# Topology Structure Analysis of High Dimensional Dataset by Flattening Deformation of Data Manifold

Xiaodong Zhuang[1], Nikos E. Mastorakis[2]
[1]Electronic Information College, Qingdao University, 266071 China (xdzhuang@qdu.edu.cn)
[2]Technical University of Sofia, Industrial Engineering Department, Kliment Ohridski 8, Sofia, 1000 Bulgaria (mastor@tu-sofia.bg)

Received: June 29, 2021. Revised: November 12, 2021. Accepted: November 23, 2021. Published: December 21, 2021.

**Abstract—A new analysis method for high dimensional sets is proposed by autonomous deforming of data manifolds. The deformation is guided by two kinds of virtual interactions between data points. The flattening of data manifold is achieved under the elastic and repelling interactions, meanwhile the topological structure of the manifold is preserved. The proposed method provides a novel geometric viewpoint on high-dimensional data analysis. Experimental results prove the effectiveness of the proposed method in dataset structure analysis.**

**Keywords—Data analysis, high dimension, data manifold, autonomous deformation**

## I. Introduction

IN current big data applications, a large part of the datasets are of high dimension such as images, videos, texts of great length, etc. It is a challenge to reveal useful underlying information in the datasets [1-6]. To face this challenge, manifold learning is one of the main non-linear methods, which provides a geometric viewpoint that the data set is regarded as samples from a data manifold. Manifold-based methods have attracted extensive research attention since the publication of the Isometric Mapping and Local Linear Embedding methods [7-14]. The existing methods usually have the form of solving optimization or linear programming under certain constraints. Although these learning methods have achieved impressive experimental results, it has been pointed out that current manifold learning methods may fail because of the extremely high dimension or high local curvature of the data manifold [15-17]. Another problem may also invalidate manifold learning that the practical data sets usually do not satisfy the ideal precondition of sufficiently dense and uniform sampling on the manifold.

In this paper, a novel method is proposed from the geometric viewpoint, which reveals the dataset topology structure by the "flattening" of the manifold in the embedding space. In the proposed method, the data manifold (in a discrete form) deforms in an autonomous self-evolution way under the virtual interaction between the data points. The flattened manifold can naturally represents the dataset topology structure, and the deforming result can naturally indicate the intrinsic dimension of the manifold. The experimental results prove the effectiveness of the proposed method.

## II. The Two Virtual Interactions between Data Points

The proposed method introduces two virtual interactions between data points, which cause the deformation of the data manifold. By proper design of the interactions, the flattening of the manifold occurs as an emergence effect. The design of the proposed method is inspired by the viewpoint of geometrically interpreting data topology structure analysis as the flattening of data manifold. In the proposed method, two virtual interactions between data points are proposed to derive an autonomous deforming process. They are the repelling and elastic interactions. The repelling vector from the data point $p_j$ to $p_i$ is defined as:

$$\vec{V}_{ij}^{r} = \begin{cases} \dfrac{(\vec{p}_i - \vec{p}_j)}{d_{ij}} & p_j \in N_i \\ \dfrac{\vec{p}_i - \vec{p}_j}{d_{ij}} & otherwise \end{cases} \quad (1)$$

where $\vec{p}_i$ and $\vec{p}_j$ are the position vectors of data points $p_i$ and $p_j$ in $R^n$. $d_{ij}$ is the distance between the two points in the deforming process. $N_i$ is the neighborhood of $p_i$. Because the vector $(\vec{p}_i - \vec{p}_j)$ points from $p_j$ to $p_i$, if $p_i$ moves along this direction it will move away from $p_j$. Therefore the vector defined in Equation (1) has a repelling effect between $p_i$ and $p_j$.

On the other hand, the elastic interaction vector between $p_i$ and $p_j$ is defined as:

$$\vec{V}_{ij}^{e} = \begin{cases} \dfrac{(d_{ij}^{0} - d_{ij}) \cdot (\vec{p}_i - \vec{p}_j)}{d_{ij}} & p_j \in N_i \\ 0 & otherwise \end{cases} \quad (2)$$

where $d_{ij}^{0}$ is the Euclidean distance between $p_i$ and $p_j$ on the original manifold (before deforming). $d_{ij}$ is the distance between $p_i$ and $p_j$ in the deforming process. Therefore, the interaction vector defined in Equation (2) will alter according to the manifold shape in the deforming process. For point $p_i$, this elastic interaction only exists for the points in the neighborhood $N_i$. For point $p_j$ in $N_i$, if it comes closer to $p_i$ in deforming (i.e. the current distance $d_{ij}$ is smaller than the original value $d_{ij}^0$), it will repel $p_i$, otherwise it will attract $p_i$. This is just an elastic effect which functions as preserving the distance between neighbor points (i.e. keep the neighbor structure in the deforming process).

The total interaction on $p_i$ from all the other data points is the weighted sum of the above two interactions:

$$\vec{V}_i = \alpha_1 \cdot \sum_{\substack{j=1 \\ j\neq i}}^{N} \vec{V}_{ij}^r + \alpha_2 \cdot \sum_{\substack{j=1 \\ j\neq i}}^{N} \vec{V}_{ij}^e \qquad (3)$$

where $N$ is the number of data points. $\alpha_1$ and $\alpha_2$ are two weight coefficients that balance the two kinds of interactions. $\vec{V}_i$ is defined as the deforming vector on $p_i$. Because $\vec{V}_i$ is entirely determined by the current position of all data points (i.e. the manifold itself), and no external influence is involved, this vector field on the manifold is intrinsic. If each $p_i$ moves according to $\vec{V}_i$ (i.e. take $\vec{V}_i$ as the displacement vector), one step of manifold deforming will happen. Moreover, if the step repeats, the deformation of data manifold will proceed step by step. Due to the intrinsic nature of the proposed vector field, the deformation under it is a kind of self-evolution of the manifold. With the two different kinds of interactions in Equation (1) and (2), the deforming process will converge to a "flattened" result, which can naturally derive the analysis result.

## III. The Algorithm

Based on the above definitions, the proposed algorithm is as follows.

**Step1**: Calculate the Euclidean distance $d_{ij}$ between each pair of data points in the data set.

**Step2**: Find the $k$ nearest neighbor points for each data point as its neighborhood point set $N_i$.

**Step3**: Initialize the count of deforming steps $C$ as zero.

**Step4**: For each data point $p_i$, calculate the current displacement vector $\vec{V}_i$ according to the current position of each point (i.e. the current manifold shape) in $R^n$.

**Step5**: Update the position of each point $p_i$ with the displacement vector $\vec{V}_i$

**Step6**: Increase $C$ by 1.

**Step7**: Check whether the termination condition is satisfied (the sum of each point's displacement is smaller than a threshold $\varepsilon$, or $C$ reaches a given value). If not, return to Step 4. Otherwise, go to **Step 8**.

**Step8**: Perform Principle Component Analysis (PCA) on the flattened manifold, and obtain the final analysis result (the estimated intrinsic dimension of manifold is the number of principle components, and the low-dimension coordinate of each data point is the projection on the principle component vectors).

The above algorithm first flattens the manifold in $R^n$, and then PCA is used to extract manifold dimension and the dataset topology structure, because the manifold has already deformed to a fairly flat geometry.

## IV. Simulation Results

The proposed method is implemented by programming simulation. Experiments have been done on simple test data sets, and also practical data sets in real world applications. Some results and analysis are as follows.

Preliminary experiments have been done for typical types of surfaces in $R^3$. Some results are shown in Fig. 1 to Fig. 3 for the half cylinder side face.

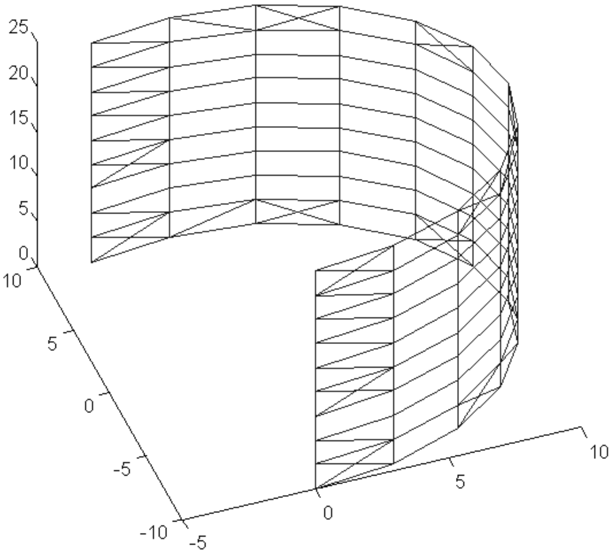

Fig. 1 The mesh of the discrete cylinder side face

Fig. 1 shows the mesh of the discrete cylinder side face, which has 120 data points. Because the neighborhood relationship is the basis of manifold topology structure, the results consist of nodes for data points, and edges for the representation of neighborhood relationship. The nodes represent the data points, and each edge connects two neighbor points in the data set.

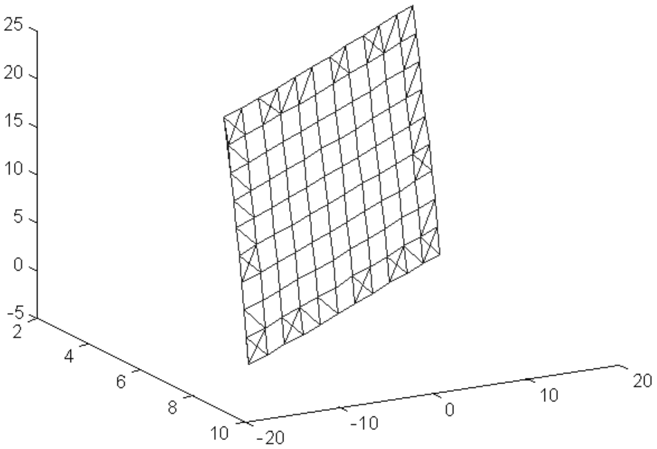

Fig. 2 The deforming result of Fig. 1 in $R^3$

Fig. 2 shows the deforming result of Fig. 1 in $R^3$. In the deforming result, the cylinder side face is totally flattened. Fig. 3 shows the final analysis result in $R^2$, which is the result of

PCA on Fig. 2. In Fig. 3, each node point represents the data after analysis, and each edge connects two data points which correspond to neighbor points in the original data set. Moreover, the length of the edge represents the Euclidean distance between the two neighbor data points. The nodes in Fig. 3 are numbered, which may facilitate further analysis. In this way, the analysis result and the topology structure of data can be clearly demonstrated. In Fig. 3, the data points are evenly distributed in $R^2$, which corresponds to the evenly sampling of the cylinder side face shown in Fig. 1. It proves the effectiveness of the proposed method on the type of curling surface.

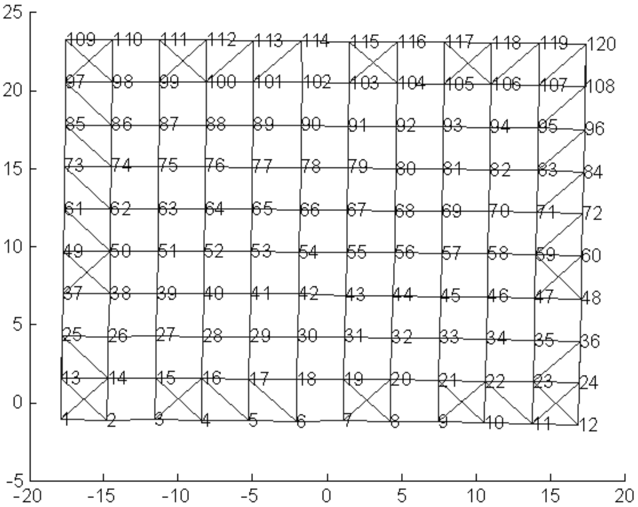

Fig. 3 The final structure analysis result of Fig. 1 in $R^2$

Fig. 4 and Fig. 5 show the experimental results on an image set from the "Object Pose Estimation Database" [18]. This image set of auto fuse is captured under different horizontal and vertical viewpoints. The image data set is from Internet [19]. Fig. 4 shows the data set with a number assigned to each image. Each image is of the size 72×90, which is a data vector of dimension 6480 in the embedded space of the data manifold. The analysis result is shown in Fig. 5. The intrinsic dimension of this image set is estimated as two. Each node point in Fig. 5 represents an image with the same number in Fig. 4. The edges connect the pairs of nodes corresponding to the neighbor data points in the original data set. The result reveals that the images change along two different dimensions. The first dimension is the $x$-axis in Fig. 5, which corresponds to the change of horizontal viewpoint. The second one is the $y$-axis in Fig. 5, which corresponds to the change of vertical viewpoint. It should be noted that the points at the lower right corner in Fig. 5 are much closer, because the method preserves the distance between neighbor points, and those distances are relatively small in the original image data set. Therefore, the topology structure of the image set is effectively extracted and represented by the result in Fig. 5.

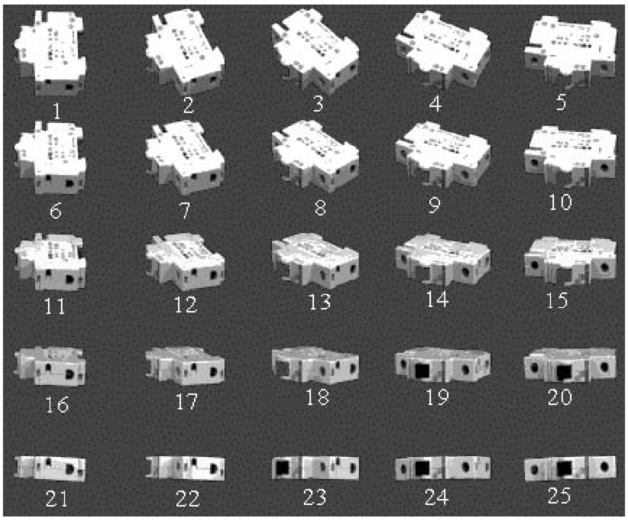

Fig. 4 The image set of auto fuse

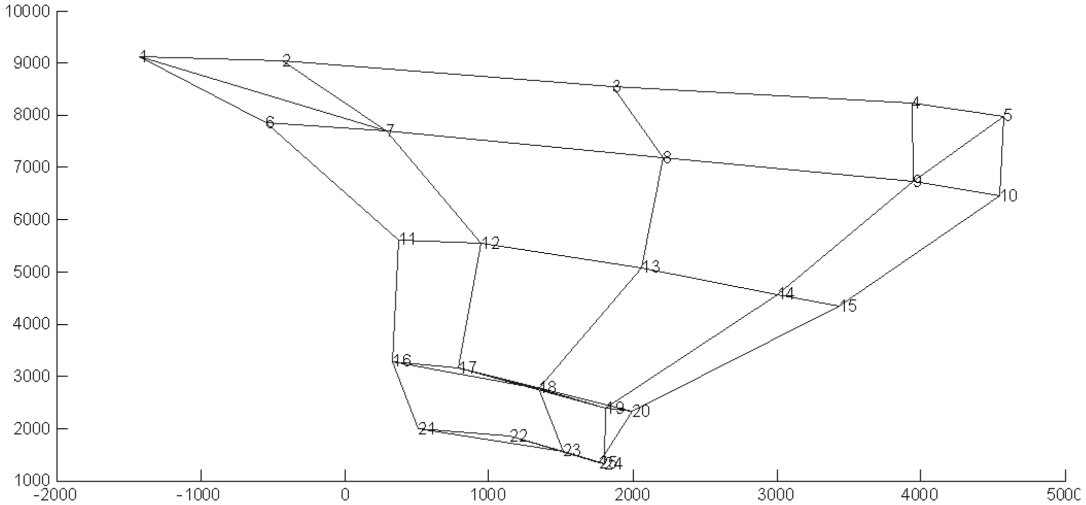

Fig. 5 The final structure analysis result of the image set of auto fuse in $R^2$

## V. Conclusion

In this paper, a novel dataset analysis method is proposed by the autonomous deforming (self-evolution) of data manifolds. The deforming is guided by the proposed deforming vector field including two kinds of interactions between data points. The elastic interaction preserves the topological structure of the data manifold, while the repelling interaction stretches and spreads the manifold. The flattening of the manifold in $R^n$ can be achieved as a result of data point interactions. The experiment results on cylindrical surface prove that the proposed method can effectively flatten the bending data manifold. The experimental results on real-world data sets prove that effective topology structure analysis can be achieved by the proposed method, the intrinsic dimension can be revealed, and there are meaningful interpretations for the analysis results. Further study will investigate detailed characteristics of the final stable shape of the deforming manifold.

## References

[1] D. L. Donoho, High-Dimensional Data Analysis: The Curses and Blessing of Dimensionality. Proceedings of AMS Math-Ematical Challenges of the 21st Century, 2000.

[2] Muhammad Habib ur Rehman, Chee Sun Liew, Assad Abbas, Prem Prakash Jayaraman, Teh Ying Wah & Samee U. Khan . Big Data Reduction Methods: A Survey. Data Science and Engineering 1, 265–284 (2016).

[3] Daniel Engel, Lars Huttenberger, Bernd Hamann, A Survey of Dimension Reduction Methods for High-dimensional Data Analysis and Visualization, Visualization of Large and Unstructured Data Sets: Applications in Geospatial Planning, Modeling and Engineering - Proceedings of IRTG 1131 Workshop 2011, pp. 135-149.

[4] Xuan Huang, Lei Wu , Yinsong Ye, A Review on Dimensionality Reduction Techniques, International Journal of Pattern Recognition and Artificial IntelligenceVol. 33, No. 10, 1950017 (2019).

[5] Shaeela Ayesha, Muhammad Kashif Hanif, Ramzan Talib, Overview and comparative study of dimensionality reduction techniques for high dimensional data, Information Fusion, Vol. 59, 2020, pp. 44-58.

[6] S. Velliangiri, S. Alagumuthukrishnan, S. Iwin Thankumar joseph, A Review of Dimensionality Reduction Techniques for Efficient Computation, Procedia Computer Science, Volume 165, 2019, pp. 104-111.

[7] Joshua B. Tenenbaum, Vin de Silva, John C. Langford, A Global Geometric Framework for Nonlinear Dimensionality Reduction, Science, Vol. 290, Issue 5500, pp. 2319-2323 (2000).

[8] Sam T. Roweis, Lawrence K. Saul, Nonlinear Dimensionality Reduction by Locally Linear Embedding, Science, Vol. 290, Issue 5500, pp. 2323-2326 (2000).

[9] Mikhail Belkin, Partha Niyogi, Laplacian Eigenmaps for dimensionality reduction and data representation, Neural Computation, Vol. 15, Issue 6, June 2003 pp. 1373–1396.

[10] K. Q. Weinberger, L. K. Saul, Unsupervised learning of image manifolds by semidefinite programming, Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004, pp. II-II.

[11] Tong Lin, Hongbin Zha, Riemannian Manifold Learning, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 30, Issue 5, May 2008, pp. 796–809.

[12] Yee Whye Teh, Sam Roweis, Automatic alignment of local representations, Proceedings of the 15th International Conference on Neural Information Processing Systems, January 2002, pp. 865–872.

[13] Laurens van der Maaten, Geoffrey Hinton, Visualizing Data using t-SNE, Journal of Machine Learning Research, Vol. 9, No. 86, pp.2579-2605, 2008.

[14] Zhenyue Zhang, Hongyuan Zha, Principal Manifolds and Nonlinear Dimensionality Reduction via Tangent Space Alignment, SIAM Journal on Scientific Computing, 26(1), pp. 313–338.

[15] Bengio Y, Larochelle H, Vincent P, Non-local manifold parzen windows Advances in Neural Information Processing Systems, pp. 115-122, 2005.

[16] Bengio Y, Monperrus M, Larochelle H, Nonlocal estimation of manifold structure, Neural Computation, 18: 2509-28, 2006.

[17] Bengio Y, Monperrus M. Non-local manifold tangent learning. In: Proceedings of Advances in Neural Information Processing Systems, 2005, 17: pp. 129–136.

[18] F. Viksten, P.-E. Forssen, B. Johansson, A. Moe, Comparison of Local Image Descriptors for Full 6 Degree-of-Freedom Pose Estimation, IEEE International Conference on Robotics and Automation, May 2009.

[19] https://www.cvl.isy.liu.se/research/objrec/posedb/

## Contribution of individual authors to the creation of a scientific article (ghostwriting policy)

Xiaodong Zhuang designed the algorithm and carried out the simulation.
Nikos E. Mastorakis helped the preparation and improvement of the manuscript writing.