ISOMAP OUT-OF-SAMPLE EXTENSION FOR NOISY TIME SERIES DATA

Hamid Dadkhahi, Marco F. Duarte

Department of Electrical
and Computer Engineering
University of Massachusetts Amherst

Benjamin Marlin *

College of Information
and Computer Sciences
University of Massachusetts Amherst

ABSTRACT

This paper proposes an out-of-sample extension framework for a global manifold learning algorithm (Isomap) that uses temporal information in out-of-sample points in order to make the embedding more robust to noise and artifacts. Given a set of noise-free training data and its embedding, the proposed framework extends the embedding for a noisy time series. This is achieved by adding a spatio-temporal compactness term to the optimization objective of the embedding. To the best of our knowledge, this is the first method for out-of-sample extension of manifold embeddings that leverages timing information available for the extension set. Experimental results demonstrate that our out-of-sample extension algorithm renders a more robust and accurate embedding of sequentially ordered image data in the presence of various noise and artifacts when compared to other timing-aware embeddings.

***Index Terms*—** Dimensionality reduction, manifold learning, time series, out-of-sample extension

1. INTRODUCTION

Recent advances in sensing technology have enabled a massive increase in the dimensionality of data captured from digital sensing systems. Naturally, the high dimensionality of data affects various stages of the digital systems, from acquisition to processing and analysis of the data. To meet communication, computation, and storage constraints, in many applications one seeks a low-dimensional embedding of the high-dimensional data that shrinks the size of the data representation while retaining the information we are interested in capturing. This problem of *dimensionality reduction* has attracted significant attention in the signal processing and machine learning communities.

The traditional method for dimensionality reduction is *principal component analysis* (PCA) [1], which successfully captures the structure of datasets well approximated by a linear subspace. However, in many applications, the data can be best modeled by a *nonlinear manifold* whose geometry cannot be captured by PCA. Manifolds are low-dimensional geometric structures that reside in a high-dimensional ambient space despite possessing merely a few degrees of freedom. *Manifold learning methods* aim to obtain a suitable nonlinear embedding into a low-dimensional space that preserves the local structure present in the higher-dimensional data in order to simplify data visualization and exploratory data analysis. Examples of manifold learning methods include Isomap [2], locally linear embedding [3], Laplacian eigenmaps [4], and Hessian eigenmaps [5].

All the aforementioned dimensionality reduction methods assume that the data points are stationary and independent. However, in many real-world applications in vision we encounter time series data. In recent years, several attempts have been made to take advantage of temporal information of data in order to discover the dynamics underlying the manifold [6–9]. In particular, spatio-temporal Isomap (ST-Isomap) [6] empirically alters the original weights in the graph of local neighbors to emphasize similarity between temporally related points. When the data to be embedded corresponds to a time series, all these temporal frameworks of dimensionality reduction generate better quality embedded spaces than the initial approaches in some sense. However, all of the mentioned methods are designed only for a given set of training points, with no extension of these timing-aware embeddings for out-of-sample points.

On the other hand, many of these dimensionality reduction algorithms (and Isomap in particular) are sensitive to noise and artifacts [10]. This is not desirable for real-world applications since the data are usually contaminated with noise and various artifacts due to imperfect sensors or human mistakes. This sensitivity is particularly relevant for emerging architectures for very low power sensing, which are subject to increased presence of noise and artifacts [11]. Note that the literature on manifold models for noisy data [12–15] fails to (i) address the out-of-sample extension problem in the presence of noise and artifacts, and (ii) take advantage of temporal information among the images.

In this paper, we address the out-of-sample extension of nonlinear manifold embeddings for the specific case where

*This work was supported by NSF Grant IIS-1239341.

978-1-4673-7454-5/15/$31.00 ©2015 IEEE

the points to be extended correspond to a time series. The aim is to use the sequential ordering of data points to improve the resilience of the embedding of out-of-sample points to various noise and artifact models. This improvement is achieved by adding a spatio-temporal compactness constraint for pairs of points within a temporal neighborhood window. Our focus in this paper is on Isomap for the sake of concreteness, but we believe that our formulation is generic enough that it can also be applied to additional nonlinear manifold learning algorithms.

The application scenario of our proposed scheme is as follows. In a training stage for nonlinear manifold learning, we have full control of the environment, which enables us to capture noise-free data. We then use the captured training data to learn the underlying manifold. In the testing stage (which corresponds to the normal operation of the sensors), we capture data points of lower quality, possibly contaminated with different artifacts/noise patterns, in a sequential order (i.e., as a time series). We then extend the original nonlinear manifold embedding to the newly acquired (noisy) samples using our proposed algorithm that leverages their timing information.

2. BACKGROUND

Manifold Models and Manifold Learning: A set of data points $\mathcal{U} = \{u_1, u_2, \dots, u_n\}$ in a high-dimensional ambient space $\mathbb{R}^d$ that have been generated by an event featuring m degrees of freedom correspond to a sampling of an m-dimensional *manifold* $\mathcal{M} \subset \mathbb{R}^d$. Given the high-dimensional data set $\mathcal{U}$, we would like to find the parameterization that has generated the manifold. One way to discover this parametrization is to embed the high-dimensional data on the manifold to a low-dimensional space $\mathbb{R}^m$ so that the local geometry of the manifold is preserved. *Dimensionality reduction methods* are devised so as to preserve such geometry, which is measured by a neighborhood-preserving criteria that varies depending on the specific algorithm.

Isomap: For nonlinear manifold learning, this paper focuses on the *Isomap* algorithm, which aims to preserve the pairwise *geodesic distances* between data points [2]. The geodesic distance is defined as the length of the shortest path between two data points u_i and u_j ($u_i, u_j \in \mathcal{M}$) along the surface of the manifold $\mathcal{M}$ and is denoted by $d_G(u_i, u_j)$. Isomap first finds an approximation to the geodesic distances between each pair of data points in a sampling of the manifold $\mathcal{U}$ by constructing a neighborhood graph in which each point is connected only to its k nearest neighbors; the edge weights are equal to the corresponding pairwise distances. For neighboring pairs of data points, the Euclidean distance provides a good approximation for the geodesic distance, i.e.,

$$d_G(u_i, u_j) \approx \|u_i - u_j\|_2 \quad \text{for} \quad u_j \in \mathcal{N}_k(u_i), \qquad (1)$$

where $\mathcal{N}_k(u_i)$ designates the set of k nearest neighbors in $\mathcal{U}$ to the point $u_i \in \mathcal{U}$. For non-neighboring points, the geodesic distance is estimated as the length of the shortest path along the neighborhood graph, which can be found via Dijkstra's algorithm. Then, multidimensional scaling is applied to the resulting geodesic distance matrix to find a set of low-dimensional points $\mathcal{L} = \{\ell_1, \ell_2, \dots, \ell_n\}$ that best match such distances. More precisely, the centralized squared geodesic distance matrix can be obtained as $\Delta_c = -\frac{1}{2} H_n \Delta_n H_n$, where Δ_n is the matrix of squared geodesic distances for $\mathcal{U}$ (i.e., $(\Delta_n)_{ij} = d_G^2(u_i, u_j)$) and H_n is the centering matrix defined by the formula $(H_n)_{ij} = \delta_{ij} - \frac{1}{n}$. Then, the m-dimensional embedding $\mathcal{L}$ for the data $\mathcal{U}$ is given as the columns of the matrix

$$L = \begin{bmatrix} \sqrt{\lambda_1} \cdot v_1^T \\ \sqrt{\lambda_2} \cdot v_2^T \\ \vdots \\ \sqrt{\lambda_m} \cdot v_m^T \end{bmatrix},$$

where λ_i and v_i are the eigenvalues and eigenvectors of the Δ_c, respectively.

Out-of-Sample Extension for Isomap: Out-of-sample extension (OoSE) generalizes the result of the nonlinear manifold embedding for new data points. Suppose we have n training data points $\mathcal{U}$ and their embedding $\mathcal{L}$. To obtain the extension of the embedding $\mathcal{L}$ to the out-of-sample (testing) set $\mathcal{Y} = \{y_1, y_2, \dots, y_N\}$, let Δ_X denote the $n \times N$ matrix of squared geodesic distances between the N out-of-sample (testing) points $\mathcal{Y}$ and the n training points $\mathcal{U}$. The out-of sample extension $\mathcal{X} = \{x_1, x_2, \dots, x_N\}$ of the embedding $\mathcal{L}$ to the points $\mathcal{Y}$ is given by the columns of the matrix [16]

$$X = \frac{1}{2} L^{\#} (\bar{\Delta}_n \mathbf{1}_N^T - \Delta_X), \qquad (2)$$

where $\mathbf{1}_N$ denotes an all-ones vector of length N, $\bar{\Delta}_n$ is the column mean of Δ_n, i.e., $\bar{\Delta}_n = \frac{1}{n} \Delta_n \mathbf{1}_n$, and $L^{\#}$ is the pseudo-inverse transpose of L, given by

$$L^{\#} = (L^T)^{\dagger} = \begin{bmatrix} v_1^T / \sqrt{\lambda_1} \\ v_2^T / \sqrt{\lambda_2} \\ \vdots \\ v_m^T / \sqrt{\lambda_m} \end{bmatrix}.$$

Note that, to the best of our knowledge, the literature on out-of-sample extension does not exploit the sequential ordering of data to mitigate the possibility of embedding errors due to the presence of noisy and contaminated data.

Spatio-Temporal Isomap: In Isomap, the neighhorhood graph is formed by linking each point in $\mathcal{U}$ to its k-nearest neighbors in the same set. *Spatio-Temporal Isomap* (ST-Isomap) leverages sequence/timing information $\{(u_i, t_i)\}$ present for each of the points $u_i \in \mathcal{U}$. ST-Isomap appends edges between pairs of *adjacent temporal neighbors* (ATN), i.e., pairs of immediate temporal neighbors where $t_j = t_i \pm 1$, to the neighborhood graph. The addition of ATN edges to the neighborhood graph introduces a first-order Markov dependency into the resulting embedding. The distances between ATNs are scaled down by a factor given by the constant parameter c_{ATN}, which emphasizes the correlation between a point and its adjacent temporal neighbors.

ST-Isomap also modifies the graph distances of a subset of the k-nearest neighbors of each point that satisfy certain spatio-temporal conditions. First, the set of points in a temporal window of size ϵ around each point u_i is considered as its *trivial matches*, denoted by $\mathcal{T}_\epsilon(u_i)$. Suppose that the point $u_i^* \in \mathcal{T}_\epsilon(u_i)$ is the closest trivial match to point u_i, i.e., $d_G(u_i, u_i^*) = \min_{u_k \in \mathcal{T}_\epsilon(u_i)} d_G(u_i, u_k)$. Now, the subset of k-nearest neighbors with distances at most $d_G(u_i, u_i^*)$ from u_i are considered as *non-trivial matches* and the resulting set is referred to as *common temporal neighbors* (CTN):

$$\mathrm{CTN}(u_i) = \{u_j \in \mathcal{N}_k(u_i) : d_G(u_i, u_j) \leq d_G(u_i, u_i^*)\}.$$

In a sense, CTN are used to identify data points in the local spatial neighborhood of each point u_i that are more likely to be analogous to u_i. Finally, the constant parameter c_{CTN} is used to emphasize the similarity between each point and its CTN set via reducing the corresponding distances by a scaling factor of c_{CTN}.

Note that ST-Isomap is devised to uncover spatio-temporal structure underlying manifold data, rather than faithfully recover the embedding of data contaminated by noise or artifact models. In addition, ST-Isomap does not address the out-of-sample extension problem. Nonetheless, we formulate an adaptation of out-of-sample extension from Isomap to ST-Isomap in Section 4.

3. PROPOSED ALGORITHM

We propose a new algorithm for out-of-sample extension of Isomap nonlinear manifold embeddings that also aims to leverage temporal information in order to improve the quality of the embedding of noisy and corrupted time series data.

Recall that the embedding of out-of-sample points is given by the matrix equation (2). Alternatively, the embedding $\mathcal{X} = \{x_1, x_2, \ldots, x_N\}$ of out-of-sample points $\mathcal{Y} = \{y_1, y_2, \ldots, y_N\}$ can be obtained via the following optimization problem:

$$X = \arg\min_X \left\| \frac{1}{2} \left(\bar{\Delta}_n \mathbf{1}_N^T - \Delta_X \right) - L^T X \right\|_F^2, \qquad (3)$$

where $\|.\|_F$ denotes the Frobenius matrix norm, and the columns of the matrix X correspond with the embeddings in $\mathcal{X}$. In our problem setup, we assume that the out-of-sample points $\{y_1, y_2, \ldots, y_N\}$ have been sampled at time instances $\{t_1, t_2, \ldots, t_N\}$, respectively. We define the set of temporal distances τ in the following way: $\tau = \{\tau_{ij} = |t_i - t_j| : i, j = 1, \ldots, N\}$.

To characterize the spatio-temporal similarity, we first define the *spatio-temporal compactness function* $\mathrm{C}_\omega(\mathcal{Y}, \tau)$ of geodesic distances and temporal information for the time series:

$$\mathrm{C}_\omega(\mathcal{Y}, \tau) \coloneqq \sum_{i=1}^{N} \sum_{j \in \mathcal{T}_K(y_i)} d_G^2(y_i, y_j) \cdot \omega(\tau_{ij}), \qquad (4)$$

where ω is a temporal weighting function and $\mathcal{T}_K(y_i)$ is the set of K nearest temporal neighbors to point y_i in $\mathcal{Y}$. Note that $K \neq k$ from (1) in general. In this work, the identity function is used for the temporal weighting function, i.e., $\omega(\tau_{ij}) = \tau_{ij}$. Note that there could be many other choices for the temporal weighting function. This particular choice is made so as to pull the more temporally distant points closer to each other.

We assume that the geodesic distances in the ambient space of each temporal neighborhood of the out-of-sample data points should be small so that the compactness term in (4) is also small. We aim to incorporate this compactness in the embedding framework. This can be achieved by leveraging the fact that the Euclidean distances in the embedding space are matched to their geodesic counterparts in the ambient space. Therefore, we modify the expression for the compactness by leveraging the expected relationship between the original geodesic distances for $\mathcal{Y}$ and the Euclidean distances of the embedded data points X: $d_G^2(y_i, y_j) \approx \|x_i - x_j\|_2^2 = \mathrm{Tr}(B_{ij} X^T X)$, where $\mathrm{Tr}(.)$ represents the trace operator and B_{ij} is an all zeros matrix of size $N \times N$ except for four elements, $B_{ii} = B_{jj} = 1$ and $B_{ij} = B_{ji} = -1$. The compactness term in (4) can then be reformulated as

$$\mathrm{C}_\omega(X, \tau) = \sum_{i=1}^{N} \sum_{j \in \mathcal{T}_K(y_i)} \left(\mathrm{Tr}(B_{ij} X^T X) \cdot \omega_{ij} \right),$$

where $\omega_{ij} = \omega(\tau_{ij})$.

We incorporate the temporal information among the sequence of out-of-sample points into the objective function of the optimization as follows:

$$\begin{aligned} X &= \arg\min_X \left\| \frac{1}{2} \left(\bar{\Delta}_n \mathbf{1}_N^T - \Delta_X \right) - L^T X \right\|_F^2 + \lambda \cdot \mathrm{C}_\omega(X, \tau) \\ &= \arg\min_X \|Q - PX\|_F^2 + \lambda \cdot \mathrm{C}_\omega(X, \tau). \end{aligned} \qquad (5)$$

where $\lambda > 0$ is a Lagrangian multiplier, and we denote $P = L^T$ and $Q = \frac{1}{2}(\bar{\Delta}_n \mathbf{1}_N^T - \Delta_X)$ for brevity. In words, the optimization balances a tradeoff between the fidelity of the embedding and the desired spatio-temporal compactness among successive points. Note that $\|A\|_F^2 = \mathrm{Tr}(A^T A)$. Hence, the first term in (5) can be written as

$$\|Q - PX\|_F^2 = \mathrm{Tr}((Q - PX)^T (Q - PX)). \qquad (6)$$

Furthermore, we can rewrite the compactness as follows:

$$\begin{aligned} \mathrm{C}_\omega(X,\tau) &= \mathrm{Tr}\left(\sum_{i=1}^{N}\sum_{j\in\mathcal{T}_K(y_i)} \omega_{ij}B_{ij}X^TX\right) \\ &= \mathrm{Tr}\left(AX^TX\right), \end{aligned} \tag{7}$$

where the matrix $A = \sum_{i=1}^{N}\sum_{j\in\mathcal{T}_K(y_i)}\omega_{ij}B_{ij}$. Plugging $\|Q - PX\|_F^2$ and $\mathrm{C}_\omega(X,\tau)$ from Equations (6) and (7) into the optimization in (5) yields

$$\begin{aligned} X &= \arg\min_X \mathrm{Tr}\left((Q-PX)^T(Q-PX)\right) + \lambda\cdot\mathrm{Tr}(AX^TX) \\ &= \arg\min_X \mathrm{Tr}\left(Q^TQ - 2Q^TPX + X^TP^TPX + \lambda AX^TX\right) \\ &\overset{(a)}{=} \arg\min_X \mathrm{Tr}\left(CX\right) + \mathrm{Tr}\left(DXX^T\right) + \lambda\mathrm{Tr}\left(AX^TX\right), \end{aligned} \tag{8}$$

where in (a) the constant matrix Q^TQ has been dropped from the objective function, we denote $C = -2Q^TP$ and $D = P^TP$, and use the fact that $\mathrm{Tr}\left(X^TDX\right) = \mathrm{Tr}\left(DXX^T\right)$ due to invariance of trace under cyclic permutations. Next, we denote the objective function in (8) by J, and take the derivative of J with respect to the embedding matrix X as follows:

$$\begin{aligned} \frac{\partial J}{\partial X} &= C^T + (D + D^T)X + \lambda\cdot X(A + A^T) \\ &\overset{(b)}{=} C^T + 2DX + 2\lambda XA, \end{aligned} \tag{9}$$

where (b) is due to the matrices D and A being symmetric. Solving $\frac{\partial J}{\partial X} = \mathbf{0}$ for X gives us the solution to the optimization in (8), where $\mathbf{0}$ is an all-zeros matrix of size $N \times m$. In order to solve the matrix equation $\frac{\partial J}{\partial X} = \mathbf{0}$, we use the Kronecker product and the vectorized format of each term:

$$\begin{aligned} \mathrm{vect}\left(\frac{\partial J}{\partial X}\right) &= \mathrm{vect}(C^T + 2DX + 2\lambda XA) \\ &= \mathrm{vect}(C^T) + 2\,(I\otimes D)\,\mathrm{vect}(X) \\ &\quad + 2\lambda\,(A^T\otimes I)\,\mathrm{vect}(X), \end{aligned} \tag{10}$$

where $\otimes$ designates the Kronecker product operator and I is the identity matrix. Setting (10) equal to zero and solving for $\mathrm{vect}(X)$ provides the embedding given by the solution to (5):

$$\mathrm{vect}(X) = -\frac{1}{2}\Big(\lambda\cdot(A^T\otimes I) + (I\otimes D)\Big)^{-1}\cdot\mathrm{vect}(C^T). \tag{11}$$

Note that the complexity of (11) is $\mathcal{O}(\max\{nmN, (mN)^t\})$, where t is determined by the complexity of matrix inversion.

4. NUMERICAL EXPERIMENTS

We investigate the robustness of the proposed algorithm to two types of artifacts. For our experiments, we use a custom eye-tracking dataset of 111×112-pixel captures from a computational eyeglasses prototype [11]. The prototype features low-power cameras mounted on a pair of eyeglasses that are trained on the user's eyes, with the goal of tracking the gaze location of the user over time. The dataset contains $n + N = 900 + 100$ images and the dimensionality of the learned manifold is $m = 2$. A uniform sequential ordering exists among the N out-of-sample points. A subset of consecutive out-of-sample points is shown in Fig. 1. We start from a dataset of original images that we treat as "clean" images, and we synthesize noisy images by adding noise to them. We consider salt-and-pepper noise and motion blurring caused by camera movement. Example noisy images are shown in Fig. 1.

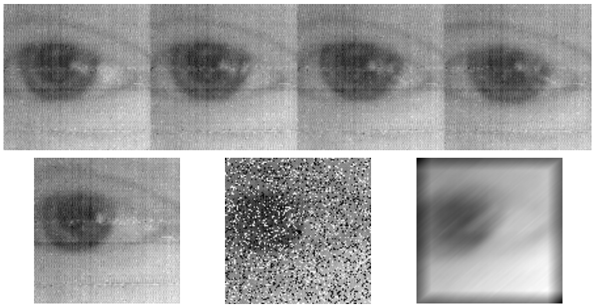


Fig. 1: Top: Example images from the eyeglasses dataset used for out-of-sample extension. The sequence depicts every second image in a sequence of 8 successive data points. Bottom: Sample noisy images. From left to right: noise-free, salt-and-pepper noise with $p = 0.3$, motion blur with length of 30 pixels and angle of 45°.

The experiments in this section pursue the following general framework. First, we obtain the Isomap embedding for n training points. Next, we obtain the OoSE of both clean and noisy N out-of-sample points. The OoSE of clean points is considered as a reference for performance measure, and we would like the embedded noisy version of out-of-sample points to be as close to that of clean data as possible. We set $k = 20$ and $K = 10$ in all the experiments. For each value of the noise parameter, we generate 6 instances of noise. We use the first instance of noise to find the value of the parameter λ by a grid search. The selected value of λ (which is dependent on the value of the noise parameters) is used with the proposed algorithm on the remaining instances of noise. Finally, we average the error over the latter instances.

Note that for Isomap we cannot directly compare the two sets of out-of-sample points [17], as the embeddings learned from different samplings of the manifold are often subject to translation, rotation, and scaling. These variations must be addressed via manifold alignment before the embedded points are compared. We find the optimal alignment of the clean and noisy embeddings via *Procrustes analysis* [18] and apply the resulting translational, rotational, and scaling components on the OoSE manifold. Finally, we measure the OoSE error as the average ℓ_2 distance between matching points in the two embeddings, i.e., the out-of-sample extension of aligned clean and denoised testing data: $E = \frac{1}{N}\sum_{i=1}^{N}\|z_i - w_i\|_2$, where $\mathcal{Z} = \{z_1, z_2, \ldots, z_N\}$ is the out-of-sample extension of clean data via Isomap OoSE, and $\mathcal{W} = \{w_1, w_2, \ldots, w_N\}$ is the OoSE from noisy data for the algorithm under test after alignment with $\mathcal{Z}$.

We compare the performance of the proposed algorithm against an adaptation of Isomap out-of-sample extension to ST-Isomap. To obtain the out-of-sample extension of ST-Isomap, we find the distances of the out-of-sample points with training points via the neighborhood graph of the ST-algorithm, and then use (2) to find the embedding for the out-of-sample points. Note that since Isomap and ST-Isomap use different graphs regardless of parameter values, the embedding of clean data via ST-Isomap differs from that of Isomap. Hence, we cannot directly compare the two embedding spaces. Thus we first align the two embedding spaces, and then evaluate the error between the aligned embedding obtained via ST-Isomap on clean and noisy data. We set the value of the parameter c_{ATN} to one as experimentally this selection produces the minimum error. We then perform a grid search over the window size ϵ and parameter c_{CTN}, in a similar manner to the evaluation of the parameter of the proposed algorithm, i.e., using 6 instances of noise. Note that we are giving an advantage to ST-Isomap out-of-sample extension over our proposed method by supplying temporal information about both training and out-of-sample points to the algorithm.

Salt-and-Pepper Noise: We consider salt-and-pepper noise where each pixel's intensity is randomly flipped with a probability of p to either zero or one, with both having equal probability $\frac{p}{2}$. We vary p from 0.1 to 0.9 with a step size of 0.2. The selected values for the parameter λ are $[0.05, 0.1, 0.5, 1.8, 2.7] \times 10^5$, respectively. Figure 4a compares the performance of the proposed algorithm at different noise levels against that of Isomap OoSE and ST-Isomap OoSE. As can be observed from this figure, the average ℓ_2-norm error of the proposed algorithm is substantially lower than those of Isomap and ST-Isomap OoSEs.

Figure 2a depicts the Isomap trajectory for the clean out-of-sample points. Its noisy counterpart with the addition of an instance of salt-and-pepper noise with $p = 0.5$ is shown in Figure 2b. In each plot, sequentially adjacent points are connected by a blue line and temporal order is color coded from blue to red. We obtained the out-of-sample extension of the noisy data by the proposed algorithm with two settings of the parameter λ; 0.3×10^5, and 20×10^5 as depicted in Figures 2c, and 2d, respectively. Among the mentioned values of the parameter λ, $\lambda = 0.3 \times 10^5$ produces the minimum ℓ_2-norm error in the recovered trajectory when compared to Isomap trajectory for the clean out-of-sample points. This can be verified by the similarity of the trajectory shown in Figure 2c to that of Figure 2a. As can be observed from the figure, increasing the value of the parameter λ eventually converts the embedding's trajectory to an almost-straight curve, as depicted in Figure 2d, where $\lambda = 20 \times 10^5$. This is to be expected since eventually the spatial information will be neglected and the embedding will retain only the temporal information of the time series. Note that setting $\lambda = 0$ in the proposed algorithm, converts it into the plain Isomap OoSE and returns the noisy trajectory, i.e., Figure 2b. Figure 3 demonstrates the aligned ST-Isomap OoSE trajectory on clean and noisy data, respectively.

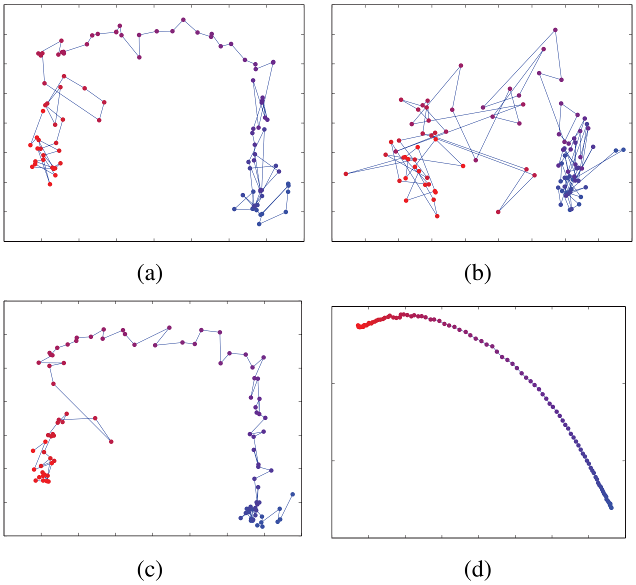


Fig. 2: Effect of the value of the parameter λ. (a) Isomap OoSE trajectory for clean data, (b) Isomap OoSE trajectory for noisy data with salt-and-pepper noise with $p = 0.5$, OoSE of noisy data via the proposed algorithm for (c) $\lambda = 0.3 \times 10^5$, and ($d$) $\lambda = 20 \times 10^5$, respectively.

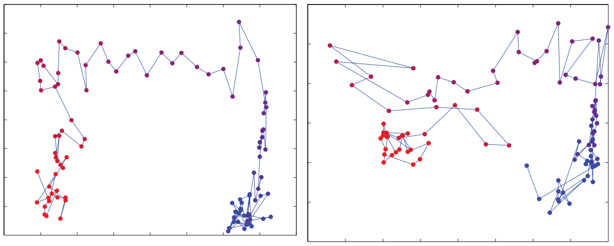
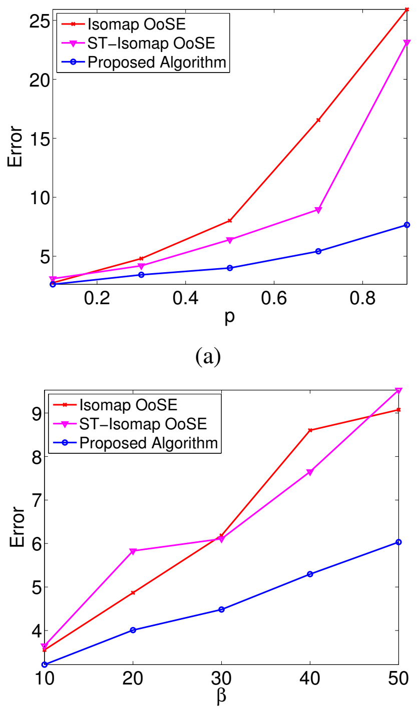


Fig. 3: ST-Isomap OoSE trajectory for (left) clean and (right) noisy data with salt-and-pepper noise with $p = 0.5$.

Motion Blur: We use a linear motion model to simulate the motion blur artifact. The artifact is applied via convolution by a filter that approximates the linear motion of a camera by η pixels, with an angle of θ degrees in a counterclockwise direction. We consider a Gaussian model for θ with zero mean and variance of $\sigma = 45$ degrees. In order to model the motion length η, we use a Gamma distribution (with PDF of $p(x) = \frac{1}{\Gamma(\alpha)\beta^\alpha} x^{\alpha-1} \exp(\frac{-x}{\beta})$, where $\Gamma(.)$ denotes the Gamma function) with shape parameter $\alpha = 1$ and vary the scale parameter β from 10 to 50 by step size of 10 in order to produce different strengths of motion blur. The selected values for the parameter λ are $[0.5, 1.5, 2, 2.5, 2.5] \times 10^4$, respectively. Figure 4b compares the performance of the proposed algorithm at different values of the scale parameter β against that of



(a)

(b)

Fig. 4: Performance evaluation for OoSE of the data points contaminated by (a) salt and pepper noise and (b) motion blur.

Isomap OoSE and ST-Isomap OoSE.

5. DISCUSSION

In this paper we devised an extension of Isomap for sequentially ordered out-of-sample data points. Numerical experiments indicate the robustness of the proposed algorithm against different noise and artifact models. The smoothness/behavior of the embedding is determined by the regularization parameter λ, the optimal value of which depends on the level of noise/artifacts. The more noise is present in the data, the larger the value of the parameter λ needs to be, in order to recover the embedding of the clean out-of-sample data. Future work can address automatic selection of the regularization parameter as a function of noise parameters.

In the future, we will evaluate the performance of the proposed algorithm on other data sets as well. In addition, it would be interesting to explore the possibility of employing the same idea of spatio-temoral compactness to other manifold learning algorithms such as LLE.

6. REFERENCES

[1] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*, Springer-Verlag, Secaucus, NJ, 2006.

[2] J. B. Tenenbaum, V. de Silva, and J. C. Langford, “A global geometric framework for nonlinear dimensionality reduction,” *Science*, vol. 290, no. 5500, pp. 2319–2323, 2000.

[3] S. T. Roweis and L. K. Saul, “Nonlinear dimensionality reduction by locally linear embedding,” *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.

[4] M. Belkin and P. Niyogi, “Laplacian eigenmaps for dimensionality reduction and data representation,” *Neural Computation*, vol. 15, no. 6, pp. 1373–1396, Mar. 2003.

[5] D. L. Donoho and C. Grimes, “Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data,” *Proc. Nat. Acad. Sciences (PNAS)*, vol. 100, no. 10, pp. 5591–5596, May 2003.

[6] O. C. Jenkins and M. J. Matarić, “A spatio-temporal extension to Isomap nonlinear dimension reduction,” in *Int. Conf. Machine Learning (ICML)*, 2004, pp. 441–448.

[7] M. Lewandowski, J. Martinez-del Rincon, D. Makris, and J. C. Nebel, “Temporal extension of Laplacian eigenmaps for unsupervised dimensionality reduction of time series,” in *Int. Conf. Pattern Recognition (ICPR)*, Washington, DC, USA, 2010, pp. 161–164.

[8] R. S. Lin, C. B. Liu, M. H. Yang, N. Ahuja, and S. E. Levinson, “Learning nonlinear manifolds from time series,” in *European Conf. Computer Vision (ECCV)*, 2006, vol. 3952, pp. 245–256.

[9] A. Rahimi, B. Recht, and T. Darrell, “Learning appearance manifolds from video,” in *Computer Vision and Pattern Recognition (CVPR)*, San Diego, CA, USA, 2005, pp. 868–875.

[10] M. Balasubramanian and E. L. Schwartz, “The Isomap algorithm and topological stability,” *Science*, vol. 295, no. 5552, pp. 7, 2002.

[11] A. Mayberry, P. Hu, B. Marlin, C. Salthouse, and D. Ganesan, “iShadow: Design of a wearable, real-time mobile gaze tracker,” in *Int. Conf. Mobile Systems, Applications and Services (MobiSys)*, Bretton Woods, NH, June 2014, pp. 82–94.

[12] H. Chen, G. Jiang, and K. Yoshihira, “Robust nonlinear dimensionality reduction for manifold learning,” in *Int. Conf. Pattern Recognition (ICPR)*, Hong Kong, 2006, vol. 2, pp. 447–450.

[13] M. Hein and M. Maier, “Manifold denoising,” in *Neural Info. Proc. Systems (NIPS)*, Vancouver, Canada, 2006, pp. 561–568.

[14] Y. Gao, K. L. Chan, and W. yun Yau, “Manifold denoising with gaussian process latent variable models,” in *Int. Conf. Pattern Recognition (ICPR)*, Tampa, FL, 2008, pp. 1–4.

[15] D. Gong, F. Sha, and G. Medioni, “Locally linear denoising on image manifolds,” in *Int. Conf. Artificial Intelligence and Statistics (AISTATS)*, Sardinia, Italy, 2010, pp. 265–272.

[16] Y. Bengio, J.-F. Paiement, and P. Vincent, “Out-of-sample extensions for LLE, Isomap, MDS, Eigenmaps, and spectral clustering,” in *Neural Info. Proc. Systems (NIPS)*, Vancouver, Canada, 2003, pp. 177–184.

[17] F. Dornaika and B. Raduncanu, “Out-of-sample embedding for manifold learning applied to face recognition,” in *IEEE Conf. Computer Vision and Pattern Recognition Workshops (CVPRW)*, Portland, OR, June 2013, pp. 862–868.

[18] C. Wang and S. Mahadevan, “Manifold alignment using procrustes analysis,” in *Int. Conf. Machine Learning (ICML)*, New York, NY, 2008, pp. 1120–1127.