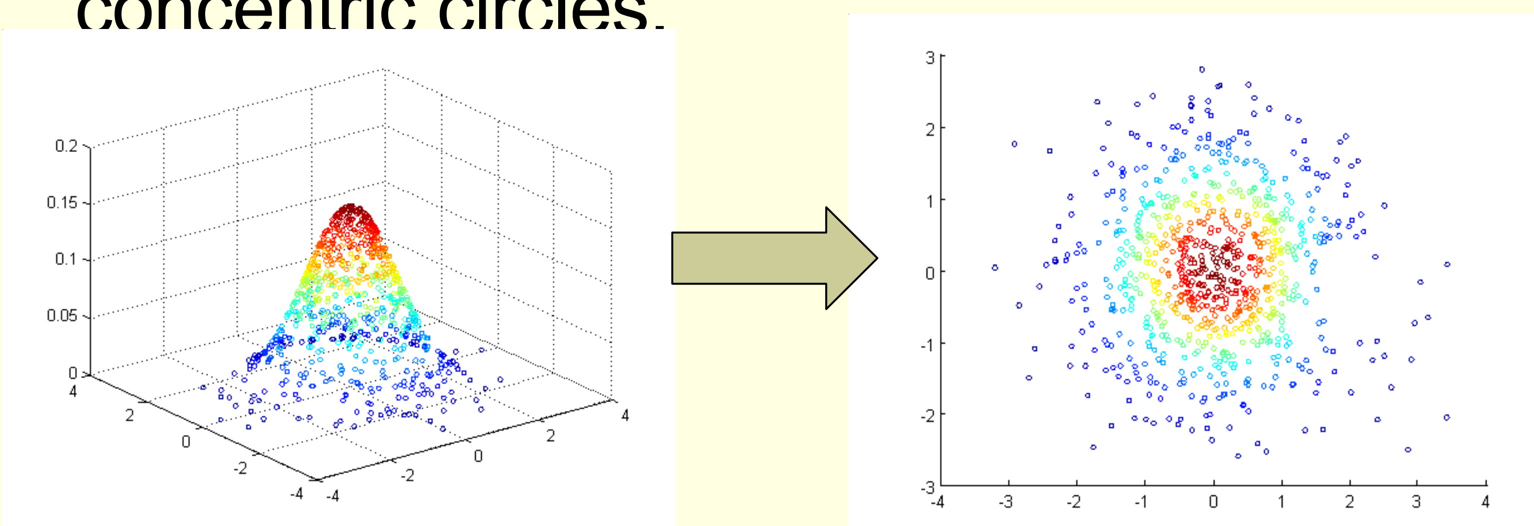


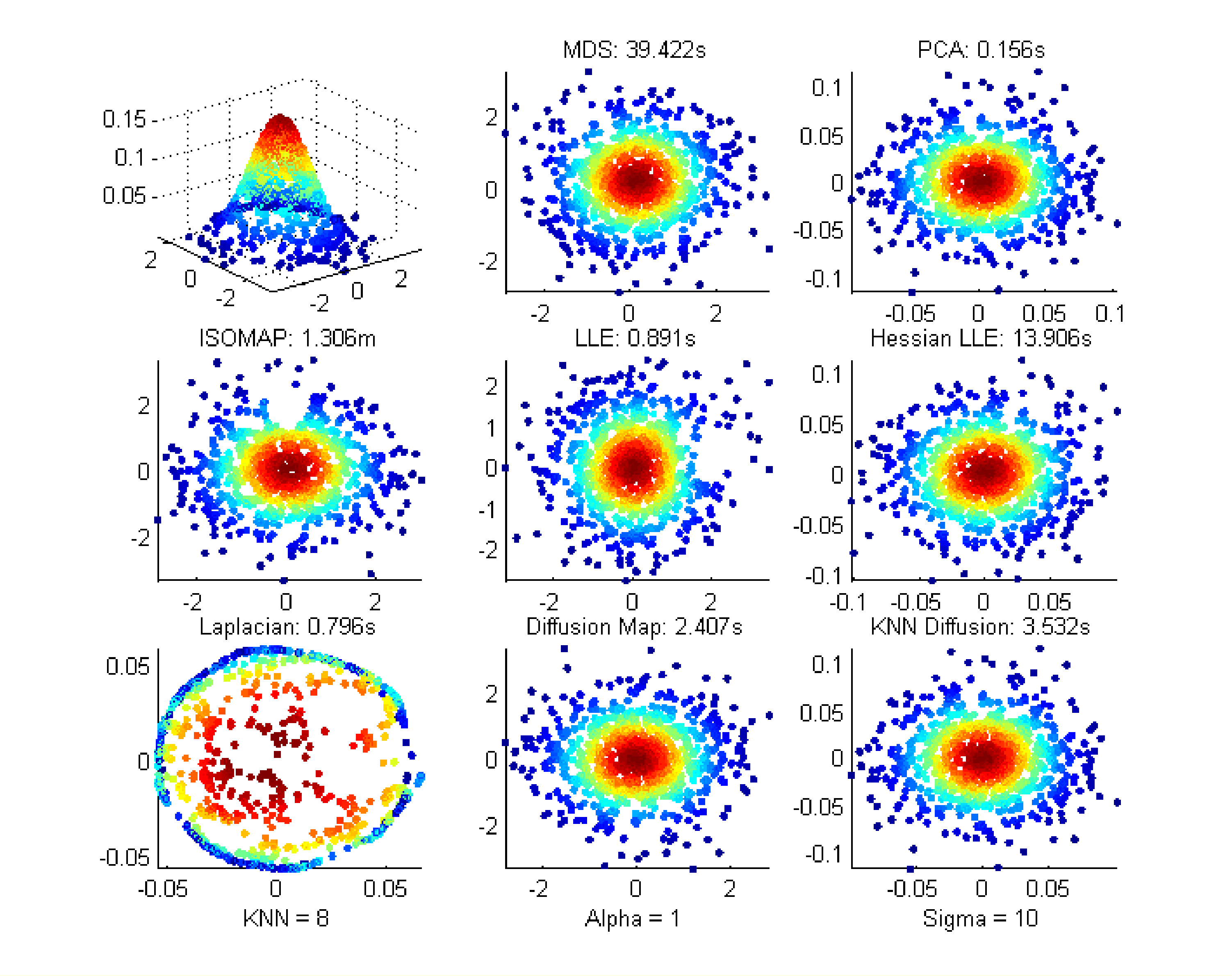


PCA, LLE, and Hessian LLE distort the mapping the most.

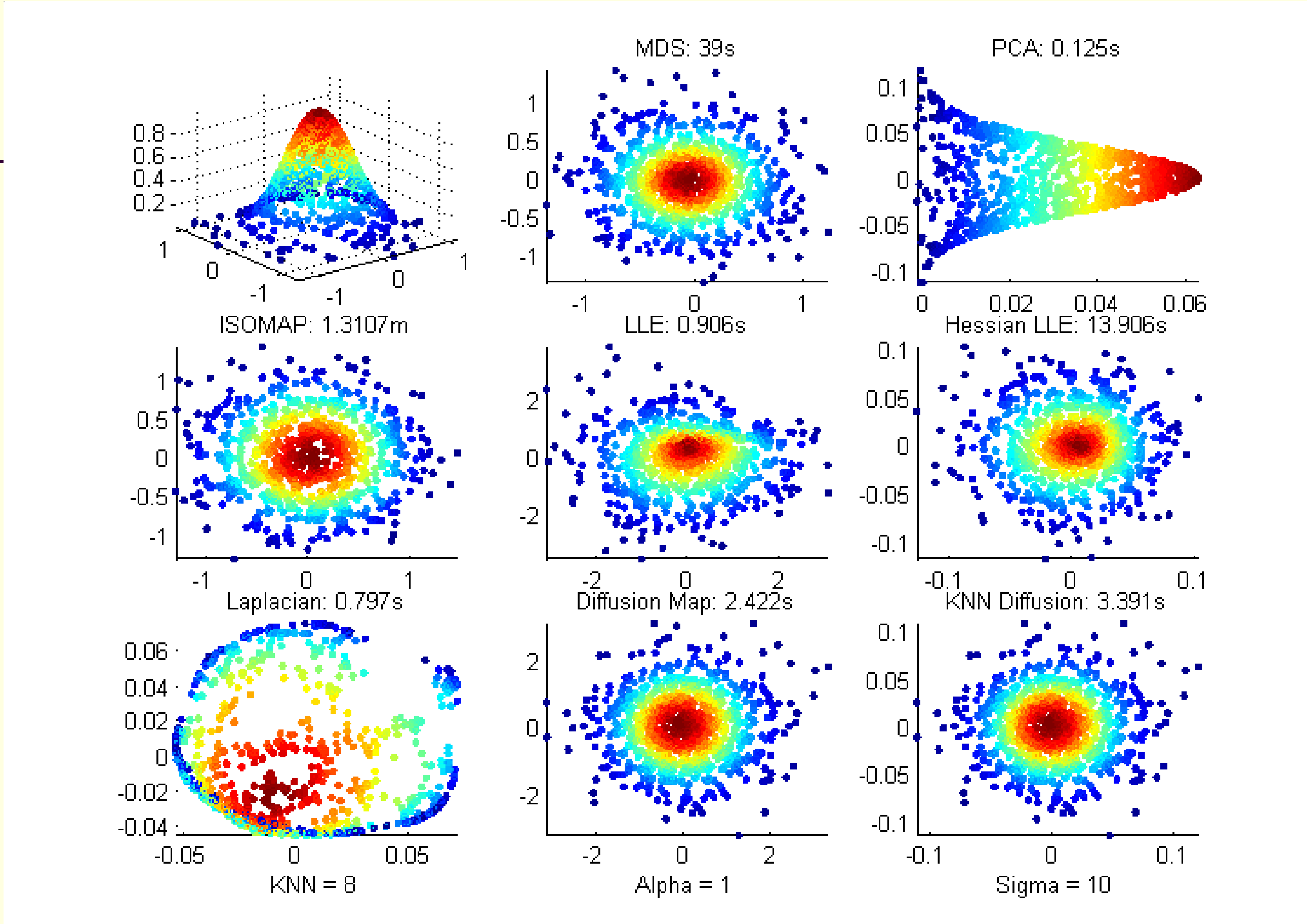
Curvature & Non-uniform Sampling

- Gaussian: We can randomly sample a Gaussian distribution.
- We increase the curvature by decreasing the standard deviation.
- Coloring on the z-axis, we should map to concentric circles.

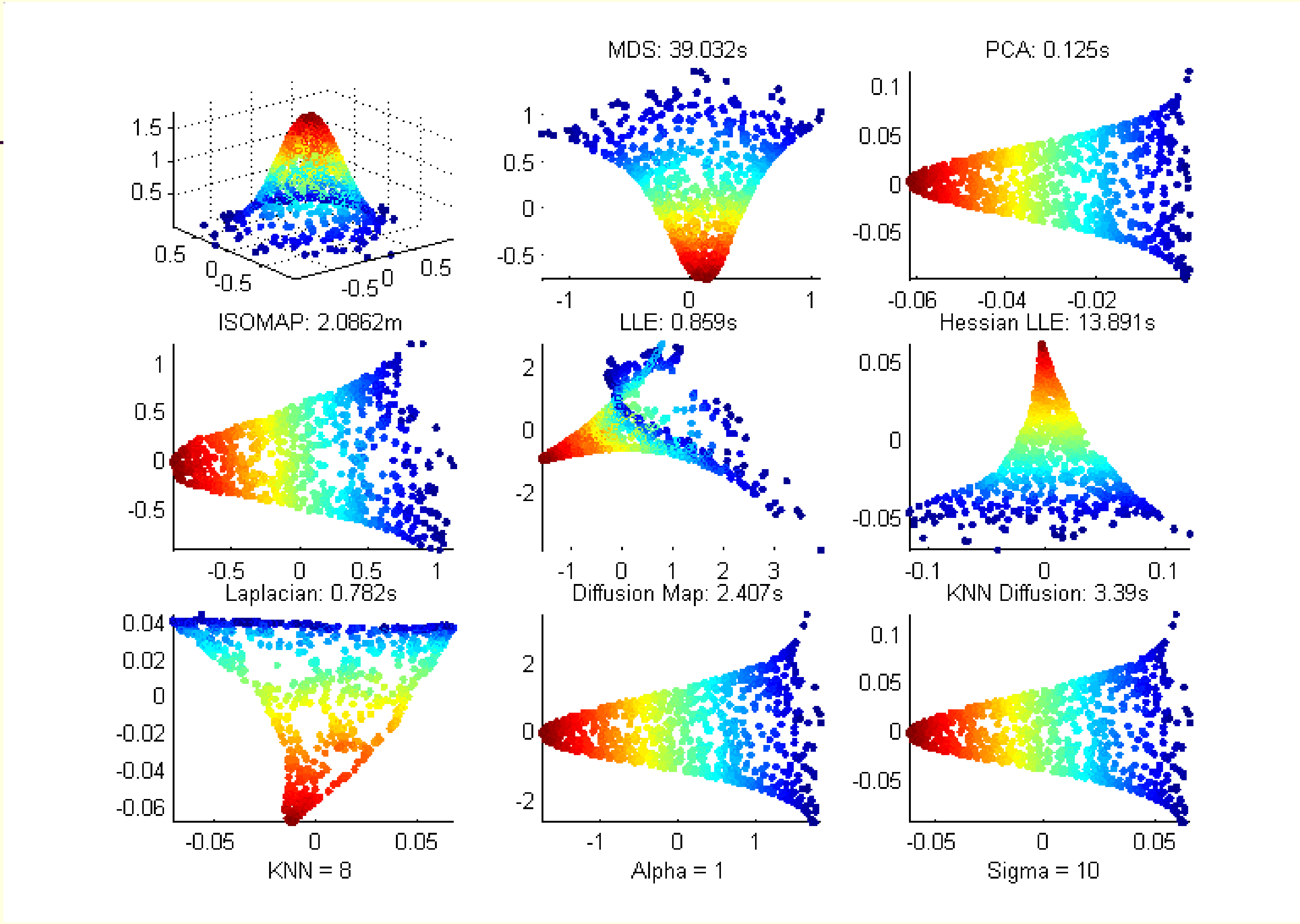




For std = 1 (low curvature), MDS and PCA can project accurately. Laplacian Eigenmap cannot handle the change in sampling.



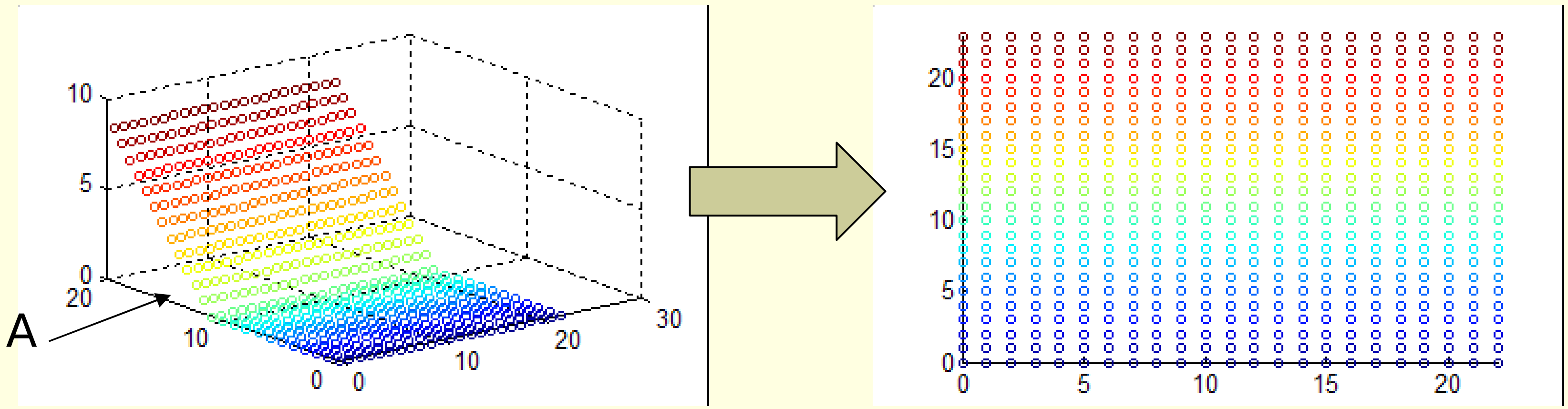
For std = 0.4 (higher curvature), PCA projects from the side rather than top-down. Laplacian looks even worse.



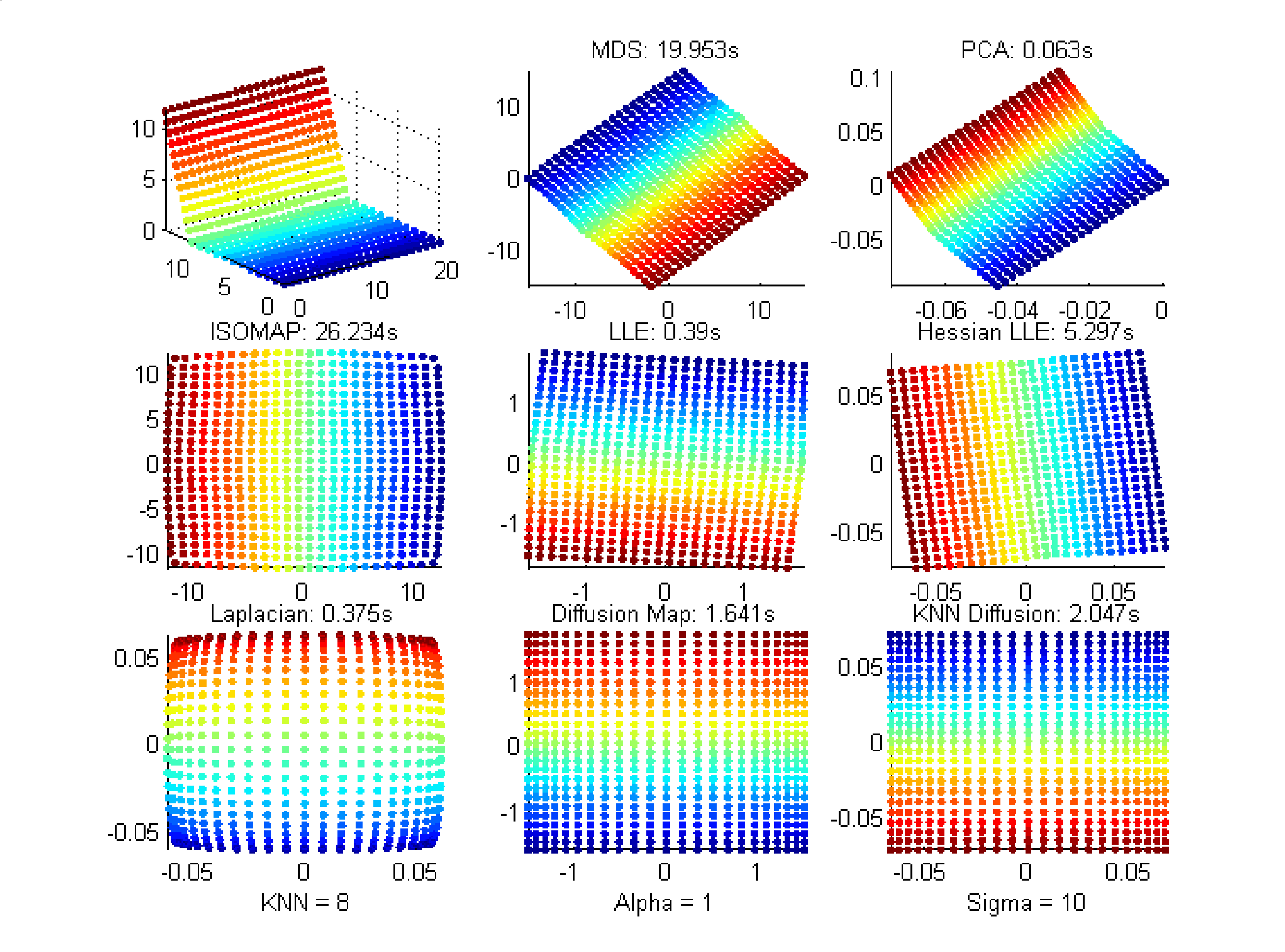
For std = 0.3 (high curvature), none of the methods can project correctly.

Corners

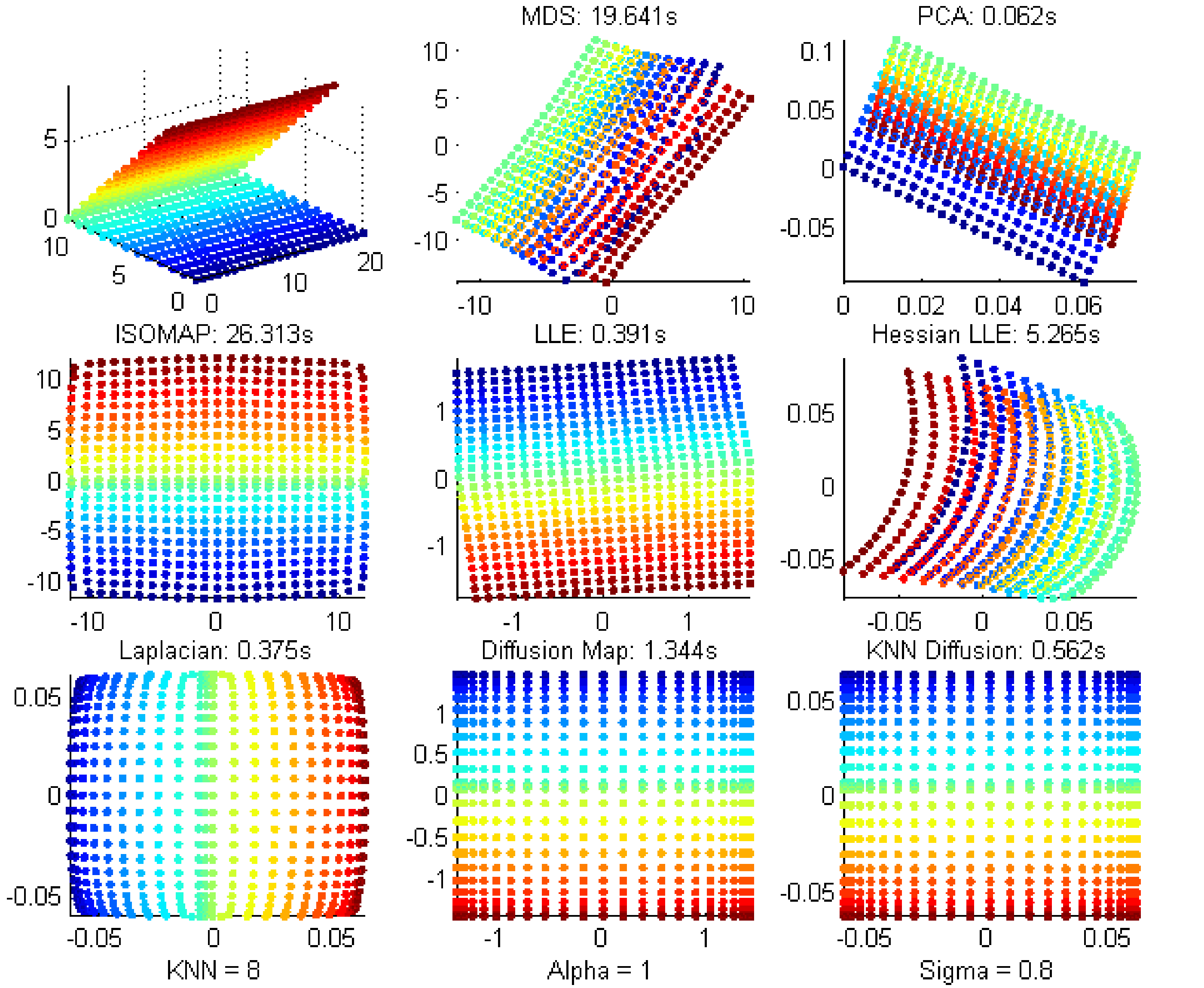
- Corner Planes: We bend a plane with a lift angle A.
- We want to bend it back down to a plane.



- If A > 90, we might see the data points written on top of each other.



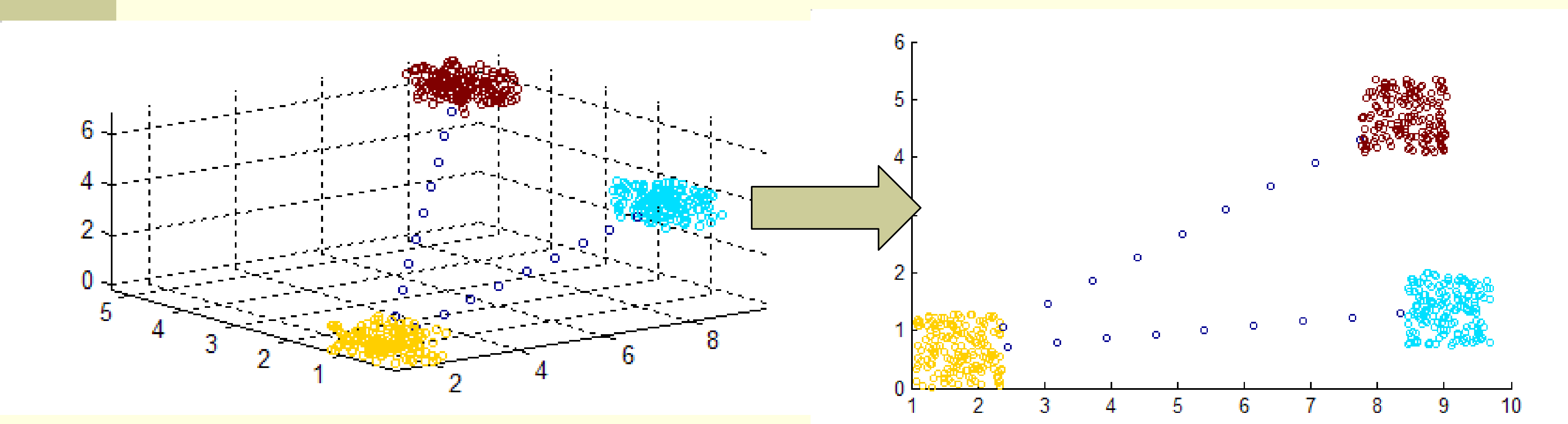
For angle A=75, we see some disortions in PCA and Laplacian.

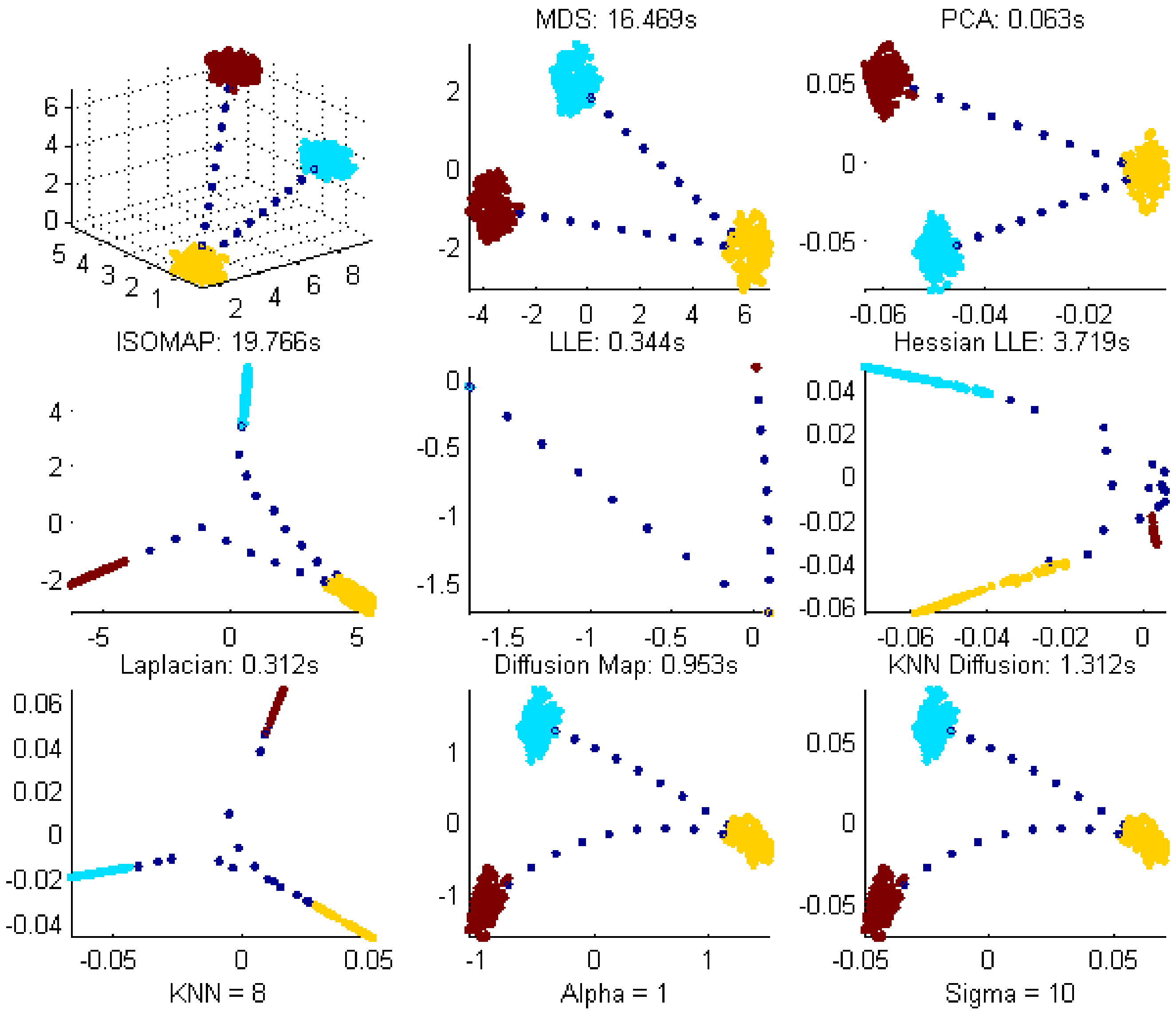


For A = 135, MDS, PCA, and Hessian LLE overwrite the data points.
Diffusion Maps work very well for Sigma < 1.
LLE handles corners surprisingly well.

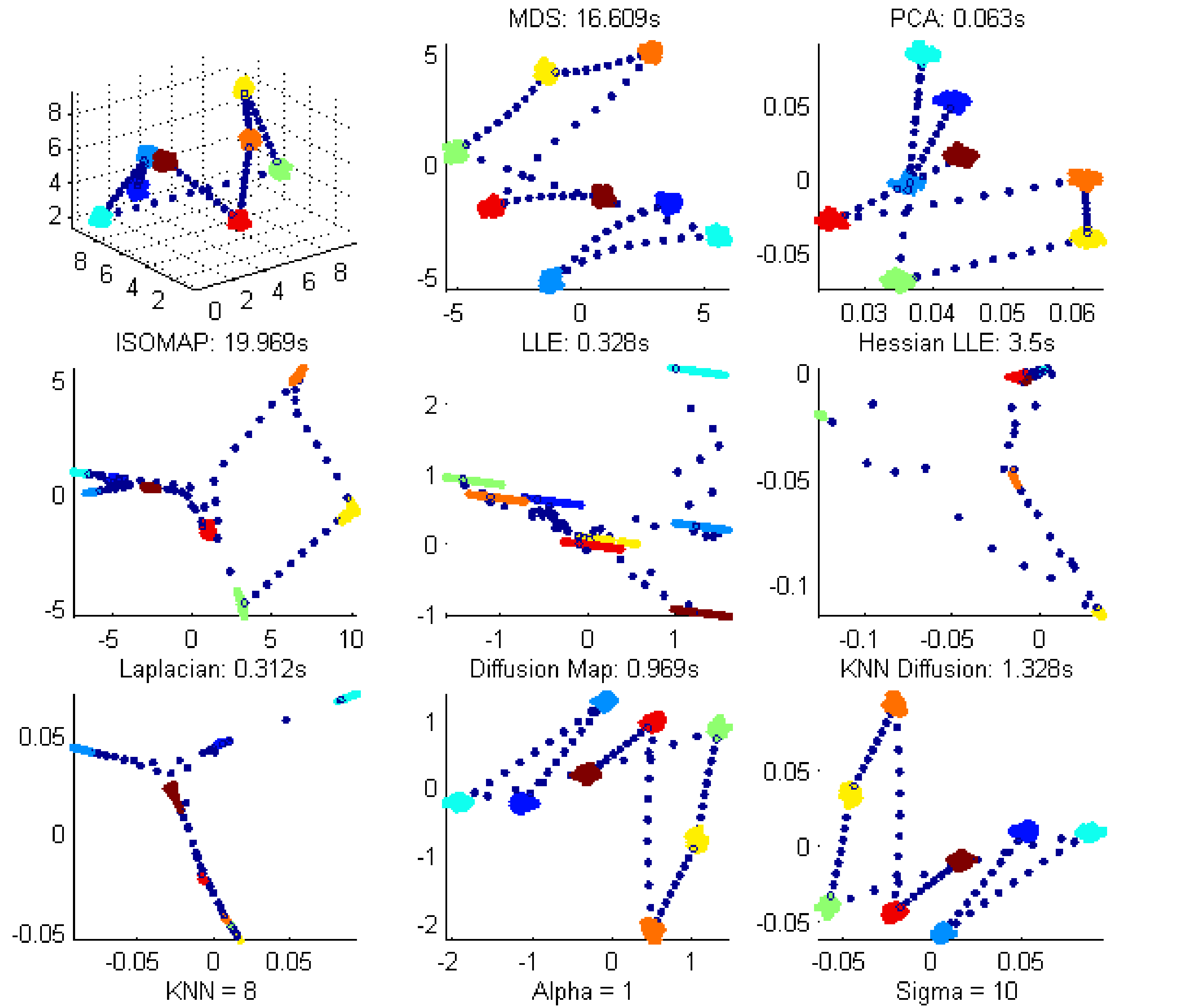
Clustering

- A good mapping should preserve clusters in the original data set.
- 3D Clusters: Generate M non-overlapping clusters with random centers. Connect the clusters with a line.





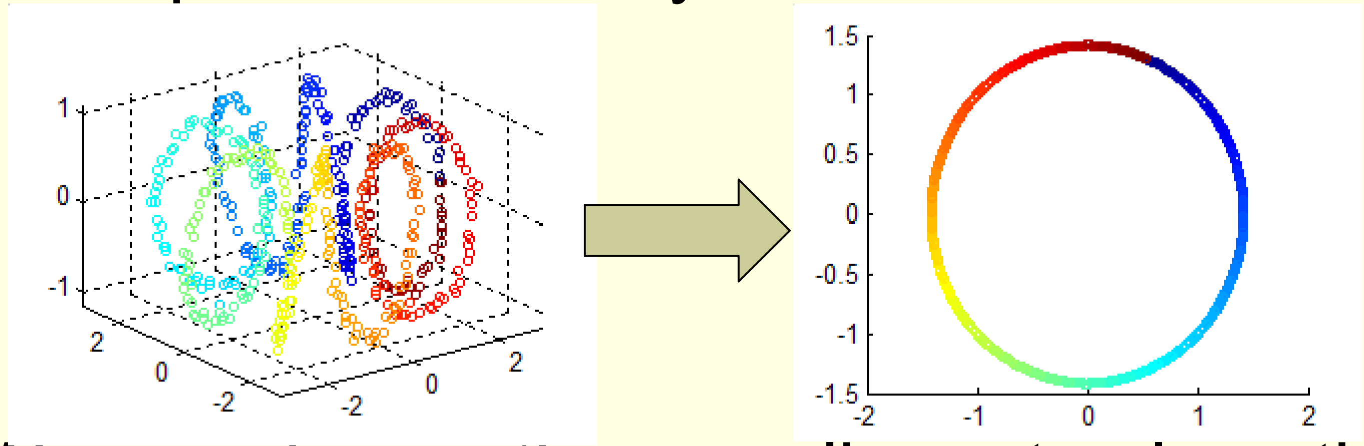
For M = 3 clusters, MDS and PCA can project correctly.
Diffusion Maps work well with large Sigma.
LLE compresses each cluster into a single point.
Hessian LLE has trouble with the sparse connecting lines.



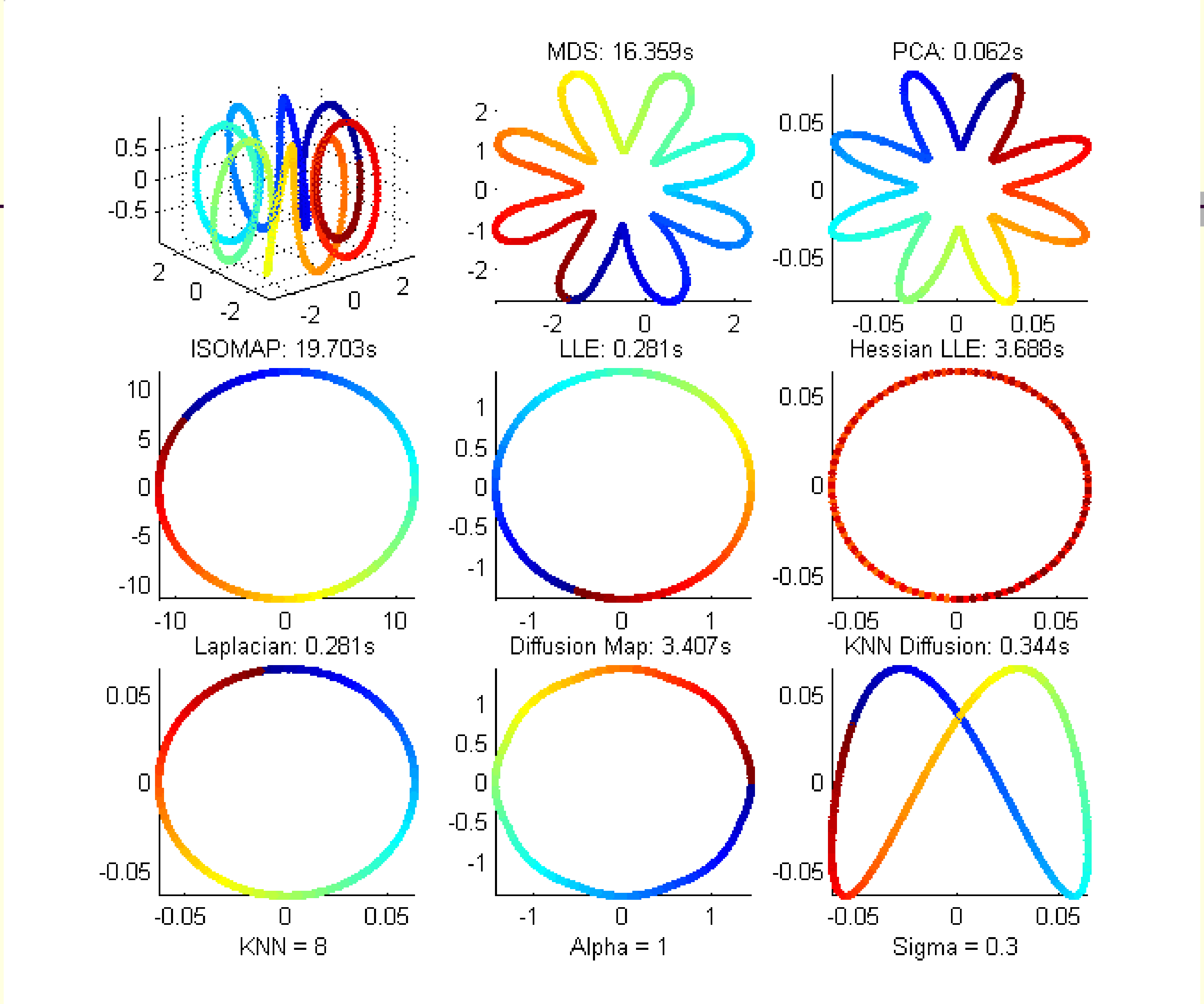
For M=8 clusters, MDS and PCA can still recover.
Diffusion Maps do quite well.
LLE and ISOMAP are decent, but Hessian and Laplacian fail.

Noise & Non-uniform Sampling

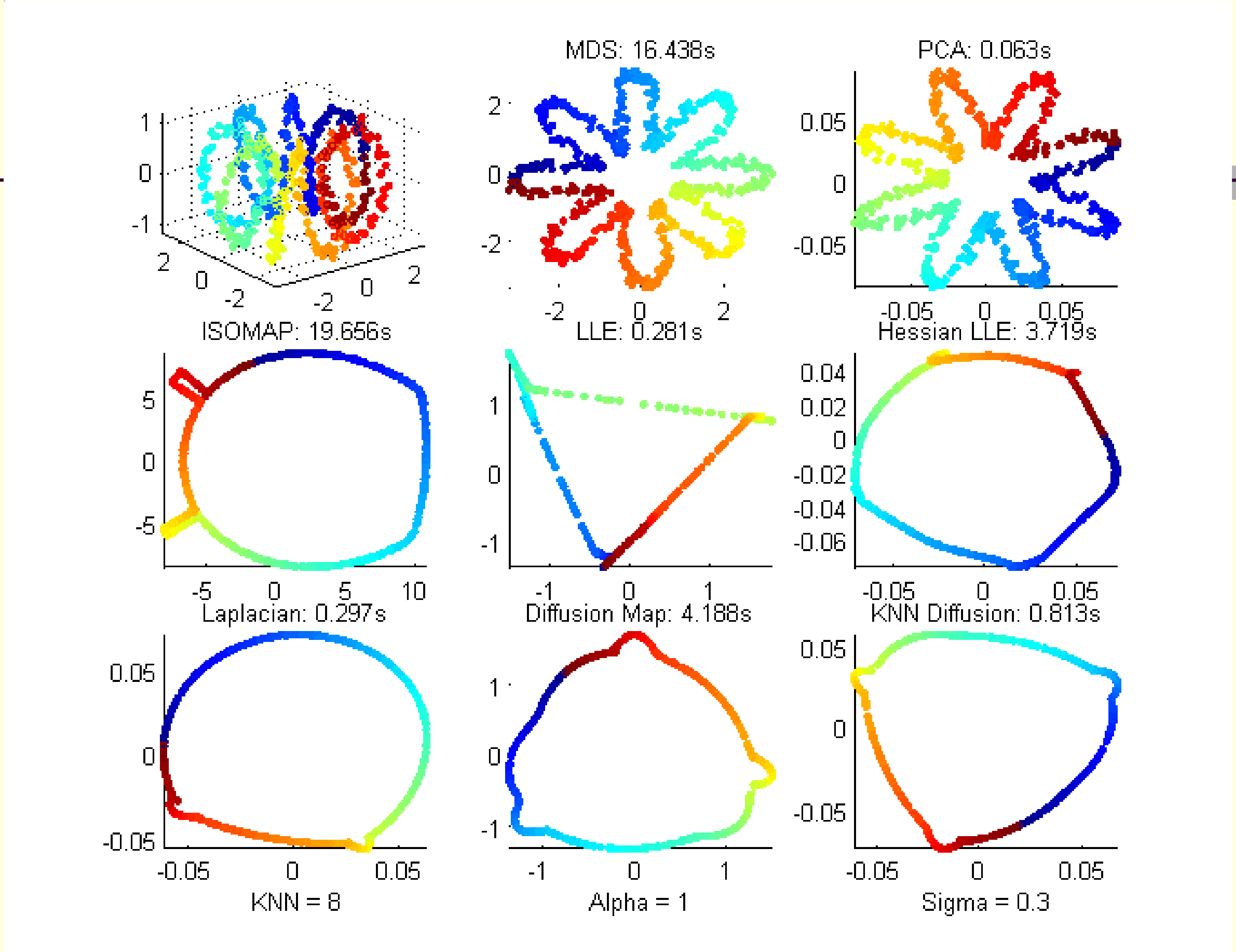
- Can the method handle changes from dense to sparse regions?
- Toroidal Helix should be unraveled into a circle parametrized by t.



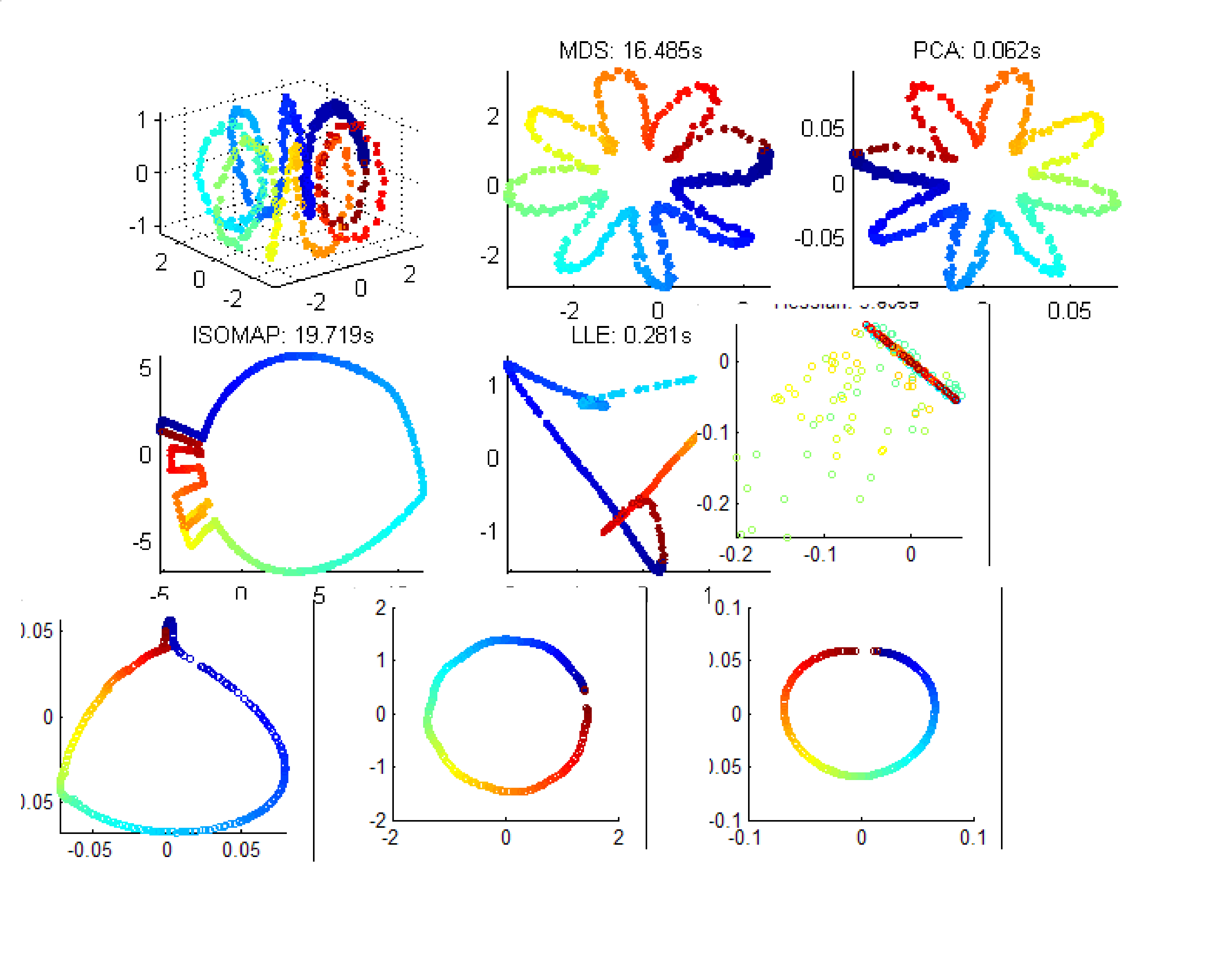
- We can change the sampling rate along the helix by changing the exponent R on the parameter t and we can add some noise.



With no noise added, ISOMAP, LLE, Laplacian, and Diffusion Map are correct.
MDS and PCA project to an asterisk.
What’s up with Hessian and KNN Diffusion?



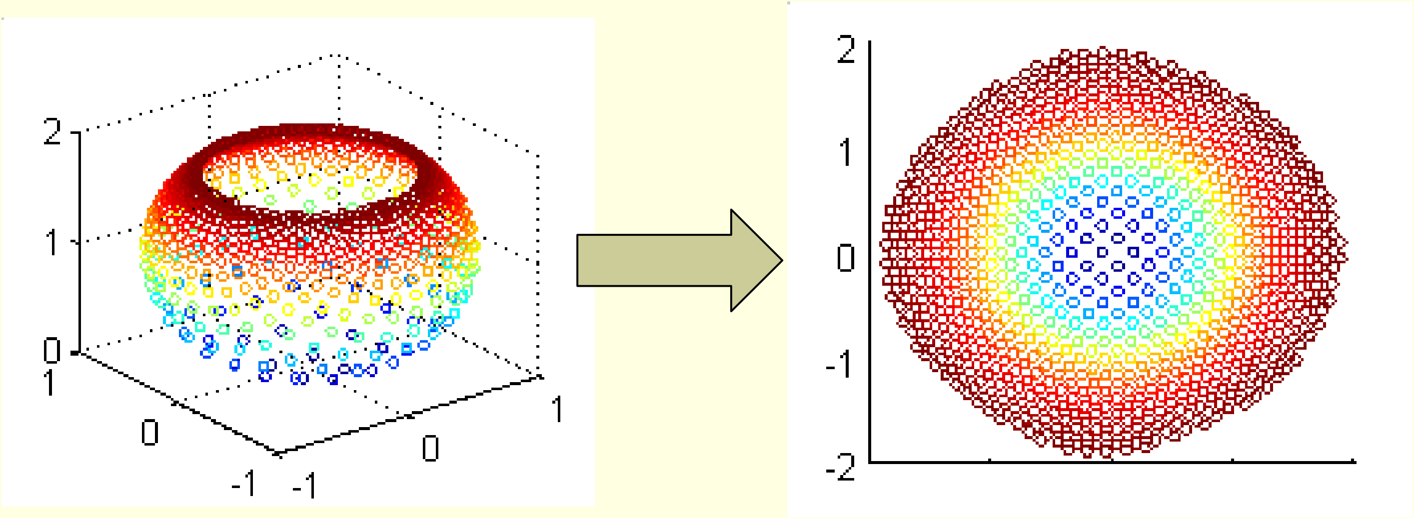
Adde noise to the Helix sampling.
LLE cannot recover the circle.
ISOMAP emphasizes outliers more than the other methods.

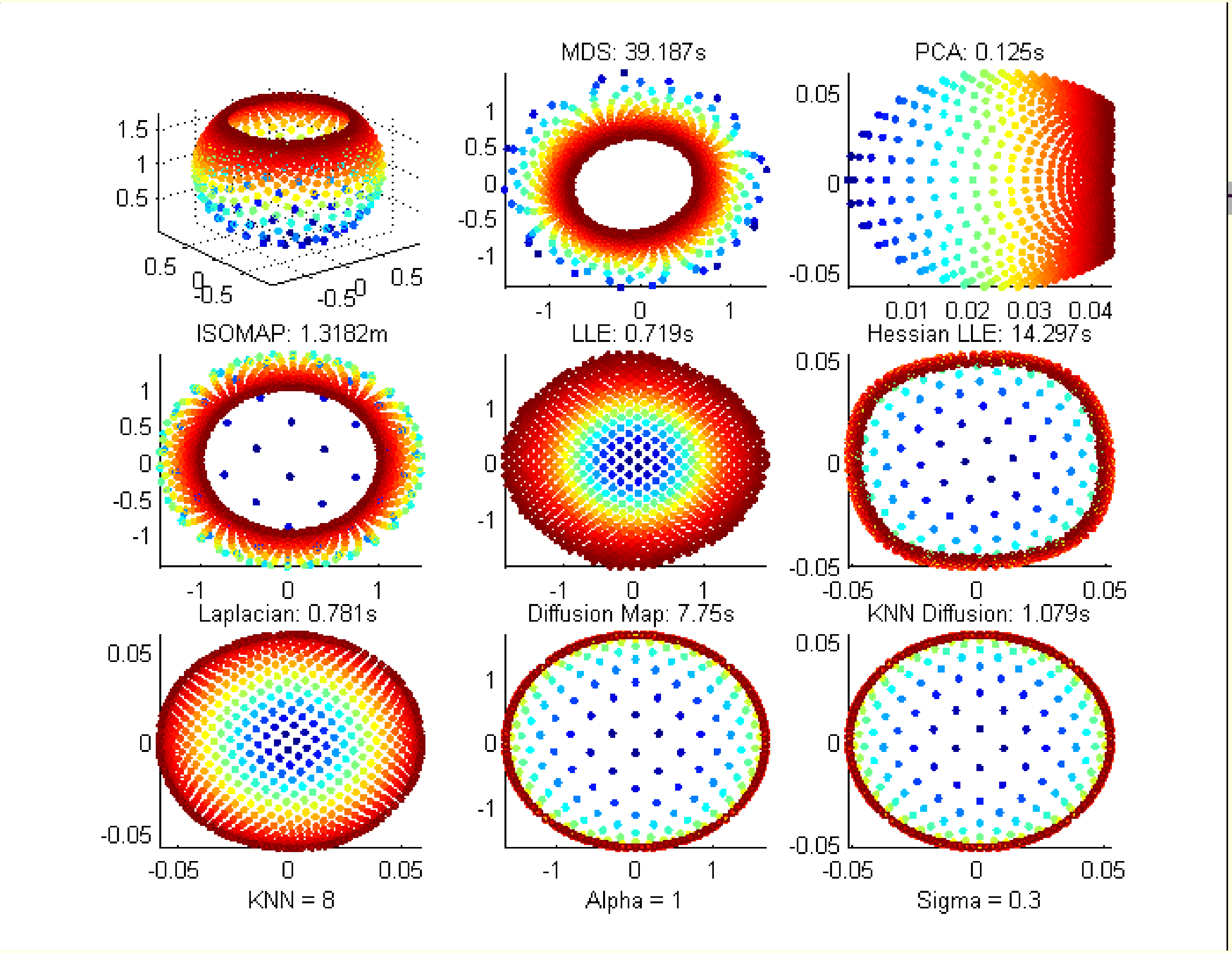


When the sampling rate is changed along the torus, Laplacian starts to mess up and Hessian is completely thrown off.
Hessian LLE code crashed frequently on this example.
Diffusion maps handle it quite well for carefully chosen Sigma=0.3.

Sparse Data & Non-uniform Sampling

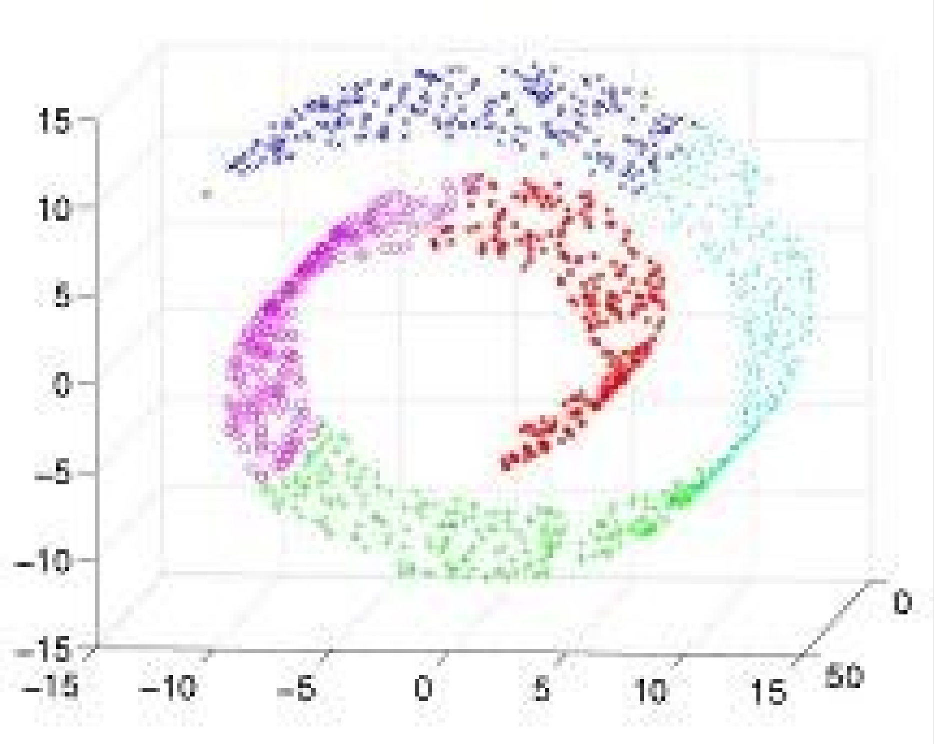
- Of course, we want as much data as possible. But can the method handle sparse regions in the data?
- Punctured Sphere: the sampling is very sparse at the bottom and dense at the top.

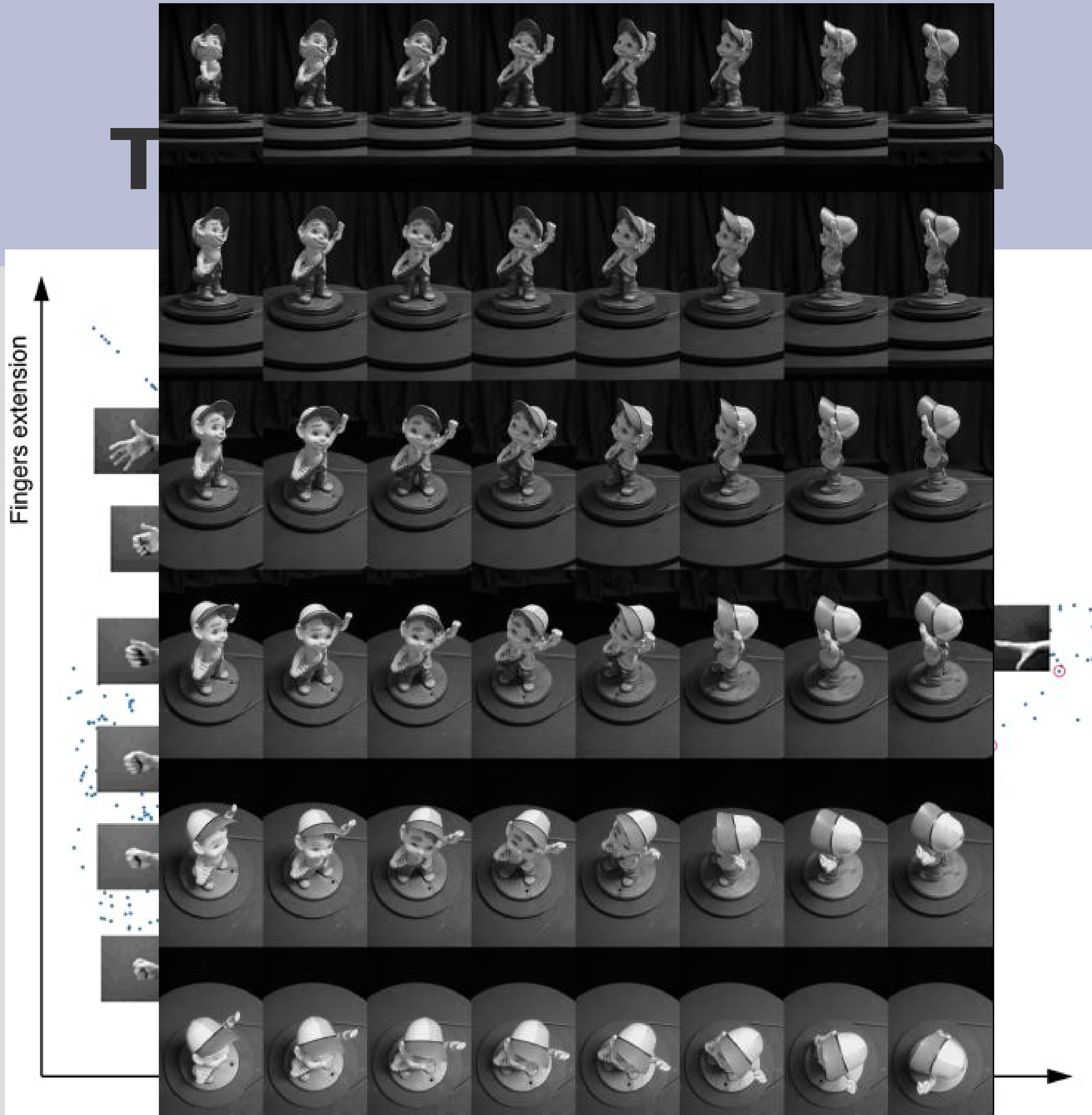




Only LLE and Laplacian get decent results.
PCA projects the sphere from the side. MDS turns it inside-out.
Hessian and Diffusion Maps get correct shape, but give too much emphasis to the sparse region at the bottom of the sphere.

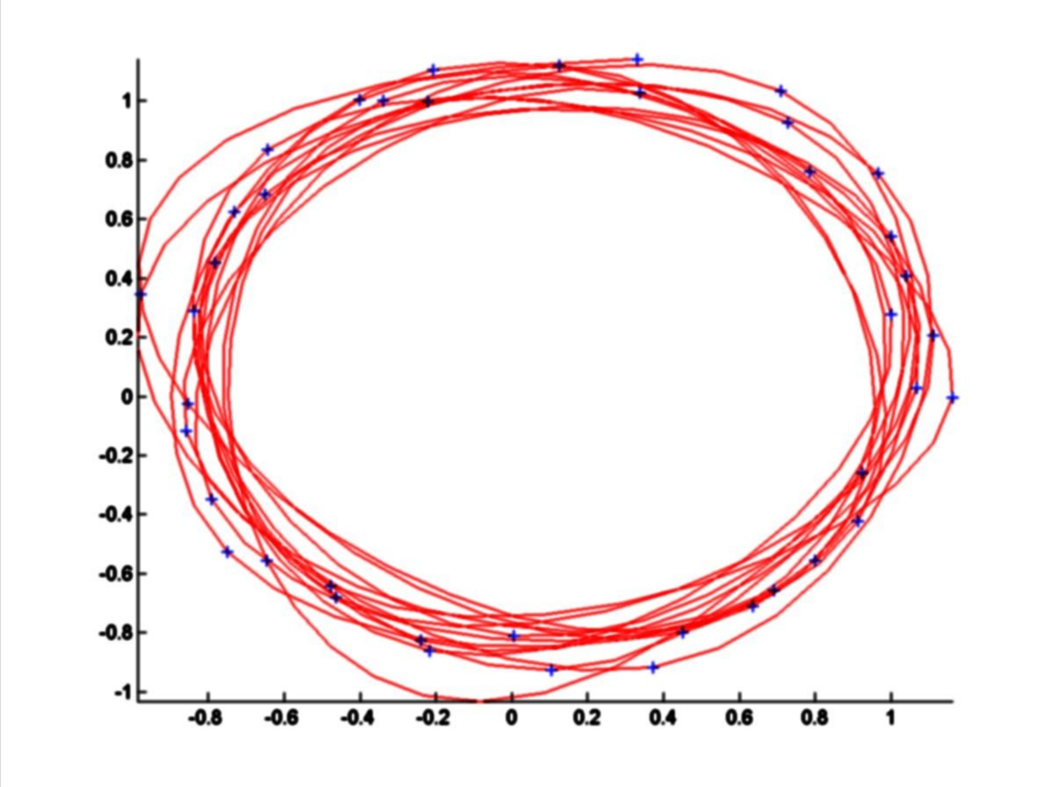
Incremental ISOMAP

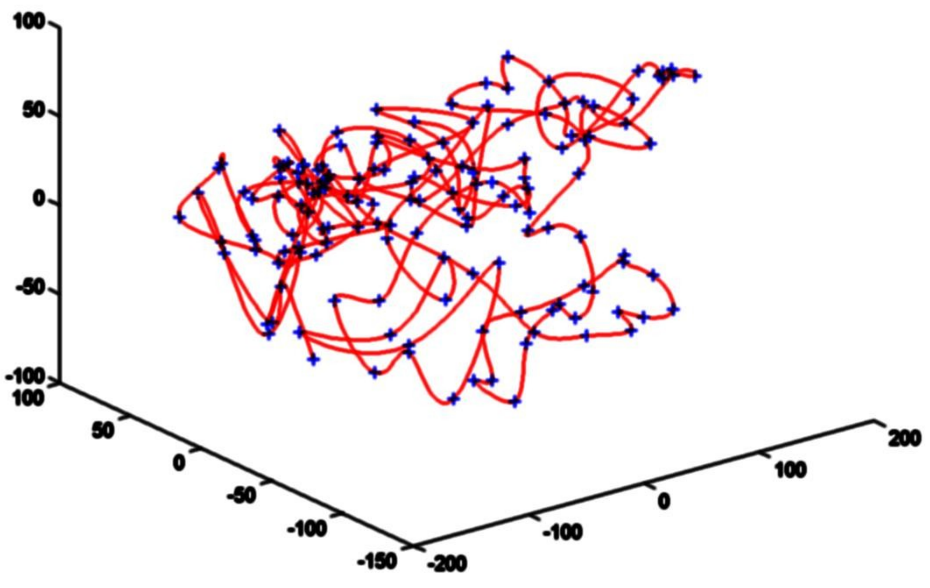



Fingers extension

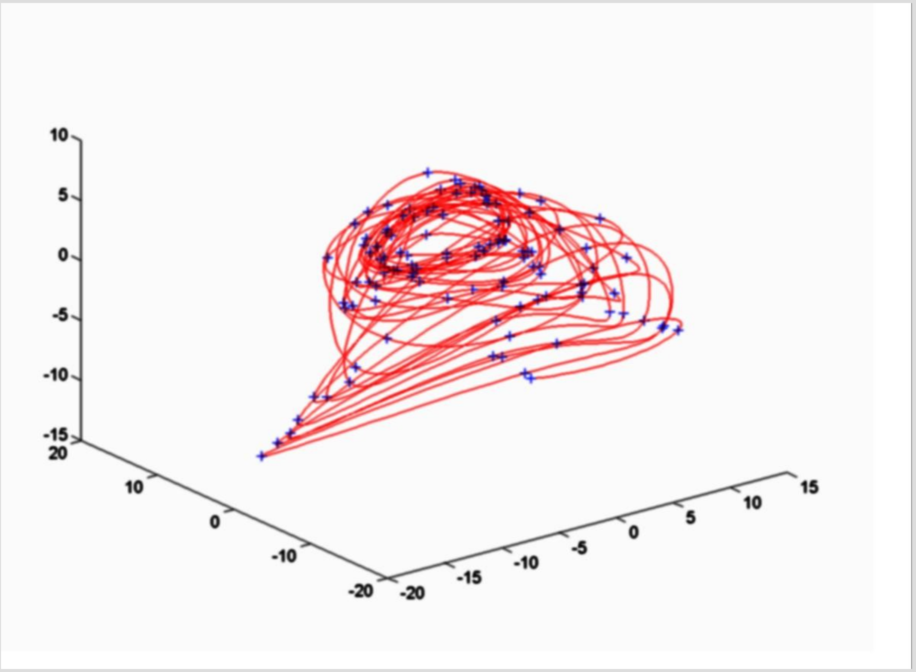
Video Textures

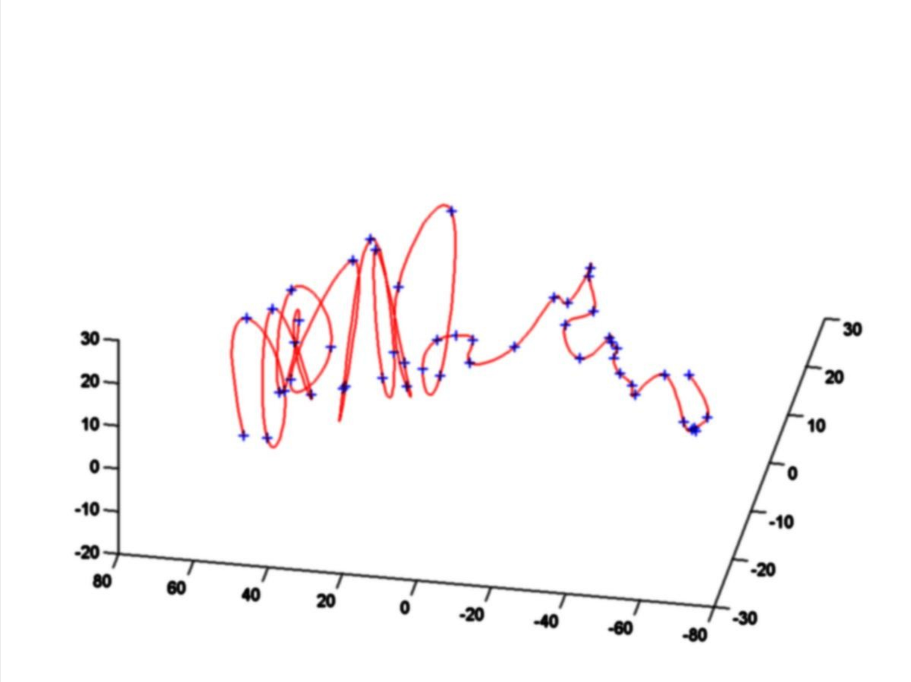
Wumap


1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1


100
50
0
-50
-100
100
50
0
-50
-100
-150
-200
-100
0
100
200

Wumap – Resampling





Wumap - Resampling