# An Extended Isomap for Manifold Topology Learning with SOINN Landmarks

Qiang Gan, Furao Shen*, Jinxi Zhao

***Abstract*—This paper presents an extended Isomap algorithm called SL–Isomap (SOINN Landmark Isomap). We adopt SOINN (Self–Organizing Incremental Neural Network) algorithm to choose the reasonable number of landmarks automatically. SOINN landmarks are able to represent topological structure of unsupervised data in the high dimensional input space. Then L–Isomap (Landmark Isomap) algorithm is used to find low dimensional manifolds from high dimensional data based on chosen landmarks. SL–Isomap solves the problem of selecting the right number and position of landmarks automatically thus reduces short–circuit errors. It also realizes data compression and nonlinear dimensionality reduction at the same time. Experiments demonstrate its promising results compared with other variants of L–Isomap.**

## I. Introduction

Dimensionality reduction algorithm consists of two parts essentially: distance calculation based on similarity in high dimensional space and dimensionality reduction based on specific goals. Traditional distance calculation (the first part) mainly uses Euclidean distance, but the goals for dimensionality reduction (the second part) are varied. For unsupervised example, PCA (Principal Component Analysis) is to maximize the variance of data mapped in the low dimensional space and MDS (Multidimensional Scaling) is to minimize the difference between corresponding distances in high and mapped low dimensional space. For supervised examples, FLD (Fisher Linear Discriminant) is to maximize the between–class distance and minimize the within–class distance mapped in low dimensional space at the same time.

Isomap [1] is the first to introduce geodesic distance to measure the similarity between data in high dimensional space. It has demonstrated its excellent results in finding meaningful low dimensional structures hidden in the high dimensional input space. As showed in Fig. 1 [1], for two points (circled) on a nonlinear manifold, their Euclidean distance in the high dimensional input space (length of dashed line) may not accurately reflect their intrinsic similarity, as measured by geodesic distance along the low–dimensional manifold (length of solid curve). Isomap replaces the Euclidean distance calculation in the first part of MDS with geodesic distance calculation and becomes a new algorithm with profound influence in manifold learning. Variants of Isomap for different applications has been continuously emerging until now. Ming-Hsuan Yang [2] [3]extends Isomap with FLD for face images classification , Zhan Dechuan and Zhou Zhihua [4] introduce ensemble learning technique to Isomap for visualization, Fengyi Lin et al. [5] combine Isomap with SVM in the prediction of business failure and Yu-Ming Liang et al. [6] combine Isomap with DTW (dynamic time warping) for human action segmentation and classification.

Qiang Gan(Email:njucsgq@gmail.com),Furao Shen(corrresponding author, Email:frshen@nju.edu.cn) and Jinxi Zhao(Email:jxzhao@nju.edu.cn) are with the Robotic Intelligence and Neural Computing Laboratory(RINC.Lab), National Key Laboratory for Novel Software Technology, and Department of Computer Science and Technology, Nanjing University, China.

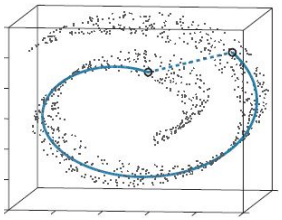


Fig. 1. Euclidean distance vs. Geodesic distance

In order to address the inefficiencies of Isomap, Silva and Tenenbaum proposes L–Isomap [7] which introduces geodesic distance to LMDS (Landmark MDS) [8]. L–Isomap consists of four steps. Firstly, $n$ landmarks are chosen randomly. Then apply MDS on $n$ landmarks and get the low dimensional embedding. Next, geodesic distances between $n$ landmarks and the rest $N-n$ data points are calculated through Dijkstra algorithm. Finally, low dimensional embedding of the rest data points are calculated based on landmarks through LMDS algorithm. Although L–Isomap works well in some databases, its performance is sensitive to the number and position of landmarks. If those two factors are not determined well, it will expand short–circuit errors. Many people use clustering algorithms in order to determine the position of landmarks better. Lukui Shi et al. [9] use SOM (Self–Organizing Map) to choose landmarks and Chi et al. [13] use k–means clustering to select landmarks. However, extended variants of L–Isomap are still not able to determine the reasonable number of landmarks to represent topological structure of data points from high dimensional space.

The proposed algorithm SL–Isomap (SOINN Landmark Isomap) uses SOINN (Self–Organizing Incremental Neural Network) to select landmarks and runs L–Isomap to get low dimensional embedding of high dimensional data points. SOINN algorithm proposed by Shen Furao et al. [11] [12] is able to represent the topological structure of unsupervised on–line data, report the reasonable number of clusters, and give typical prototype patterns of every cluster without prior conditions. It solves the problem of selecting the right number and position of landmarks. SL–Isomap combines the advantages of

data compression from SOINN and nonlinear dimensionality reduction from Isomap. Compared with other variants of L–Isomap, experiments of SL–Isomap demonstrate its promising results on swiss_roll_data set, swiss_roll_data set with Gauss noise and AT&T face database .

## II. SL–ISOMAP

### A. Motivation

Due to the sparse distribution of landmarks, L–Isomap with inappropriate position or number of landmarks will magnify short–circuit errors. Essentially it is equivalent to the situation with a too large neighborhood [15].

Fig. 2 shows the importance of the position of landmarks. Fig. 2(a) and Fig. 2(b) has the same number of landmarks. Obviously, landmark $L_1$, $L_2$ and $L_3$ in Fig. 2(a) represent the fold in the manifold more reasonably. The latter "eliminates" the fold completely through direct links between landmarks.

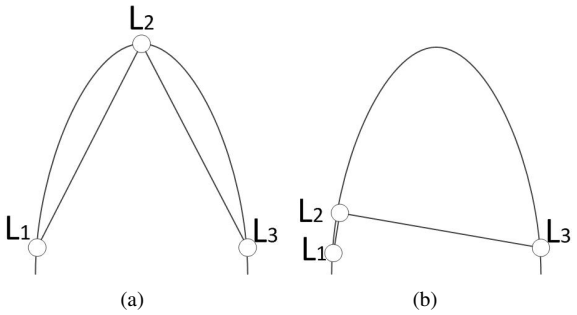



Fig. 2. The importance of the position of landmarks

Fig. 3 shows the importance of the number of landmarks. If we select too many landmarks, L–Isomap reduces to Isomap. However, if the number of landmarks is too small (Fig. 3(b)), direct link between $L_1$ and $L_2$ causes severe short–circuit and cannot represent the fold in the manifold. This problem need to be solved by adding a landmark (Fig. 3(a)).

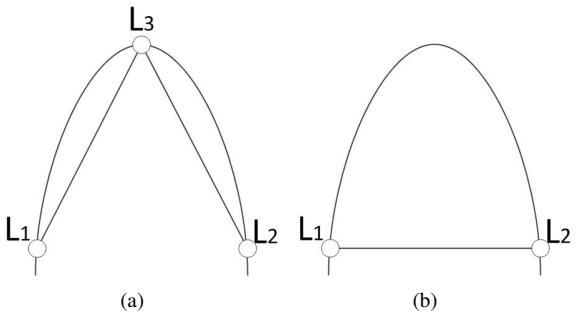



Fig. 3. The importance of the number of landmarks

As discussed above, the number and position of landmarks play a key role in L–Isomap. Random number and position of landmarks in [7] is obviously not a good solution. Although the determination of position can be optimized through clustering algorithm such as SOM [9] and k–means [13], the number of landmarks is still determined by users, which is highly demanding for users and unreliable in many situations.

The proposed SL–Isomap is able to solve the problem. It determines the reasonable number and position of landmarks automatically.

### B. SL–Isomap Algorithm

Consider a set of $N$ data points $\{x_1, x_2, ..., x_N\}$ from high dimensional input space. SL–Isomap consists of four steps. Algorithm for the SL–Isomap algorithm is described in Algorithm 1. Details of LMDS algorithm are in [7] and [8].

**Algorithm 1** SL–Isomap

1: **Selecting the SOINN landmarks**
Use SOINN to select landmarks $\{x_{l_1}, x_{l_2}, ..., x_{l_n}\}$ from high dimensional input space.
The number of landmarks $n$ is determined by SOINN automatically without any prior knowledge.

2: **MDS on SOINN landmarks**
Apply MDS on the landmarks-only distance matrix $D_G$ $(n * n)$, and get low–dimensional coordinates of $n$ landmarks.

3: **LMDS based on SOINN landmarks**
Use Dijkstra algorithm to compute shortest path matrix $D_G{}'$ $(n * N)$ between landmarks and data points.
Then construct a low dimensional embedding of the rest $N - n$ data points through LMDS.

4: **Normalization**
Use PCA to reorient the axes to reflect the distribution of all data points, instead of the much smaller distribution of SOINN landmarks.

### C. Step1: Selecting the SOINN landmarks

We adopt SOINN to choose landmarks. It determines the reasonable number and position of landmarks automatically.

1) Initialization
   - The output node set A is initialized to contain two nodes, $x_1$ and $x_2$: $A = \{x_1, x_2\}$.
   - Local accumulated number of signals $M_{x_1}$ and $M_{x_2}$ is initialized to 1.
   - Their thresholds $T_{x_1} = T_{x_2} = D_{E(1,2)}$
   - Their connection set $C$ is set to $\emptyset$ and their connection age $age_{(1,2)}$ is set to 0.

2) Input and Find Winner.
Input data point $x_i (i \in [3, N])$ one by one. Search node set A to determine the winner $s_1$, and second-nearest node (second winner) $s_2$ by

$$s_1 = \arg\min_{n_c \in A} \|x_i - n_c\|$$
$$s_2 = \arg\min_{n_c \in A \setminus \{s_1\}} \|x_i - n_c\|$$

3) Nodes Update

- If $\|x_i - n_{s_1}\| > T_{s_1}$ or $\|x_i - n_{s_2}\| > T_{s_2}$ , that means $x_i$ is a new node, then $A = A \cup x_i$ and go to 2).
- If a connection between $s_1$ and $s_2$ does not exist already, create it and add it to connection set $C$. Set the age of the connection $age_{(s_1,s_2)}$ to 0.
- Add 1 to the age of all edges emanating from $s_1$.
- Increase the local accumulated number of signals $M_{s_1}$ by 1.
- Adapt the winner $s_1$ to input data $x_i$ by a certain fraction $\varepsilon$.
- Remove edges with an age greater than a predefined threshold age $age_{dead}$ and delete its connection from $C$.

4) Thresholds Update
Update the threshold $T_{s_1}/T_{s_2}$ to the largest distance between $s_1/s_2$ and its neighbors.
5) Noise Elimination
For all nodes in $A$, if a node $n_i$ has only one neighbor and $M_{n_i}$ is less than an adaptive threshold, remove it from the node set.

After the above processes, we get the node set $A = \{n_1, n_2, \cdots, n_n\}$. The number of landmarks $n$ is determined by SOINN automatically without any priori knowledge. N-earest neighbors of $A$ in $Data$ is exactly the landmark set $LM = \{x_{l_1}, x_{l_2}, ..., x_{l_n}\}$ we need. $x_{l_i}$ in $LM$ is calculated by:

$$x_{l_i} = \arg\min_{x_c \in Data} \|n_i - x_c\|$$

We take data points in $LM$ as landmarks because raw results of SOINN may move slightly off the manifold. It happens when adapting winner $s_1$ to the input data.

### D. Step2: MDS on SOINN landmarks

This step applies MDS on $n$ landmarks that are selected in the first step. The landmarks-only distance matrix is denoted by $D_G$.

Construct matrix $B_n = -H_n \Delta_n H_n/2$. $\Delta_n$ is the matrix of squared $D_G$ and $H_n$ is defined by $(H_n)_{ij} = \delta_{ij} - 1/n$. We write $\lambda_i$ for the $i$th biggest eigenvalues of $B_n$ and $\overrightarrow{v_i}$ for the corresponding eigenvector. The $l$–dimensional coordinates of $n$ landmarks are given as the columns of matrix $L$:

$$L = \begin{bmatrix} \sqrt{\lambda_1} \cdot \overrightarrow{v_1}^T \\ \sqrt{\lambda_2} \cdot \overrightarrow{v_2}^T \\ \vdots \\ \sqrt{\lambda_l} \cdot \overrightarrow{v_l}^T \end{bmatrix}$$

### E. Step3: LMDS based on SOINN landmarks

We compute embedding coordinates for the remaining data points based on their distances from the SOINN landmarks.

This step includes:

1) Run Dijkstra algorithm $n$ rounds to calculate single-source shortest path matrix $D'_G$ $(n*N)$. $D'_G$ $(n*N)$ is the approximation of geodesic distance between landmarks and remaining data points. In each round of Dijkstra algorithm, $n$ landmarks acts as the source point one by one.
2) Construct a low dimensional embedding through LMDS: Let $\Delta_x$ denote the column vector of squared distances between data point $x$ and $n$ landmarks (one column vector in squared $D'_G$). The embedding of $x$ is calculated by:

$$\overrightarrow{x} = \frac{1}{2} L^{\#} (\overline{\Delta_n} - \Delta_x)$$

where $\overline{\Delta_n}$ is the column mean of $\Delta_n$ and $L^{\#}$ is:

$$L^{\#} = \begin{bmatrix} \overrightarrow{v_1}^T/\sqrt{\lambda_1} \\ \overrightarrow{v_2}^T/\sqrt{\lambda_2} \\ \vdots \\ \overrightarrow{v_l}^T/\sqrt{\lambda_l} \end{bmatrix}$$

### F. Step4: Normalization

Note that the axes are determined by landmarks $\{x_{l_1}, x_{l_2}, ..., x_{l_n}\}$ in step2. They may not reflect the overall distribution of all data points in $Data$.

We need to use PCA to reorient the axes to reflect the overall distribution of $\{x_1, x_2, ..., x_N\}$, after the full data set $Data$ has been embedded in low dimensional space.

## III. Experiments

In this section, we demonstrate the results of SL–Isomap on three data sets compared with other variants of L–Isomap. All experiments are implemented in MATLAB R2010a on a PC with 3.2GHz CPU. L–Isomap algorithm is available in http://isomap.stanford.edu/. Note that SOINN is an on–line incremental algorithm, and the input order of $Data$ may affect the result. Consider a set of $N$ data points $Data = \{x_1, x_2, ..., x_N\}$ from high dimensional input space. Our experiments include five steps:

1) Compute the residual variance of SL–Isomap to determine the intrinsic dimensionality $d$ of $Data$;
2) Run SOINN on high dimensional data set to determine the reasonable number $n$ of landmarks;
3) Run SL–Isomap to get the $d-$dimensional embedding of $Data$;
4) Take $n$ as a parameter to run L–Isomap based on Random/Kmeans/SOM landmarks and get the $d --$dimensional embedding of $Data$;
5) Calculate the distance error $DisErr$:

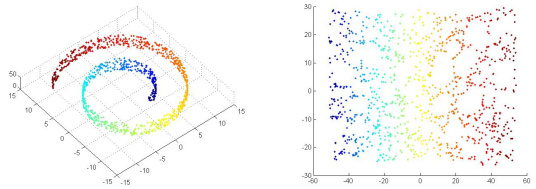
$$DisErr = \sum_i^N \sum_j^N \left| D_{G(i,j)} - D_{LE(i,j)} \right|$$

$D_{G(i,j)}$ is the geodesic distance between $x_i$ and $x_j$ in high dimensional input space.
$D_{LE(i,j)}$ is the Euclidean distance between $x_i$ and $x_j$ in their $d$–dimensional embedding.
$DisErr$ is more accurate for error analysis in two aspects. It not only corresponds to the cost function of MDS, but also measures the short–circuit errors.

While in [9], error function calculates the difference between low dimensional coordinates out of Isomap and L-Isomap, which is less accurate because Isomap has its inherent errors.

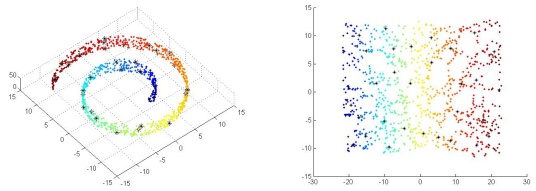
### A. *Swiss_roll_data*

In this experiment, we test the performance of SL–Isomap in data set swiss_roll_data. Swiss_roll_data is one of the most commonly used data set in manifold learning. The input data set $Data$ consists of 1000 data points from the swiss_roll_data (Fig. 4).



(a) (b)

Fig. 4. (a) Swiss_roll_data and (b) its 2–dimensional embedding by Isomap (k=7)



(a) (b)

Fig. 5. (a) Swiss_roll_data and (b) its 2–dimensional embedding by SL–Isomap (k=7)

Firstly, we need to know the reasonable number of landmarks. We run SOINN 500 times on $Data$ and the number of landmarks concentrates around 90.

We test SL–Isomap against L–Isomap based on random, k–means and SOM landmarks. As illustrated above, the input order of $Data$ may affect the result of SOINN, thus the number of landmarks of SL-Isomap is automatically generated to $90 \pm 20$. The 2–dimensional embedding of SL–Isomap is shown in (Fig. 5(b)). Black star points in Fig. 5 are SOINN landmarks. In other methods, the number of landmarks is set to 90.

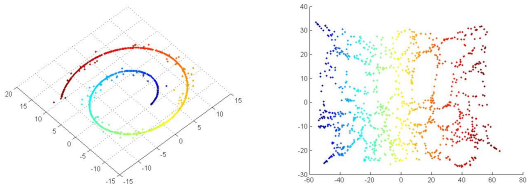
Table I. shows the mean $DisErr$ of 100 repeated experiments by SL–Isomap against other methods. The result shows that SL–Isomap performs better in preserving the similarity between data points in high dimensional space than other variants of L–Isomap on swiss_roll_data set. That also means SL–Isomap reduces more short–circuit errors.

### B. *Swiss_roll_data with Gauss noise*

Isomap algorithm is vulnerable to short–circuit errors if noise in the data moves the points slightly off the manifold [15]. The solution proposed by Tenenbaum is to decrease the neighborhood parameter $k$ from 7 to 5. In this experiment, we want to know whether landmark methods are vulnerable to short–circuit errors, and the performance between SL–Isomap and other methods in noise environment. We take 1000 data points from the swiss_roll_data together with 100 points of Gauss noise (Fig. 6) to be the input data set $Data$.

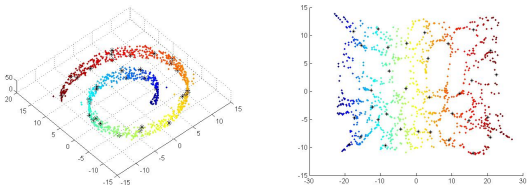
TABLE I
DISERR ON SWISS_ROLL_DATA (K=7)

| Method | Number of Landmarks | DisErr |
|---|---|---|
| Random Landmark Isomap | 90 | 19.8622 |
| Kmeans Landmark Isomap | 90 | 19.4859 |
| SOM Landmark Isomap | 9*10 | 19.3545 |
| SOINN Landmark Isomap | 90±20 | 19.3502 |



(a) (b)

Fig. 6. (a) Swiss_roll_data with Gauss noise and (b) its 2–dimensional embedding by Isomap (k=5)



(a) (b)

Fig. 7. (a) Swiss_roll_data with Gauss noise and (b) its 2–dimensional embedding by SL–Isomap (k=5)

The 2–dimensional embedding of SL–Isomap is shown in (Fig. 7(b)). SOINN landmarks are represented with black star points in Fig. 7. The number of landmarks of SL-Isomap is automatically generated to $90 \pm 20$. In other methods, the number of landmarks is set to 90.

TABLE II
DISERR ON SWISS_ROLL_DATA WITH GAUSS NOISE (K=5)

| Method | Number of Landmarks | DisErr |
|---|---|---|
| Random Landmark Isomap | 90 | 22.2251 |
| Kmeans Landmark Isomap | 90 | 19.2218 |
| SOM Landmark Isomap | 9*10 | 19.2500 |
| SOINN Landmark Isomap | 90±20 | 19.1390 |

Table II. shows the mean $DisErr$ of 100 repeated experiments by SL–Isomap and other variants of L–Isomap. Compared with Table I, there is a noteworthy phenomenon that the result of Random Landmark Isomap becomes worse obviously,

while other methods performs better in noise environment. One plausible explanation is that noise data points constitute clusters (black circles in Fig. 8). They help clustering methods to determine the position of landmarks better and reduces the short–circuit errors. However, for Random Landmark Isomap, the noise data points may be selected as landmarks and ruin the result. We can also see that SL–Isomap outperforms its counterparts in reducing the short–circuit errors in the noise environment.

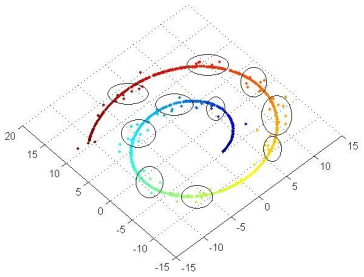
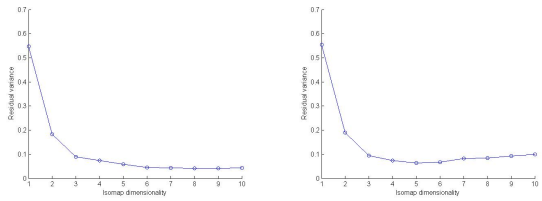


Fig. 8. Noise data points constitute clusters

### C. AT&T face

In this experiment, we test the performance of SL–Isomap on real world data. We take 100 images of 10 people from the AT&T face to be the input data set $Data$. To reduce computational complexity, each face image is downsampled to 23 * 28 pixels in this experiment.

The residual variance by Isomap and SL–Isomap on $Data$ is shown as Fig. 9. We can see their curves change roughly consistently with the dimension $d$ and the knee point at both curves is when $d = 3$. When $d \geq 3$, the residual variance ceases to decrease significantly with added dimensions.



(a) (b)

Fig. 9. AT&T face residual variance by (a) Isomap (k=10) and (b) SL–Isomap (k=10)

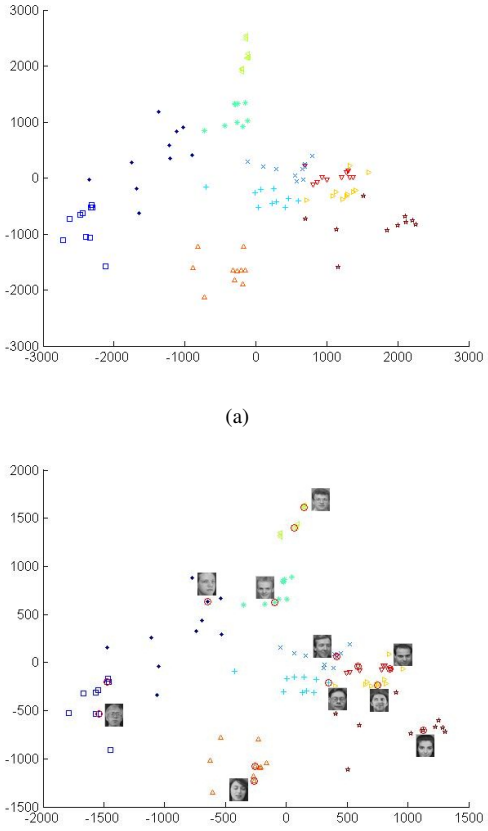
We run SOINN 500 times at first and the number of landmarks concentrates around 14. We get the 3–dimensional embedding of $Data$ by SL–Isomap and other landmark methods, and calculate the mean $DisErr$ of 100 repeated experiments. The experimental result is shown in Table III. It shows that SL–Isomap performs better in reducing short–circuit errors than other variants of L–Isomap in real world data.

For visualization, we run Isomap and SL–Isomap to get the 2–dimensional embedding of $Data$ (Fig. 10). Points with red circle in Fig. 10(b) are SOINN landmarks. We can see

TABLE III
DISERR ON AT&T FACE (K=10)

| Method | Number of Landmarks | DisErr |
|---|---|---|
| Random Landmark Isomap | 14 | 1113.5 |
| Kmeans Landmark Isomap | 14 | 1054.6 |
| SOM Landmark Isomap | 2*7 | 1092.5 |
| SOINN Landmark Isomap | 14±4 | 1047.1 |

that distributions of 2–dimensional embedding by Isomap and SL–Isomap are almostly consistent, and 14 SOINN landmarks are able to cover 10 classes without prior knowledge.



(a)

(b)

Fig. 10. AT&T face 2–dimensional embedding by (a) Isomap (k=10) and (b) SL–Isomap (k=10)

### D. Discussion

In all variants of L–Isomap, short–circuit errors are inevitable. The cause is the direct links between sparse distributed landmarks. Fig. 11 illustrates the cause of short–circuit errors. In Isomap, the geodesic distance $\mathrm{D}_{\mathrm{G}(\mathrm{L}_1,\mathrm{L}_2)}$ between adjacent landmarks $L_1$ and $L_2$ is calculated with the shortest path distances (sum of Euclidean distances $D_E$) in neighborhood graph: $\mathrm{D}_{\mathrm{G}(\mathrm{L}_1,\mathrm{L}_2)}=\mathrm{D}_{\mathrm{E}(\mathrm{L}_1,\mathrm{d}_1)}+\mathrm{D}_{\mathrm{E}(\mathrm{d}_1,\mathrm{d}_2)}+\cdots+\mathrm{D}_{\mathrm{E}(\mathrm{d}_4,\mathrm{d}_5)}+\mathrm{D}_{\mathrm{E}(\mathrm{d}_5,\mathrm{L}_2)}$. However, in all variants of L–Isomap, $\mathrm{D}_{\mathrm{G}(\mathrm{L}_1,\mathrm{L}_2)}$ is approximated by Euclidean distance between adjacent

landmarks $L_1$ and $L_2$: $\mathrm{D}_{\mathrm{G}(\mathrm{L}_1,\mathrm{L}_2)} \approx \mathrm{D}_{\mathrm{E}(\mathrm{L}_1,\mathrm{L}_2)}$. Although SL–Isomap is able to determine the reasonable number and position of landmarks automatically and reduce short–circuit errors, it does not solve the problem completely. As shown in three experiments above, the embedding of $Data$ with SL–Isomap is "shrinked" with a certain scale compared to that with Isomap. It's a tradeoff between number of landmarks and short–circuit errors.

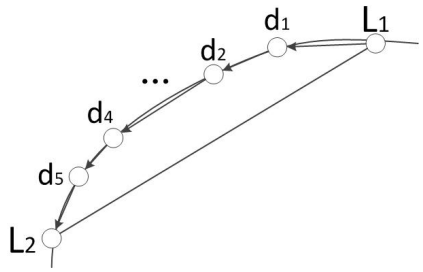



Fig. 11. Short-circuit errors caused by landmarks

## IV. Conclusion

In this paper, we present an extended Isomap algorithm called SL–Isomap. It solves the problem of selecting the right number and position of landmarks automatically, and realizes data compression and nonlinear dimensionality reduction at the same time. It outperforms its counterparts in preserving the geodesic distances in high dimensional space.

Our future work will focus on further reducing the short–circuit errors of SL–Isomap. We plan to determine the neighborhood parameter $k$ or $\varepsilon$ of each landmark adaptively with SOINN.

## Acknowledgment

This work is supported in part by the National Science Foundation of China under Grant Nos. (61375064, 61373001, 61321491), 973 program (2010CB327903), and Jiangsu NSF grant (BK20131279, BK2011567).

## References

[1] Tenenbaum J B, De Silva V, Langford J C. A global geometric framework for nonlinear dimensionality reduction[J]. Science, 2000, 290(5500): 2319-2323.

[2] Yang M H. Extended isomap for classification[C]//Pattern Recognition, 2002. Proceedings. 16th International Conference on. IEEE, 2002, 3: 615-618.

[3] Yang M H. Extended isomap for pattern classification[C]//AAAI/IAAI. 2002: 224-229.

[4] Dechuan Z, Zhihua Z. Ensemble-Based Manifold Learning for Visualization [J][J]. Journal of Computer Research and Development, 2005, 9: 012

[5] Lin F, Yeh C C, Lee M Y. The use of hybrid manifold learning and support vector machines in the prediction of business failure[J]. Knowledge-Based Systems, 2011, 24(1): 95-101.

[6] Liang Y M, Shih S W, Shih A C C. Human action segmentation and classification based on the Isomap algorithm[J]. Multimedia Tools and Applications, 2013, 62(3): 561-580.

[7] Silva V D, Tenenbaum J B. Global versus local methods in nonlinear dimensionality reduction[C]//Advances in neural information processing systems. 2002: 705-712.

[8] De Silva V, Tenenbaum J B. Sparse multidimensional scaling using landmark points[R]. Technical report, Stanford University, 2004.

[9] Shi L, He P, Liu E. An incremental nonlinear dimensionality reduction algorithm based on ISOMAP[M]//AI 2005: Advances in Artificial Intelligence. Springer Berlin Heidelberg, 2005: 892-895.

[10] Law M H C, Jain A K. Incremental nonlinear dimensionality reduction by manifold learning[J]. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 2006, 28(3): 377-391.

[11] Furao S, Hasegawa O. An incremental network for on-line unsupervised classification and topology learning[J]. Neural Networks, 2006, 19(1): 90-106.

[12] Furao S, Ogura T, Hasegawa O. An enhanced self-organizing incremental neural network for online unsupervised learning[J]. Neural Networks, 2007, 20(8): 893-903.

[13] Chi J, Crawford M M. Selection of Landmark Points on Nonlinear Manifolds for Spectral Unmixing Using Local Homogeneity[J]. 2013.

[14] F. Samaria and A. Harter "Parameterisation of a stochastic model for human face identification" 2nd IEEE Workshop on Applications of Computer Vision December 1994, Sarasota (Florida).

[15] Balasubramanian M, Schwartz E L. The isomap algorithm and topological stability[J]. Science, 2002, 295(5552): 7-7.