SIAM J. Imaging Sciences
Vol. 12, No. 3, pp. 1347–1391

Intrinsic Isometric Manifold Learning with Application to Localization*

Ariel Schwartz† and Ronen Talmon†

Abstract. Data living on manifolds commonly appear in many applications. Often this results from observing an inherently latent low-dimensional system via higher-dimensional measurements. We show that, under certain conditions, it is possible to construct an intrinsic and isometric data representation for such data which respects an underlying latent intrinsic geometry. Namely, we view the observed data only as a proxy and learn the structure of a latent unobserved intrinsic manifold, whereas common practice is to learn the manifold of the observed data. For this purpose, we build a new metric and propose a method for its robust estimation by assuming mild statistical priors and by using artificial neural networks as a mechanism for metric regularization and parametrization. We show a successful application to unsupervised indoor localization in ad hoc sensor networks. Specifically, we show that our proposed method facilitates accurate localization of a moving agent from imaging data it acquires. Importantly, our method is applied in the same way to two different imaging modalities, thereby demonstrating its intrinsic and modality-invariant capabilities.

Key words. manifold learning, intrinsic, isometric, metric estimation, inverse problem, sensor invariance, positioning

AMS subject classifications. 57R40, 57M50, 65J22, 57R50

DOI. 10.1137/18M1198752

1. Introduction.

Making measurements is an integral part of everyday life and, particularly, exploration and learning. However, we are usually interested not in the values of the acquired measurements themselves but rather in understanding the latent system which drives these measurements and might not be directly observable. For example, when considering a radar system, we are interested not in the pattern of the electromagnetic wave received and measured by the antenna but rather in the location, size, and velocity of the object captured by the wave reflection pattern. This example highlights the difference between *observed* properties, which can be directly measured, and *intrinsic* properties corresponding to the latent driving system. The importance of this distinction becomes central when observing the same latent system in multiple ways; the different measurements and observation modalities may have different observed properties, yet they share some common intrinsic properties and structures since they are all generated by the same driving system.

Different observation modalities are not all equally suitable for examining the underlying driving system. It is often the case that systems, which are governed by a small number

*Received by the editors July 5, 2018; accepted for publication (in revised form) April 8, 2019; published electronically July 30, 2019.
https://doi.org/10.1137/18M1198752
Funding: This research was supported by the European Union's Seventh Framework Programme (FP7) under Marie Curie grant 630657, by the Technion Hiroshi Fujiwara Cyber Security Research Center, and by the Israel Cyber Bureau.
†Viterbi Faculty of Electrical Engineering, Technion – Israel Institute of Technology, Haifa, 3200003, Israel (ariels@campus.technion.ac.il, ronen@ee.technion.ac.il).

of variables (and are hence inherently low-dimensional), are observed via redundant complex high-dimensional measurements, obscuring the true low-dimensional and much simpler nature of the latent system of interest. As a result, when trying to tackle signal processing and machine learning tasks, a considerable amount of effort is initially put into choosing or building an appropriate representation of the observed data.

Much research in the fields of machine learning and data mining has been devoted to methods for constructing data representations in an unsupervised fashion, directly from observed data, aiming to lower the dimensionality of the observations, simplify them, and uncover properties and structures of their latent driving system [6]. This field of research has recently gained considerable attention due to the ability to acquire, store, and share large amounts of information, leading to the availability of large scale data sets. Such large data sets are often generated by systems which are not well understood and for which tailoring data representations based on prior knowledge is impossible.

A particular representation learning subclass of methods is manifold learning [6, 56, 15, 5, 46, 63, 42]. In manifold learning, it is assumed that observed data lie near a low-dimensional manifold embedded in a higher-dimensional ambient observation space. Typically the goal then is to embed the data in a low-dimensional Euclidean space while preserving some of their properties.

Most manifold learning methods operate directly on the observed data and rely on their geometric properties [56, 6, 15, 18, 27, 42, 43, 48, 58]. This observed geometry, however, can be significantly influenced by the observation modality, which is often arbitrary and without a clear known connection to the latent system of interest. Consequently, it is subpar to preserve observed properties, which might be irrelevant, and instead, one should seek to rely on intrinsic properties of the data. Such properties are inherent to the latent system of interest and invariant to the observation modality.

Intrinsic geometry can be especially beneficial when it adheres to some useful structure, such as that of a vector space. Indeed, there exist many systems whose dynamics and state space geometry can be simply described within an ambient Euclidean space. Unfortunately, when observing such systems via an unknown nonlinear observation function, this structure is typically lost. In such cases, it is advantageous to not only locally circumvent the influence of the observation function but also explicitly preserve the global intrinsic geometry of the latent system.

We claim that in many situations, methods which are both intrinsic and globally isometric are better suited for manifold learning. In this paper, we propose a dimensionality reduction method which robustly estimates the push-forward metric of an unknown observation function using a parametric estimation implemented by an artificial neural network (ANN). The estimated metric is then used to calculate local intrinsic isometric geometric properties of the latent system state space directly from observed data. Our method then uses these geometric properties as constraints for embedding the observed data into a low-dimensional Euclidean space. The proposed method is thus able to uncover the underlying geometric structure of a latent system from its observations without explicit knowledge of the observation model.

The structure of this paper is as follows. In section 2 we present the problem of mapping and localization based on measurements acquired using an unknown observation model. This problem motivates the development of an intrinsic isometric manifold learning method. In

section 3 we formulate the problem mathematically. In section 4 we present our dimensionality reduction method, which makes use of the push-forward metric as an intrinsic metric. In section 5 we propose a robust method for estimating this intrinsic metric directly from the observed data using an ANN. In section 6 we present the results of our proposed algorithm on synthetic data sets. We also revisit the localization problem described in section 2 for the specific case of image data. We show that the proposed intrinsic isometric manifold learning approach allows for sensor invariant positioning and mapping in realistic conditions using highly nonlinear observation modalities. In section 7 we conclude the work, discuss some key issues, and outline possible future research directions.

2. The localization problem and motivation. Our motivating example is that of mapping a 2-dimensional region and positioning an agent within that region based on measurements it acquires. We denote the set of points belonging to the region by $\mathcal{X} \subseteq \mathbb{R}^2$ and the location of the agent by $\mathbf{x} \in \mathcal{X}$ (as illustrated in Figure 1). At each position the agent makes measurements $\mathbf{y} = f(\mathbf{x})$, which are functions of its location. The values of the measurements are observable to us; however, we cannot directly observe the shape of $\mathcal{X}$ or the location of the agent $\mathbf{x}$; therefore, they represent a latent system of interest.

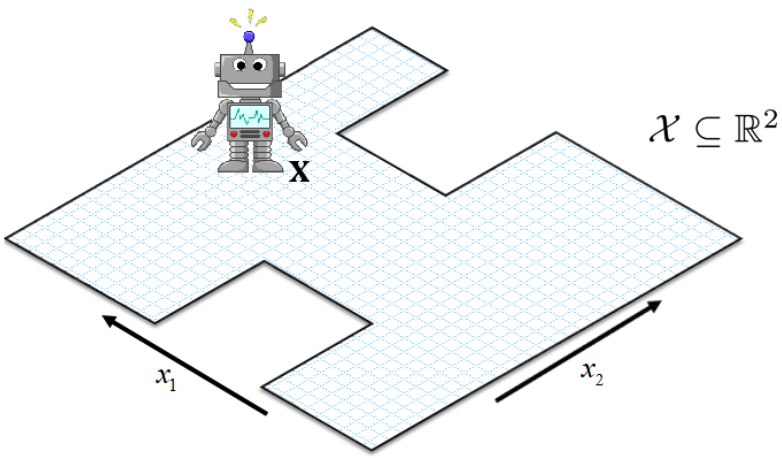


Figure 1. *Agent in a 2-dimensional space.*

We simultaneously consider two distinct possible measurement models as described in Figure 3. The first measurement modality described in Figure 3(a) is conceptually similar to measuring received signal strength (RSS) values at $\mathbf{x}$ from several transmitting antennas located at different locations. Such measurements typically decay as the agent is further away from the signal origin. The second possible measurement model, visualized in Figure 3(b), consists of measurements which are more complex and do not have an apparent interpretable connection to the location of the agent. Although these measurement modalities are 3-dimensional, the set of acquired measurements reside on a 2-dimensional manifold embedded in 3-dimensional ambient observation space (as visualized in Figure 2) due to the inherent 2-dimensional nature of the latent system.

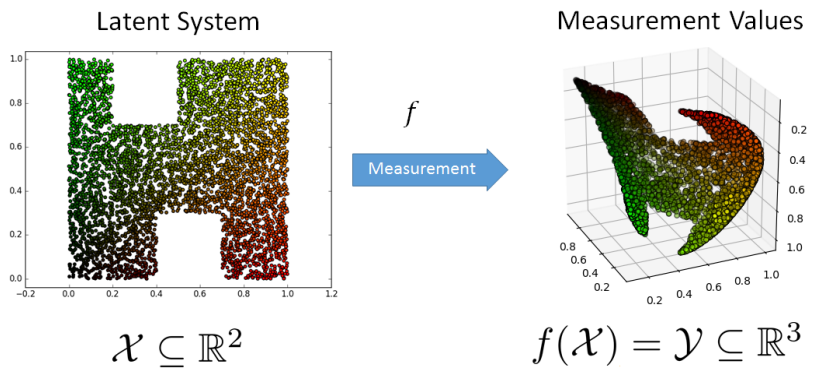
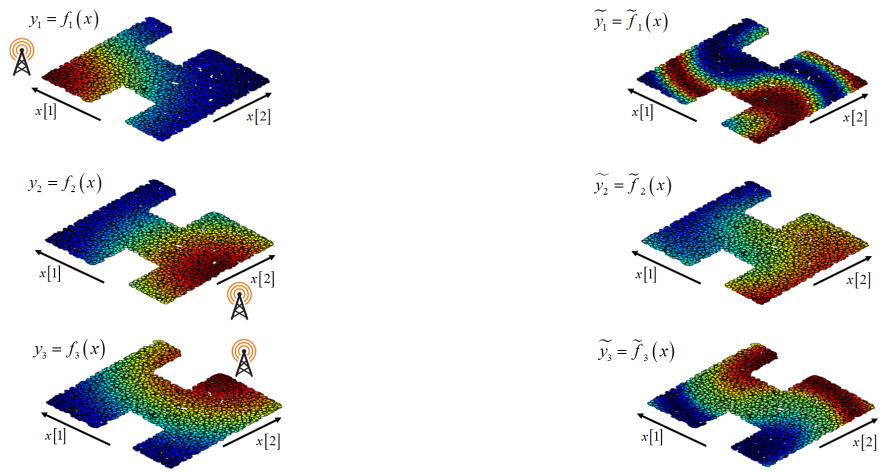


Figure 2. *Creation of the observed manifold.*



(a) First measurement modality. (b) Second measurement modality.

Figure 3. *Two different observation modalities.* (a) *A "free-space" signal strength decay model. The antenna symbol represents the location from which the signal originates.* (b) *A second measurement modality of arbitrary and unknown nature. In both cases the variation in color represents different measurement values.*

The observation functions, f and $\widetilde{f}$, are both bijective on $\mathcal{X}$; i.e., no two locations in $\mathcal{X}$ correspond to the same set of measurement values. Therefore, one can ask whether it is possible to infer the location of the agent $\mathbf{x}$ given the corresponding measurement values. If the measurement model represented by the observation function is known, this becomes a standard problem of model-based localization, which is a specific case of nonlinear inverse problems [23]. However, if the observation function is unknown or depends on many unknown parameters, recovering the location from the measurements becomes much more challenging. The latter case is representative of many real-life scenarios, for example, acoustic and electromagnetic wave propagation models, which are complex and depend on many a priori unknown factors (such as room geometry, locations of walls, reflection, transmission, and absorption of materials, etc.). Image data can also serve as an indirect measurement of position, as we

discuss in subsection 6.2. Although the image acquisition model is well known, the actual image acquired also depends on the geometry and look of the surrounding environment. If the surrounding environment is not known a priori, then this observation modality also falls into the category of observation via an unknown model.

In this work, we will consider the case of observation via an unknown model and address the following question: is it possible to retrieve $\mathbf{x}$ from $f(\mathbf{x})$ without knowing f?

Given that the described problem involves the recovery of a low-dimensional structure from an observable manifold embedded in a higher-dimensional ambient space, we are inclined to attempt a manifold learning approach. Unfortunately, standard manifold learning algorithms do not yield satisfactory results, as can be seen in Figure 4. While some of the methods provide an interpretable low-dimensional parametrization of the latent space by preserving the general topology of the manifold, none of the tested methods recover the actual location of the agent or the structure of the region it moves in.

This motivating example demonstrates the inadequacy of existing manifold learning algorithms for uncovering the intrinsic geometric structure of a latent system of interest from its observations. We attribute this inadequacy to two main factors: the lack of *intrinsicness* and the lack of *geometry preservation* or *isometry*.

Existing manifold learning algorithms generate new representations of data while preserving specific *observed* properties. When data originates from a latent system measured via an observation function, the observed measurements are affected by the specific (and often arbitrary) observation function, which in turn affects the learned low-dimensional representation. As a consequence, the same latent system observed via different observation modalities is represented differently when manifold learning methods are applied (as can be seen in Figure 4). This dependence on the observation modality fails to capture the intrinsic characteristics of the different observed manifolds, which originate from the same latent low-dimensional system.

Settings in which one is interested only in the properties of a latent system observed using an unknown measurement model necessitate manifold learning algorithms which are *intrinsic*; i.e., their learned representation is independent of the observation function and invariant to the way in which the latent system is measured.

Intrinsicness by itself, however, is not enough to retrieve the latent geometric structure of the data. As can be seen in Figure 4, even when manifold learning methods are applied directly to the low-dimensional latent space, thus avoiding any observation function, most methods do not guarantee the preservation of the geometric structure of the latent low-dimensional space. In order to explicitly preserve the latent geometry, the learned representation needs to also be *globally isometric*, i.e., distance preserving not only on a local but also on a global scale.

In this paper, we present a manifold learning method which is *isometric with respect to the latent intrinsic geometric structure*. We will show that, under certain assumptions, our method allows for the retrieval of $\mathbf{x}$ and $\mathcal{X}$ from observed data without requiring explicit knowledge of the specific observation model, thus solving a "blind inverse problem." In the context of our motivating problem, we show that using the suggested method we can achieve sensor-invariant localization and mapping, enabling the application of the same positioning algorithm to problems from a broad range of signal domains with multiple types of sensors.

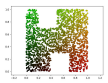
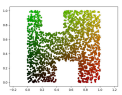
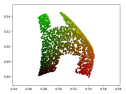
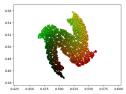
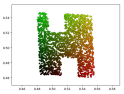
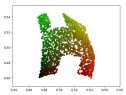
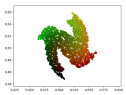
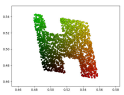
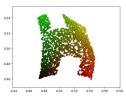
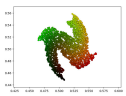
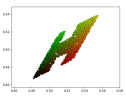
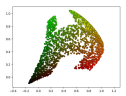
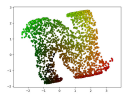
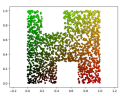
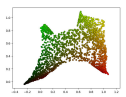
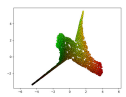
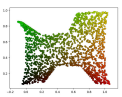
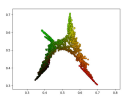
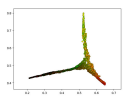
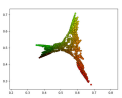
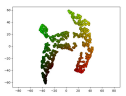
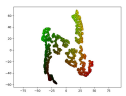
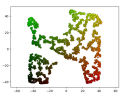
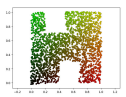
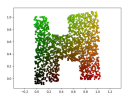
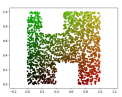
Intrinsic space			
Observation modality	Modality 1	Modality 2	Direct
Observed space			
MLLE			
HLLE			
LTSA			
MDS			
Isomap			
Diffusion Map			
t-SNE			
Proposed method			

Figure 4. *Manifold learning for positioning and mapping based on observed measurements. The first two columns correspond to the two different observation modalities. The third column represents a direct observation of the latent space, circumventing the observation function. The different rows represent the application of various popular manifold learning methods to the observed manifold. The bottom row represents the results achieved by our method.*

3. Problem formulation. Let $\mathcal{X}$ be a path-connected n-dimensional manifold embedded in $\mathbb{R}^n$. We refer to $\mathcal{X}$ as the intrinsic manifold, and it represents the set of all the possible latent states of a low-dimensional system. These states are only indirectly observable via an observation function $f : \mathcal{X} \to \mathbb{R}^m$, which is a continuously differentiable injective function that maps the latent manifold into the observation space $\mathbb{R}^m$. The image of f is denoted by $\mathcal{Y} = f(\mathcal{X})$ and is referred to as the observed manifold. Since f is injective, we have that $n \leq m$. Here, we focus on a discrete setting and define two finite sets of sample points from $\mathcal{X}$ and $\mathcal{Y}$. Let $\mathbf{X} = [\mathbf{x}_1|\mathbf{x}_2|\dots|\mathbf{x}_N]$ be a data matrix whose columns are N intrinsic points sampled from $\mathcal{X}$, and let $\mathbf{Y} = [\mathbf{y}_1|\mathbf{y}_2|\dots|\mathbf{y}_N] = [f(\mathbf{x}_1)|f(\mathbf{x}_2)|\dots|f(\mathbf{x}_N)]$ be the corresponding observation matrix whose columns are observations of the columns of $\mathbf{X}$ via f.

Under the above setting and given access only to observed data, we wish to generate a new embedding $\widetilde{\mathbf{X}} = [\tilde{\mathbf{x}}_1|\tilde{\mathbf{x}}_2|\dots|\tilde{\mathbf{x}}_N]$ of the observed points $\mathbf{Y}$ into n-dimensional Euclidean space while respecting the geometric structure of the latent data $\mathbf{X}$.

In order to provide a quantitative measure for intrinsic geometric structure preservation, we utilize the Kruskal stress function commonly used for multidimensional scaling (MDS) [33, 10, 17, 14]. This cost function penalizes the discrepancies between the pairwise distances in the constructed embedding and some target distances or dissimilarities. In our case, the target distances are the pairwise distances in the intrinsic space. This results in the following cost function:

$$\sigma\left(\widetilde{\mathbf{X}}\right) = \sum_{i<j} \left(\|\widetilde{\mathbf{x}}_i - \widetilde{\mathbf{x}}_j\| - \|\mathbf{x}_i - \mathbf{x}_j\|\right)^2. \tag{1}$$

Namely, low-stress values imply that the geometric structure of the embedding respects the intrinsic geometry conveyed by the pairwise distances in $\mathcal{X}$. Our goal then is to find an embedding $\widetilde{\mathbf{X}}$ that minimizes $\sigma(\widetilde{\mathbf{X}})$.

Minimizing the stress function is seemingly a standard least squares multidimensional scaling (LS-MDS) problem and is considered challenging since it leads to a nonconvex optimization problem, typically requiring a good initialization point located within the main basin of attraction and hence facilitating convergence to a global minimum [17, 24]. However, a significant additional challenge in the case of our specific task is approximating the ground-truth pairwise Euclidean distances $\|\mathbf{x}_i - \mathbf{x}_j\|$ from the observed data, despite the observation function f being unknown.

4. Intrinsic isometric manifold learning. In this section, we describe a manifold learning algorithm, which is both intrinsic and isometric.

Let $\mathbf{M}(\mathbf{y}_i)$ be defined by

$$\mathbf{M}(\mathbf{y}_i) = \left[\frac{df}{dx}\left(f^{-1}(\mathbf{y}_i)\right)\frac{df}{dx}\left(f^{-1}(\mathbf{y}_i)\right)^T\right]^\dagger, \tag{2}$$

where $\frac{df}{dx}\left(f^{-1}(\mathbf{y}_i)\right)$ is the Jacobian of the observation function f with respect to the intrinsic variable at the intrinsic point $\mathbf{x}_i = f^{-1}(\mathbf{y}_i)$. In subsection 4.1 we identify $\mathbf{M}(\mathbf{y}_i)$ as the push-forward metric tensor on $\mathcal{Y}$ with respect to the diffeomorphism between $\mathcal{X}$ and $\mathcal{Y}$.

The algorithm requires, as a prerequisite, that $\mathbf{M}(\mathbf{y}_i)$ be known for all $\mathbf{y}_i$. This requirement appears to be a restrictive condition that requires knowledge of the observation function

f. Nevertheless, we show in section 5 and section 6 that in several scenarios this tensor can be robustly approximated from the available sample set $\mathbf{Y}$ without explicit knowledge of f.

4.1. Local intrinsic geometry approximation. The minimization of the cost function given in (1) first requires an approximation of the intrinsic Euclidean pairwise distances $d_{i,j} = \|\mathbf{x}_i - \mathbf{x}_j\|$ without direct access to the latent intrinsic data points. We overcome this problem by approximating these distances using the $\mathbf{M}(\mathbf{y}_i)$ as follows:

$$d_{i,j}^2 = \tilde{d}_{iE,,j}^2 + \mathcal{O}\left(\|\mathbf{y}_i - \mathbf{y}_j\|^4\right), \tag{3}$$

where the approximation $\tilde{d}_{i,j}^2$ is given by

$$\tilde{d}_{i,j}^2 = \frac{1}{2}\left[\mathbf{y}_i - \mathbf{y}_j\right]^T \mathbf{M}(\mathbf{y}_i)\left[\mathbf{y}_i - \mathbf{y}_j\right] + \frac{1}{2}\left[\mathbf{y}_i - \mathbf{y}_j\right]^T \mathbf{M}(\mathbf{y}_j)\left[\mathbf{y}_i - \mathbf{y}_j\right]. \tag{4}$$

This approximation has been used in several papers [21, 20, 19, 22, 34, 48, 54, 52, 53, 55], and error analysis for it was presented in [20, 48]. This approximation can be viewed as a result of the relation between the intrinsic and observed manifolds. If one recognizes that $\mathcal{X}$ can be viewed as a Riemannian manifold, then $\mathbf{M}(\mathbf{y}_i)$ is the push-forward metric tensor on $\mathcal{Y}$ with respect to the diffeomorphism between $\mathcal{X}$ and $\mathcal{Y}$ [50, 37]. By using this specific metric on $\mathcal{Y}$, an isomorphism is established between the two manifolds $\mathcal{X}$ and $\mathcal{Y}$ which enables us to use $\mathcal{Y}$ as a proxy for making intrinsic geometric calculations on the latent manifold $\mathcal{X}$, just as if they were calculated directly on the latent intrinsic manifold $\mathcal{X}$. This makes the method ideally invariant to the observation function f and the distortions it induces on the geometry of the observed manifold. In practice, the push-forward metric changes at each point on the manifold, inducing an error in any nondifferential calculation performed and making the distance approximation (4) only locally accurate. The local validity of the approximation is demonstrated empirically in section 6. In addition, rigorous error analysis was provided in [20], where it was shown that the approximation is accurate for pairs of points on the observed manifold for which $\|\mathbf{y}_i - \mathbf{y}_j\|^4$ is sufficiently small with respect to the higher derivatives of the observation function.

4.2. Embedding via partial-stress minimization. Once local pairwise intrinsic Euclidean distances are estimated, our goal then is to build an embedding of the observed manifold into n-dimensional Euclidean space, which respects those distances. However, due to the local nature of the distance estimation, we only have valid approximation of short intrinsic distances, and we cannot directly minimize the stress function presented in (1). Instead we use the following weighted variant of the stress function:

$$\sigma_w\left(\widetilde{\mathbf{X}}\right) = \sum_{i<j} w_{i,j}\left(\|\widetilde{\mathbf{x}}_i - \widetilde{\mathbf{x}}_j\| - \tilde{d}_{i,j}\right)^2, \tag{5}$$

where unknown (or unreliable) distance estimations are assigned with zero weight and are thus excluded from the construction of the low-dimensional embedding.

Since the observation function is unknown, one cannot calculate the error term in the approximation (3). As a result, determining the scale at which distance estimations are

deemed reliable from the observed data itself is still an open issue which is similar to local scale determination issues for many other common nonlinear manifold learning methods. A good rule of thumb is to select for each point its distance to its k nearest neighbors, where, for most data sets tested, the use of 10–30 nearest neighbors provided the best results.

Construction of the embedding is achieved by minimizing (5) via an iterative descent algorithm. While many optimization methods can be used, the unique structure of the stress function allows for an efficient minimization using majorization of the cost function by a simple convex surrogate function whose exact minimum can be analytically calculated. This optimization method, referred to as scaling by majorizing a complicated function (SMACOF), has been shown to outperform regular strategies for minimizing the stress function [17, 24] and is commonly used in the context of MDS. For our purpose, SMACOF requires some adaptation in order to allow weighting of the stress function (5). This adaptation is also described in [41, 12, 40] and is briefly presented here for the convenience of the reader.

We start by expanding the weighted stress cost function:

$$\sigma_w(\widetilde{\mathbf{X}}) = \sum_{i<j} w_{i,j} \left\|\widetilde{\mathbf{x}}_i - \widetilde{\mathbf{x}}_j\right\|^2 - 2\sum_{i<j} w_{i,j}\tilde{d}_{i,j} \left\|\widetilde{\mathbf{x}}_i - \widetilde{\mathbf{x}}_j\right\| + \sum_{i<j} w_{ij}\tilde{d}_{i,j}^2.$$

The first term is quadratic in $\widetilde{\mathbf{X}}$ and can be recast using the weighted Laplacian:

$$\sum_{i<j} w_{i,j} \left\|\widetilde{\mathbf{x}}_i - \widetilde{\mathbf{x}}_j\right\|^2 = \operatorname{Tr}(\widetilde{\mathbf{X}}^T \mathbf{L} \widetilde{\mathbf{X}}),$$

where $\mathbf{L}$ is defined by

$$\mathbf{L}_{i,j} = \begin{cases} -w_{i,j}, & i \neq j, \\ \sum_{k=i} w_{i,k}, & i = j. \end{cases}$$

The second term can be rewritten as

$$-2\sum_{i<j} w_{i,j}\tilde{d}_{i,j} \left\|\widetilde{\mathbf{x}}_i - \widetilde{\mathbf{x}}_j\right\| = \operatorname{Tr}\left(\widetilde{\mathbf{X}}^T \mathbf{L}^{\widetilde{\mathbf{X}}} \widetilde{\mathbf{X}}\right),$$

where $\mathbf{L}^{\widetilde{\mathbf{X}}}$ is defined by

$$\mathbf{L}^{\widetilde{\mathbf{X}}}_{i,j} = \begin{cases} -\frac{w_{i,j}\tilde{d}_{i,j}}{\|\tilde{\mathbf{x}}_i - \tilde{\mathbf{x}}_j\|}, & \|\tilde{\mathbf{x}}_i - \tilde{\mathbf{x}}_j\| \neq 0, i \neq j, \\ 0, & \|\tilde{\mathbf{x}}_i - \tilde{\mathbf{x}}_j\| = 0, i \neq j, \\ -\sum_{j=1, j\neq i}^{N} \mathbf{L}^{\widetilde{\mathbf{X}}}_{i,j}, & i = j. \end{cases}$$

The third term is constant with respect to the embedding $\widetilde{\mathbf{X}}$. Now consider some other possible embedding $\mathbf{Z}$. According to the Cauchy–Schwarz inequality

$$\left\|\widetilde{\mathbf{x}}_i - \widetilde{\mathbf{x}}_j\right\| \|\mathbf{z}_i - \mathbf{z}_j\| \geqslant \left[\widetilde{\mathbf{x}}_i - \widetilde{\mathbf{x}}_j\right]^T [\mathbf{z}_i - \mathbf{z}_j]$$

for $\|\mathbf{z}_i - \mathbf{z}_j\| \neq 0$, we divide both sides by $\|\mathbf{z}_i - \mathbf{z}_j\|$:

$$\|\widetilde{\mathbf{x}}_i - \widetilde{\mathbf{x}}_j\| \geqslant [\widetilde{\mathbf{x}}_i - \widetilde{\mathbf{x}}_j]^T \frac{[\mathbf{z}_i - \mathbf{z}_j]}{\|\mathbf{z}_i - \mathbf{z}_j\|}.$$

Since this holds for any i, j, it also holds for the following positive weighted sum:

$$\sum_{i<j} w_{ij}\tilde{d}_{i,j} \|\widetilde{\mathbf{x}}_i - \widetilde{\mathbf{x}}_j\| \geqslant \sum_{i<j} w_{ij}\tilde{d}_{i,j} [\widetilde{\mathbf{x}}_i - \widetilde{\mathbf{x}}_j]^T \frac{[\mathbf{z}_i - \mathbf{z}_j]}{\|\mathbf{z}_i - \mathbf{z}_j\|} = \mathrm{Tr}\left(\widetilde{\mathbf{X}}^T \mathbf{L}^{\mathbf{Z}} \mathbf{Z}\right).$$

Denoting

$$\tau(\widetilde{\mathbf{X}}, \mathbf{Z}) = \mathrm{Tr}\left(\widetilde{\mathbf{X}}^T \mathbf{L} \widetilde{\mathbf{X}}\right) - 2\mathrm{Tr}\left(\widetilde{\mathbf{X}}^T \mathbf{L}^{\mathbf{Z}} \mathbf{Z}\right) + \sum_{i<j} w_{ij}\tilde{d}_{i,j}^2,$$

we conclude that $\tau(\widetilde{\mathbf{X}}, \mathbf{Z})$ is a quadratic function in $\widetilde{\mathbf{X}}$ that serves as an upper bound to our cost term for any $\mathbf{Z}$. Additionally, $\tau(\widetilde{\mathbf{X}}, \widetilde{\mathbf{X}}) = \sigma_w(\widetilde{\mathbf{X}})$. This suggests the following iterative minimization procedure: at the nth step we set $\mathbf{Z} \leftarrow \widetilde{\mathbf{X}}^{n-1}$. Then we reach a new embedding by optimizing the quadratic cost $\tau(\widetilde{\mathbf{X}}, \mathbf{Z})$ over $\widetilde{\mathbf{X}}$, i.e., $\widetilde{\mathbf{X}}^n \leftarrow \arg\max_{\widetilde{\mathbf{X}}} \tau(\widetilde{\mathbf{X}}, \mathbf{Z})$. This optimization process decreases stress monotonically since we have

$$\sigma_w(\widetilde{\mathbf{X}}^{n-1}) = \tau(\widetilde{\mathbf{X}}^{n-1}, \widetilde{\mathbf{X}}^{n-1}) \geqslant \tau(\widetilde{\mathbf{X}}^{n}, \widetilde{\mathbf{X}}^{n-1}) \geqslant \tau(\widetilde{\mathbf{X}}^{n}, \widetilde{\mathbf{X}}^{n}) = \sigma_w(\widetilde{\mathbf{X}}^{n}).$$

The stopping criterion is set once the reduction in stress between consecutive iterations is beneath some predefined threshold.

4.3. Embedding initialization using intrinsic Isomap. Since the cost function (5) is nonconvex with respect to the embedding coordinates, the iterative optimization process described in subsection 4.2, although efficient, is only guaranteed to converge to some local minimum [59]. In order for this process to converge to a good minimum, the optimization requires a good initialization point that represents a possible embedding of the data into low-dimensional Euclidean space which is roughly similar to the actual low-dimensional intrinsic structure. To construct such an initial embedding, we propose the use of an intrinsic variant of the Isomap method [56]. This variant uses intrinsic, instead of observed, geodesic distances as constraints for the embedding process. These intrinsic geodesic distances are approximated using the already calculated approximations of local intrinsic Euclidean distances. This approximation is accomplished by first constructing a graph from the data points, where only properly approximated intrinsic distances are represented by weighted edges, and then solving an "all-to-all" shortest path problem.

To see why this process approximates intrinsic geodesic distances, we look at the definition of curve length on a Riemannian manifold $\mathcal{X}$ endowed with a smooth Riemannian metric g. Let $\gamma(t) \in \mathcal{X}$, where $t \in [a, b]$ is a curve on the manifold $\mathcal{X}$. The length of this curve is defined by the following integral [38, 37]:

$$L_{\mathcal{X},g}\{\gamma\} = \int_a^b \sqrt{g\left(\gamma(t)', \gamma(t)'\right)} dt. \tag{6}$$

By definition of the push-forward metric, denoted by $f_*(g)$, we have that [38, 37]

$$f_*(g)\left(f(\gamma(t))', f(\gamma(t))'\right) = g\left(\gamma(t)', \gamma(t)'\right). \tag{7}$$

If both $\mathcal{X}$ and γ are observed via an injective function f, this integral can be equivalently calculated in observed space using the push-forward metric and plugging (7) into (6):

$$L_{\mathcal{X},g}\{\gamma\} = L_{\mathcal{Y},f_*(g)}\{f(\gamma)\} = \int_a^b \sqrt{f_*(g)\left(f(\gamma(t))', f(\gamma(t))'\right)}dt. \tag{8}$$

Equation (8) states that all length calculations for corresponding paths on the pair of diffeomorphic manifolds $\mathcal{X}$ and $\mathcal{Y}$ will be identical if one uses the push-forward metric on the observed manifold $\mathcal{Y}$. In this calculation, the push-forward metric accounts for local stretching and contraction of the manifold due to the observation function and changes the way we measure distances in order to compensate for this. This gives us a practical method to calculate intrinsic path lengths using only observed data. Since for our setting the metric used on $\mathcal{X}$ is endowed by an ambient Euclidean space, both these integrals receive a much simpler form:

$$L_{\mathcal{X},g}\{\gamma\} = \int_a^b \|\gamma(t)'\|dt = \int_a^b \sqrt{f(\gamma(t))'^T\left[\frac{df}{dx}\left(f^{-1}(\gamma(t))\right)\frac{df}{dx}\left(f^{-1}(\gamma(t))\right)^T\right]^\dagger f(\gamma(t))'}dt.$$

Using intrinsic curve length calculation, one can calculate geodesic distances as the infimum intrinsic length over all paths in $\mathcal{Y}$ which connect $f(\mathbf{x}_i)$ and $f(\mathbf{x}_j)$. Since only a final set of points sampled from this manifold is given, this minimization over all continuous paths can be approximated by minimization over all paths passing through the limited set of sample points. This presents geodesic distance approximation as a shortest path problem where only short distances, for which we have proper distance approximations, are represented by edges in the graph.

Once all pairwise intrinsic geodesic distances are approximated, classical scaling methods can be used to create an n-dimensional embedding in a Euclidean space, which respects the approximated geodesic geometry. This can be accomplished, as in [56], by assuming that the symmetric matrix of squared estimated intrinsic geodesic distances $\mathbf{D_{geo}}^{(2)}$ originates from pairwise distances of a set of N points in some Euclidean space. Then a Gram matrix of inner products can be calculated by the process of double centering:

$$\mathbf{G} = -\frac{1}{2}\mathbf{J}\mathbf{D_{geo}}^{(2)}\mathbf{J},$$

where $\mathbf{J} = \mathbf{I} - \frac{1}{N}\mathbf{1}\mathbf{1}^T$. An embedding into a low-dimensional Euclidean space $\widetilde{\mathbf{X}}$, which optimally approximates this inner product matrix in the Frobenius norm sense by minimizing $\|\mathbf{G} - \widetilde{\mathbf{X}}^T\widetilde{\mathbf{X}}\|_F$, can be obtained by principal component analysis (PCA). If the specified distances exactly correspond to a Euclidean distance matrix, then this results in an embedding which respects all pairwise distance constraints. Otherwise, this is only a heuristic procedure for constructing an embedding which is commonly used and typically attains a low-stress cost [12].

For the restricted class of convex latent intrinsic manifolds, the Euclidean and geodesic metrics coincide, allowing the intrinsic variant of Isomap, which uses intrinsic geodesic distances, to directly recover a proper embedding of the points into a low-dimensional space. However, many problems of interest have a nonconvex intrinsic state-space.

For intrinsic manifolds which are not extremely nonconvex, the discrepancy between geodesic and Euclidean intrinsic distances can be small, and the resulting embedding produced is often quite close to a proper intrinsic isometric embedding; this is also noticed in [18]. This observation suggests that it is possible to use the embedding generated by this intrinsic variant of Isomap as an initial point for further iterative optimization of the partial stress function (5) even when the intrinsic manifold is not strictly convex.

This approach harnesses the ability of the eigendecomposition to construct an initial embedding which is global in the sense that it respects the inferred inner products and optimally minimizes the strain function, and the ability of iterative optimization to use a partial stress function to consider only local intrinsic Euclidean structure. Figure 5 illustrates the flow of the algorithm.

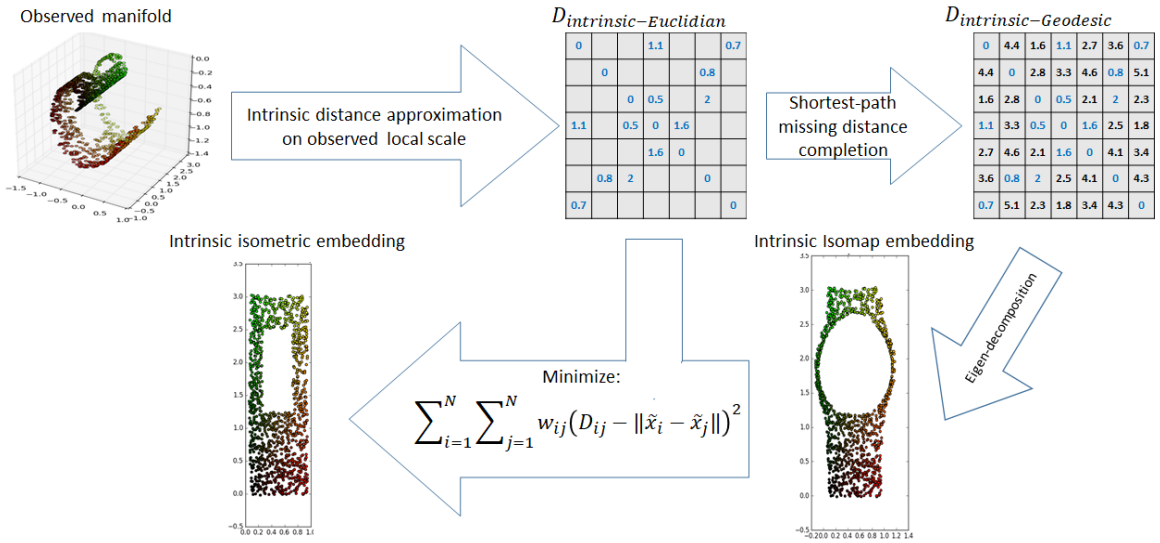


Figure 5. *Flow graph of our intrinsic isometric manifold learning method applied to an observed manifold.*

In the case of a highly nonconvex intrinsic manifold, the discrepancy between the two metrics results in a distorted embedding which does not properly respect the intrinsic Euclidean geometry [41]. One can calculate a quantitative measure for the level of the nonconvexity of the intrinsic manifold based on the amount discrepancy between true intrinsic pairwise Euclidean and geodesic distances, which can also be defined by a Kruskal stress term. However, this is a quantity that is a priori unknown and would, therefore, serve little purpose. Instead, we refer to an intrinsic manifold as highly nonconvex when the LS-MDS optimization process, when started at the initial embedding which respects the estimated intrinsic geodesic geometry, does not converge to an embedding which has a low weighted/short range stress value. A procedure to handle such a situation is proposed in subsection 4.4.

4.4. A multiscale scheme and failure detection. A convergence of the described iterative optimization process to a low-stress embedding is achieved, provided that the initial embedding obtained by the intrinsic variant of the Isomap algorithm is sufficiently close to the actual intrinsic structure and that we have a sufficient number of accurate pairwise distance estimates so that the embedding problem is appropriately constrained and well-posed. We remark that the method might fail due to various adverse conditions such as extremely noisy samples, insufficient data, an observation function which is not sufficiently smooth, etc. These weaknesses are common to most nonlinear manifold learning methods. In addition, there are two possible failure scenarios which are unique to the proposed method and can occur even when the local intrinsic geometric estimation is accurate.

The first can occur when the stress optimization converges to a stress value which is large with respect to the approximated intrinsic distances. Such an outcome indicates that the embedding used to initialize the LS-MDS optimization differs significantly from the intrinsic latent structure, and as a result, the process converges to a local minimum. This is demonstrated in the example presented in Figure 10. Such a failure occurs for highly nonconvex latent manifolds, which leads to a large discrepancy between the Euclidean and geodesic intrinsic distances and can lead to an initial embedding which is quite "far" from the desired embedding and drives the LS-MDS to converge to an unsatisfactory local minimum. This type of undesired final embedding can be detected by a relatively high-stress value.

To handle such manifolds, we devise an iterative multiscale operation scheme aimed to take advantage of the fact that small patches of the manifold are typically convex or at least approximately convex, thus increasing the chance that LS-MDS optimization converges to a good solution. In this multiscale scheme, large patches of points on the manifold are split into partially overlapping patches. Each of the subpatches can be further recursively split into smaller and smaller patches until small enough patches, that are not extremely nonconvex and allow for the successful application of our algorithm, are reached. We implement this division of the entire manifold into patches by using the already calculated estimates of the intrinsic geodesic distances between points and a k-means algorithm based on geodesic distances as in [2, 29]. Once two subpatches are successfully embedded into a low-dimensional Euclidean space, they are registered using a rigid rotation and translation transformation, which optimally aligns the embeddings of the common points. The alignment is implemented using the singular value decomposition (SVD) of the inner-product matrix of the two different embeddings as described in [49].

After registration, these patches are merged into a larger scale patch by averaging the location of the points over their locations in all the registered patches. To avoid the accumulation of error stemming from this bottom-up procedure, the embedding obtained by merging the embeddings of the subpatches is used only as an initial embedding that is further iteratively optimized as by LS-MDS. The proposed multiscale scheme is illustrated on a nonconvex manifold depicted in Figure 6. We remark that while the visualization is in the intrinsic space, the estimations of the geodesic distances, and hence the scale selection, are performed directly on the observed manifold.

This multiscale scheme also exhibits significant computational advantages for large data sets since the initialization point of each LS-MDS optimization is close to its final minimum and SMACOF only requires a small number of iterations on only a subset of the points to

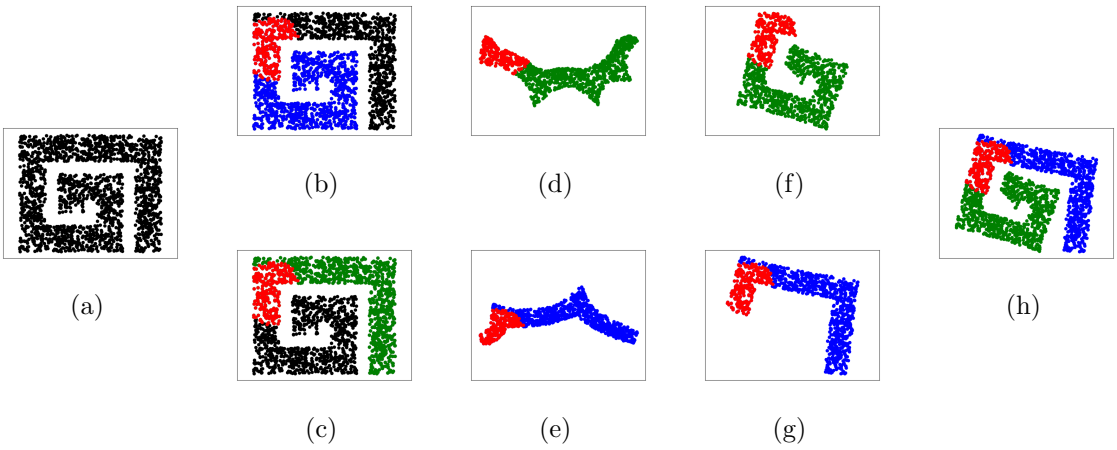


Figure 6. *A highly nonconvex manifold (a) is split into two partially overlapping subpatches colored blue and green (b) and (c). Their overlap is colored red. The initial embeddings of the subpatches are obtained by two separate applications of the intrinsic Isomap (d) and (e). Based on these two initial embeddings, intrinsic isometric embeddings are constructed separately for each subpatch (f) and (g). The subpatch embeddings are then registered and merged according to their overlapping areas to attain an embedding of the entire manifold, which is then further optimized using the LS-MDS optimization process. (h) Final intrinsic isometric embedding of entire manifold.*

converge.

The second type of failure manifests itself by the convergence of the optimization process to a relatively low-stress value, but the resulting embedding "folds" on itself, and intrinsically distant points are embedded into the same location. Such an embedding respects the local geometry but completely violates the global geometry. This can happen since only local scale pairwise distances are used for the final optimization of the stress function and the construction of the embedding, resulting in the stress value being unaffected by such a folding. Detection of such situations can be achieved by tracking changes in the topology of the manifold. Recalculating the geodesic distances of a "folded" embedded manifold results in a large discrepancy when compared to the initially estimated distances. To circumvent such situations we propose utilizing common LS-MDS heuristics, including multiple slightly perturbed initialization of the optimization process. If the folding repeats across multiple realizations, one can also increase the scale of distances which are incorporated into the weighted stress function; this is done at the risk of including distance estimation affected by nonconvexity. A more sophisticated approach for incorporating larger scale estimated distances is presented in [41]. There, detection of the edges of the manifold is harnessed to formulate criteria for detecting geodesics distances which are not affected by nonconvexity and are equivalent to corresponding Euclidean distances; these include long-range distances which can better condition the embedding problem and prevent folding.

4.5. Making other manifold learning methods intrinsic. In Figure 4 we considered a few popular existing manifold learning methods and showed that they do not result in an intrinsic isometric embedding when applied to observed data. To increase the fairness of

the comparison, we adapted some of these popular methods to make use of local metric information.

Most nonlinear manifold learning methods use a local kernel, including popular methods such as Isomap, Hessian Locally Linear Embedding (HLLE), Modified Locally Linear Embedding (MLLE), Linear Tangent Space Alignment (LTSA), t-distributed Stochastic Neighbor Embedding (t-SNE), Laplacian eigenmaps, and diffusion maps [37, 7]. Such methods can seemingly be adapted to also be intrinsic by simply locally incorporating available metric information. Such an adaptation was presented in this work for the Isomap method and in [48] for diffusion maps.

Methods such as HLLE, MLLE, and LTSA operate on a local scale by first choosing the k nearest neighbors of each point and then calculating their local covariance matrix. These two stages can be easily be adapted to use a local metric at each point. Other methods which only require local pairwise distances, such as t-SNE and Laplacian eigenmaps, can also be similarly adapted to use the notion of a local intrinsic metric. A comprehensive study of the possible adaptation of existing manifold learning methods in this manner is beyond the scope of this work. However, in section 6 we implement and test some intrinsic variants of a few popular manifold learning methods on both a toy problem and a realistic localization setting. We show that competing methods are indeed unsuitable for the tested problems.

In the following discussion, we attempt to provide a qualitative explanation for this empirical result. In LTSA [64], for example, local neighborhoods of data points are considered. For each neighborhood, the local covariance matrix is computed, and its leading eigenvectors are recovered and assumed to span the local tangent space to the manifold. The observed points in each neighborhood are then projected onto their corresponding low-dimensional tangent space, resulting in multiple separate local low-dimensional embeddings. A global embedding for all data points is then constructed by optimally linearly aligning each of these local embeddings. This local degree of freedom originates from the local ambiguity up to a linear transformation in the local projection of points to their tangent space. Incorporating a local metric on the tangent plane in the calculation of the local covariance matrix changes the covariances but does not change the subspace spanned by its leading eigenvalues. This makes the incorporation of a local metric ineffective in recovering any intrinsic structure. Methods such as MLLE and HLLE follow the same principle but implement it in different ways [63, 18]. All these locally linear methods encode linear relations between points in small neighborhoods of the observed manifold and define a cost term for a low-dimensional embedding which penalizes violations of the relations. Since a change of the local metric does not change the local linear relations between points, such a change does not affect the constructed embedding.

The assumption behind these methods is that nonlinear distortion to the data can be viewed as locally linear, and thus the locally linear relations between points are approximately unchanged. It is then assumed that constructing a low-dimensional embedding which respects these relations should result in an embedding which respects the intrinsic structure of the data. In practical settings, however, the observation function can exhibit nonlinearity even on local scales. Therefore, what eventually determines the embedding constructed using these methods is the nonlinearity of the observation function and its distorting effect on the encoded linear relations. This distortion induces an inherent dependence of these methods on the observation function, which provides an explanation for the results observed in Figures 4 and 11.

Methods such as t-SNE, Laplacian eigenmaps, and diffusion maps encode metric information implicitly via pairwise distances. Therefore, the notion of a local intrinsic metric can be naturally incorporated. However, these methods do not guarantee the preservation of the global relations between points, and they do not directly preserve geometry but rather other derived quantities; thus they do not provide global isometry.

In Table 1 we briefly summarize the reasons for the failure of these methods in achieving intrinsic isometry, addressing both their original and their intrinsic variants (if applicable).

The uniqueness of Isomap's suitability for the purpose of a globally isometric intrinsic embedding stems from the fact that it not only encodes local metric information but also aggregates it via the calculation of geodesics to construct global constraints which stabilize the embedding produce. This not only keeps points which are intrinsically near close to each other in the embedding but also prevents distant points from being embedded into the same location. Although these may be inaccurate due to the nonconvexity of the intrinsic manifold, they still serve as a stabilizing factor (this is true assuming that the nonconvexity is not too significant, which would require the application of our suggested multiscale approach).

5. Intrinsic metric estimation.

In the algorithm presented in section 4, we assumed that the "push-forward" metric $\mathbf{M}(\mathbf{y}_i)$ is known at every point $\mathbf{y}_i$. This metric allows us to approximate the intrinsic Euclidean and geodesic distances, and it has a key role in recovering the intrinsic latent manifold. Therefore, to apply the algorithm proposed in section 4, $\mathbf{M}(\mathbf{y}_i)$, which is typically unknown, needs to be robustly approximated from the observed data. This approximation is the subject of this section.

Several possible settings and models allow for estimation of the intrinsic metric from observed data. In this work, we focus on an intrinsic isotropic Gaussian Mixture Model (GMM), described in subsection 5.1, for a few reasons. First, such a model has been shown to be valid for real-world data when measurements are acquired in clusters of approximately normally distributed samples centered around a mean intrinsic state, e.g., [51]. Second, such a model can also be viewed as an approximation of a latent Itô process model, a model commonly used for dynamical systems in many fields, as discussed in subsection 5.2. Finally, the GMM allows for the development of a robust metric approximation scheme which is optimal in a probabilistic sense, as described in subsection 5.1.

We also wish to stress that these locally Gaussian settings are not unique, and one can think of a multitude of scenarios in which prior assumptions about the latent system allows for the estimation of the local distortion imposed on data by the unknown observation function. We provide an example of such an approximation method in subsection 6.2. Although the resulting estimator is optimal only in this specific GMM setting, it can also be viewed as the fitting of a smooth model to noisy local metric estimations. This interpretation suggests that it can be a reasonable heuristic for intrinsic metric estimation, even in cases where the GMM assumption does not hold, as is the case in subsection 6.2.

5.1. Intrinsic isotropic Gaussian mixture model.

Assume that the intrinsic data are sampled from an n-dimensional GMM consisting of N isotropic Gaussian distributions, all with the same known variance σ_{int}^2, where the means of these distributions are points on the intrinsic manifold, denoted by $\{\mathbf{x}_i\}_{i=1}^N \subseteq \mathcal{X}$. To distinguish between the model means and the

Table 1

Intrinsicness and global isometry properties of popular manifold learning methods and their intrinsic variants.

Property	Intrinsicness	Global isometry
LTSA, MLLE, HLLE	No. These methods preserve the local linear relations which are invariant to linear transformations and local changes of metric. As a result, there is a local linear ambiguity at each point and local metric information cannot be incorporated. Linear relations are also distorted by nonlinearities, making these methods dependent on the observation function [63].	No
MDS	No. The method preserves Euclidean distances in the observed space. Since all pairwise distances are used, local metric information is not sufficient to make the method intrinsic.	Yes, but only with respect to observed geometry.
Isomap	Geodesics are approximated in the observed space. Can be made to be intrinsic by use of a local metric.	No. Method is locally, but not globally, isometric.
t-SNE	The standard variant is not intrinsic since it uses probability distributions over points based on the observed space geometry. However, because the distributions are constructed using local pairwise affinities it can be made intrinsic by use of local metric information.	No. The method preserves the Kullback–Leibler divergence between probability distributions. This is not equivalent to preservation of global pairwise Euclidean distances.
Diffusion map	The standard variant is not intrinsic since it constructs a diffusion operator over points based on the observed space geometry. However, because construction is made using local pairwise affinities it can be made intrinsic by use of local metric information [48].	No. Constructed embedding preserves diffusion distances. This is not equivalent to preservation of global pairwise Euclidean distances.
Suggested Intrinsic Isometric Embedding	Yes	Yes

observed data points we use double indexing, i.e., $\mathbf{x}_{i,j} \in \mathcal{X}$ and

$$\mathbf{x}_{i,j} \sim \mathcal{N}\left(\mathbf{x}_i, \sigma_{int}^2 \mathbf{I}\right), \quad j = 1, \ldots, N_i,$$

where N_i is the number of points sampled from the ith Gaussian and $\mathbf{I}$ is the identity matrix. The intrinsic points are observed via an observation function $f : \mathbb{R}^n \to \mathbb{R}^m$ and are subject to additive white Gaussian noise (AWGN) with variance σ_{obs}^2 introduced in the observation

process. This results in the following observation model:

$$\mathbf{y}_{i,j} = f\left(\mathbf{x}_{i,j}\right) + w_{i,j},$$

where $w_{i,j} \sim \mathcal{N}\left(0, \sigma_{obs}^2 \mathbf{I}\right)$. Broadly, if σ_{int}^2 is small with respect to the second derivative of f, the observed data follows a Gaussian distribution with transformed mean and covariance:

$$\mathbf{y}_{i,j} = f\left(\mathbf{x}_{i,j}\right) + w_{i,j} \sim \mathcal{N}\left(f\left(\mathbf{x}_i\right), \sigma_{int}^2 \frac{df}{dx}\left(\mathbf{x}_i\right) \frac{df}{dx}\left(\mathbf{x}_i\right)^T + \sigma_{obs}^2 \mathbf{I}\right). \tag{9}$$

For a more rigorous analysis, see [20]. Namely, the observed data follows a GMM distribution, with anisotropic covariance, induced by the Jacobian of the observation function. Each Gaussian in this GMM can be interpreted according to probabilistic PCA approach [58], and it can be shown that the optimal maximum-likelihood estimator of the matrix $\frac{df}{dx}\left(\mathbf{x}_i\right) \frac{df}{dx}\left(\mathbf{x}_i\right)^T$ can be computed by applying PCA and finding the n most significant principal directions of the observed sample covariance matrix $\mathbf{S}_i$, i.e.,

$$\mathbf{S}_i = \frac{1}{N_i} \sum_{j=1}^{N_i} \left[\mathbf{y}_{i,j} - \bar{\mathbf{y}}_i\right] \left[\mathbf{y}_{i,j} - \bar{\mathbf{y}}_i\right]^T,$$

where $\bar{\mathbf{y}}_i$ is empirical observed sample mean

$$\bar{\mathbf{y}}_i = \frac{1}{N_i} \sum_{j=1}^{N_i} \mathbf{y}_{i,j}.$$

This matrix can be decomposed using spectral decomposition:

$$\mathbf{S}_i = \sum_{k=1}^{m} \sigma_k^2 \mathbf{u}_k \mathbf{u}_k^T.$$

The maximum-likelihood estimator is given by [58]

$$\frac{df}{dx}\left(\mathbf{x}_i\right) \frac{df}{dx}\left(\mathbf{x}_i\right)^T \approx \sum_{k=1}^{n} \left(\sigma_k^2 - \bar{\sigma^2}\right) \mathbf{u}_k \mathbf{u}_k^T,$$

$$\bar{\sigma^2} = \frac{1}{m-n} \sum_{k=n+1}^{m} \sigma_k^2.$$

Assuming that the clustering structure of observed measurements is known (i.e., we are able to group together observation points originating from the same Gaussian component), $\mathbf{M}\left(\mathbf{y}_i\right) = \frac{df}{dx}\left(\mathbf{x}_i\right) \frac{df}{dx}\left(\mathbf{x}_i\right)^T$ can be estimated directly from $\mathbf{S}_i$, separately for each model i. This setting is similar to the setting presented in [48], where short bursts of intrinsic diffusion processes with isotropic diffusion were used, resulting in an approximate intrinsic GMM and leading to the same estimator of the intrinsic metric using local applications of PCA. Although this setting can be a good approximation of real-world data (e.g., [51]), it is applicable mainly for dimensionality reduction of complex nonlinear systems which admit simulations according to a known model since this allows for the acquisition of multiple observations distributed around the same intrinsic state.

5.2. Intrinsic isotropic Gaussian mixture model as an approximation. Data often originates from observing a latent dynamical system over time. In such situations, information is encoded not only in the measurement values but also in the time at which these measurements were acquired. Prior knowledge about the dynamical model of the latent system combined with this temporal information can be used to estimate the intrinsic metric. Here we present a setting where the state dynamics of the latent system is governed by the following n-dimensional stochastic differential equation (SDE):

$$\mathbf{x}(t) = \mu_{int}(\mathbf{x}(t))\, dt + \sigma_{int} dB(t),$$

where μ_{int} is a drift term which only depends on the intrinsic state of the latent system and $dB(t)$ is the differential of a Brownian motion. Such processes are used to model many phenomena in a broad range of fields, including finance, physics, statistical mechanics, chemistry, and biology [36, 25, 1, 52]. If such a process is observed via an m-dimensional vector-valued function $f : \mathbb{R}^n \to \mathbb{R}^m$, by the Itô lemma [48], the observed process is governed by the following SDE:

$$df(\mathbf{x}(t)) = \mu_{obs}(\mathbf{x}(t))\, dt + \sigma_{int} \frac{df}{dx}(\mathbf{x}(t))\, dB(t).$$

Observing this process at equally spaced time intervals results in a discrete-time process which, according to the Euler–Maruyama scheme [31], can be approximately described by the following discrete SDE:

$$f(\mathbf{x}[k+1]) = f(\mathbf{x}[k]) + \mu_{obs}(\mathbf{x}[k])\, \triangle t + \sigma_{int} \frac{df}{dx}(\mathbf{x}[k])\, W[k]\, \triangle t,$$

where $\triangle t$ is the sampling interval and $W[k]$ is a discrete white Gaussian noise with unit variance. To further simplify this, we redenote

$$\mu_{obs}(\mathbf{x}[k]) = \mu_{obs}(\mathbf{x}[k])\, \triangle t \sigma_{int} = \sigma_{int} \triangle t$$

and get the following difference equation:

$$f(\mathbf{x}[k+1]) - f(\mathbf{x}[k]) = \mu_{obs}(\mathbf{x}[k]) + \sigma_{int} \frac{df}{dx}(\mathbf{x}[k])\, W[k].$$

This states that the difference process of the discrete observation process approximately admits to a normal distribution with the following parameters:

$$f(\mathbf{x}[k+1]) - f(\mathbf{x}[k]) \sim \mathcal{N}\left(\mu_{obs}(\mathbf{x}[k]), \sigma_{int}^2 \frac{df}{dx}(\mathbf{x}[k]) \frac{df}{dx}(\mathbf{x}[k])^T\right).$$

When AWGN is introduced into the observation process, we obtain the following distribution of the observed process:

$$\mathbf{y}[k+1] - \mathbf{y}[k] \sim \mathcal{N}\left(\mu_{obs}(\mathbf{x}[k]), \sigma_{int}^2 \frac{df}{dx}(\mathbf{x}[k]) \frac{df}{dx}(\mathbf{x}[k])^T + \sigma_{obs}^2 \mathbf{I}\right),$$

where $y[k] = f(x[k])$. This indicates that the difference process of the observations can be viewed as samples from a Gaussian distribution whose covariance and drift are dependent only on the intrinsic state of the system.

Another case in which one can approximately model the observed data with an intrinsic GMM is when the observation noise is negligible in comparison to the variation in the observation function as a result of changes in the intrinsic state of the system. In such a case we can cluster together sample points originating from approximately the same intrinsic state. For a specific observation point $\mathbf{y}_i$, we look for all observation time indexes k for which $\mathbf{y}[k] \approx \mathbf{y}_i$ and denote this set by $K_{\mathbf{y_i}} : K_{\mathbf{y_i}} = \{k|\mathbf{y}[k] \approx \mathbf{y}_i\}$. If the observation function is sufficiently smooth and invertible, observations which are close enough to each other in the observation space originate from similar intrinsic points and therefore experience similar observation function Jacobians. As a result, the observed process "jumps" $f(\mathbf{x}[k+1]) - f(\mathbf{x}[k])$ for all $k \in K_{\mathbf{y_i}}$ will be distributed with very similar Gaussian distributions. This clustering, illustrated in Figure 7, achieves an approximate GMM structure directly from observed data.

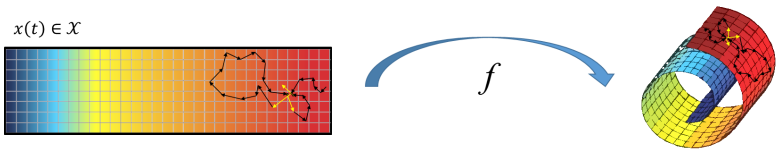


Figure 7. *GMM clustering in the observed space. Observed difference process "jumps" starting from approximately the same observed value originate from approximately the same intrinsic state.*

5.3. Maximum-likelihood intrinsic metric estimation. We note that a common characteristic of local estimation methods, such as the one described above, is that these methods typically operate on a scale at which the observation function f is approximately linear and can be well approximated by its Jacobian. The curvature of the manifold dictates how quickly the Jacobian and the metric change along the manifold and therefore restricts the scale at which samples are relevant for estimation of the metric at a specific point. If the data is not sufficiently sampled, the number of data points that will be incorporated into the estimation of a single local metric will be small, resulting in a poor estimation. This nonrobustness is not unique to these specific metric estimation methods, and many nonlinear estimation problems and manifold learning techniques are nonrobust due to their local nature, as discussed in [7]. To overcome this limitation, the remainder of this section introduces global (parametric) regularization of the metric estimation, imposing a smooth variation of the estimated metrics along the intrinsic manifold.

In subsection 5.1, the observed data were modeled by a GMM with covariance matrices that depend on the Jacobian of the observation function (9). Under this statistical model, an unconstrained maximum-likelihood estimation of the metric $\mathbf{M}(\mathbf{y}_i)$ is a local estimator which only uses data from a single Gaussian and does not impose any smoothness on the metric, limiting the number of samples incorporated in the estimation and the estimation accuracy.

To overcome this, we propose a constrained nonlocal maximum-likelihood estimator; the intrinsic isotropic GMM described in subsection 5.1 results in the following approximate global

log-likelihood function for the observed data:

$$\mathcal{L}\left(\left\{\frac{df}{dx}(\mathbf{x}_i)\frac{df}{dx}(\mathbf{x}_i)^T\right\}_{i=1}^{N}\right) = -\frac{1}{2}\sum_{i=1}^{N} N_i\left\{d\ln(2\pi) + \ln|\mathbf{C}_i| + \mathrm{Tr}\left(\mathbf{C}_i^{-1}\mathbf{S}_i\right)\right\},$$

where N is the number of Gaussians in the GMM, N_i is the number of points sampled from the ith Gaussian, $\mathbf{S}_i$ is the sample covariance of the ith observed Gaussian distribution, and $\mathbf{C}_i$ denotes the (population) covariance, which is given by

$$\mathbf{C}_i = \sigma_{int}^2\frac{df}{dx}(\mathbf{x}_i)\frac{df}{dx}(\mathbf{x}_i)^T + \sigma_{obs}^2\mathbf{I}.$$

This log-likelihood is also presented in [57].

We restrict the Jacobian of the observation function to a parametric family of functions, which we denote by $\mathbf{J}(\mathbf{y}_i|\theta) : \mathbb{R}^m \to \mathbb{R}^{m\times n}$, where θ represents the parametrization of the family. Estimation is then performed by choosing θ that maximizes the parametrized log-likelihood function:

$$\mathcal{L}(\{\mathbf{y}_{i.j}\}|\theta) = -\frac{1}{2}\sum_{i=1}^{N} N_i\left\{d\ln(2\pi) + \ln|\mathbf{C}(\mathbf{y}_i|\theta)| + \mathrm{Tr}\left(\mathbf{C}(\mathbf{y}_i|\theta)^{-1}\mathbf{S}_i\right)\right\}, \tag{10}$$

where

$$\mathbf{C}(\mathbf{y}_i|\theta) = \sigma_{int}^2\mathbf{J}(\mathbf{y}_i|\theta)\mathbf{J}(\mathbf{y}_i|\theta)^T + \sigma_{obs}^2\mathbf{I}.$$

We note that the log-likelihood in (10) only differs by a constant from the following function:

$$\begin{aligned} R(\{\mathbf{y}_{i.j}\}|\theta) &= -\sum_{i=1}^{N} N_i D_{KL}(\mathbf{C}_i||\mathbf{S}_i) \\ &= -\sum_{i=1}^{N} N_i\left\{\ln\frac{|\mathbf{C}_i|}{|\mathbf{S}_i|} + \mathrm{Tr}\left(\mathbf{C}_i^{-1}\mathbf{S}_i\right)\right\} \\ &= \mathcal{L}(\{\mathbf{y}_{i.j}\}|\theta) + \frac{1}{2}\sum_{i=1}^{N} N_i\left\{d\ln(2\pi) - \ln|\mathbf{S}_i|\right\} \\ &= \mathcal{L}(\{\mathbf{y}_{i.j}\}|\theta) + constant, \end{aligned} \tag{11}$$

where $D_{KL}(\mathbf{C}_i||\mathbf{S}_i)$ is the well-known Kullback–Leibler divergence between Gaussian distributions with the same mean and two different covariances matrices $\mathbf{C}_i$ and $\mathbf{S}_i$. This divergence reaches its minimal value when $\mathbf{C}_i$ and $\mathbf{S}_i$ are equal. Therefore, maximization of the likelihood is equivalent to minimization of the Kullback–Leibler divergence, over the whole manifold, between the covariance explained by the parametric model and the noisy local sample covariance calculations.

5.4. Parametric estimation using regularized artificial neural networks. To increase the robustness of the estimation and, specifically, to avoid large variations in the metric estimates, we use a parametric family of smooth functions, implemented by an ANN. Neural networks are

compositions of linear transformations and nonlinear operation nodes which can be adapted to approximate a specific function by choosing the values of the linear transformations in the network. These values are called the *weights* of the network. Thus, in the context of our work, the parameter vector θ consists of the weights of the network. By choosing the nonlinear operation nodes to be smooth functions, e.g., the standard sigmoid function $s(x) = \frac{1}{1+e^{-x}}$, we can explicitly impose smoothness.

The inputs of the network are points on the observed data manifold, and the outputs are $n \times m$ matrices $\mathbf{J}(\mathbf{y}_i|\theta)$, which estimate the Jacobian of the observation function at the input points. Using the output of the network to model the Jacobian of the observation function does not require any special network architecture—as opposed to directly modeling the intrinsic metric of the observed covariance, which is restricted to the manifold of symmetric positive definite (SPD) metrics and would require specialized network architecture.

Training data for the network consists of pairs of locations on the observed manifold and locally estimated, and thus also noisy, covariances at those locations. The training of the metric estimation network is carried out by optimizing the weights of the network using the likelihood (10) as a cost function.

The neural network structure used in this work contains two hidden layers and an additional linear layer at the output, which is added in order to allow for automatic scaling of the output as part of the optimization process.

The use of ANNs has several advantages when compared to other smooth parametric function families. First, ANNs are general and have proven to perform well in a broad spectrum of applications. Second, many methods for network regularization exist, facilitating the convenient implementation of the smoothness constraint. Third, due to the extreme popularity of ANNs, there are many software and hardware solutions allowing for efficient and fast construction and optimization.

This estimation method also has some secondary advantages in addition to providing robustness to the intrinsic metric estimation when compared to local metric estimation methods. As opposed to local methods, which only estimate the intrinsic metric for observed sample points, the regression approach gives rise to an estimation of the intrinsic metric and the tangent plane on the whole observed space. These estimations can be used to generate additional points between existing sampled points, thus artificially increasing the sample density. This can improve both the short-range intrinsic distance estimation described in subsection 4.1 and the geodesic distance estimation described in subsection 4.2. Both effects could improve the results of the algorithm presented in this paper. In addition, it was shown in [7] that by learning the tangent plane to the manifold at each point, one could move on the manifold by making infinitesimal steps each time in the tangent plane. Since, with the method proposed in this section, we estimate not only the tangent plane but the intrinsic metric as well, we can know how far we have gone on the manifold in terms of an intrinsic notion of distance, an ability that might be relevant to applications such as nonlinear interpolation [13].

5.5. Implementation. In order to optimize the network weights with respect to the nonstandard cost function presented in (10), we implemented the described metric estimation network in Python using Theano [8], which allows for symbolic gradient calculation. To facilitate better and faster optimization, the stochastic gradient descent method ADAptive Moment

estimation (ADAM) [30] was used. This method uses momentum and moment estimation over time to automatically and adaptively scale the learning rate of the optimization process for faster convergence.

We control the estimated function complexity by limiting the number of nodes in the network. In addition, we use a weight decay penalty term, which encourages small weights and, in general, prevents excessively large variation of the network outputs with respect to small changes in the input [32].

In unsupervised settings, choosing the network hyperparameters (e.g., the number of nodes and the emphasis put on the weight decay term) is not straightforward since there are no labels or ground-truth values for the intrinsic metric (or the Jacobian). Here, it is carried out by cross-validation using the log-likelihood function as the cost function. In our tests, we used k-fold cross-validation where 20% of the observed samples were used for validation purposes.

6. Experimental results.

6.1. Simulated data. Consider the intrinsic manifold $\mathcal{X}$ corresponding to a 2-dimensional square with a cross-shaped hole at its center. The latent sample set $\mathbf{X}$ is generated by uniformly sampling points from $\mathcal{X}$. These points are then observed via the following observation function:

$$\mathbf{y}(\mathbf{x}) = f(\mathbf{x}) = \begin{bmatrix} y_1(\mathbf{x}) \\ y_2(\mathbf{x}) \\ y_3(\mathbf{x}) \end{bmatrix} = \begin{bmatrix} \sin(2.5 \cdot x_1) \cdot \sin(x_2) \\ \sin(2.5 \cdot x_1) \cdot \cos(x_2) \\ -\sin(x_2) \end{bmatrix}. \tag{12}$$

This observation function embeds the data into 3-dimensional Euclidean space by wrapping it around the unit sphere, resulting in the observed data set $\mathbf{Y}$.

Ideal conditions. First, we evaluate the performance of the proposed method under ideal conditions. To do so, we analytically calculate $\frac{df}{dx}(\mathbf{x}_i)\frac{df}{dx}(\mathbf{x}_i)^T$ by taking the derivative of the observation function given in (12) with respect to the intrinsic latent variable. Results are displayed in Figure 8.

In Figure 8(a) we present the intrinsic latent manifold in the intrinsic space, and in Figure 8(b) we present the observed manifold embedded in the observation space. In Figure 8(c) we visualize the intrinsic metric by plotting corresponding ellipses at several sample points. These ellipses represent the images, under the observation function, of equally sized circles in the intrinsic space according to the intrinsic metric. Such a visualization shows the amount of local stretch and contraction that the observed manifold experiences with respect to the latent intrinsic geometry.

In Figure 8(d) we scatter-plot the ground truth intrinsic Euclidean pairwise distances and the approximated pairwise distances using (4). In such a plot, points near the identity map (colored red) represent good intrinsic Euclidean distance approximations. In Figure 8(e), we scatter-plot the same distances but restrict the depicted pairs of points to only include the k nearest neighbors in the observation space. For this example, we use $k = 10$. These two plots indicate that the distance approximation is indeed valid at short distances, for which the manifold is approximately only linearly distorted. The short distances to the nearest neighbors are then used in order to approximate all of the pairwise intrinsic geodesic distances on the

manifold, as described in subsection 4.3. We scatter-plot these approximations against their true value in Figure 8(f). Again, the concentration of points near the identity map indicates a good approximation.

Finally, we construct an initial embedding, presented in Figure 8(h), using the approximated intrinsic geodesic distances and the intrinsic variant of Isomap described in subsection 4.3. This embedding is then refined as described in subsection 4.2 to produce the final intrinsic isometric embedding presented in Figure 8(i). To also provide a nonintrinsic reference, we construct the standard Isomap embedding and present it in Figure 8(g). For each of these 3 embedding methods, we also plot a scatter plot of the true intrinsic Euclidean distances against those calculated from the embedding, Figure 8(j), Figure 8(k), and Figure 8(l). Above these scatter plots, we note the calculated embedding stress value according to (1), which serves as a quantitative measure of the embedding success.

As can be observed from these results, our method successfully recovers the structure of the data in the intrinsic space and achieves a low-stress value. Standard Isomap and the intrinsic variant of Isomap fail to do so and construct a distorted embedding, which does not respect the intrinsic geometric structure, resulting in higher stress values. Standard Isomap fails to recover the intrinsic data structure due to its dependence on the observation function. Intrinsic Isomap fails due to the nonconvexity of the intrinsic manifold and the discrepancy it induces between Euclidean and geodesic intrinsic distances.

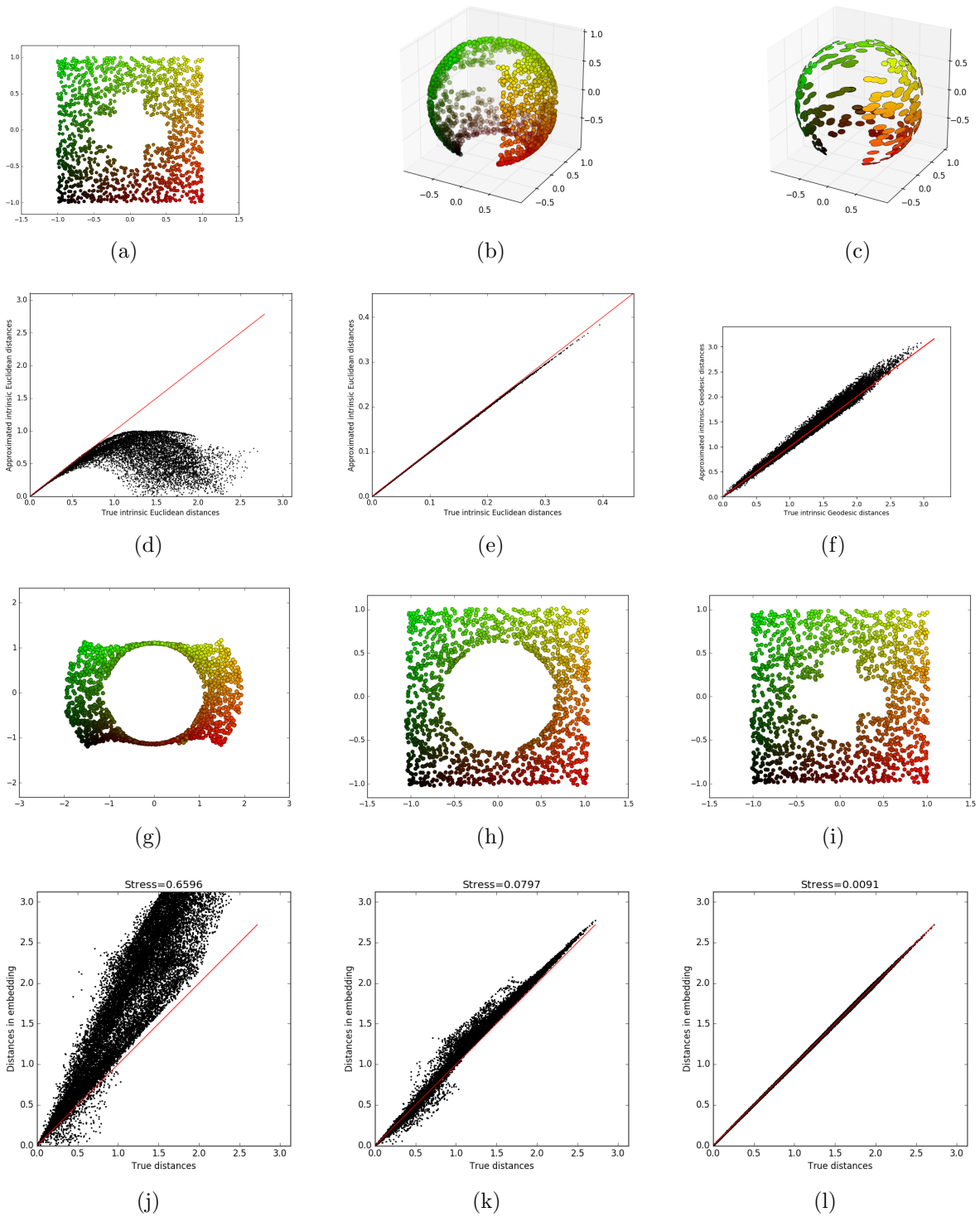


Figure 8. *Punctured severed sphere (embedding).* (a) *Intrinsic space.* (b) *Observed space.* (c) *Intrinsic metric.* (d) *Intrinsic Euclidean distance approximation.* (e) *Intrinsic Euclidean distance approximation—k-NN only.* (f) *Intrinsic geodesic distance approximation—k-NN only.* (g) *Standard Isomap embedding.* (h) *Intrinsic Isomap embedding.* (i) *Intrinsic isometric embedding.* (j) *Standard Isomap stress.* (k) *Intrinsic Isomap stress.* (l) *Intrinsic isometric stress.*

Metric estimation. Next, we analyze the effect of using an intrinsic metric which is estimated from the observed data, as opposed to using the exact intrinsic metric. To do so, we extend the previous example to include estimation of the metric under the setting described in subsection 5.1. Specifically, we use either $N_i = 5$ or $N_i = 200$ random measurements sampled from an isotropic Gaussian with intrinsic variance $\sigma_{int}^2 = 0.03^2$ centered at each of the data points in the intrinsic space. Additionally, observation noise is added with variance $\sigma_{obs}^2 = 0.03^2$.

In Figure 9 we compare the trivial local estimation (as described in subsection 5.1) and the global estimation approach implemented via an ANN, as proposed in section 5. For each metric estimation method, we plot, similarly to the previous example, in Figure 9(a), Figure 9(b), and Figure 9(c) a visualization of the metric using ellipses, in Figure 9(d), Figure 9(e), and Figure 9(f) a scatter plot of the true intrinsic distance against the approximated intrinsic distances, in Figure 9(g), Figure 9(h), and Figure 9(i) the resulting low-dimensional embedding, and in Figure 9(j), Figure 9(k), and Figure 9(l) the scatter plot of the Euclidean distances in the embedding compared to the true intrinsic distances, including a calculation of the actual stress value. These are attained for the case of local estimation with dense sampling $N_i = 200$ (Figure 9(a), Figure 9(d), Figure 9(g), and Figure 9(j)), for the case of local estimation with sparse sampling $N_i = 5$ (Figure 9(b), Figure 9(e), Figure 9(h), and Figure 9(k)), and finally for our proposed ANN estimation method with sparse sampling $N_i = 5$ (Figure 9(c), Figure 9(f), Figure 9(i), and Figure 9(l)).

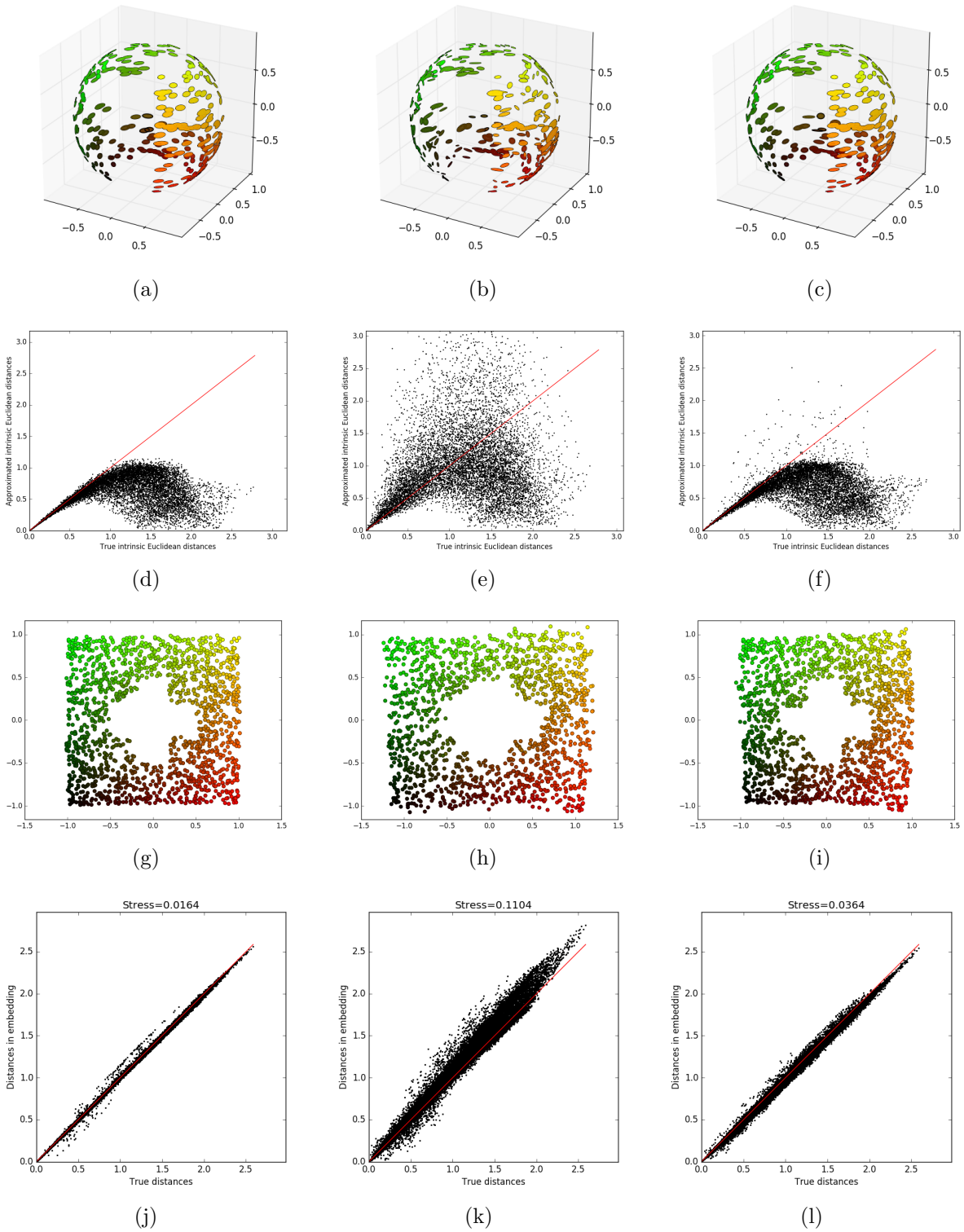


Figure 9. *Punctured severed sphere (metric estimation).* (a) *True intrinsic metric.* (b) *Locally estimated intrinsic metric.* (c) *Net learned intrinsic metric.* (d) *Distance estimation using true metric.* (e) *Distance estimation using locally estimated metric.* (f) *Distance estimation using net estimated metric.* (g) *Embedding using true intrinsic metric.* (h) *Embedding using locally learned intrinsic metric.* (i) *Embedding using net learned intrinsic metric.* (j) *Embedding stress using true intrinsic metric.* (k) *Embedding stress using locally estimated intrinsic metric.* (l) *Embedding stress using net learned intrinsic metric.*

It can be seen that the local metric estimations in the sparse case are "noisy" when compared to those estimated in the dense sampling case and can sometimes change abruptly between nearby locations on the manifold. This inaccuracy in the estimated local metrics adversely affects the intrinsic distance estimation and, in turn, hinders the recovery of the intrinsic isometric learned representation.

It is evident that our neural network based estimation method outperforms the local metric estimation and enables us to learn intrinsic isometric representation given sparse sampling and in the presence of observation noise.

Multiscale approach. We demonstrate the need for the multiscale approach presented in subsection 4.4. To do so, we use the same observation function as above but change the intrinsic sample point to the spiral data set presented in Figure 6(h). We attempt to embed this data set both in a one-shot approach using all available data simultaneously and also by using two stages of subdivision of the entire data set into overlapping subpatches as described in subsection 4.4.

We see in Figure 10 that the one-shot approach fails to recover the structure of the latent system, while the multiscale approach does not. If we look at the initial embedding recovered by the intrinsic variant of Isomap on the whole data set, we see that it is significantly distorted with respect to the intrinsic structure;thus it does not serve as a suitable starting point for the LS-MDS optimization; this is of course due to the extreme nonconvexity of the intrinsic manifold. The most inner and outer points of the spiral have a large geodesic distance, leading Isomap to embed them at a large distance from one another; however, in the intrinsic Euclidean sense they are not that far apart.

Another advantage of the multiscale approach is the reduced computation time. While the one-shot approach can require a large number of iterations to ensure convergence from the initial starting point, the multiscale approach, on the other hand, is never too far from the local minimum at each scale and thus only requires a few iterations at each stage.

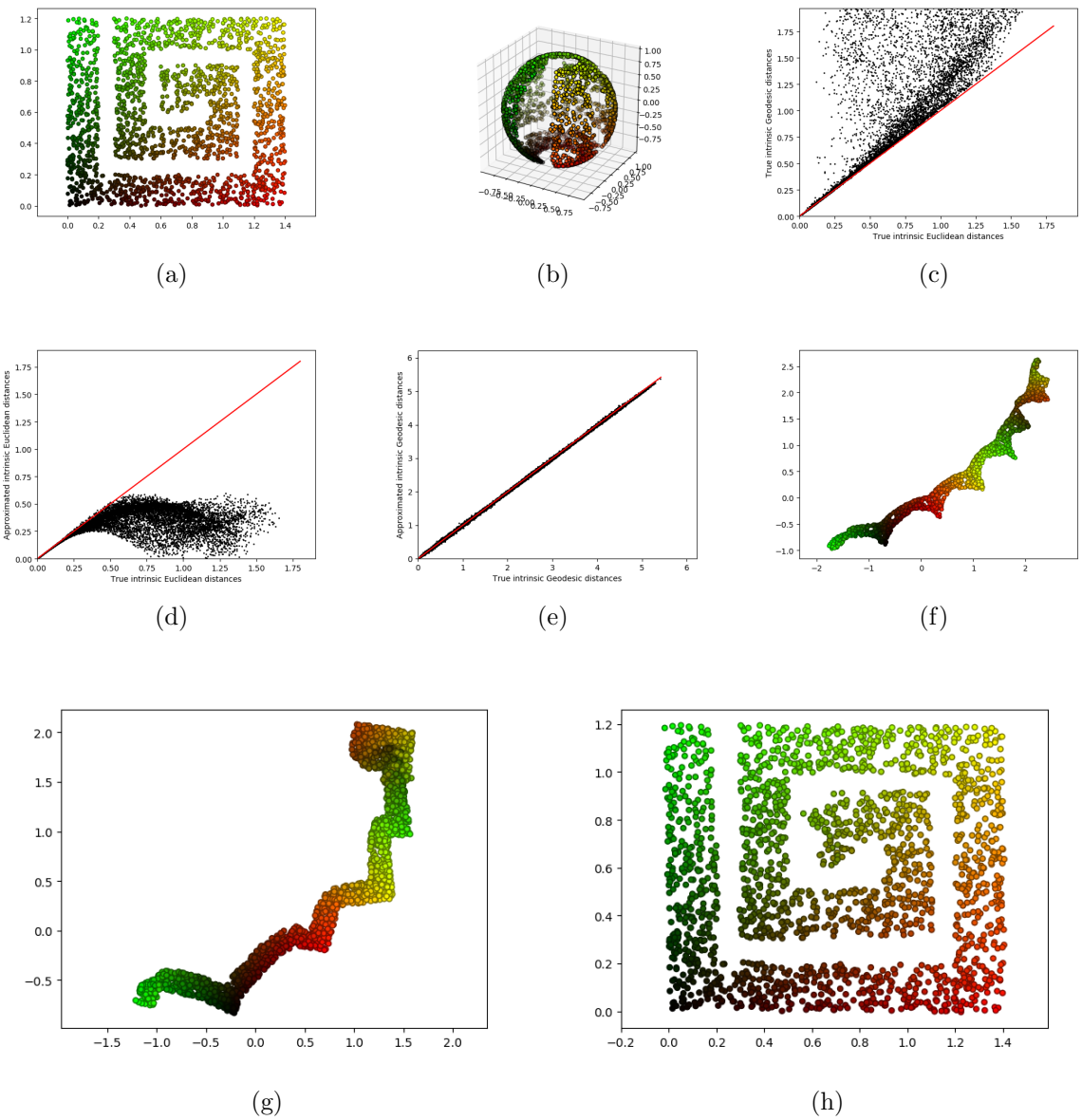


Figure 10. *Multiscale approach.* (a) *Intrinsic space.* (b) *Observed space.* (c) *Euclidean versus geodesic distances in intrinsic space.* (d) *Approximated versus true intrinsic geodesic distances.* (e) *Distance estimation using locally estimated metric.* (f) *Intrinsic Isomap embedding.* (g) *One-shot intrinsic isometric embedding.* (h) *Multiscale intrinsic isometric embedding.*

Intrinsic variants of other manifold learning methods. Here we also tried applying other manifold learning methods both in their standard form and in an intrinsic variant as discussed in subsection 4.5. As can be seen in Figure 11, the standard methods were affected by distortions induced by the observation function, while their intrinsic variants were not as

affected. However, they still did not recover the globally intrinsic geometry and structure.

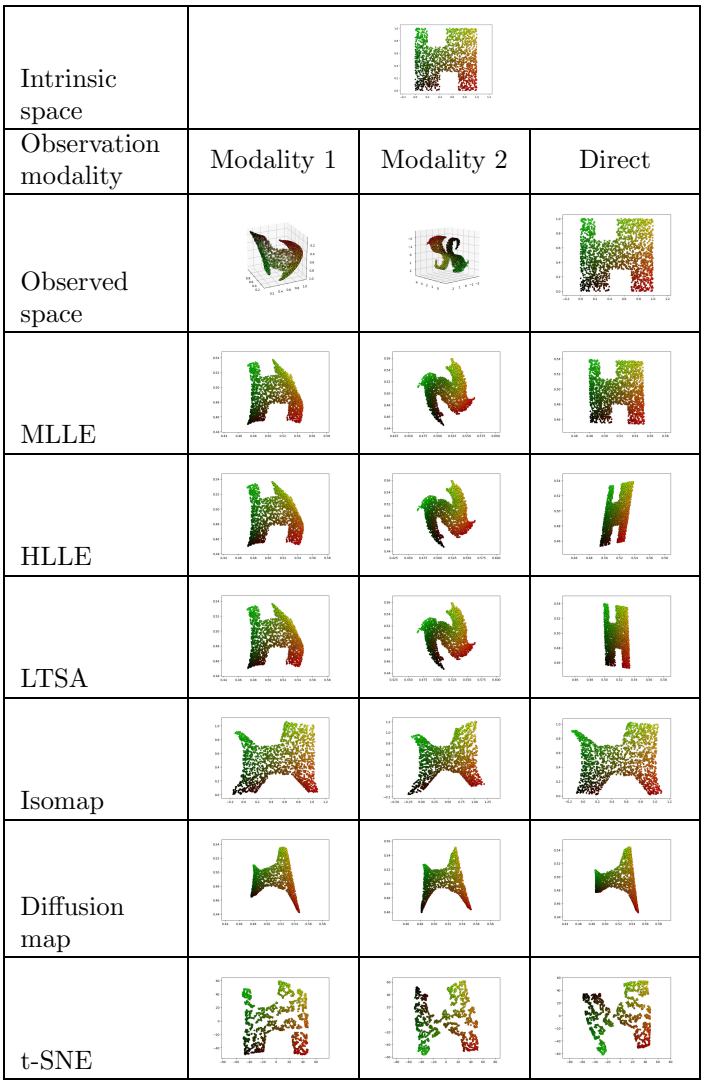


Figure 11. *Intrinsic variants of local kernel manifold learning methods. The first two columns correspond to the two different observation modalities. The third column represents a direct observation of the latent space. The different rows represent the application of various popular manifold learning methods to the observed manifold.*

6.2. Localization from image data. In section 2 we provided initial motivation for our work through the simple and intuitive example of localization using observations acquired via an unknown model. We now revisit this example and discuss in detail how the algorithm proposed in this work can be applied to the problem of positioning using image-based measurements. With this experiment, we examine the advantages of intrinsic geometry preservation and demonstrate its relevance to high-dimensional and complex scenarios.

The experimental setting is as follows: an agent is located within a compact, path-

connected subset $\mathcal{X}$ of $\mathbb{R}^2$ which represents a closed indoor environment. The outlines of $\mathcal{X}$ used for this experiment are depicted in Figure 12. At each sample point $\mathbf{x} \in \mathcal{X}$, an image is acquired by the agent. These images serve as an indirect observation or measurement of the location of the agent. Such observations are made in a sufficient number of different locations so that $\mathcal{X}$ is fully covered. The dimension of the intrinsic manifold for this problem is $n = 2$, corresponding to the two-dimensional physical space, and the dimension of the observation space corresponds to the dimensionality of the generated images and is in general high-dimensional.

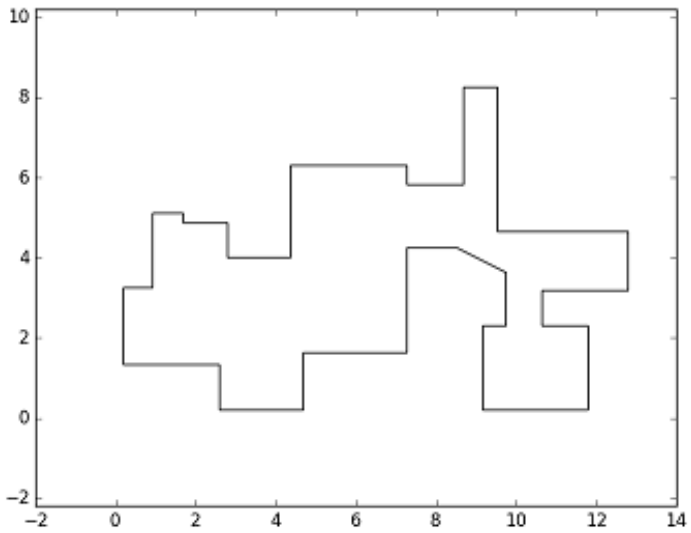


Figure 12. *Outline of the latent manifold which represents a confined indoor environment.*

To simulate this setting, we used "Blender,"[1] a professional, freely available and open-source modeling software. Using "Blender," we constructed a 3-dimensional model mimicking the interior of an apartment. The created model is shown in Figure 13. The region of the model in which there are no objects and in which the simulated agent is allowed to move corresponds to the shape of $\mathcal{X}$ presented in Figure 12. "Blender" allows us to render an image of the 3D model as seen via a virtual camera. Using this ability, we produce a set of 360-degree panoramic color images of size 128×256 pixels, taken from the point of view of the agent as seen in Figure 14.

In order for the observation to depend solely on the location of the agent, as is assumed by our proposed algorithm, we were required to omit the effect of the orientation of the agent. This was implemented in the frequency domain by applying a Fourier transform to each frame and then estimating and removing the linear phase in the horizontal direction. Since cyclic rotations of images are equivalent to an addition of linear phase in the Fourier domain, this

[1] https://www.blender.org

makes the observations invariant to the agent orientation.

In order to reduce the initial dimensionality of the data, PCA was applied, and only the first 100 principal components were taken. This number of principal components was chosen since empirically 100 components embodied most of the power of the data.

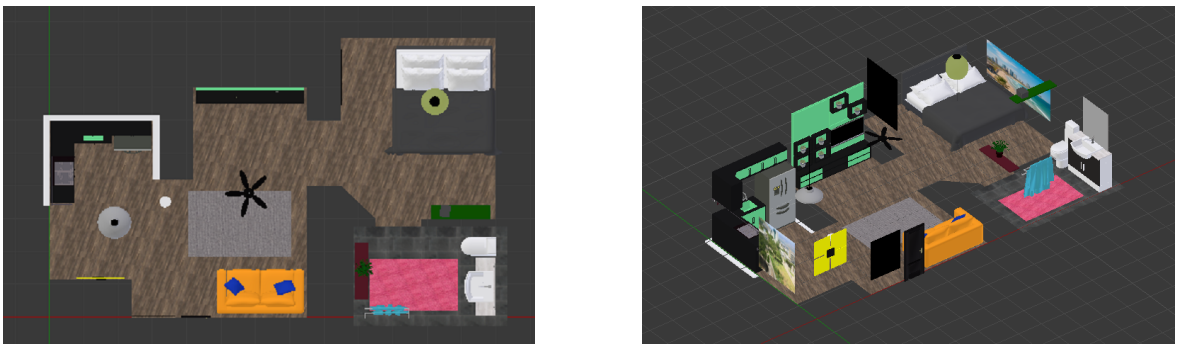


Figure 13. 3-*dimensional model in Blender.*

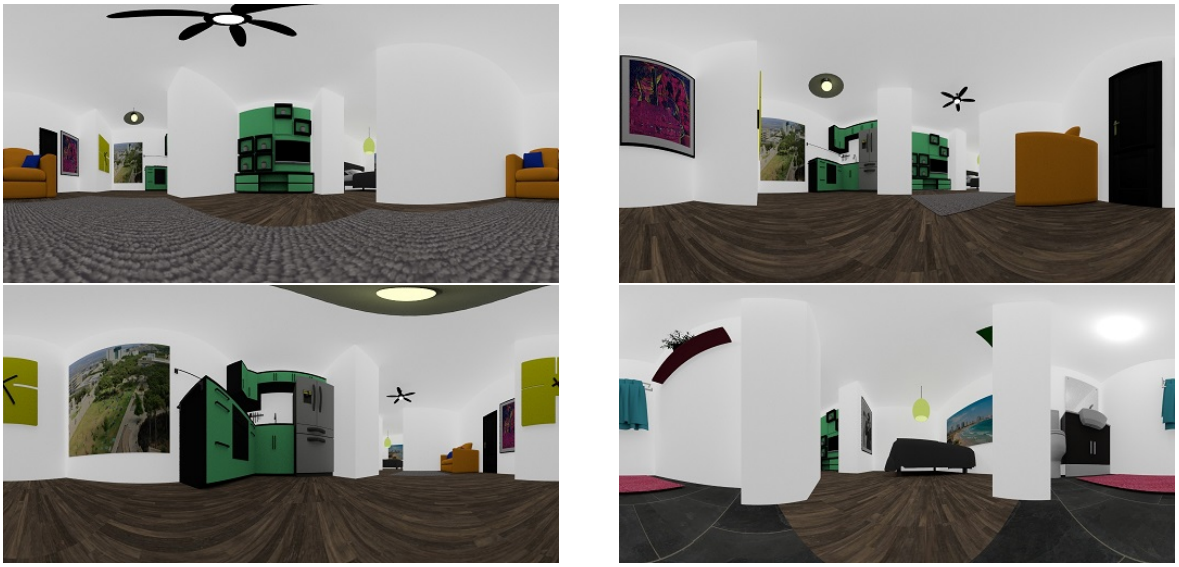


Figure 14. *Samples of generated panoramic color images.*

To stress the fact that our algorithm is invariant with respect to the sensor modality, we also used an additional different image modality. The second modality, which is also simulated using "Blender," is a gray-scale depth map. For this modality, the gray level at each pixel represents the distance from the camera to the nearest object in the direction of that pixel. Several examples of such images are shown in Figure 15. This observation modality was also preprocessed similarly to the color images in order to impose invariance to cyclic rotations in the horizontal axis and to perform initial dimensionality reduction.

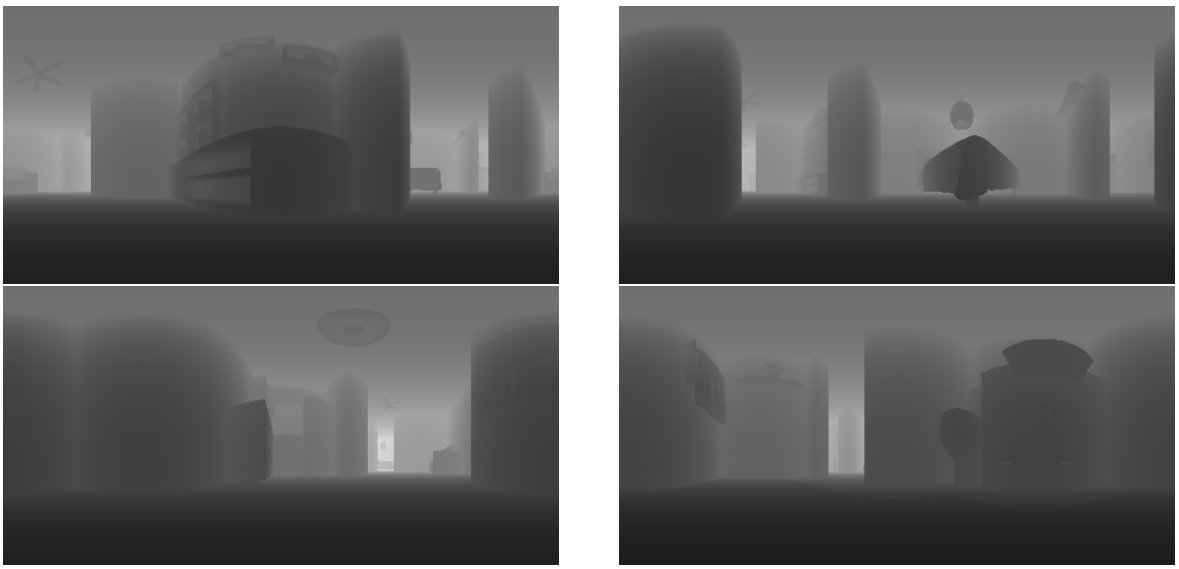


Figure 15. *Samples of generated panoramic depth maps.*

Images were used as inputs to our algorithm since this represents a possible realistic setting and since it is a nonlinear sensor modality which is easy to simulate using 3-dimensional modeling software. We wish to emphasize, however, that after the preprocessing stage, in which these images are made invariant to cyclic rotations, the input is no longer treated as an image, and our proposed algorithm uses no additional image or computer vision related computation on the input. This invariance of the algorithm to the input type enables us, for example, to apply a dimensionality reduction using PCA as a preprocessing, although it distorts the image structure.

As discussed in section 5, in order to uncover the intrinsic metric of the manifold from the observed data, we require the intrinsic data sampling to adhere to some known structure which allows for its estimation from observed data. For this experiment, a different intrinsic acquisition model than the one suggested in subsection 5.1 is used.

Metric estimation using a rigid sensor array. In this setting, the intrinsic sampling is assumed to be acquired by the use of a rigid sensor array. Consider the values of an observation function $f : \mathbb{R}^n \to \mathbb{R}^m$ observed at $\mathbf{x} + \hat{\mathbf{u}}$ for a point $\mathbf{x} \in \mathcal{X}$ and a small displacement $\hat{\mathbf{u}}$. According to the Taylor series approximation,

$$f(\mathbf{x} + \hat{\mathbf{u}}) = f(\mathbf{x}) + \frac{df}{dx}(\mathbf{x})\,\hat{\mathbf{u}} + \mathcal{O}\left(\|\hat{\mathbf{u}}\|^2\right).$$

If the difference between the two measurements $\|\hat{\mathbf{u}}\|^2$ is small with respect to the higher derivatives of the function f, we get that

$$f(\mathbf{x} + \hat{\mathbf{u}}) - f(\mathbf{x}) \approx \frac{df}{dx}(\mathbf{x})\,\hat{\mathbf{u}}.$$

If one observes the function at $\mathbf{x}$ and at k displacements, $\mathbf{x} + \hat{\mathbf{u}}_1, \ldots, \mathbf{x} + \hat{\mathbf{u}}_k$, we get the following set of equations:

$$\begin{aligned} f(\mathbf{x}+\hat{\mathbf{u}}_1)-f(\mathbf{x}) &= \frac{df}{dx}(\mathbf{x})\,\hat{\mathbf{u}}_1 \\ &\vdots \\ f(\mathbf{x}+\hat{\mathbf{u}}_k)-f(\mathbf{x}) &= \frac{df}{dx}(\mathbf{x})\,\hat{\mathbf{u}}_k. \end{aligned}$$

Recasting in matrix form gives

$$\mathbf{D} = \frac{df}{dx}(\mathbf{x})\,\hat{\mathbf{U}},$$

where

$$\begin{aligned} \hat{\mathbf{U}} &= [\hat{\mathbf{u}}_1,\ldots,\hat{\mathbf{u}}_k], \\ \mathbf{D} &= [f(\mathbf{x}+\hat{\mathbf{u}}_1)-f(\mathbf{x}),\ldots,f(\mathbf{x}+\hat{\mathbf{u}}_k)-f(\mathbf{x})]. \end{aligned}$$

If the vectors $\hat{\mathbf{u}}_1,\hat{\mathbf{u}}_2,\ldots,\hat{\mathbf{u}}_k$ span the n-dimensional intrinsic space (i.e., the matrix $\hat{\mathbf{U}}$ has a full row rank), we can invert this relationship using the pseudoinverse of $\hat{\mathbf{U}}$ in order to recover the Jacobian of the observation function by

$$\mathbf{D}\hat{\mathbf{U}}^T\left[\hat{\mathbf{U}}\hat{\mathbf{U}}^T\right]^{-1} = \frac{df}{dx}(\mathbf{x}).$$

This suggests a method for estimating the Jacobian of an unknown observation function using an array of measurements with a known structure. The metric $\mathbf{M}(\mathbf{y})$ can then be estimated by

$$\mathbf{M}(\mathbf{y}) = \frac{df}{dx}(\mathbf{x})\frac{df}{dx}(\mathbf{x})^T \approx \mathbf{D}\hat{\mathbf{U}}^T\left[\hat{\mathbf{U}}\hat{\mathbf{U}}^T\right]^{-1}\left[\hat{\mathbf{U}}\hat{\mathbf{U}}^T\right]^{-T}\hat{\mathbf{U}}\mathbf{D}. \tag{13}$$

We notice that if this array is rotated around the point $\mathbf{x}$ with a rotation matrix $\mathbf{R}$, we get a rotation of the estimated Jacobian:

$$\begin{aligned} \mathbf{D}\left[\mathbf{R}\hat{\mathbf{U}}\right]^T\left[\mathbf{R}\hat{\mathbf{U}}\left[\mathbf{R}\hat{\mathbf{U}}\right]^T\right]^{-1} &= \mathbf{D}\hat{\mathbf{U}}^T\mathbf{R}^T\left[\mathbf{R}\hat{\mathbf{U}}\hat{\mathbf{U}}^T\mathbf{R}^T\right]^{-1} \\ &= \mathbf{D}\hat{\mathbf{U}}^T\mathbf{R}^T\mathbf{R}^{-T}\left[\hat{\mathbf{U}}\hat{\mathbf{U}}^T\right]^{-1}\mathbf{R}^{-1} \\ &= \mathbf{D}\hat{\mathbf{U}}^T\left[\hat{\mathbf{U}}\hat{\mathbf{U}}^T\right]^{-1}\mathbf{R}^{-1} \\ &= \frac{df}{dx}(\mathbf{x})\,\mathbf{R}^{-1}. \end{aligned}$$

However, the estimation of $\mathbf{M}(\mathbf{y})$ in (13) is unaffected by this rotation since

$$\frac{df}{dx}(\mathbf{x})\,\mathbf{R}^{-1}\left[\frac{df}{dx}(\mathbf{x})\,\mathbf{R}^{-1}\right]^T = \frac{df}{dx}(\mathbf{x})\,\mathbf{R}^{-1}\mathbf{R}^{-T}\cdot\left[\frac{df}{dx}(\mathbf{x})\right] = \frac{df}{dx}(\mathbf{x})\frac{df}{dx}(\mathbf{x})^T.$$

This shows that the estimation proposed in (13) is invariant to a rigid rotation of the measurement directions $\hat{\mathbf{u}}_1,\ldots,\hat{\mathbf{u}}_k$. The invariance to rotations makes this measurement setting

practical for intrinsic metric estimation since the sensor array at different points can be rotated and the respective observations do not need to be aligned. In our experiment, the sensor array has an "L" shape and consists of three sensors with a 15cm distance between measurements, as illustrated in Figure 16.

Slight variations in the location of the agent cause slight variations in the point of view of the camera and therefore in the rendered image, as seen in Figure 16. These slight observed variations, combined with our assumption about the intrinsic structure of the data, allow us to infer the local intrinsic metric.

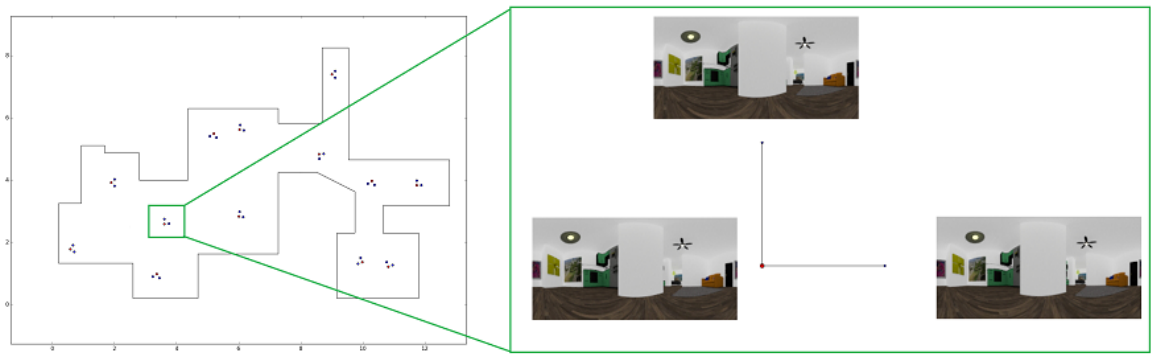


Figure 16. *Observation points using a rigid sensor array for a subset of* 13 *sample points. On the right, one can see the effect of slight variations in the agent's position on the viewed panoramic images.*

For both modalities, $N = 2000$ locations were sampled, and three measurements were made at each such location using the sensor array described above. A 2-dimensional embedding was then constructed based on these measurements and using the proposed algorithm. The constructed embedding was compared to the embedding obtained using the standard Isomap method. Results are presented in Figures 17 and 18.

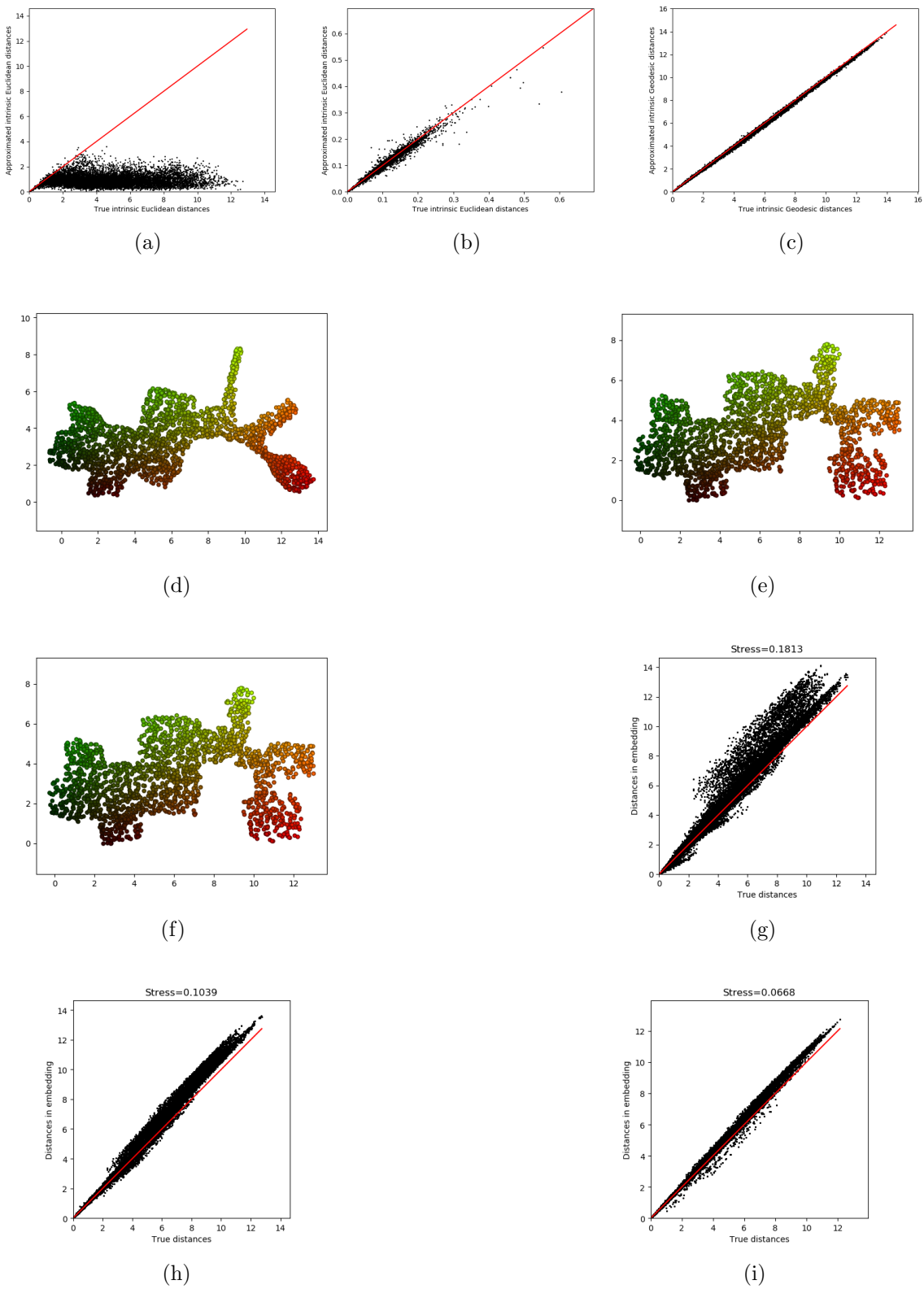


Figure 17. *Mapping and localization from color image observations.* (a) *Intrinsic Euclidean distance estimation.* (b) *Intrinsic Euclidean distance estimation (k-NN).* (c) *Intrinsic geodesic distance estimation (k-NN).* (d) *Embedding using intrinsic Isomap.* (e) *Embedding using proposed intrinsic isometric method (one-shot).* (f) *Embedding using proposed intrinsic isometric method (multiscale).* (g) *Embedding using intrinsic Isomap.* (h) *Stress of embedding using proposed intrinsic isometric method (one-shot).* (i) *Stress of embedding using proposed intrinsic isometric method (multiscale).*

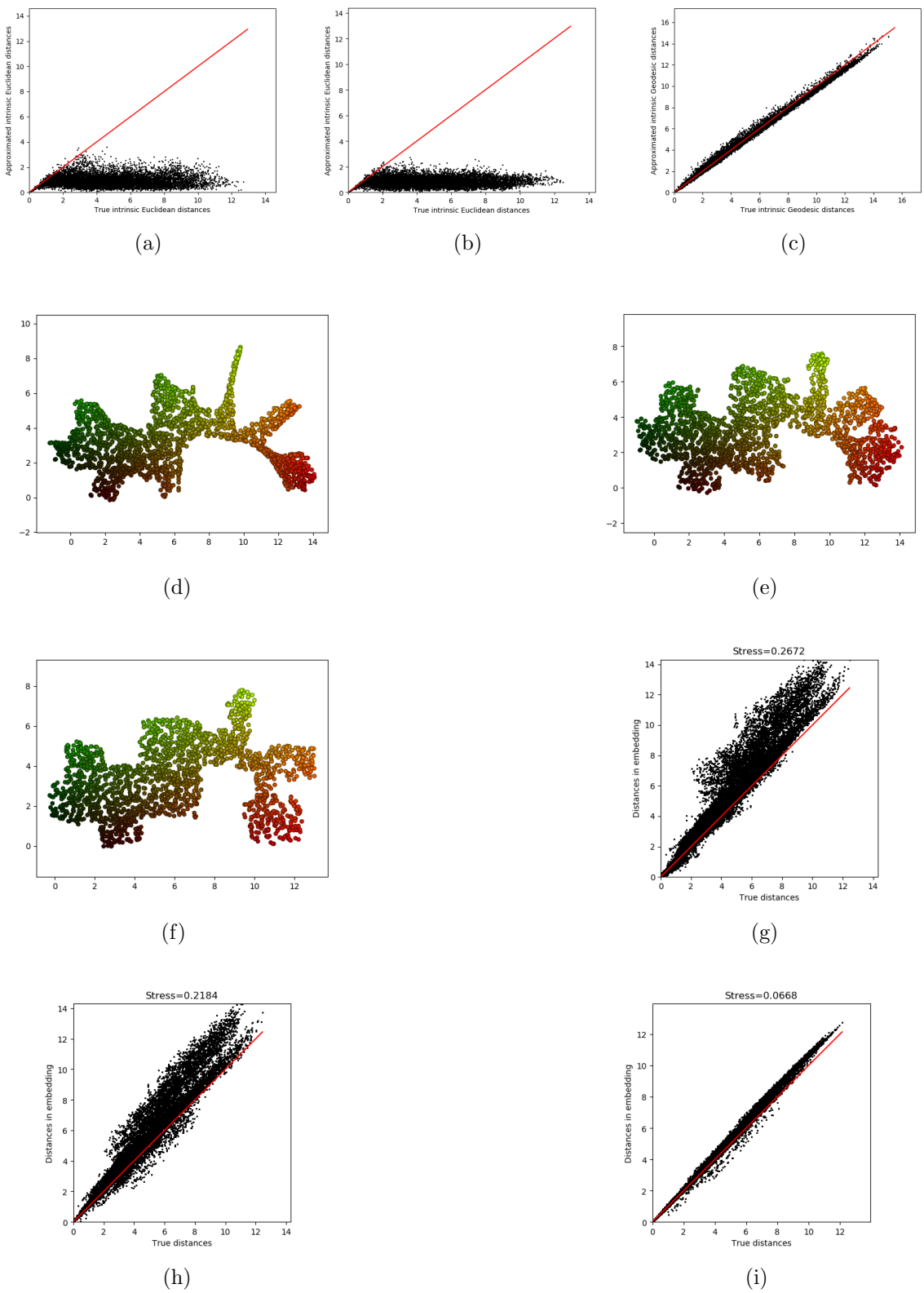


Figure 18. *Mapping and localization from depth image observations.* (a) *Intrinsic Euclidean distance estimation.* (b) *Intrinsic Euclidean distance estimation (k-NN).* (c) *Intrinsic geodesic distance estimation (k-NN).* (d) *Embedding using intrinsic Isomap.* (e) *Embedding using proposed intrinsic isometric method (one-shot).* (f) *Embedding using proposed intrinsic isometric method (multiscale).* (g) *Embedding using intrinsic Isomap.* (h) *Stress of embedding using proposed intrinsic isometric method (one-shot).* (i) *Stress of embedding using proposed intrinsic-isometric method (multiscale).*

The proposed algorithm accurately retrieves the intrinsic structure for the two observation modalities. This is evident by the recovered intrinsic isometric embedding structure, which is almost identical to the structure of the points in the latent space. This can be observed for both modalities by comparing Figure 17(a) to Figure 17(e) and by comparing Figure 18(a) to Figure 18(e), as well as from the fact that most pairwise distances in the final embedding closely approximate the true intrinsic Euclidean distances. This also leads to a low-stress value for the embedding (as can be observed for both modalities in Figure 17(f) and Figure 18(f)). The success of the embedding stems from the fact that the estimated intrinsic metric allows for a good estimation of short-range intrinsic distances (as can be observed for both the modalities in Figure 17(c), Figure 17(d), Figure 18(c), and Figure 18(d)). We notice that since the observation functions used are not locally isometric, standard Isomap fails to retrieve the intrinsic structure of the data or to even provide a 2-dimensional parametrization of the latent space (as can be observed for the two modalities in Figure 17(b) and Figure 18(b)). Isomap does, for the most part, preserve proximity on a local scale (as evidenced by points with similar colors being embedded close to each other), yet the global structure is not preserved (as can be observed for the two modalities by comparing Figure 17(a) and Figure 17(b) and Figure 18(a) and Figure 18(b)). Since standard Isomap is nonintrinsic, it is affected by the modality of the observation, and we receive different embeddings for different sensor modalities (as can be observed by comparing Figure 17(b) and Figure 18(b)).

One noticeable weak point of our algorithm, which manifests slightly in these examples, occurs around the coordinate $(9, 4)$ in the true intrinsic space (Figure 17(a) and Figure 18(a)). This region corresponds to a narrow area in the apartment model in which there are not a lot of sample points. Typically, errors in distance estimations are attenuated by global self-consistency. However, the distance estimations in this particular region are not sufficiently redundant due to the small number of sample points (and pairwise short distances). This leads to a slight distortion in the embedding of this region, which causes a "bend" in the global structure of the embedding (as seen in Figure 17(f) and Figure 18(f)).

We also attempt to apply other manifold learning methods both in their standard form and in their intrinsic variant. Results are presented in Figure 19. None of these methods recover the intrinsic structure of the data in either their standard form or their intrinsic variant.

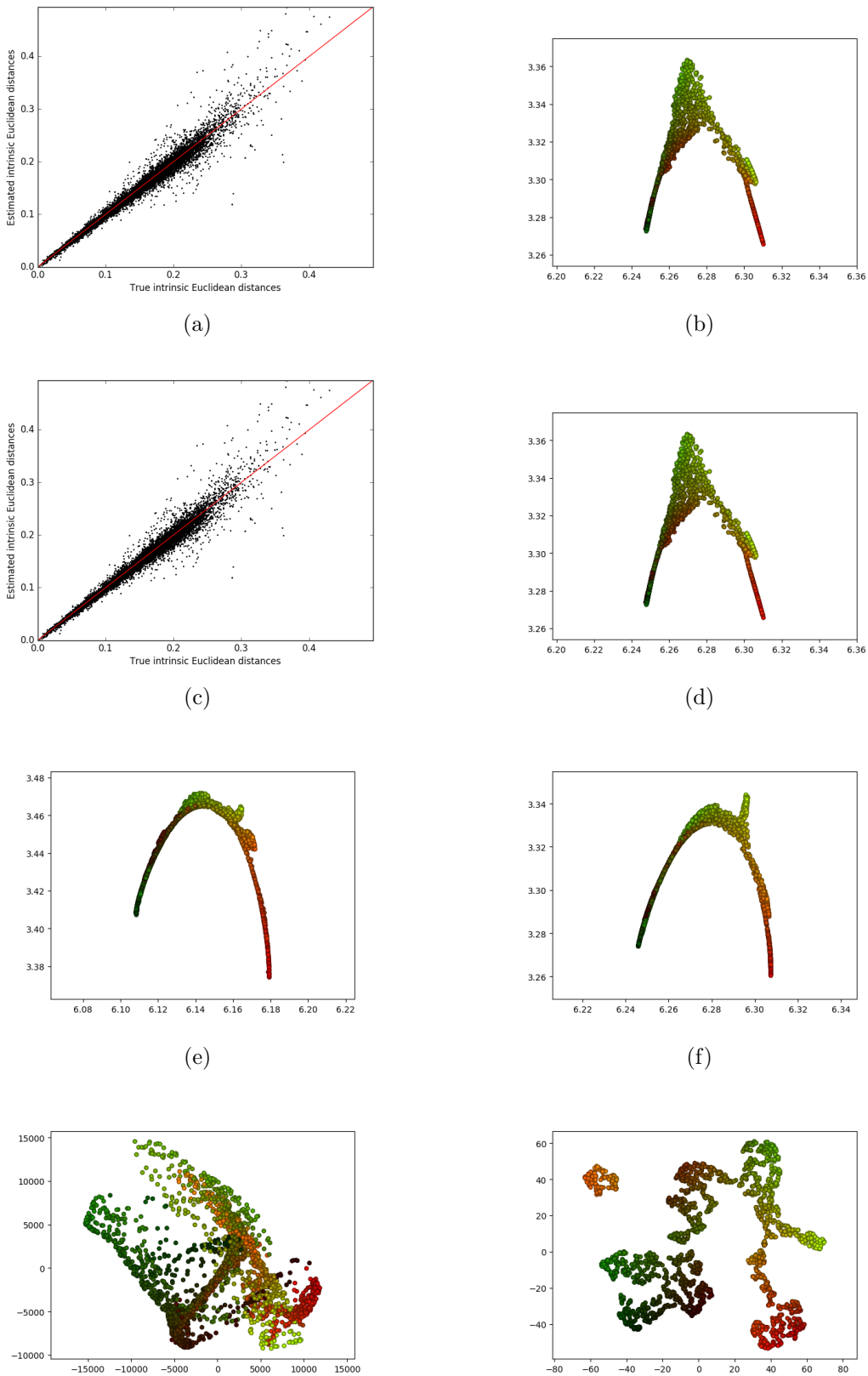


Figure 19. *Intrinsic variants of various local scale manifold learning methods.* (a) *Standard LTSA.* (b) *Intrinsic LTSA.* (c) *Standard HLLE.* (d) *Intrinsic HLLE.* (e) *Standard diffusion map.* (f) *Intrinsic diffusion map.* (g) *Standard t-SNE.* (h) *Intrinsic t-SNE.*

These results show that our intrinsic isometric dimensionality reduction algorithm was successfully applied to the problem of localization and mapping. Basic physical experiments using a randomly walking agent in 2-dimensional space are underway. In these experiments, a robotic iRobot Roomba programmable robotic vacuum cleaner was controlled by a Raspberry-Pi minicomputer and performed a random walk inside a 2-dimensional region. Signal acquisition was performed either by measuring the RSS from WiFi stations or via a wide angle lens which produced 360-degree images. Results from the experiments will be published when concluded.

The advantage of our algorithm is that it exploits a simple prior assumption on the unlabeled measurement acquisition process in order to provide latent system recovery in a manner which is both unsupervised and modality invariant. These two qualities make it especially suitable for settings where one wants to use an automatic algorithm for localization and/or when the observation model is unknown, which is often the case with indoor localization. However, we wish to remark that we make no claim that this algorithm is superior or even comparable to existing algorithms tailored to specific measurement modalities (if the measurement modality is known a priori) or to machine learning tools trained on labeled data sets.

Relation to the sensor network localization problem. The intrinsicness of our method is achieved by using the push-forward metric for the approximation of short-range pairwise distances. Once these distances have been calculated, an embedding of the data into a lower-dimensional Euclidean space which respects these distances is constructed. This construction implies that manifold learning problems (with local intrinsic metrics) can be recast as a sensor network localization (SNL) problem [16, 62, 3, 11, 39, 35, 4, 9, 26, 47, 44, 45, 28], in which we are given a number of scattered sensors, each only able to measure distances to a limited number of neighboring sensors, and the task then is to generate a global mapping or embedding of all the sensors in order to accurately localize each sensor.

Perhaps the leading approach to SNL problems is based on relaxation to a semidefinite programming (SDP) problem [61, 9]. This approach suffers from two main shortcomings. First, SDP methods scale poorly as the number of points increases, since SDP solvers scale cubically with the number of points and constraints. A possible way to reduce the complexity of this approach is to consider representing it in the basis of the leading eigenvectors of the graph Laplacian [61]. However, this tends to oversmooth the constructed embedding and poorly models the boundaries of the sensor network. A second problem is that the optimal inner-product matrix, found by solving the corresponding SDP, is not necessarily of rank corresponding to the intrinsic dimension. To reduce the chance of non–low-rank solutions, a maximum-variance regularization term can be added [60, 61]. This, however, still does not strictly constrain the rank of the embedding, and after the fact dimension truncation of the embedding results in an embedding which violates the distance restrictions.

The SNL problem does not have a prevailing solution or method since each method suffers from some drawbacks. As a result, most methods take a similar approach to the one taken in this paper and regard the constructed embedding only as an initialization point to be further optimized using local descent methods. We note that a similar method to the one suggested in this paper for constructing an embedding once short pairwise distances are given appears in [47].

The significant difference between SNL methods and this work is the broadness of the scope of applicability. For the SNL problem the starting point is the availability of true short-range pairwise distances between points in a low-dimensional Euclidean space (calculated by specific hardware in the case of the SNL problem), whereas in this work the measurement of pairwise distances is not immediate and the main challenge is to first expose the latent low-dimensional Euclidean structure of systems which initially appear to have high-dimensional and complex structure.

7. Conclusions. In this work, we addressed intrinsic isometric manifold learning. We first showed that the need for methods which preserve the underlying Euclidean geometric structure of data manifolds naturally arises when observing inherently low-dimensional systems via high-dimensional nonlinear measurements. We presented an approach which uses local properties of the observation function to calculate a local intrinsic metric on the observed manifold (the "push-forward" metric) used to estimate intrinsic pairwise distances between data points. We discussed several settings under which estimation of the metric is possible from the observed data themselves; however, we recognized that, due to their local nature, many metric estimation methods are not sufficiently robust to noise and sparse sampling when the observation function is highly nonlinear. To overcome this, we proposed a nonlocal metric estimation method that uses an ANN as a parametric regressor for the intrinsic metric. This computational approach was justified based on a maximum-likelihood estimation under a particular statistical model. We showed that by using smooth nonlinearities in the network and additionally restricting the network structure and its weights, we can provide sufficient regularization for the metric estimation, making it robust to noise and sparse sampling. By combining a robust intrinsic metric estimation method and an algorithm which can use these metrics to build an intrinsic and isometric embedding, we devised an algorithm which can recover the geometric structure of a latent low-dimensional manifold from its observations via an unknown nonlinear high-dimensional function. Finally, we focused on the example of mapping and positioning an agent using a sensor network of unknown nature and modalities. We showed that our proposed algorithm can recover the structure of the space in which the agent moves and can accurately position the agent within that space. Due to the intrinsic nature of our method, this mapping and positioning do not require prior knowledge of the measurement model. This invariance to the type of measurement used was shown to be suitable and required in a setting such as indoor positioning where the exact measurement model is usually unknown.

The manifold learning method proposed in this paper does not require any assumptions regarding the structure of the observed data but rather only regarding the structure of the data in the intrinsic space, making our method invariant to additional deformations and manipulations applied to the observed date. This invariance has the advantage of enabling one to perform preprocessing stages, e.g., for reducing the dimensionality of the data or for removing noise. For example, in subsection 6.2, when localization and mapping were performed using image data, our algorithm was not applied directly to images, but rather to their low-dimensional projections on the principal components of the data. Importantly, such preprocessing could not be used prior to application of vision-based algorithms since these exploit the specific structure of image data which is deformed by the PCA.

Another benefit stemming from the invariance of our method to the observation function is its the natural ability to facilitate a sensor. Measurements from different measurement modalities can be simply concatenated, creating a higher-dimensional observation function. Further research is required, however, in order to determine the proper weighting of different measurement modalities in order to achieve optimal results.

Supervised machine learning uses labeled data (consisting of pairs of input and desired output values) and attempts to "learn" or infer a functional connection between the two. Unfortunately, acquisition of labeled data is usually nontrivial and expensive and requires an already existing method to correctly label data. Our method, when viewed end-to-end, can label a large set of data points by only making an assumption about the intrinsic structure of the measurement process. Our proposed approach can, therefore, be used as a method for automatic acquisition of labeled data, as it operates in a completely unsupervised manner and produces data labeling as an output.

REFERENCES

[1] E. Allen, *Modeling with Itô Stochastic Differential Equations*, Math. Model. Theory Appl. 22, Springer, New York, 2007.

[2] N. Asgharbeygi and A. Maleki, *Geodesic k-means clustering*, in Proceedings of the 19th International Conference on Pattern Recognition (ICPR), IEEE, Washington, DC, 2008, pp. 1–4.

[3] J. Ash and L. Potter, *Sensor network localization via received signal strength measurements with directional antennas*, in Proceedings of the 2004 Allerton Conference on Communication, Control, and Computing, Monticello, IL, 2004, pp. 1861–1870.

[4] M. Bal, M. Liu, W. Shen, and H. Ghenniwa, *Localization in cooperative wireless sensor networks: A review*, in Proceedings of the 13th International Conference on Computer Supported Cooperative Work in Design (CSCWD) IEEE, Washington, DC, 2009, pp. 438–443.

[5] M. Belkin and P. Niyogi, *Laplacian eigenmaps and spectral techniques for embedding and clustering*, in Proceedings of Advances in Neural Information Processing Systems (NIPS), vol. 14, Montreal, Canada, 2001, pp. 585–591.

[6] Y. Bengio, A. Courville, and P. Vincent, *Representation learning: A review and new perspectives*, IEEE Trans. Pattern Anal. Mach. Intell., 35 (2013), pp. 1798–1828.

[7] Y. Bengio and M. Monperrus, *Non-local manifold tangent learning*, in Proceedings of Advances in Neural Information Processing Systems (NIPS), Vancouver, Canada, 2004, pp. 129–136.

[8] J. Bergstra, O. Breuleux, F. Bastien, P. Lamblin, R. Pascanu, G. Desjardins, J. Turian, D. Warde-Farley, and Y. Bengio, *Theano: A CPU and GPU math compiler in Python*, in Proceedings of the 9th Python in Science Conference, 2010, pp. 1–7.

[9] P. Biswas, T.-C. Lian, T.-C. Wang, and Y. Ye, *Semidefinite programming based algorithms for sensor network localization*, ACM Trans. Sensor Networks, 2 (2006), pp. 188–220.

[10] I. Borg and P. J. Groenen, *Modern Multidimensional Scaling: Theory and Applications*, Springer, New York, 2005.

[11] A. Boukerche, H. A. Oliveira, E. F. Nakamura, and A. A. Loureiro, *Localization systems for wireless sensor networks*, IEEE Wireless Commun., 14 (2007), pp. 6–12.

[12] A. Boyarski, A. M. Bronstein, and M. M. Bronstein, *Subspace least squares multidimensional scaling*, in International Conference on Scale Space and Variational Methods in Computer Vision, Springer, New York, 2017, pp. 681–693.

[13] C. Bregler and S. M. Omohundro, *Nonlinear image interpolation using manifold learning*, in Proceedings of Advances in Neural Information Processing Systems (NIPS), Denver, CO, 1995, pp. 973–980.

[14] A. Buja, D. F. Swayne, M. L. Littman, N. Dean, H. Hofmann, and L. Chen, *Data visualization with multidimensional scaling*, J. Comput. Graph. Statist., 17 (2008), pp. 444–472.

[15] R. R. Coifman and S. Lafon, *Diffusion maps*, Appl. Comput. Harmon. Anal., 21 (2006), pp. 5–30.

[16] J. A. Costa, N. Patwari, and A. O. Hero III, *Distributed weighted-multidimensional scaling for node localization in sensor networks*, ACM Trans. Sensor Networks, 2 (2006), pp. 39–64.
[17] J. de Leeuw, *Applications of Convex Analysis to Multidimensional Scaling*, Department of Statistics Papers, UCLA, Los Angeles, CA, 2005.
[18] D. L. Donoho and C. Grimes, *Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data*, Proc. Natl. Acad. Sci. USA, 100 (2003), pp. 5591–5596.
[19] C. J. Dsilva, R. Talmon, R. R. Coifman, and I. G. Kevrekidis, *Parsimonious representation of nonlinear dynamical systems through manifold learning: A chemotaxis case study*, Appl. Comput. Harmon. Anal., 44 (2018), pp. 759–773.
[20] C. J. Dsilva, R. Talmon, C. W. Gear, R. R. Coifman, and I. G. Kevrekidis, *Data-Driven Reduction for Multiscale Stochastic Dynamical Systems*, preprint, https://arxiv.org/abs/1501.05195, 2015.
[21] C. J. Dsilva, R. Talmon, N. Rabin, R. R. Coifman, and I. G. Kevrekidis, *Nonlinear intrinsic variables and state reconstruction in multiscale simulations*, J. Chem. Phys., 139 (2013), 184109.
[22] D. Duncan, R. Talmon, H. P. Zaveri, and R. R. Coifman, *Identifying preseizure state in intracranial EEG data using diffusion kernels*, Math. Biosci. Engrg., 10 (2013), pp. 579–590.
[23] H. W. Engl and P. Kügler, *Nonlinear inverse problems: Theoretical aspects and some industrial applications*, in Multidisciplinary Methods for Analysis Optimization and Control of Complex Systems, Springer, 2005, pp. 3–47.
[24] E. R. Gansner, Y. Koren, and S. North, *Graph drawing by stress majorization*, in International Symposium on Graph Drawing, Springer, New York, 2004, pp. 239–250.
[25] C. Gardiner, *Stochastic Methods*, Springer Ser. Synergetics 13, Springer, Berlin, 2009.
[26] S. Gepshtein and Y. Keller, *Sensor network localization by augmented dual embedding*, IEEE Trans. Signal Process., 63 (2015), pp. 2420–2431.
[27] T. Kamada and S. Kawai, *An algorithm for drawing general undirected graphs*, Inform. Process. Lett., 31 (1989), pp. 7–15.
[28] Y. Keller and Y. Gur, *A diffusion approach to network localization*, IEEE Trans. Signal Process., 59 (2011), pp. 2642–2654.
[29] J. Kim, K.-H. Shim, and S. Choi, *Soft geodesic kernel k-means*, in Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, Washington, DC, 2007, pp. 429–432.
[30] D. Kingma and J. Ba, *Adam: A Method for Stochastic Optimization*, preprint, https://arxiv.org/abs/1412.6980, 2014.
[31] E. Kloeden and E. Platen, *Numerical Solution of Stochastic Differential Equations*, Springer, Berlin, 1992.
[32] A. Krogh and J. A. Hertz, *A simple weight decay can improve generalization*, in Proceedings of Advances in Neural Information Processing Systems (NIPS), vol. 4, Denver, CO, 1991, pp. 950–957.
[33] J. B. Kruskal, *Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis*, Psychometrika, 29 (1964), pp. 1–27.
[34] G. Mishne, R. Talmon, and I. Cohen, *Graph-based supervised automatic target detection*, IEEE Trans. Geosci. Remote Sensing, 53 (2015), pp. 2738–2754.
[35] R. L. Moses, D. Krishnamurthy, and R. M. Patterson, *A self-localization method for wireless sensor networks*, EURASIP J. Adv. Signal Process., 2003 (2003), 839843.
[36] B. Øksendal, *Stochastic differential equations*, in Stochastic Differential Equations: An Introduction with Applications, Springer, New York, 2003, pp. 65–84.
[37] D. Perraul-Joncas and M. Meila, *Non-linear Dimensionality Reduction: Riemannian Metric Estimation and the Problem of Geometric Discovery*, preprint, https://arxiv.org/abs/1305.7255, 2013.
[38] P. Petersen, S. Axler, and K. Ribet, *Riemannian Geometry*, Grad. Texts in Math. 171, Springer, New York, 2006.
[39] V. Ramadurai and M. L. Sichitiu, *Localization in wireless sensor networks: A probabilistic approach*, in International Conference on Wireless Networks, Las Vegas, NV, 2003, pp. 275–281.
[40] G. Rosman, A. M. Bronstein, M. M. Bronstein, A. Sidi, and R. Kimmel, *Fast Multidimensional Scaling Using Vector Extrapolation*, Tech. report, Computer Science Department, Technion, Haifa, Israel, 2008.

[41] G. Rosman, M. M. Bronstein, A. M. Bronstein, and R. Kimmel, *Nonlinear dimensionality reduction by topologically constrained isometric embedding*, Internat. J. Comput. Vis., 89 (2010), pp. 56–68.
[42] S. T. Roweis and L. K. Saul, *Nonlinear dimensionality reduction by locally linear embedding*, Science, 290 (2000), pp. 2323–2326.
[43] L. K. Saul and S. T. Roweis, *Think globally, fit locally: Unsupervised learning of low dimensional manifolds*, J. Mach. Learn. Res., 4 (2003), pp. 119–155.
[44] A. Savvides, C.-C. Han, and M. B. Strivastava, *Dynamic fine-grained localization in ad-hoc networks of sensors*, in Proceedings of the 7th Annual International Conference on Mobile Computing and Networking, ACM, New York, 2001, pp. 166–179.
[45] A. Savvides, H. Park, and M. B. Srivastava, *The bits and flops of the n-hop multilateration primitive for node localization problems*, in Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications, ACM, New York, 2002, pp. 112–121.
[46] B. Schölkopf, A. Smola, and K.-R. Müller, *Kernel principal component analysis*, in Proceedings of the International Conference on Artificial Neural Networks, Springer, New York, 1997, pp. 583–588.
[47] Y. Shang, W. Ruml, Y. Zhang, and M. P. Fromherz, *Localization from mere connectivity*, in Proceedings of the 4th ACM International Symposium on Mobile Ad Hoc Networking & Computing, ACM, New York, 2003, pp. 201–212.
[48] A. Singer and R. R. Coifman, *Non-linear independent component analysis with diffusion maps*, Appl. Comput. Harmon. Anal., 25 (2008), pp. 226–239.
[49] O. Sorkine-Hornung and M. Rabinovich, *Least-Squares Rigid Motion Using SVD*, Technical note, Department of Computer Science, ETH Zurich, Zurich, Switzerland, 2017; available online at https://igl.ethz.ch/projects/ARAP/svd_rot.pdf.
[50] M. Spivak, *A Comprehensive Introduction to Differential Geometry, Vol.* IV[a], Publish or Perish, Houston, TX, 1981.
[51] R. Talmon, I. Cohen, and S. Gannot, *Supervised source localization using diffusion kernels*, in Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA), IEEE, Washington, DC, 2011, pp. 245–248.
[52] R. Talmon and R. R. Coifman, *Empirical intrinsic geometry for nonlinear modeling and time series filtering*, Proc. Natl. Acad. Sci. USA, 110 (2013), pp. 12535–12540.
[53] R. Talmon and R. R. Coifman, *Intrinsic modeling of stochastic dynamical systems using empirical geometry*, Appl. Comput. Harmon. Anal., 39 (2015), pp. 138–160.
[54] R. Talmon, D. Kushnir, R. R. Coifman, I. Cohen, and S. Gannot, *Parametrization of linear systems using diffusion kernels*, IEEE Trans. Signal Process., 60 (2012), pp. 1159–1173.
[55] R. Talmon, S. Mallat, H. Zaveri, and R. R. Coifman, *Manifold learning for latent variable inference in dynamical systems*, IEEE Trans. Signal Process., 63 (2015), pp. 3843–3856.
[56] J. B. Tenenbaum, V. de Silva, and J. C. Langford, *A global geometric framework for nonlinear dimensionality reduction*, Science, 290 (2000), pp. 2319–2323.
[57] M. E. Tipping and C. M. Bishop, *Mixtures of probabilistic principal component analyzers*, Neural Comput., 11 (1999), pp. 443–482.
[58] M. E. Tipping and C. M. Bishop, *Probabilistic principal component analysis*, J. Roy. Statist. Soc. Ser. B Statist. Methodol., 61 (1999), pp. 611–622.
[59] M. W. Trosset and R. Mathar, *On the existence of nonglobal minimizers of the stress criterion for metric multidimensional scaling*, in Proceedings of the Statistical Computing Section, American Statistical Association, Alexandria, VA, 1997, pp. 158–162.
[60] K. Q. Weinberger and L. K. Saul, *Unsupervised learning of image manifolds by semidefinite programming*, Internat. J. Comput. Vis., 70 (2006), pp. 77–90.
[61] K. Q. Weinberger, F. Sha, Q. Zhu, and L. K. Saul, *Graph Laplacian regularization for large-scale semidefinite programming*, in Proceedings of Advances in Neural Information Processing Systems (NIPS), Vancouver, Canada, 2007, pp. 1489–1496.
[62] J. Yang and Y. Chen, *Indoor localization using improved RSS-based lateration methods*, in Proceedings of the 2009 Global Telecommunications Conference (GLOBECOM), IEEE, Washington, DC, 2009, pp. 1–6.

[63] Z. Zhang and J. Wang, *MLLE: Modified locally linear embedding using multiple weights*, in Proceedings of Advances in Neural Information Processing Systems 19 (NIPS 19), Vancouver, Canada, 2006, pp. 1593–1600.

[64] Z. Zhang and H. Zha, *Principal manifolds and nonlinear dimensionality reduction via tangent space alignment*, SIAM J. Sci. Comput., 26 (2004), pp. 313–338, https://doi.org/10.1137/S1064827502419154.