

# The Hierarchical Isometric Self-Organizing Map for Manifold Representation

Haiying Guan and Matthew Turk  
Computer Science Department  
University of California, Santa Barbara, CA, USA  
{haiying, mturk}@cs.ucsb.edu

## Abstract

*We present an algorithm, Hierarchical ISometric Self-Organizing Map (H-ISOSOM), for a concise, organized manifold representation of complex, non-linear, large scale, high-dimensional input data in a low dimensional space. The main contribution of our algorithm is threefold. First, we modify the previous ISOSOM algorithm by a local linear interpolation (LLI) technique, which maps the data samples from low dimensional space back to high dimensional space and makes the complete mapping pseudo-invertible. The modified-ISOSOM (M-ISOSOM) follows the global geometric structure of the data, and also preserves local geometric relations to reduce the nonlinear mapping distortion and make the learning more accurate. Second, we propose the H-ISOSOM algorithm for the computational complexity problem of Isomap, SOM and LLI and the nonlinear complexity problem of the highly twisted manifold. H-ISOSOM learns an organized structure of a non-convex, large scale manifold and represents it by a set of hierarchical organized maps. The hierarchical structure follows a coarse-to-fine strategy. According to the coarse global structure, it “unfolds” the manifold at the coarse level and decomposes the sample data into small patches, then iteratively learns the nonlinearity of each patch in finer levels. The algorithm simultaneously reorganizes and clusters the data samples in a low dimensional space to obtain the concise representation. Third, we give quantitative comparisons of the proposed method with similar methods on standard data sets. Finally, we apply H-ISOSOM to the problem of appearance-based hand pose estimation. Encouraging experimental results validate the effectiveness and efficiency of H-ISOSOM.*

## 1. Introduction

Modeling and classifying images of articulated visual objects, such as the pose of human hands under camera viewpoint variations and self-occlusion conditions [1] [4], is a challenging problem in computer vision and human

computer interaction [7]. Two main approaches have been proposed to the problem: one is the class of discriminative approaches, which try to solve the problem by learning the mapping function from the visual input [2] [8] to the 3D configuration output. However, the mapping function from the visual input to 3D poses might be too complex to be learned in practice.

Other approaches seek to learn manifolds in a generative way, i.e., learn the mapping from a large training data set of visual input to a low dimensional manifold representation. An advantage of generative mapping is that it is possible to create a large data set, spanning a wide range of expected configurations, from synthesized images. Thus, it is particularly promising for subsequent tracking or dynamic recognition. The critical problem, however, is the manifold learning and representation.

Many linear and nonlinear methods have been developed for visual manifold learning. Classical techniques such as principle component analysis (PCA), multidimensional scaling (MDS), and independent component analysis (ICA) are suitable for the case where the sub-manifolds can be embedded linearly, or almost linearly, in the observation space.

Recently, nonlinear dimensionality reduction techniques, such as Isomap, Locally Linear Embedding, Laplacian Eigenmap, Hessian Eigenmap, Semidefinite Embedding, Kernel PCA, and Kernel ICA have been proposed for nonlinear manifold learning. Tenenbaum’s Isomap algorithm [12] represents remote distances of sample points as sums of a trusted set of distances between immediate neighbors, then uses MDS to compute a low-dimensional embedding by minimizing the global error between Euclidean distance in embedded space and geodesic distances of each pair of points in the original space. Isomap may distort the local structure of the data because MDS does a much better job in representation large distances (the global structure) than small ones (the local structure).

Roweis’ Local Linear Embedding (LLE) algorithm [9] represents each point as a weighted combination of a trusted set of nearest neighbors by solving a least-squares problem, and minimizes the distortion (reconstruction error) of neighborhood relationships from high dimensional data to

the embedding. Because LLE estimates the global geometry by the local geometry, it may distort the global structure.

Other techniques involve iterative optimization procedures, such as self-organized maps (SOM) [6], and generative topographic mapping (GTM). Kohonen’s SOM is an unsupervised clustering algorithm for dimensionality reduction, which is an effective tool for the visualization of high dimensional data in a low dimensional (normally 2D) space. It is used to build a mapping from many to few dimensions by preserving the topological order of the data. However Euclidian distance in high dimensional space may not be the best way to measure the nonlinear manifold. Growing hierarchical self-organizing map (GHSOM) [3] is an extended version of SOM with dynamic and hierarchical structure. The algorithm starts with a “virtual” layer 0, which consists of a single unit, and continuously trains the sub-layer by minimizing the mean quantization errors.

By integrating the self-organizing model with the geometric graph distance of Isomap, Guan [5] proposed an Isometric Self-Organizing Map (ISOSOM) method for concise, nonlinear manifold representation, which was applied to the problem of 3D hand pose estimation. ISOSOM implicitly models the learned manifold through an organized map. The algorithm not only reduces the input dimension of the data samples, but also effectively organizes and clusters the samples in the low dimensional map to reduce the size of representative samples<sup>1</sup>. However, there main issues exist for the algorithm: first, because Isomap has distortions on local property, ISOSOM also has distortions. Second, because the computational complexity of Isomap, the algorithm can only handle a middle scale of data set (less than 20,000 generally). It is nearly intractable for the algorithm to learn a large scale data set.

The main challenges of learning a concise, organized manifold representation of a large scale, non-convex, high intrinsic dimension, complex manifold are the following:

†**Computation complexity:** Most of the algorithms become practically intractable if the sample size is larger than twenty thousand, which might not be enough to learn the complex, large scale data set such as the data set for hand pose estimation.

†**Accuracy:** Most of the algorithms introduce some global or local distortion. For example, LLE is susceptible to placing faraway points nearby to each other and Isomap has problem with non-convex manifolds.

†**Sampling problem:** Isomap and LLE assume the data are “well-sampled”, but the global smoothness condition is a strong assumption for some sparse sampled data sets.

†**Invertibility problem:** The objective is to calculate the high dimensional coordinates in the input sample space of any point on the low-dimensional manifold to make the mapping invertible.

†**Organized representation:** An effectively organized structure is useful for the further tasks such as recognition, fast indexing or tracking [11].

<sup>1</sup>The main point is that the scale of the final representation should be decided by the complexity of the manifold and the required accuracy instead of the scale of the training samples.

In this paper, we propose a Hierarchical ISometric Self-Organizing Map (H-ISOSOM) to address the problems mentioned above. The main contributions of the paper are the following: first, we present an modified-ISOSOM (M-ISOSOM) algorithm with a local linear interpolation (LLI) technique for better accuracy. LLI constructs a mapping from low to high dimensional space and makes the whole mapping pseudo-invertible. Since it preserves local and global geometric relations simultaneously, the M-ISOSOM is more accurate than the nearest neighbor technique of the previous ISOSOM. Second, due to the computational complexity limitation of Isomap, ISOSOM or M-ISOSOM is incapable to learn large scale data sets. We propose an hierarchical version of the M-ISOSOM to address two complexity problems: the computational complexity, and the manifold complexity<sup>2</sup> for better accuracy. H-ISOSOM follows a coarse-to-fine strategy to build the mapping between the high dimensional input space and the low dimensional space. According to coarse global structure, it “unfolds” the manifold in the coarse level and decomposes the sample data into small patches, then iteratively learns the nonlinear sub-manifolds. Consequently, for a very complex nonlinear manifold, H-ISOSOM divides it to small patches, each small patch is less complex and makes the algorithm more accurate, effective and trackable for large scale data set. Third, we compare the performance of SOM and GHSOM with ISOSOM and HISOSOM on five different kinds of manifolds, reporting quantitative measurements of signal-to-noise ratio (SNR), mean quantization error (MQE), and the standard derivation of quantization error (STDQE). These experimental results show that our algorithm outperforms SOM and GHSOM.

Finally, we apply H-ISOSOM to exemplar-based pose estimation problem [1]. The hand gesture images can be approximate as a high dimensional manifold embedded in the visual input space, which twists significantly depending on the viewpoint, the hand shape, self-occlusion, geometry, and the lighting condition. The manifold learning algorithm not only requires high discriminative ability to distinguish different hand images with different pose and posture, but also requires the generative ability to group similar images together for concise representation in order to reduce the size of nodes for further retrieval. As Vassilis *et al.* mentioned in [1], the main difficulty of exemplar-based hand pose estimation is the complexity problem; that is, if the pose angle accuracy requirement is increased, the size of the database is exponentially increased<sup>3</sup>. The large scale of the data set is a challenge for further indexing or retrieval [13] [10]. In such cases, the concise, accurate manifold

<sup>2</sup>For example, non-convex shapes such as the data samples contain holes, or shapes with very detailed sharp curvatures locally.

<sup>3</sup>For example, in the case that the angle interval is 36° for each DOF and the total number of DOF is 24, the number of sample points in the data set is  $O((360/36)^{24})$ . However, if the accuracy requirement is 12°, the number of exemplars in the data set is  $O((360/12)^{24})$  which is  $O(3^{24})$  times greater than the original data set.

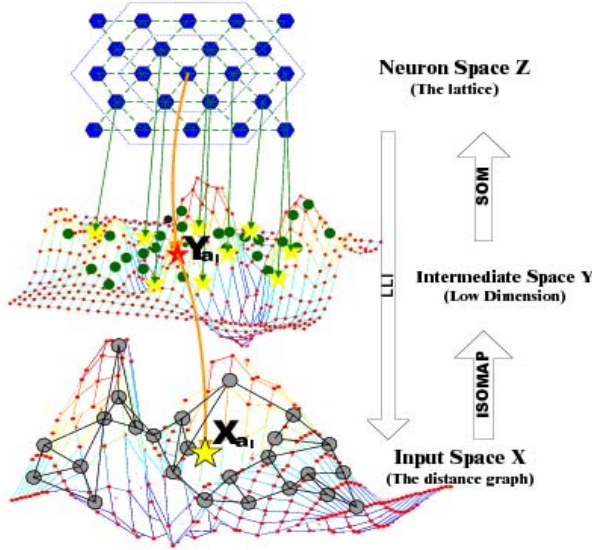


Figure 1. The Modified Isometric Self-Organizing Map

representation of the complex, nonlinear, high dimensional manifold is necessary. H-ISOSOM attempts to reduce the size of the data set by a hierarchical organized map structure with relatively small number of nodes.

## 2. Manifold learning with M-ISOSOM

The objective of M-ISOSOM is to learn the organized map in a low dimensional lattice for a set of observations by inverting the following generative model. Let  $X$  be a high dimensional domain in the Euclidean space  $R^{d_x}$ . Let  $Y$  be a  $d_y$ -dimensional domain contained in the Euclidean space  $R^{d_y}$ , where  $d_x \gg d_y$ . Let  $f_{xy} : Y \rightarrow R^{d_x}$  be a smooth embedding, where  $\{x_i = f_{xy}(y_i)\} \subset R^{d_x}$ . Let  $Z$  be a  $d_z$ -dimensional lattice (normally the dimension is two) contained in the Euclidean space  $R^{d_z}$ . Let  $f_{yz} : Z \rightarrow R^{d_y}$  be an exemplar-based embedding represented (encoded) by the vector associated with each node, where  $\{y_i = f_{yz}(z_i)\} \subset R^{d_y}$ . The objective of M-ISOSOM is to learn the organized map in the low dimensional lattice  $Z$  of dimension  $d_z$  from the samples in high dimensional space  $X$  of dimension  $d_x$  through the intermediate space  $Y$  of dimension  $d_y$ , where  $d_x \gg d_y \geq d_z$ .

### 2.1. M-ISOSOM

We modify Guan’s ISOSOM algorithm [5] with the local linear interpolation techniques to take both global and local relationships into account and make the whole mapping pseudo-invertible. The intuitive depiction of the M-ISOSOM is illustrated in Figure 1.

The learning process of the M-ISOSOM algorithm contains three main steps: First, we construct a distance graph  $G$  of the manifold over all data points in input space  $X$ . The graph takes all points as nodes and adds an edge between two nodes  $i$  and  $j$  if  $i$  is one of the  $k$  nearest neighbors of

$j$ . The cost of the edge is measured by the Euclidean distance between  $i$  and  $j$ . The distance of any two nodes on the graph is defined by the cost of the shortest path between them. The low dimensional embedding  $Y$  and  $f_{xy}^{-1}$  is constructed by the classic MDS algorithm based on the distance graph as in the Isomap algorithm (see Alg. 1, Part (I)). This step focuses on the global relationships of the data samples and encodes it to the dimensional embedding.

The second step is the exemplar learning and organization by the SOM algorithm. The data samples in the intermediate space  $Y$  are used as the training samples to train the organized structure. Similar to the map structure of SOM, the M-ISOSOM map is a lower dimensional lattice,  $A$ , formed by a set of organized processing units, called nodes. The nodes are connected with their neighbors on the lattice. Each representative  $a_i$  is labeled by an index  $i \in \{1 \dots \text{size}(A)\}$  and has reference vectors  $Y_{a_i}$  attached.

In each M-ISOSOM training step, we randomly choose a sample vector  $y$  from the training samples, and the response of a representative to the vector  $y$  is determined by the comparisons between  $y$  and the reference vector  $Y_{a_i}$  of each representative with the geometric distance defined by the distance graph. The Best Matching Unit (BMU) is defined as the winner representative  $a_i$  which has its reference vector  $Y_{a_i}$  closest to the given input  $y$ . After obtaining the BMU, its prototype vectors  $Y_{a_i}$  and its topological neighbors are updated and moved closer to the input vector  $y$ .

Isomap maps a set of points from the high dimensional space  $X$  to a set of points in the low dimension space  $Y$ . For each sample point in  $X$ , there is a point in  $Y$  corresponding to it. After SOM training, the associate vector  $Y_{a_i}$  of each representative is the new point in  $Y$  space generated by SOM iterative training. We need its corresponding point  $X_{a_i}$  in the high dimension space  $X$ . Since Isomap is a nonlinear dimension reduction techniques, it is hard to find the exact inverse mapping. Thus, we approximate its corresponding points in the original space  $X$  by the local linear interpolation (LLI) algorithm (see Alg. 1, Part (III)).

The LLI algorithm (see Alg. 2) is motivated by the idea of LLE [9], but the objective of LLI is to project the points from the low dimensional space  $Y$  to the high dimensional space  $X$  rather than reducing the dimensionality.

Given a query vector with full components or partial components with the mask  $w$ , the best match nodes are retrieved by the following similarity measurement:

$$BMU = \underset{\forall a \in A}{\operatorname{argmin}} \operatorname{Distance}(w(x_a), w(x)) \quad (1)$$

where  $w(x_a)$  is the mask function defined by  $W$  representing the existing components.

### 2.2. Performance comparisons of SOM, ISOSOM and M-ISOSOM

To validate the effectiveness of M-ISOSOM, we compare the performance of SOM, previous ISOSOM [5] and



---

**Algorithm 1** The Isometric Self-Organizing Map

---

(I) **Nonlinear dimension reduction using Isomap** The data samples in the high dimensional input space  $X$  are mapped into to the low dimensional intermediate space  $Y$ , and the mapping,  $f_{xy}^{-1} : X \rightarrow R^{d_y}$ , is learned by the Isomap algorithm.

(II) **Clustering and organization using SOM** The data samples in the intermediate space  $Y$  is used for the SOM training and the mapping,  $f_{yz}^{-1} : Y \rightarrow R^{d_x}$  is learned by the SOM algorithm. The associate vector of each nodes,  $Y_{a_i}$ , is a new sample in the low dimension  $Y$  space.

(III) **Local linear interpolation**  $X_{a_i}$ , the inverse mapping point of  $Y_{a_i}$ , is approximate by the local linear interpolation of  $x_1, \dots, x_k$ , which are the inverse mapping of  $y_1, \dots, y_k$ , the effective nearest points of  $Y_{a_i}$ . The mapping  $f_{xy} : Y \rightarrow R^{d_x}$  is learned by local linear approximation.

(IV) **M-ISOSOM Retrieval** In the retrieval stage, the best matching unit (BMU) is the closest representative on the M-ISOSOM map to the query vector.

---

---

**Algorithm 2** The Local Linear Interpolation

---

(I) **Effective neighbors** For each point described by the learned Isomap representative vector, and each point from the original point set in low dimension  $Y$  space, their  $K$  effective neighbors are identified by choosing the nearest points from the original point set in  $Y$  space (The neighbors which comes from the new nodes on the map are not effective neighbors).

(II) **Local weight** The weights for all points is calculated by minimizing the reconstruction errors given in Eq. 2:

$$\varepsilon(W) = |\vec{Y}_i - \sum_j^K W_{ij} \vec{Y}_j|^2 \quad (2)$$

where  $\vec{Y}_j \in$  the effective neighbor set, and  $\sum_j W_{ij} = 1$ . The error function adds up the squared distance between all the data points and their effective reconstructions.

(III) **Reconstruction** We use the weights to reconstruct a point on the map by its effective neighbors' corresponding points in  $X$  space,

$$\vec{X}_i = \sum_j^K W_{ij} \vec{X}_j \quad (3)$$

---

M-ISOSOM results using five different data sets: a non-convex roll surface data set A, a roll surface data set B with intersections, a roll surface data set C, an open box data set, and a fishbowl data set. The roll surface data sets A and B are constructed by a group of functions with different parameters. The open box data set are constructed by the five flat sheets. The fishbowl data set are constructed by a part of 3D sphere and a flat disc.

We use three parameters to measure the learning performance: mean quantization error (MQE), signal-to-noise ratio (SNR) (see Eq. 4-6<sup>4</sup>), and standard derivation of quantization error (STDQE), which is the standard derivation from the testing samples to its BMU.

The comparisons of SOM and M-ISOSOM with the same training data sets are shown in Table 1. The two algorithms are tested with the same five data sets. The testing sample sizes of these five data sets are around 3295-

<sup>4</sup>SNR (dB) is calculated by the following equations:

$$SNR=10\lg(\text{Signal Power}/\text{Noise Power}) \quad (4)$$

$$\text{Signal Power}=1/N\sum_i \|x_i - \bar{x}\|^2, \text{ where, } \bar{x}=1/N\sum_i x_i \quad (5)$$

$$\text{Noise Power}=1/N\sum_i \|x_i - x_{BMU}\|^2, \text{ where, } x_{BMU} = \text{BMU of } x_i \quad (6)$$

For every testing data point, the noise error is the Euclidean distance between itself and its best matching unit (BMU) in the map.

3300. The training results of the map size is around 285-294. The lattices of SOM and ISOSOM are hexagonal. Generally, more nodes have a greater ability to represent the map accurately. In order to verify that M-ISOSOM is better than SOM, we guarantee that the map sizes of SOM are all slightly greater or equal to the map sizes of M-ISOSOM. According to our experiments, different initializations of SOM have very slightly effects on the results and could be neglected.

SOM utilizes the Euclidean distance as measure function in the original data set space. M-ISOSOM utilizes the geometric distance defined by the distance graph of Isomap. An inspection of (larger versions of) the figures of the learned maps in Table 1 shows that the SOM map does not follow the roll surface and has lots of interpolation between the roll surface gaps. Thus, the SOM map doesn't represent the training data well. At the same time, the best matching unit (BMU) of the SOM map can not represent the testing data samples accurately. On the contrary, the M-ISOSOM map follows the roll surface nicely and represents the training or testing data set well. This intuition is also verified by the quantitative measurements: the MQE of M-ISOSOM are smaller than the MQE of SOM for all five data sets. The SNR of M-ISOSOM are greater than SOM for all five data sets. The STDQE of M-ISOSOM are also smaller than SOM for the first four data sets; for the fishbowl data set, the STDQE is slightly larger for M-ISOSOM. The comparison results also show that M-ISOSOM is very good at Swiss Roll-like surfaces, such as A and C, for which the performance differences between M-ISOSOM and SOM are greater than the other cases. Roll Surface B has intersections and M-ISOSOM has more distortions in calculating the distance graph than A and C.

We also compare the performance of previous ISOSOM [5] and M-ISOSOM. The results shows that the SNR of M-ISOSOM is 0.557dB better than the previous ISOSOM on average for the five data sets with training samples contains holes.

One of the objectives of our manifold learning is that we not only want to accurately represent the map with a limited number of organized nodes, but we also want to do interpolations on this organized map in order to obtain better accuracy with the same set of training samples<sup>5</sup>. The map interpolation ability of SOM and M-ISOSOM are compared in Table 2. For ease in implementing the interpolation, the lattices of SOM and M-ISOSOM are rectangular. We do four times interpolation scale for the first three data sets and two times interpolation scale for the open box and fish bowl data sets. Compared with Table 1, the results show that the interpolated maps are more accurate than the non-interpolated maps. In general, the interpolated M-ISOSOM maps are more accurate than the interpolated SOM maps.

<sup>5</sup>Such property is very crucial for hand pose estimation, when it is intractable to generate the data set with small viewpoints intervals, we hope the algorithm has the interpolation ability to approximate it in finer level.

Table 1. SOM vs. M-ISOSOM

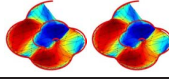
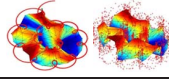
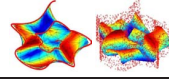
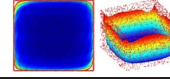
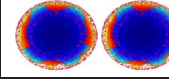
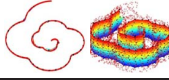
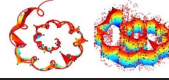

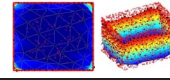
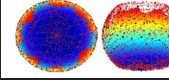
	Roll Surface A	Roll Surface B	Roll Surface C	Open Box	Fish Bowl
SOM					
M-ISOSOM					
SOM(MQE)	0.1101	0.1048	0.1223	0.0652	0.1086
M-ISOSOM (MQE)	0.0774	0.0851	0.0878	0.0616	0.0936
SOM (SNR)	15.0924	13.1661	14.4780	18.6338	18.1267
M-ISOSOM (SNR)	17.9543	15.1709	17.0674	18.9520	18.8602
SOM (STDQE)	0.0689	0.0660	0.0748	0.0375	0.0562
M-ISOSOM (STDQE)	0.0535	0.0493	0.0601	0.0390	0.0622*

Table 2. SOM vs. M-ISOSOM with interpolation

	Roll A	Roll B	Roll C	Open Box	Fish Bowl
SOM(MQE)	0.0910	0.0909	0.1023	0.0485	0.0869
M-ISOSOM (MQE)	0.0512	0.0591	0.0682	0.0433	0.0640
SOM (SNR)	16.0916	13.8285	15.5631	20.4134	19.6408
M-ISOSOM (SNR)	19.3687	18.0018	16.1853	20.7378	20.8217
SOM (STDQE)	0.0728	0.0700	0.0744	0.0375	0.0547
M-ISOSOM (STDQE)	0.0614	0.0653	0.0547	0.0402*	0.0628*

Intrinsically, M-ISOSOM utilizes the global geometric distance and the local relationship to perform the nonlinear dimension reduction. The global geometric distance is defined by the metric relationship between the training samples and preserves the relationship of the samples in high dimension space. In the SOM learning process, this geometric relationship is also preserved in the ISOSOM map's organized structure, where similar nodes are closer to each other in the grid than dissimilar ones. The local linear interpolation preserves the local details and improves the algorithm accuracy. Above all, ISOSOM preserves the spatial relationships in high dimensional input space  $X$  to the low dimensional ISOSOM lattice map  $Z$ , and follows better the topology of the underlying data set than SOM. In addition, the organized map of ISOSOM discretely represents the manifold by exemplars, which to some extent relaxes the global smoothness constraints of Isomap and LLE.

### 3. Hierarchical-ISOSOM

#### 3.1. H-ISOSOM

Due to the computational complexity of ISOSOM and M-ISOSOM, they cannot handle large scale data sets (for example, more than one million training samples). In order to solve this problem, we present a hierarchical version of M-ISOSOM. The intuitive depiction of the Hierarchical-ISOSOM is illustrated in Figure 2. H-ISOSOM aims to defuse the computation complexity problem for large scale data sets, to improve the accuracy, and to construct a hierarchical structure for the fast retrieval or indexing.

Hierarchical algorithms can be either divisive or agglomerative, i.e., top-down or bottom-up. Divisive hierarchical algorithms begin with the coarsest possible partition and split groups apart step by step. Alternatively, agglomerative hierarchical algorithms, which are widely used in clustering, start from the finest possible structure (each data point

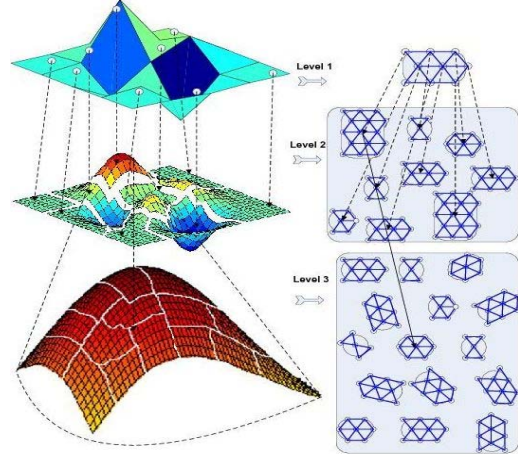


Figure 2. The Hierarchical Isometric Self-Organizing Map forms a cluster) and merge together at different levels by a certain criterion. We adopt the divisive strategy.

The complexity of the Isomap is  $O(N^3)$  (where  $N$  is the size of the training sample)<sup>6</sup>. The complexity for SOM is  $O(NMD)$ <sup>7</sup>. Both of them have difficulties if the training sample size is more than 20,000. In order to make the training of large scale data sets tractable, H-ISOSOM follows a coarse-to-fine strategy. Let  $N_{max}$  be the upper limit of the sample size that the algorithm of  $O(N^3)$  complexity can handle in practice. H-ISOSOM (Alg. 3) first randomly samples the whole data set and obtains the subset with  $m$  samples, where  $m < N_{max}$ , to train the first layer of the H-ISOSOM map using the M-ISOSOM algorithm. After that, for each representative on the map, it collects the training samples from the original data set and sub-samples them to train the next layer. The algorithm iteratively trains the sub-maps with M-ISOSOM until a certain criterion is reached.

The criteria used in the paper are the mean quantization error (MQE) and signal-noise-ratio (SNR) between the map representative data set  $S_{map}$  and the training sample data set  $S_{trn}$ . MQE is the average distance of each sample in the data set  $S_{trn}$  to its nearest points in the map data set  $S_{map}$

<sup>6</sup>Even for Landmark-Isomap algorithm, with the complexity of  $O(n^2N)$  and the tradeoff between the complexity and distortion, the complexity still intractable for the complex, large scale data set.

<sup>7</sup>where  $N$  is the number of training samples,  $M$  is the number of the nodes, and  $D$  is the dimension of the input space.

(the quantization error).

In the retrieval stage, given the input data, H-ISOSOM retrieves the top  $k$  BMUs, and continues to find the refined best match results in the next layer until the last layer is reached. Then it reorders all the retrieved nodes from the hierarchical structure according to the distance metric to obtain the final BMU set.

---

**Algorithm 3** The Hierarchical ISOSOM

---

**(I) Sampling** ( $O(N)$ ) If the number of the current data set,  $DS_{cur}$ , in the high dimensional input space  $X$  is larger than  $N_{max}$ , the algorithm randomly sample a small training data set,  $DS_{trn}$ , with  $N_{max}$  number of training data; or else, the algorithm take the whole data set as training data set.

**(II) ISOSOM training** ( $O(N^3)$ ) Given the training data set,  $DS_{trn}$ , with  $m$  number of data samples, where  $m < N_{max}$ , the ISOSOM map of current layer,  $map_{cur}$ , is trained.

**(III) Criterion Judgment** ( $O(N)$  or  $O(N^2)$ ) For each trained ISOSOM,  $map_{cur}$ , the MQE and SNR between the current data set,  $DS_{cur}$ , and the trained map,  $map_{cur}$ , are measured. If the criterion is satisfied (for example, MQE is less than a threshold or/and SNR is greater than a threshold), the algorithm stops iterative dividing. If not, for each node in current ISOSOM map,  $map_{cur}$ , the algorithm continues to collect the samples for the next layer from the current data set,  $DS_{cur}$ . The next layer data collection algorithm for each node is the following: for each sample from the current data set,  $DS_{cur}$ , find it’s BMU in the current ISOSOM map,  $map_{cur}$ , and labeled the sample to it’s BMU, the data samples for each node in the next layer are all the samples whose BMU is this node. The algorithm then repeats step (I) and (II) to iteratively trains the ISOSOM map in the following layers until the certain criterion is reached or the number of the training samples in the finer layer is less than a fixed number.

**(IV) H-ISOSOM Retrieval** ( $O(\log N')$ ) For a H-ISOSOM map with  $N'$  nodes, we find the  $k$  number of BMU in the first level, the retrieval iteratively search on the sub-layers associated with retrieved nodes until the finest layer is touched. Then, all the retrieved nodes are reordered and the top  $k$  BMU set is obtained.

---

According to Alg. 3, H-ISOSOM will reduce to M-ISOSOM if the size of the data sample is tractable for M-ISOSOM and the criterion is large enough. The criterions used in H-ISOSOM indicate how well the map fits the training data. With the criterions, the algorithm can provide a concise representation of a large-scale training data set.

**3.2. GHSOM vs. H-ISOSOM**

In this section, we compare the performance of two hierarchical nonlinear clustering algorithm, GHSOM and H-ISOSOM. Generally, although GHSOM and H-ISOSOM are hierarchical algorithms, they are intrinsically different in two main aspects. First, the objective of GHSOM is to retrieve the hierarchical structure of the data with accuracy. H-ISOSOM seeks to handle the large scale data sets that ISOSOM cannot handle in practice. Second, the structures of the two algorithms are different. GHSOM starts from a single unit, splits to a 4-unit map, and iteratively splits until the criterion is reached. H-ISOSOM starts from the global coarse structure of the data (a large map) and then refines it in the following layers.

The comparisons of GHSOM and H-ISOSOM with the same training data sets are shown in Table 3. The two al-

Table 4. GHSOM vs. H-ISOSOM with interpolation

	Roll A	Roll B	Roll C	Open Box	Fish Bowl
GHSOM(MQE)	0.0407	0.1501	0.1277	0.0525	0.0824
H-ISOSOM(MQE)	0.0156	0.0283	0.0221	0.0148	0.0482
GHSOM(SNR)	19.5035	9.4517	11.8738	18.1149	17.9299
H-ISOSOM(SNR)	27.4414	22.4388	25.4192	26.2211	22.5248
GHSOM(STDQE)	0.1820	0.1159	0.1454	0.0607	0.0943
H-ISOSOM(STDQE)	0.0274	0.0317	0.0136	0.0279	0.0599
GHSOM (# of nodes)	303, 102	180, 768	172, 860	268, 468	36, 708
H-ISOSOM (# of nodes)	300, 198	167, 947	173, 708	228, 749	28, 324

gorithms are also tested with the same five data sets. The training results of the node map size are shown in the table. The lattices of GHSOM and H-ISOSOM are rectangular. In order to verify that H-ISOSOM performs better than GHSOM, we guarantee that the node sizes of GHSOM are all slightly greater or equal to the node sizes of H-ISOSOM. Both the GHSOM map and the H-ISOSOM map are two-layer structures.

Compared with Table 1, the performance of GHSOM and H-ISOSOM are better than that of SOM and ISOSOM.

From the structure point of view, GHSOM starts from a single “virtual” node and iteratively divides (splits) the map according to the MQE criterion in the sublayer. H-ISOSOM follows a global coarse-to-fine strategy with distance graph of Isomap and refines the local accuracy by ISOSOM. Intuitively, H-ISOSOM is better able to follow the data manifold, which is also shown in Table 3, where GHSOM still interpolates in the gaps between the roll, while H-ISOSOM follows the surface nicely. The quantitative data also verifies that H-ISOSOM generally performs better than GHSOM.

In addition, for the open box and fish bowl cases, H-ISOSOM handles the rims much better than M-ISOSOM. According to the algorithm of H-ISOSOM (Alg. 2), for each node on the first layer of the H-ISOSOM, it re-gathers the nearby samples (all the samples whose BMU is this node) and re-trains the sub-layer M-ISOSOM. This procedure greatly helps the learning on the rims.

Table 4 shows the interpolation results of the GHSOM map and H-ISOSOM. It shows that H-ISOSOM follows the data manifold much better than GHSOM, so the interpolation results are more accurate than GHSOM. The reason is that M-ISOSOM utilizes nonlinear reduction techniques with geodesic distance instead of Euclidean distance to measure the manifold structure.

Finally, we test the computational complexity of H-ISOSOM algorithm on the large scale data sets with more than one million training samples. Such large scale data sets are intractable for ISOSOM or M-ISOSOM. Table 5 shows the learning performance of three layer H-ISOSOM. It shows that H-ISOSOM follows the coarse-to-fine strategy nicely and it can approximate the training samples and represented them nearly perfectly, provided enough layers in the structure and enough nodes.



Table 3. GHSOM vs. H-ISOSOM

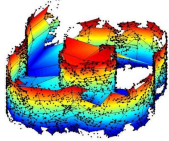
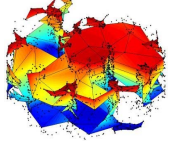
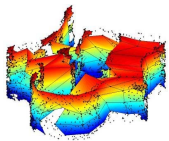
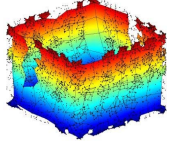
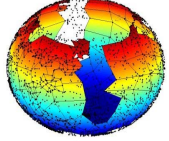
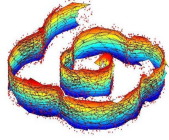
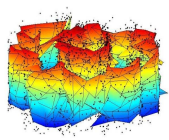
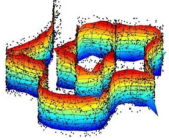
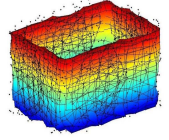
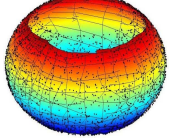
	Roll Surface A	Roll Surface B	Roll Surface C	Open Box	Fish Bowl
GHSOM					
H-ISOSOM					
GHSOM(MQE)	0.0329	0.0553	0.0547	0.0346	0.0741
H-ISOSOM (MQE)	0.0323	0.0517	0.0435	0.0282	0.0545
GHSOM (SNR)	25.1429	18.8966	21.2024	23.6207	21.1413
H-ISOSOM (SNR)	25.1251*	19.1739	23.0723	25.7078	23.8273
GHSOM (STDQE)	0.1820	0.1348	0.0371	0.0294	0.0447
H-ISOSOM (STDQE)	0.0256	0.0341	0.0307	0.0248	0.0326
GHSOM (# of nodes)	2217	1158	1307	1762	981
H-ISOSOM (# of nodes)	2211	1151	1294	1739	974

Table 5. H-ISOSOM with three layers

H-ISOSOM	Roll A	Roll B	Roll C
training sample size	1, 000, 000	7, 363, 381	2, 000, 001
L1(MQE)	0.1126	0.0911	0.1121
L2(MQE)	0.0067	0.0089	0.0067
L3(MQE)	0.0012	0.0016	0.0011
L1(SNR)	14.8667	14.5895	15.0211
L2(SNR)	39.1014	32.7398	39.1653
L3(SNR)	53.8508	41.6249	54.9354
L1(STDQE)	0.0732	0.0520	0.0745
L2(STDQE)	0.0048	0.0094	0.0050
L3(STDQE)	$8.8806e - 4$	0.0044	$7.9719e - 4$
L1 (# of nodes)	156	258	195
L2 (# of nodes)	39, 118	67, 076	50, 529
L3 (# of nodes)	945, 250	3, 052, 340	1, 499, 956

#### 4. 3D hand pose estimation using H-ISOSOM

We apply the H-ISOSOM algorithm to the problem of 3D hand pose estimation. The main challenge for hand pose estimation is that the mapping from hand angle space to hand image feature space is a many-to-many nonlinear mapping. In order to unfold the significantly twisted manifold due to the viewpoint twists and other twists caused by image formulation or feature extraction, we combine the ground truth of the hand configuration and the viewpoint parameters together with the image feature representation to learn the manifold. Generally speaking, in this way, we convert the supervised learning problem to unsupervised learning by learning the joint distribution between both the input (image features) and the target (hand configurations with camera view points) information<sup>8</sup>.

Another main challenge is that if we generate a synthesized data set with a small viewpoint sampling interval, the size of the data set will exponentially increase. H-ISOSOM thus is used to handle the large scale data set which is in-

<sup>8</sup>Given a hand image, due to many-to-many mapping property, there exist many hand configurations with different hand postures. The algorithm should output all those correct configurations. M-ISOSOM with top  $N$  retrieval is a suitable solution for such problem.

tractable to M-ISOSOM. The Correct Retrieval Rate curve generally will decrease greatly for a large dense data set compared with small sparse data set. The Recall-Precision curve will also decrease if the feature’s discriminative ability is not strong enough to distinguish the small viewpoint changes, and even worse if the feature has no distinguishing ability for different gesture or viewpoints.

In the experiments, we generated a synthesized hand image data set with 15 gestures; each gesture is sampled from 15376 viewpoints from a 3D view sphere (the viewpoint interval for each DOF is  $12^\circ$ ). In the experiment, instead of focusing on feature extraction, we aim at improving the retrieval accuracy given a commonly used feature, Hu moments.

First we test the algorithms with another dense synthesized data set, whose pitch and yaw camera viewpoint is sampled at  $8^\circ$  intervals. Figure 3 shows the hand images randomly picked up from the testing dataset. The size for the testing data set is 861 for each gesture; Hu moment features are computed, which are in-plane rotation invariant (corresponding to the roll parameter of the camera). We compare the performance of the H-ISOSOM, GHSOM, SOM, and K-Nearest Neighbor for pose estimation with a single gesture. The Recall-Precision Graph of the “pick” gesture is shown in Figure 4. The correct retrieval rate chart for the same gesture with the same training results is shown in Figure 5. The percentage of correct retrieval is calculated this way: for a given  $N$  number of top matches, if the viewpoint angle parameters are within  $15^\circ$  of the ground truth, it is considered to be correctly retrieved. All 861 test samples are tested and the percentage of correct retrieval is calculated<sup>9</sup>. Both Figure 4 and 5 show that H-ISOSOM

<sup>9</sup>In most retrieval applications only the first  $N$  retrievals are considered, regardless the size of the data set. Such information is not easy to deduce from the Recall-Precision graph. For the hand pose estimation problem, for example, only the first 100 retrievals might be needed for the further

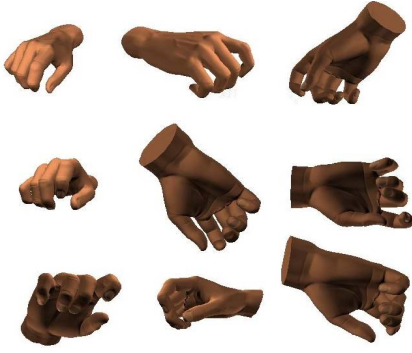


Figure 3. The hand images for the “pick” gesture

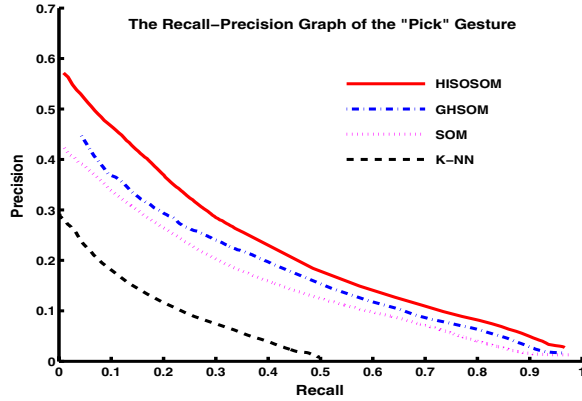


Figure 4. The Recall Precision graph for the “pick” gesture

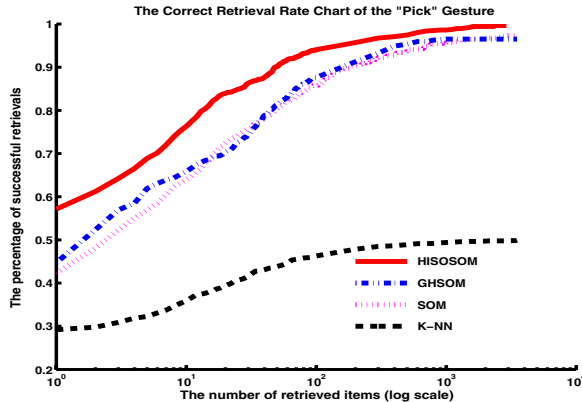


Figure 5. The Correct Retrieval Rate Chart for the “pick” gesture

performs better than GHSOM<sup>10</sup>, SOM<sup>11</sup>, and a K-nearest neighbor (K-NN) algorithm.

analysis.

<sup>10</sup>The size of the GHSOM map is around 6000, which is slightly larger than the size of the H-ISOSOM map

<sup>11</sup>The size of SOM is 3577, which is much larger than the default map size given the data set. It is also the largest map we can obtain with our PC. From the intuition during the test experiments with different map size for SOM, even the map size could be enlarged further, the performance is not going to increase any more.

## 5. Conclusion

We have presented the H-ISOSOM algorithm for the concise representation of the nonlinear manifold. We mainly address two issues: the computational complexity and accuracy problem of the complex, highly twisted, large-scale, nonlinear manifold learning. We verified the accuracy and effectiveness of H-ISOSOM quantitatively by three parameters: MQE, SNR and STDQE on five standard and challenging synthetic data sets. We also applied it to hand pose estimation. The experiments show that H-ISOSOM generally outperforms GHSOM and SOM.

## References

- [1] V. Athitsos and S. Sclaroff. Estimating 3D hand pose from a cluttered image. In *Computer Vision and Pattern Recognition (CVPR)*, pages 432–439, June 2003. 1, 2
- [2] T. D. Campos. Regression-based hand pose estimation from multiple cameras. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 782 – 789, June 2006. 1
- [3] M. Dittenbach, D. Merkl, and A. Rauber. The growing hierarchical self-organizing map. In *International Joint Conference on Neural Networks (IJCNN)*, pages 24 – 27, June 2000. 2
- [4] A. Elgammal and C.-S. Lee. Inferring 3d body pose from silhouettes using activity manifold learning. In *IEEE International Conference on Computer Vision and Pattern Recognition*, volume 2. 1
- [5] H. Guan, R. S. Feris, and M. Turk. The isometric self-organizing map for 3d hand pose estimation. In *The 7th International Conference on Automatic Face and Gesture Recognition (FG06)*, pages 263 – 268, April 2006. 2, 3, 4
- [6] T. Kohonen. *Self-Organizing Maps*. Springer Series in Information Sciences, 2001. 2
- [7] M. Kolsch, R. Bane, T. Hoellerer, and M. Turk. Multimodal interaction with a wearable augmented reality system. *IEEE Computer Graphics and Applications*, 26(3):62–71, 2006. 1
- [8] R. Rosales and S. Sclaroff. Inferring body pose without tracking body parts. In *CVPR*, 2000. 1
- [9] S. Roweis and L. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, (5500):2323 –2326, Dec 2000. 1, 3
- [10] G. Shakhnarovich, P. Viola, and T. Darrell. Fast pose estimation with parameter-sensitive hashing. In *IEEE International Conference on Computer Vision*, pages 750–757, Nice, France, 2003. 2
- [11] B. Stenger, A. Thayananthan, P. H. S. Torr, and R. Cipolla. Model-based hand tracking using a hierarchical bayesian filter. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(9):1372–1384, Sept. 2006. 2
- [12] J. B. Tenenbaum, V. D. Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, (5500):2319 –2323, Dec 2000. 1
- [13] H. Zhou and T. S. Huang. Okapi-chamfer matching for articulated object recognition. In *Proc. IEEE International Conference on Computer Vision*, page 1026C1033, Oct 2005. 2