Discrete Hessian Eigenmaps Method for Dimensionality Reduction

Qiang Ye

Department of Mathematics, University of Kentucky, Lexington, KY 40506, USA

Weifeng Zhi*

Department of Computer Science, University of California, Davis, Davis, CA 95616, USA

Abstract

For a given set of data points lying on a low-dimensional manifold embedded in a high-dimensional space, the dimensionality reduction is to recover a low-dimensional parametrization from the data set. The recently developed Hessian Eigenmaps is a mathematically rigorous method that also sets a theoretical framework for the nonlinear dimensionality reduction problem. In this paper, we develop a discrete version of the Hessian Eigenmaps method and present an analysis, giving conditions under which the method works as intended. As an application, a procedure to modify the standard constructions of k-nearest neighborhoods is presented to ensure that Hessian LLE can recover the original coordinates up to an affine transformation.

Keywords: Hessian Eigenmaps, Dimensionality Reduction, Null Space, Hessian Matrix

1. Introduction

High dimensional data sets arise in many real-world applications. They may be obtained from various measurements or sensing systems, such as systems of digital cameras, video surveillance, text document processing, and digital sound analysis. These data points may lie approximately on a low

*Corresponding author

Email addresses: qye3@uky.edu (Qiang Ye), wzhi@ucdavis.edu (Weifeng Zhi)

dimensional manifold embedded in a high dimensional space. Dimensionality reduction (or called manifold learning) is to recover a set of low-dimensional parametric representations for the high-dimensional data points, which may be used for further processing or visualization of the data.

Mathematically, consider a d-dimensional parameterized manifold $\mathcal{M}$ embedded in $\mathbb{R}^n$ $(d < n)$ characterized by a possibly nonlinear map ψ: $\mathcal{C} \subset \mathbb{R}^d \to \mathbb{R}^n$, where $\mathcal{C}$ is a compact and connected subset of $\mathbb{R}^d$. Here $\mathbb{R}^n$ is the high-dimensional data space with $\mathcal{M} = \psi(\mathcal{C})$ being the manifold containing data points and $\mathbb{R}^d$ is the low-dimensional parameter space. Suppose we have a set of data points $m_1, \ldots, m_N$ sampled from the manifold $\mathcal{M}$ with

$$m_i = \psi(\tau_i), \quad i = 1, \ldots, N, \tag{1}$$

for some $\tau_i \in \mathcal{C}$. Then the dimensionality reduction (or manifold learning) problem is to recover the parameter points τ_i's and/or the map ψ from m_i's. Clearly, this problem is not well defined for a general nonlinear map ψ. However, as is shown by Donoho and Grimes in the derivation of the Hessian Eigenmaps method [1], if ψ is a local isometric map, then $\tau = \psi^{-1}(m)$ is uniquely determined up to a rigid motion and hence captures the geometric structure of the data set. In this paper, we consider the noise-free data (1) and assume for the theoretical purpose that ψ is a local isometry.

Traditionally, the linear dimensionality reduction problem has been considered where the data set lies close to an affine subspace, i.e. ψ is a linear map. Such a problem can be solved by the principal component analysis (PCA) for example. However, only limited data problems admit a linear structure. Recent interests have focused on nonlinear dimensionality reductions where ψ is nonlinear. Since the publications of two methods for nonlinear dimensionality reductions called Locally Linear Embedding (LLE) [2] and Isometric Mapping (Isomap) [3] in 2000, several other methods and generalizations have been proposed, which include Laplacian Eigenmaps [4], Hessian Eigenmaps [1], and Local Tangent Space Alignment (LTSA) [5] among many others [6, 7, 8, 9]; see also [10] for a review. While theoretical analysis has been developed for some of these methods [11, 12, 13, 14, 15, 16, 17, 18, 19, 20], theoretical understanding of many of them remains limited.

The Hessian Eigenmaps [1] is a mathematically rigorous method based on a theory that also sets a theoretical framework for the nonlinear dimensionality reduction problem. For each function $f : \mathcal{M} \mapsto \mathbb{R}$, a Hessian operator

$H_f(m)$ of f can be defined at $m \in \mathcal{M}$ as the Hessian matrix of an extension of f through a projection on the tangent plane at m. From this, the quadratic form $\mathcal{H}(f) = \int_{\mathcal{M}} \|H_f(m)\|_F^2 dm$ called $\mathcal{H}$-functional is introduced. If ψ is a local isometric map, it is proved in [1] that the $\mathcal{H}$-functional $\mathcal{H}(f)$ has a $(d+1)$-dimensional null space spanned by the constant function and the d component functions of ψ^{-1}; see [1] or Section 2 for details. Note that for each $m \in \mathcal{M}$, $\tau = \psi^{-1}(m)$ is the original coordinate. Hence, the locally isometric coordinate, up to a linear transformation[1], is uniquely determined and can be recovered by computing the null space of the $\mathcal{H}$-functional. This method called the Hessian Eigenmaps is however only a theoretical method as it is based on the continuous setting.

For a practical problem with a discrete data set (1), a numerical procedure called Hessian LLE is derived in [1] to construct some form of approximate $\mathcal{H}$-functional. This involves constructing local neighborhoods and computing local tangent space coordinates for points in a small neighborhood for the use of constructing some equivalent form of the local Hessian operator, from which an $\mathcal{H}$-functional is constructed; see [1] or Section 2 for details. As a result of various approximations and necessary modifications of definition to the discrete setting, it is not clear whether an approximate $\mathcal{H}$-functional so obtained still approximately recovers the original isometric coordinates. Indeed, for 1-dimensional problems ($d = 1$), Hessian LLE usually fails with some commonly used constructions of local neighborhoods such as k-nearest neighborhoods; see discussion in Section 4 and numerical examples in Section 5. One way to address this difficulty is a modified construction of k-nearest neighborhoods introduced in the implementation in [21].

In this paper, we present a discrete version of the Hessian Eigenmaps method, which is theoretically equivalent to the Hessian LLE method, an implementation of the Hessian Eigenmap in the discreet setting introduced in [1]. The main feature of our formulation is the introduction of a discrete Hessian operator defined from points in a small neighborhood. Given N data points in (1), by using the discrete Hessian operators defined for a set of small neighborhoods, a matrix Ψ is constructed that represents a quadratic form generalizing the $\mathcal{H}$-functional and is called a Hessian alignment matrix. Provided that local coordinates used to define the discrete Hessian operators

[1] With an additional step of matching with local coordinates in one local neighborhood [1, 16], the uncertainty can be reduced to an orthogonal transformation, i.e. a rigid motion.

are computed correctly, we will show under certain conditions on the local neighborhoods that

$$\mathsf{null}(\Psi) = \mathsf{span}([e, T^T]),$$

where $T = [\tau_1, \ldots, \tau_N]$ and e is the vector of all 1's. Then, the matrix T or the original coordinates τ_i can be recovered, up to a linear transformation, from the null space of Ψ. This generalizes Donoho and Grimes' theorem on the null space of $\mathcal{H}(f)$ and provides a theoretical analysis of the Hessian LLE . As an implication of this analysis, we propose a modified neighborhood construction to improve robustness of Hessian LLE. Numerical examples will be presented to illustrate the improvements.

The paper is organized as follows. In Section 2, we describe the continuous Hessian Eigenmaps method and the associated numerical procedure for implementation as presented in [1]. In Section 3, we formulate Hessian LLE as a discrete version of the Hessian Eigenmaps and provide an analysis for Hessian alignment matrix. In Section 4, we discuss constructions of local neighborhoods and present a method to expand neighborhoods for proper recovery of original coordinates based on our result. We will also present a numerical example in Section 5, followed by some concluding remarks in Section 6.

Notation. Throughout, e denotes a column vector of all ones, the dimension of which should be determined from the context. $\mathsf{null}(\cdot)$ is the null space of a matrix, and $\mathsf{span}(\cdot)$ denotes the subspace spanned by all the columns of argument matrices. $A^\dagger$ denotes the Moore-Penrose pseudoinverse of A.

For two matrices $A = (a_{ij})_{m\times n} \in \mathbb{R}^{m\times n}$ and $B = (b_{ij})_{p\times q} \in \mathbb{R}^{p\times q}$, $A \bigotimes B \in \mathbb{R}^{mp\times nq}$ represents the Kronecker product of A and B [22, p.274]. For two row vectors $a = [a_1, \ldots, a_n] \in \mathbb{R}^{1\times n}$ and $b = [b_1, \ldots, b_n] \in \mathbb{R}^{1\times n}$, we also define

$$a \bigodot b = [c_1, \ldots, c_{\frac{n(n+1)}{2}}] \in \mathbb{R}^{1\times \frac{n(n+1)}{2}}, \tag{2}$$

where $c_{\frac{k(k-1)}{2}+\ell} = a_k b_\ell$ for $1 \le \ell \le k \le n$.

We call a finite subset $\boldsymbol{S}$ of $\mathbb{R}^d$ an ordered set if an order is associated with its elements. Then the set can be represented by an ordered list as $\boldsymbol{S} = \{\tau_1, \ldots, \tau_k\}$. The matrix

$$T = [\tau_1, \ldots, \tau_k] \tag{3}$$

with its columns given by the vectors in $\boldsymbol{S}$ in the given order is called the corresponding matrix.

2. Hessian Eigenmaps

In this section, we first describe the Hessian Eigenmaps method of [1] in the continuous setting. Given that the map ψ defined in (1) is a local isometric embedding, the map $\phi = \psi^{-1} : \mathcal{M} \subset \mathbb{R}^n \to \mathbb{R}^d$ provides a (locally) isometric coordinate system for $\mathcal{M}$. Each component of ϕ is a function defined on $\mathcal{M}$ that provides one coordinate. The main idea of the Hessian Eigenmaps is to introduce a Hessian operator and a functional called the $\mathcal{H}$-functional defined for functions on $\mathcal{M}$, for which the null space consists of the d coordinate functions and the constant function.

Let $f : \mathcal{M} \mapsto \mathbb{R}$ be a function defined on $\mathcal{M}$ and let m_0 be an interior point of manifold $\mathcal{M}$. We can define a function $h : \mathcal{C} \mapsto \mathbb{R}$ as $h(\tau) = f(\psi(\tau))$, where $\mathcal{C} = \phi(\mathcal{M}) \subset \mathbb{R}^d$ and $\tau = [t_1, \dots, t_d]^T \in \mathcal{C}$. h is called a pullback of f to $\mathcal{C}$. Let $\tau_0 = \phi(m_0)$. We call the Hessian matrix of h at τ_0 the Hessian matrix of function f at m_0 in the isometric coordinate and we denote it by $H_f^{iso}(m_0)$. Then $(H_f^{iso})_{i,j}(m_0) = \frac{\partial^2 h(\tau_0)}{\partial t_i \partial t_j}$. From the Hessian matrix, we define a $\mathcal{H}$-functional of f in isometric coordinates, denoted by $\mathcal{H}^{iso}(f)$, as

$$\mathcal{H}^{iso}(f) = \int_{\mathcal{M}} \|H_f^{iso}(m)\|_F^2 dm, \tag{4}$$

where dm is a probability measure on $\mathcal{M}$ which has strictly positive density everywhere on the interior of $\mathcal{M}$. It is clear that $\mathcal{H}^{iso}$ of the d component functions of ϕ are zero as their pullbacks to $\mathcal{C}$ are linear functions. Indeed, $\mathcal{H}^{iso}(\cdot)$ has a $(d+1)$-dimensional null space, consisting of the span of the constant functions and the d component functions of ϕ; see [1, Corollary 4].

The Hessian matrix and the $\mathcal{H}$-functional in isometric coordinates introduced above are unfortunately not computable without knowing the isometric coordinate system ϕ first. To obtain a functional with the same property but independent of the isometric coordinate system ϕ, a Hessian matrix and the $\mathcal{H}$-functional in local tangent coordinate systems are introduced in [1]. We describe it now.

For a smooth manifold $\mathcal{M}$ and an interior point $m_0 \in \mathcal{M}$, let $T_{m_0}(\mathcal{M})$ denote the tangent space at m_0. Let $\mathcal{N}_{m_0}$ be the set of points $m \in \mathcal{M}$ in a small neighborhood of m_0. Consider the tangent space as a plane at m_0 (or a linear subspace $\mathbb{R}^n$ with the origin at m_0). There is an orthonormal basis $\{v_i, 1 \le i \le d\}$ for $T_{m_0}(\mathcal{M})$, where $v_i \in \mathbb{R}^n$. If $\mathcal{N}_{m_0}$ is a sufficiently small neighborhood, then for any point $m \in \mathcal{N}_{m_0}$, there is a unique point

$v(m) \in T_{m_0}(\mathcal{M})$ that is closest to m. For m_0, the closest point in $T_{m_0}(\mathcal{M})$ is m_0 itself. We can write $v(m)$ in the basis $\{v_i\}$ as

$$v(m) = x_1^{(tan,m_0)}(m)v_1 + \cdots + x_d^{(tan,m_0)}(m)v_d.$$

In this way, each $m \in \mathcal{N}_{m_0}$ is uniquely defined by

$$x^{(tan,m_0)}(m) = [x_1^{(tan,m_0)}(m), \ldots, x_d^{(tan,m_0)}(m)]^T \in \mathbb{R}^d,$$

which we call a local tangent coordinate (parametrization) of $m \in \mathcal{N}_{m_0}$.

Now, let $f \in C^2(\mathcal{M}) : \mathcal{M} \mapsto \mathbb{R}$. It induces $g(x) : x \in U_0 \to \mathbb{R}$ defined by

$$g(x) = f(m),$$

where $x = x^{(tan,m_0)}(m) \in \mathbb{R}^d$ for $m \in \mathcal{N}_{m_0}$ and $U_0 \subset \mathbb{R}^d$ is a small neighborhood of $0 \in \mathbb{R}^d$ such that there is a one-to-one correspondence between $x \in U_0$ and $m \in \mathcal{N}_{m_0}$. From this, we define the Hessian matrix of f at m_0 in the local tangent coordinates as the ordinary Hessian matrix of $g(x)$ at $0 \in \mathbb{R}^d$ and denote it by $H_f^{tan}(m_0) = \big((H_f^{tan})_{i,j}(m_0)\big)$. Then,

$$(H_f^{tan})_{i,j}(m_0) = \frac{\partial^2 g}{\partial x_i \partial x_j}(x)|_{x=0} \quad i, j = 1, \ldots, d.$$

While the Hessian defined above depends on the coordinate systems and the basis chosen for the tangent space, it is easy to see that the Hessians defined under different coordinate systems are orthogonally similar. Thus, it is uniquely defined up to an orthogonal similarity transformation. In particular,

$$\mathcal{H}(f) = \int_{\mathcal{M}} \|H_f^{tan}(m)\|_F^2 dm$$

is well defined and is called $\mathcal{H}$-functional. The following is the main theorem of [1].

Theorem 2.1. *(Donoho and Grimes [1]) Suppose $\mathcal{M} = \psi(\mathcal{C})$ where $\mathcal{C}$ is an open connected subset of $\mathbb{R}^d$, and ψ is a locally isometric embedding of $\mathcal{C}$ into $\mathbb{R}^n$. Then $\mathcal{H}(f) = \mathcal{H}^{iso}(f)$ and they have a $d+1$ dimensional null space consisting of the constant function and a d-dimensional space of functions spanned by the original isometric coordinates (i.e the component functions of ϕ).*

In a practical setting, we are given N high dimensional data points $\boldsymbol{M} = \{m_1, \cdots, m_N\} \subset \mathbb{R}^n$. The following five step numerical procedure called Hessian LLE is introduced in [1] to implement Theorem 2.1.

Step 1 **Identify Local Neighbors.** For every m_i, identify its k nearest point neighborhood $\mathcal{N}_{m_i} = \{m_{i_1}, \ldots, m_{i_k}\}$, $M_i = [m_{i_1}, \ldots, m_{i_k}]^T$. Let $\bar{M}_i = [m_{i_1} - \bar{m}_i, \ldots, m_{i_k} - \bar{m}_i]^T$, where $\bar{m}_i = \frac{1}{k}\sum_{j=1}^{k} m_{i_j}$.

Step 2 **Obtain Local Tangent Coordinates.** Let the singular value decomposition of $\bar{M}_i$ be $\bar{M}_i = U^{(i)}\Sigma^{(i)}{V^{(i)}}^T$, where $U^{(i)} = [u_1^{(i)}, \ldots, u_k^{(i)}] \in \mathbb{R}^{k\times k}$, $\Sigma^{(i)} = \mathrm{diag}(\sigma_1, \ldots, \sigma_k) \in \mathbb{R}^{k\times n}$ with $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_k$ and $V^{(i)} \in \mathbb{R}^{n\times n}$. The first d columns of $V^{(i)}$ span approximately the tangent space at m_i and the tangent coordinates of points in $\mathcal{N}_{m_i}$ are

$$[\theta_1^{(i)}, \ldots, \theta_k^{(i)}] := \mathrm{diag}(\sigma_1, \ldots, \sigma_d)[u_1^{(i)}, \ldots, u_d^{(i)}]^T. \tag{5}$$

Step 3 **Develop Hessian Estimator.** For each i, construct

$$X^{(i)} = \begin{array}{c} \begin{array}{ccc} \scriptstyle 1 & \scriptstyle d & \scriptstyle q \end{array} \\ \begin{pmatrix} 1 & {\theta_1^{(i)}}^T & {\theta_1^{(i)}}^T \bigodot {\theta_1^{(i)}}^T \\ 1 & {\theta_2^{(i)}}^T & {\theta_2^{(i)}}^T \bigodot {\theta_2^{(i)}}^T \\ \vdots & \vdots & \vdots \\ 1 & {\theta_k^{(i)}}^T & {\theta_k^{(i)}}^T \bigodot {\theta_k^{(i)}}^T \end{pmatrix} \end{array},$$

where $q = d(d+1)/2$ and the operation $\bigodot$ is defined according to (2). Perform the Gram-Schmidt orthonormalization process on the columns of $X^{(i)}$ and let $G_i^T \in \mathbb{R}^{k\times q}$ be the matrix of the last q orthonormal columns. G_i is called a Hessian estimator.

Step 4 **Construct Approximate $\mathcal{H}$-functional.** $\Psi = \Sigma_{i=1}^{N} E_i G_i^T G_i E_i^T$, where $E_i = [e_{i_1}, \ldots, e_{i_k}] \in \mathbb{R}^{N\times k}$.

Step 5 **Find Basis of Null Space.** Compute the $d+1$ eigenvectors corresponding to the $d+1$ smallest eigenvalues of Ψ with e being the one corresponding to the eigenvalue 0. Let Z be the matrix consisting of the d eigenvectors corresponding to the 2nd to $(d+1)$-st eigenvalues, where $Z \in \mathbb{R}^{d\times N}$. Find an orthogonal matrix U such that the restriction of ZU to a fixed local neighborhood has orthonormal columns. Then the columns of $T = U^T Z^T$ are approximate global coordinates.

In light of the use of Hessian estimators as an ad hoc approximation of the Hessian operator, one of the theoretical difficulties with the above procedure is how well they approximate the continuous counterparts. In the next section, we formally define a discrete Hessian operator and analyze a discrete version of the Hessian Eigenmaps.

3. Discrete Hessian Eigenmaps Method

In this section, we present a discrete version of the Hessian Eigenmaps method. Specifically, we introduce the discrete Hessian operator and a generalization of the $\mathcal{H}$-functional and prove generalizations of Theorem 2.1. The discrete Hessian Eigenmaps method is essentially the same as the numerical procedure of Hessian LLE [1] described in Section 2, but is formulated as a direct generalization of the original Hessian Eigenmaps method. By establishing a discrete version of Theorem 2.1 with analysis, we directly provide a theoretical basis of the discrete procedure.

We are interested in reconstructing the coordinate set $\{\tau_1, \tau_2, \cdots, \tau_N\}$ for a given data set $\boldsymbol{M} = \{m_1, \ldots, m_N\}$. We partition $\boldsymbol{M}$ into subsets $\{\boldsymbol{M}_i, i = 1, \ldots, s\}$ with $\boldsymbol{M}_i = \{m_{i_1}, \ldots, m_{i_{k_i}}\}$ $(i_1 < i_2 < \cdots < i_{k_i})$ consisting of points in a small neighborhood so that a coordinate system on the local tangent space can be approximately obtained. For theoretical purpose, we assume that $\boldsymbol{M}_i$ is a subset such that a local isometric coordinate $\boldsymbol{\Theta}_i = \{\theta_1^{(i)}, \ldots, \theta_{k_i}^{(i)}\}$ can be constructed such that $\|\theta_p^{(i)} - \theta_q^{(i)}\|_2 = \|\tau_{i_p} - \tau_{i_q}\|_2$ (which is also equal to the geodesic distance between m_{i_p}, m_{i_q} along $\mathcal{M}$) for any $1 \leq p, q \leq k_i$. (In the context of Hessian Eigenmaps, the local tangent coordinate is an approximation of the local coordinate defined here). The essence of the Hessian Eigenmaps method is to reconstruct τ_i's from the local coordinates using Hessian operators.

3.1. Hessian Operator

Let $\boldsymbol{S}_0 = \{\tau_1, \ldots, \tau_k\} \subset \mathbb{R}^d$ be an ordered set and let $T_0 = [\tau_1, \ldots, \tau_k]$ be the corresponding matrix as in (3). For our purpose, $\boldsymbol{S}_0$ is a coordinate set and we first define its dimension.

Definition 3.1. The coordinate set $\boldsymbol{S}_0 = \{\tau_1, \ldots, \tau_k\} \subset \mathbb{R}^d$ is said to be of dimension p if

$$\mathsf{rank}[\tau_1 - \bar{\tau}, \tau_2 - \bar{\tau}, \ldots, \tau_k - \bar{\tau}] = p$$

where $\bar{\tau} = (\Sigma_{j=1}^k \tau_j)/k$. We write $\dim(\boldsymbol{S}_0) = p$.

Clearly, $\dim(\boldsymbol{S}_0) = p$ is the dimension of the space spanned by centered $\boldsymbol{S}_0$ or an affine plane through $\boldsymbol{S}_0$. Recall that e is a column vector of all ones. The following lemma is shown in [16].

Lemma 3.1. $\dim(\boldsymbol{S}_0) = p$ *if and only if* $\mathsf{rank}([e, T_0^T]) = 1 + p$.

Definition 3.2. Let $\boldsymbol{S}_0 = \{\tau_1, \ldots, \tau_k\} \subset \mathbb{R}^d$ be an ordered subset and let $T_0 = [\tau_1, \tau_2, \cdots, \tau_k]$ be the corresponding matrix. Let

$$Z_0 = Y_0 - [e, T_0^T][e, T_0^T]^\dagger Y_0, \tag{6}$$

where

$$Y_0 = \begin{pmatrix} \tau_1^T \bigodot \tau_1^T \\ \tau_2^T \bigodot \tau_2^T \\ \vdots \\ \tau_k^T \bigodot \tau_k^T \end{pmatrix} \in \mathbb{R}^{k \times q} \text{ with } q = \frac{d(d+1)}{2} \tag{7}$$

and the operation $\bigodot$ is defined in (2). We say

$$H_0 = Z_0^\dagger$$

is the discrete Hessian operator as defined by $\boldsymbol{S}_0$.

Remark 3.1. The discrete Hessian operator H_0 depends on the order in $\boldsymbol{S}_0$. If we permute the vectors in $\boldsymbol{S}_0$ to get another ordered set $\hat{\boldsymbol{S}}_0$ with the corresponding matrix $\hat{T}_0 = T_0 P$ for some permutation matrix P (see [22, p.38]), it is easy to check that $\hat{H}_0 = H_0 P$ where $\hat{H}_0$ is the discrete Hessian operator defined by $\hat{\boldsymbol{S}}_0$. Therefore, permuting the vectors in $\boldsymbol{S}_0$ results in permuting the columns of the discrete Hessian operator.

The justification for this definition of discrete Hessian operator will be given in Theorem 3.1 below. We note first that the columns of $[e, T_0^T]$ consist of the vectors $[f(\tau_i)]_{i=1}^k$ with $f : \mathbb{R}^d \to \mathbb{R}$ being the constant or the d basic linear functions. Y_0 consists of $[f(\tau_i)]_{i=1}^k$ with $f : \mathbb{R}^d \to \mathbb{R}$ being the basic quadratic functions. The next lemma demonstrates a simple property of the Hessian.

Lemma 3.2. Let $\boldsymbol{S}_0 = \{\tau_1, \ldots, \tau_k\} \subset \mathbb{R}^d$ be an ordered subset and let $T_0 = [\tau_1, \tau_2, \cdots, \tau_k]$. Let H_0 be the discrete Hessian operator for $\boldsymbol{S}_0$. We have $\mathsf{span}([e, T_0^T]) \subset \mathsf{null}(H_0)$.

Proof: From Definition 3.2, we have $[e, T_0^T]^T Z_0 = 0$ where $Z_0 = H_0^\dagger = Y_0 - [e, T_0^T][e, T_0^T]^\dagger Y_0$ and Y_0 is defined in (7). Then $Z_0^T[e, T_0^T] = 0$. Hence $Z_0^\dagger[e, T_0^T] = 0$. ∎

Now, consider a function $h(\tau) : \mathbb{R}^d \mapsto \mathbb{R}$ and we are interested in an approximation of the Hessian of h at some point τ_0 using the values of $h(\tau_i)$. Performing the Taylor expansion for $h(\tau)$ at τ_0, we have

$$h(\tau) = h(\tau_0) + (\tau - \tau_0)^T \nabla h(\tau_0) + \frac{1}{2}(\tau - \tau_0)^T H_h(\tau_0)(\tau - \tau_0) + R(\tau),$$

where $R(\tau)$ is the remainder term and $H_h(\tau_0)$ is the Hessian matrix of the function $h(\tau)$ at $\tau = \tau_0$. Let

$$\mathfrak{h}_h(\tau) = [a_1(\tau), \ldots, a_q(\tau)]^T \in \mathbb{R}^q, \tag{8}$$

where

$$a_{\frac{k(k-1)}{2}+\ell}(\tau) = \begin{cases} \frac{1}{2}\frac{\partial^2 h}{\partial t_k^2}(\tau) & \text{if } k = \ell; \\ \frac{\partial^2 h}{\partial t_k \partial t_\ell}(\tau) & \text{if } k > \ell; \end{cases}$$

and $q = d(d+1)/2$. Then $\mathfrak{h}_h(\tau)$ is a vector form of the Hessian matrix containing the entries of the lower triangular part (including diagonal) of the Hessian. Now, considering the Taylor expansion of $h(\tau_i)$ for $i = 1, 2, \cdots, k$ and combining them together, we can write

$$\begin{pmatrix} h(\tau_1) \\ h(\tau_2) \\ \vdots \\ h(\tau_k) \end{pmatrix} = \begin{pmatrix} 1 & (\tau_1 - \tau_0)^T & (\tau_1 - \tau_0)^T \bigodot (\tau_1 - \tau_0)^T \\ 1 & (\tau_2 - \tau_0)^T & (\tau_2 - \tau_0)^T \bigodot (\tau_2 - \tau_0)^T \\ \vdots & \vdots & \vdots \\ 1 & (\tau_k - \tau_0)^T & (\tau_k - \tau_0)^T \bigodot (\tau_k - \tau_0)^T \end{pmatrix} \begin{pmatrix} h(\tau_0) \\ \nabla h(\tau_0) \\ \mathfrak{h}_h(\tau_0) \end{pmatrix} + \begin{pmatrix} r_1 \\ r_2 \\ \vdots \\ r_k \end{pmatrix}, \tag{9}$$

where $r_i = R(\tau_i)$. Note that if $h \in C^3(\mathbb{R}^d)$, $|r_i| \leq C\|\tau_i - \tau_0\|^3$ for some constant $C > 0$.

Theorem 3.1. Let $h \in C^3(\mathbb{R}^d) : \mathbb{R}^d \mapsto R$ and H_0 be the discrete Hessian operator as defined by the ordered set $\boldsymbol{S}_0 = \{\tau_1, \ldots, \tau_k\} \subset \mathbb{R}^d$. Let $\mathfrak{h}_h(\tau_0)$ be the column form of the Hessian matrix of function $h(\tau)$ at τ_0 as defined in (8). If H_0 has full row rank, we have

$$H_0 \begin{pmatrix} h(\tau_1) \\ h(\tau_2) \\ \vdots \\ h(\tau_k) \end{pmatrix} = \mathfrak{h}_h(\tau_0) + H_0 \begin{pmatrix} r_1 \\ r_2 \\ \vdots \\ r_k \end{pmatrix}, \tag{10}$$

where r_i is such that $|r_i| \le C\|\tau_i - \tau_0\|^3$ for some constant $C > 0$.

Proof: Set

$$\mathbf{h} = \begin{pmatrix} h(\tau_1) \\ h(\tau_2) \\ \vdots \\ h(\tau_k) \end{pmatrix} \in \mathbb{R}^k, \quad \text{and} \quad \mathbf{r} = \begin{pmatrix} r_1 \\ r_2 \\ \vdots \\ r_k \end{pmatrix} \in \mathbb{R}^k.$$

By noting that each element of $(\tau - \tau_0)^T \bigodot (\tau - \tau_0)^T$ is a basic quadratic function plus a linear function in the entries of τ, we rewrite (9) as

$$\mathbf{h} = [e, T_0^T - e\tau_0^T] \begin{pmatrix} h(\tau_0) \\ \nabla h(\tau_0) \end{pmatrix} + \left(Y_0 + [e, T_0^T]R_0\right) \mathfrak{h}_h(\tau_0) + \mathbf{r},$$

where Y_0 is defined in (7) and $R_0 \in \mathbb{R}^{(d+1)\times q}$ is some matrix. Equivalently,

$$\begin{aligned}\mathbf{h} = {} & [e, T_0^T - e\tau_0^T] \begin{pmatrix} h(\tau_0) \\ \nabla h(\tau_0) \end{pmatrix} + \left(I - [e, T_0^T][e, T_0^T]^\dagger\right) (Y_0 + [e, T_0^T]R_0)\mathfrak{h}_h(\tau_0) \\ & + [e, T_0^T][e, T_0^T]^\dagger (Y_0 + [e, T_0^T]R_0)\mathfrak{h}_h(\tau_0) + \mathbf{r}.\end{aligned} \tag{11}$$

Multiplying (11) by H_0, we obtain

$$\begin{aligned}H_0\mathbf{h} = {} & H_0[e, T_0^T - e\tau_0^T] \begin{pmatrix} h(\tau_0) \\ \nabla h(\tau_0) \end{pmatrix} + H_0(I - [e, T_0^T][e, T_0^T]^\dagger)(Y_0 + [e, T_0^T]R_0)\mathfrak{h}_h(\tau_0) \\ & + H_0[e, T_0^T][e, T_0^T]^\dagger (Y_0 + [e, T_0^T]R_0)\mathfrak{h}_h(\tau_0) + H_0\mathbf{r}.\end{aligned}$$

Noticing $H_0[e, T_0^T] = 0$ by Lemma 3.2 and $(I - [e, T_0^T][e, T_0^T]^\dagger)Y_0 = H_0^\dagger$, we have

$$H_0\mathbf{h} = H_0 H_0^\dagger \mathfrak{h}_h(\tau_0) + H_0\mathbf{r}.$$

Since H_0 has full row rank, we have $H_0 H_0^\dagger = I$. (10) is proved. ∎

This theorem shows that applying H_0 to the function values at the k discrete points $[h(\tau_1), h(\tau_2), \cdots, h(\tau_k)]^T$ gives an approximation to the Hessian at τ_0. This justifies the definition of the discrete Hessian operator.

The discrete Hessian operator H_0 as defined is not easy to compute. However, what is really needed in the Hessian Eigenmaps method is $P = H_0^\dagger H_0$, which is the projection operator onto $\mathsf{span}(H_0^T)$ and has the same null space as H_0. This projection operator can be easily obtained by computing an orthonormal basis for $\mathsf{span}(Z_0)$, where Z_0 is defined in (6). Specifically, assuming $\boldsymbol{S}_0$ has dimension d, perform the Gram-Schmidt orthogonalization process on the columns of $Y^{(0)} = [e, T_0^T, Y_0]$, where Y_0 is given in (7). Noting that $[e, T_0^T]$ has full column rank, let Q_0 be matrix whose columns are obtained from the columns of Y_0 in the Gram-Schmidt process. The next theorem shows that we can compute $P = H_0^\dagger H_0$ from Q_0. Indeed, Q_0 is the Hessian estimator in Hessian LLE.

Theorem 3.2. Given an ordered subset $\boldsymbol{S}_0 \subset \mathbb{R}^d$ of dimension d, perform the Gram-Schmidt orthogonalization process on $[e, T_0^T, Y_0]$, where T_0 is the matrix corresponding to $\boldsymbol{S}_0$ and Y_0 is defined in (7), and let V_0 be the matrix of the first $d+1$ orthonormal vectors obtained and Q_0 be the matrix of the remaining orthonormal vectors. Then we have

$$\mathsf{span}(Q_0) = \mathsf{span}(H_0^T) \text{ and } H_0^\dagger H_0 = Q_0 Q_0^T,$$

where $H_0 \in \mathbb{R}^{q\times k}$ is the discrete Hessian operator for $\boldsymbol{S}_0$.

Proof: Since $\dim(\boldsymbol{S}_0) = d$, $[e, T_0^T]$ has full column rank and hence V_0 is obtained from the Gram-Schmidt process on $[e, T_0^T]$ and $\mathsf{span}(V_0) = \mathsf{span}([e, T_0^T])$. Then, the columns of Q_0 are obtained from the Gram-Schmidt process on $(I - V_0 V_0^T)Y_0$. As $(I - V_0 V_0^T)Y_0 = Z_0$, we have $\mathsf{span}(Q_0) = \mathsf{span}(Z_0)$, where $Z_0 = (I - [e, T_0^T][e, T_0^T]^\dagger)Y_0$ as defined in (6). It follows from $H_0 = Z_0^\dagger$ that $\mathsf{span}(Z_0) = \mathsf{span}(H_0^\dagger) = \mathsf{span}(H_0^T)$. Therefore $\mathsf{span}(Q_0) = \mathsf{span}(H_0^T)$. This immediately implies that $H_0^\dagger H_0 = Q_0 Q_0^T$. ∎

We now discuss two properties of the discrete Hessian operators which will be used in later sections. First, the discrete Hessian operator is defined from a given coordinate set but, as in the continuous case, its column space is invariant under a linear transformation on the coordinate set (i.e. change of basis), as is shown in the next lemma.

Lemma 3.3. Let G_0 and H_0 be the discrete Hessian operators for the ordered subsets $\boldsymbol{\Theta}_0 = \{\theta_1, \ldots, \theta_k\}$ and $\boldsymbol{S}_0 = \{\tau_1, \ldots, \tau_k\}$, respectively. Set $\Theta_0 = [\theta_1, \ldots, \theta_k]$ and $T_0 = [\tau_1, \ldots, \tau_k]$. If $\Theta_0 = V_1 T_0 + ce^T$ where $c \in \mathbb{R}^d$ and $V_1 \in \mathbb{R}^{d\times d}$ is a nonsingular matrix, we have $G_0 = V_2 H_0$ for some nonsingular matrix V_2.

Proof: Let

$$X^{(0)} = [e, \Theta_0^T, X_0] \quad \text{and} \quad Y^{(0)} = [e, T_0^T, Y_0], \tag{12}$$

where

$$X_0 = \begin{pmatrix} {\theta_1}^T \bigodot {\theta_1}^T \\ {\theta_2}^T \bigodot {\theta_2}^T \\ \vdots \\ {\theta_k}^T \bigodot {\theta_k}^T \end{pmatrix} \text{ and } Y_0 = \begin{pmatrix} {\tau_1}^T \bigodot {\tau_1}^T \\ {\tau_2}^T \bigodot {\tau_2}^T \\ \vdots \\ {\tau_k}^T \bigodot {\tau_k}^T \end{pmatrix}.$$

To simplify some calculations, we also let

$$\widetilde{X}_0 = \begin{pmatrix} {\theta_1}^T \bigotimes {\theta_1}^T \\ {\theta_2}^T \bigotimes {\theta_2}^T \\ \vdots \\ {\theta_k}^T \bigotimes {\theta_k}^T \end{pmatrix} \text{ and } \widetilde{Y}_0 = \begin{pmatrix} {\tau_1}^T \bigotimes {\tau_1}^T \\ {\tau_2}^T \bigotimes {\tau_2}^T \\ \vdots \\ {\tau_k}^T \bigotimes {\tau_k}^T \end{pmatrix}.$$

Noting that, for two vectors a and b, $a \bigodot b$ and $a \bigotimes b$ have the same set of elements with $a \bigotimes b$ containing some repeated entries. Then, $\widetilde{X}_0$ and X_0 contain the same set of column vectors with $\widetilde{X}_0$ containing some repeated ones. Therefore,

$$\mathsf{span}(X_0) = \mathsf{span}(\widetilde{X}_0) \quad \text{and} \quad \mathsf{span}(Y_0) = \mathsf{span}(\widetilde{Y}_0). \tag{13}$$

Using $\theta_i = V_1\tau_i + c$ for $i = 1, \ldots, k$ and the properties of the Kronecker product [22, Lemma 6.3], we have

$$\begin{aligned} &\theta_i^T \bigotimes \theta_i^T \\ =&(\tau_i^T V_1^T + c^T) \bigotimes (\tau_i^T V_1^T + c^T) \\ =&(\tau_i^T \bigotimes \tau_i^T)(V_1^T \bigotimes V_1^T) + c^T \bigotimes (\tau_i^T V_1^T) + (\tau_i^T V_1^T) \bigotimes c^T + c^T \bigotimes c^T \\ =&(\tau_i^T \bigotimes \tau_i^T)(V_1^T \bigotimes V_1^T) + (c^T \bigotimes \tau_i^T)(I \bigotimes V_1^T) + (\tau_i^T \bigotimes c^T)(V_1^T \bigotimes I) \\ &+c^T \bigotimes c^T, \end{aligned}$$

for $i = 1, \ldots, k$. It follows that

$$\widetilde{X}_0 = \widetilde{Y}_0(V_1^T \bigotimes V_1^T) + [e, T_0^T]R \tag{14}$$

for some matrix $R \in \mathbb{R}^{(d+1)\times d^2}$. Since V_1 is a nonsingular matrix, we have $V_1^T \bigotimes V_1^T$ is a nonsingular matrix by [22, Lemma 6.3]. It also follows from $\Theta_0 = V_1 T_0 + ce^T$ that

$$I - [e, \Theta_0^T][e, \Theta_0^T]^\dagger = I - [e, T_0^T][e, T_0^T]^\dagger. \tag{15}$$

Combining (14) and (15), we have

$$\mathsf{span}\left((I-[e,\Theta_0^T][e,\Theta_0^T]^\dagger)\widetilde{X}_0\right)=\mathsf{span}\left((I-[e,T_0^T][e,T_0^T]^\dagger)\widetilde{Y}_0\right). \tag{16}$$

$\mathsf{span}\left((I-[e,\Theta_0^T][e,\Theta_0^T]^\dagger)X_0\right)=\mathsf{span}\left((I-[e,T_0^T][e,T_0^T]^\dagger)Y_0\right)$ from (13) and (16). Then $\mathsf{span}(G_0^\dagger)=\mathsf{span}(H_0^\dagger)$. It follows that $\mathsf{span}(G_0^T)=\mathsf{span}(H_0^T)$ or $G_0=V_2H_0$ for some nonsingular matrix V_2. ∎

We finally discuss possibly zero discrete Hessian operator H_0. Note that if $\boldsymbol{S}_0$ contains only $d+1$ points and is of dimension d, then its discrete Hessian operator H_0 is necessarily 0 as $[e,T_0^T]$ is a square nonsingular matrix; see the lemma below. Therefore, we need to have at least $d+2$ points to define a nontrivial Hessian. The following lemma shows that this is also sufficient.

Lemma 3.4. Given an ordered subset $\boldsymbol{S}_0=\{\tau_1,\ldots,\tau_k\}$ of dimension d with k distinct points, let H_0 be the discrete Hessian operator for $\boldsymbol{S}_0$. We have that $H_0\neq 0$, i.e. $\mathsf{rank}(H_0)\geq 1$ if and only if $k\geq d+2$.

Proof: First, since $\boldsymbol{S}_0$ is of dimension d, we have $k\geq d+1$ by Lemma 3.1. If $\mathsf{rank}(H_0)\geq 1$ and $k<d+2$, we have $k=d+1$. Then $[e,T_0^T]$ is a square matrix with full column rank, where $T_0=[\tau_1,\ldots,\tau_k]$. Thus it follows from the definition (or Lemma 3.2) that its discrete Hessian operator H_0 is necessarily 0, which is a contradiction. Therefore $k\geq d+2$.

Now, assume $k\geq d+2$. Since $\mathsf{rank}([e,T_0^T])=d+1$ (Lemma 3.1), we can find a permutation matrix P, a nonsingular matrix R and a vector c such that

$$P[e,T_0^T]\begin{array}{cc} & \begin{array}{cc}1 & d\end{array}\\ \begin{array}{c}1\\ d\end{array} & \begin{pmatrix}1 & c^T\\ 0 & R\end{pmatrix}\end{array}=\begin{array}{cc} & \begin{array}{cc}1 & d\end{array}\\ \begin{array}{c}1\\ d\\ k-(d+1)\end{array} & \begin{pmatrix}1 & 0\\ e & I_d\\ e & A\end{pmatrix}\end{array}\equiv[e,\widetilde{T}_0^T], \tag{17}$$

i.e. we can reduce $[e,T_0^T]$ to the form (17) through a row permutation and a column Gaussian elimination. Let $\widetilde{\boldsymbol{S}}_0$ be the ordered subset consisting of all the column vectors of $\widetilde{T}_0$. We generate a matrix $\widetilde{Y}^{(0)}=[e,\widetilde{T}_0^T,\widetilde{Y}_0]\in\mathbb{R}^{k\times(1+d+q)}$ for $\widetilde{\boldsymbol{S}}_0$ as in (12), where $\widetilde{Y}_0\in\mathbb{R}^{k\times q}$ and $q=d(d+1)/2$. Let $\widetilde{H}_0$ be the discrete Hessian operator defined by $\widetilde{\boldsymbol{S}}_0$. We now prove $\widetilde{H}_0\neq 0$ by showing there is a column in $\widetilde{Y}_0$ that can not be expressed as a linear combination of the columns of $[e,\widetilde{T}_0^T]$.

Since T_0^T has distinct rows, so does $\widetilde{T}_0^T$. Consider the first row of $A=[a_{ij}]$ (i.e. the $(d+2)$-nd row of $\widetilde{T}_0^T$) and let it be $\mathbf{a}_1=[a_{11},\ldots,a_{1d}]$, which must be different from the top $d+1$ rows of matrix $\widetilde{T}_0^T$. Then $\mathbf{a}_1 \neq 0$ as otherwise the row is the same as the first row of $\widetilde{T}_0^T$. Without loss of generality, assume $a_{11} \neq 0$. We consider the following two cases.

Case 1 If $a_{11} = 1$, there is at least another element of $\{a_{1i}, 2 \leq i \leq d\}$ is nonzero since the row $\mathbf{a}_1$ is different from the second row of $\widetilde{T}_0^T$. Without loss of generality, we assume that $a_{12} \neq 0$. Then, the entrywise multiplication of the first two columns of $\widetilde{T}_0^T$, which forms one of the columns of $\widetilde{Y}_0$, is

$$
\begin{array}{c} {\scriptstyle (d+1)} \\ {\scriptstyle 1} \\ {\scriptstyle 1} \\ \vdots \\ {\scriptstyle 1} \end{array}
\begin{pmatrix} \mathbf{0} \\ a_{11}a_{12} \\ a_{21}a_{22} \\ \vdots \\ a_{(k-(d+1)),1}a_{(k-(d+1)),2} \end{pmatrix}
=
\begin{array}{c} {\scriptstyle (d+1)} \\ {\scriptstyle 1} \\ {\scriptstyle 1} \\ \vdots \\ {\scriptstyle 1} \end{array}
\begin{pmatrix} \mathbf{0} \\ a_{12} \\ a_{21}a_{22} \\ \vdots \\ a_{(k-(d+1)),1}a_{(k-(d+1)),2} \end{pmatrix}
$$

By inspecting the first $d+2$ entries, this column vector can not be represented as a linear combination of the columns of $[e, \widetilde{T}_0^T]$. Therefore $\widetilde{H}_0 \neq 0$.

Case 2 If $a_{11} \neq 1$, then the entrywise square of the first column of $\widetilde{T}_0^T$, which forms one of the columns of $\widetilde{Y}_0$, is

$$
\begin{bmatrix} 0, & e_1^T, & a_{11}^2, & a_{21}^2, & \cdots, & a_{(k-(d+1)),1}^2 \end{bmatrix}^T
$$

where $e_1 = (1, 0, \ldots, 0)^T \in \mathbb{R}^d$. Again, by inspecting the first $d+2$ entries, this column vector can not be represented as a linear combination of the columns of $[e, \widetilde{T}_0^T]$. Therefore $\widetilde{H}_0 \neq 0$.

Thus, we have proved that $\widetilde{H}_0 \neq 0$. Finally, from (17), we have $\widetilde{T}_0 = R^T T_0 P^T + ce^T$. Then $\mathsf{rank}(H_0^T) = \mathsf{rank}(\widetilde{H}_0^T)$ by Lemma 3.3 and Remark 3.1. It follows that $H_0 \neq 0$. ∎

3.2. Hessian Alignment Matrix

We now generalize the definition of the $\mathcal{H}$-functional in the continuous case to the discrete case and present a generalization of Theorem 2.1. In the discrete setting, the construction of the quadratic form or the corresponding symmetric matrix is more closely related to that of the alignment matrix in the LTSA method [5, 16]. Indeed, they have some similar spectral properties as well. Hence we call it the Hessian alignment matrix.

Let $\{\boldsymbol{S}_i, 1 \le i \le s\}$ be a collection of ordered subsets of a given ordered set $\boldsymbol{S} = \{\tau_1, \dots, \tau_N\}$. Write

$$\boldsymbol{S}_i = \{\tau_{i_1}, \dots, \tau_{i_{k_i}}\}, \quad i_1 < i_2 < \dots < i_{k_i},$$

and let

$$T = [\tau_1, \cdots, \tau_N] \in \mathbb{R}^{d\times N} \quad \text{and} \quad T_i = [\tau_{i_1}, \cdots, \tau_{i_{k_i}}]$$

be the corresponding matrices. We say T_i is a *section* of T. Let

$$E_i = [e_{i_1}, \dots, e_{i_{k_i}}] \in \mathbb{R}^{N\times k_i}, \tag{18}$$

where $e_i \in \mathbb{R}^N$ is the i-th column of I_N. Then we have $TE_i = T_i$. E_i is called the selection matrix for $\boldsymbol{S}_i$.

In the context of manifold learning, each $\boldsymbol{S}_i$ is a coordinate set for points in a small neighborhood, from which a discrete Hessian operator can be defined. Assembling them together, the following is a generalization of the definition of the $\mathcal{H}$-functional.

Definition 3.3. Given an ordered set $\boldsymbol{S} = \{\tau_1, \dots, \tau_N\}$ and a collection of ordered subsets $\{\boldsymbol{S}_i, 1 \le i \le s\}$, let

$$\Phi = \sum_{i=1}^{s} E_i H_i^{\dagger} H_i E_i^T,$$

where H_i $(1 \le i \le s)$ is the discrete Hessian operator for $\boldsymbol{S}_i$ $(1 \le i \le s)$ and E_i is the selection matrix for $\boldsymbol{S}_i$ (see (18)). We call Φ the Hessian alignment matrix for $\{\boldsymbol{S}_i, 1 \le i \le s\}$.

Remark 3.2. A direct generalization of the $\mathcal{H}$-functional (4) is actually $\hat{\Phi} = \sum_{i=1}^{s} E_i H_i^T H_i E_i^T$, with which all the results we present later on Φ will also be valid. However, we have adopted Definition 3.3 in the discrete case because $H_i^T H_i$ is not easily computable while $H_i^{\dagger} H_i$ is by Theorem 3.2.

Remark 3.3. We note that H_i depends on the order of the vectors in $\boldsymbol{S}_i$ but it is easy to check that $E_i H_i^{\dagger} H_i E_i^T$ is not. Hence Φ is independent of the order of the vectors in $\boldsymbol{S}_i$. Thus, the discrete Hessian operator is defined for a collection of subsets $\boldsymbol{S}_i$ $(1 \le i \le s)$ that are not necessarily ordered.

Lemma 3.5. Given an ordered set $\boldsymbol{S} = \{\tau_1, \ldots, \tau_N\}$ and a collection of subsets $\{\boldsymbol{S}_i, 1 \le i \le s\}$, let T and T_i be their corresponding matrices. Let E_i be the selection matrix for $\boldsymbol{S}_i$ (see (18)) such that $T_i = TE_i$. If H_i is the discrete Hessian operator for $\boldsymbol{S}_i$ and Φ is the Hessian alignment matrix for $\{\boldsymbol{S}_i, 1 \le i \le s\}$, then we have

$$\mathsf{null}(H_i E_i^T) \supset \{x | E_i^T x \in \mathsf{span}([e, T_i^T])\} \tag{19}$$

and

$$\mathsf{null}(\Phi) = \bigcap_{i=1}^{s} \mathsf{null}(H_i E_i^T).$$

Moreover, we have $\mathsf{span}([e, T^T]) \subset \mathsf{null}(\Phi)$.

Proof: First, it is easily checked that (19) follows from Lemma 3.2. The rest of the proof for this lemma is the same as the proof of Lemma 2.1 of [16]. We omit it here. ∎

The main result of this section is to determine under what conditions that we have $\mathsf{span}([e, T^T]) = \mathsf{null}(\Phi)$. For that, we need to introduce some definitions.

Definition 3.4. Let $\boldsymbol{S}_1$ and $\boldsymbol{S}_2$ be two ordered subsets of $\mathbb{R}^d$ with a non-trivial intersection. Let T_1 and T_2 be the corresponding matrices. Let H_2 be the discrete Hessian operator for $\boldsymbol{S}_2$ and let $H_{2\backslash 1}$ be the submatrix of H_2 consisting of the columns of H_2 that correspond[2] to the vectors in $\boldsymbol{S}_2 \setminus \boldsymbol{S}_1$ (i.e. $H_{2\backslash 1} = H_2 E_{2\backslash 1}$ where $E_{2\backslash 1}$ is a selection matrix (18) such that $T_2 E_{2\backslash 1}$ is the submatrix of T_2 corresponding to the set $\boldsymbol{S}_2 \setminus \boldsymbol{S}_1$). We say $\boldsymbol{S}_2$ is rigidly connected to $\boldsymbol{S}_1$ if $H_{2\backslash 1}$ has full column rank.

In the definition, an empty matrix $H_{2\backslash 1}$ is considered to have full column rank. Based on the above relation of two subsets, we can associate a directed graph with $\{\boldsymbol{S}_1, \boldsymbol{S}_2, \ldots, \boldsymbol{S}_s\}$ as follows.

Definition 3.5. Given a collection of ordered subsets $\{\boldsymbol{S}_1, \boldsymbol{S}_2, \ldots, \boldsymbol{S}_s\}$ with $\boldsymbol{S}_i \subset \mathbb{R}^d$, its associate graph is a directed graph $\mathcal{G}$ with s vertices representing the s subsets, where there is a directed edge from vertex i to vertex j if the subset $\boldsymbol{S}_i$ is rigidly connected to the subset $\boldsymbol{S}_j$.

[2]Note that each column of the discrete Hessian H_2 corresponds to a column of T_2 (or a vector in $\boldsymbol{S}_2$).

Definition 3.6. Given an ordered set $\boldsymbol{S} = \{\tau_1, \ldots, \tau_N\} \subset \mathbb{R}^d$ of dimension d and a collection of subsets $\{\boldsymbol{S}_i, 1 \leq i \leq \ell\}$ with $\boldsymbol{S} = \bigcup_{i=1}^{\ell} \boldsymbol{S}_i$, let Φ be the Hessian alignment matrix for $\{\boldsymbol{S}_1, \ldots, \boldsymbol{S}_\ell\}$. We say the collection $\{\boldsymbol{S}_1, \ldots, \boldsymbol{S}_\ell\}$ is a full spanning collection, if $\mathsf{rank}(\Phi) = N - (d+1)$.

Assuming that $\boldsymbol{S}$ is of dimension d, it follows from Lemma 3.5 that $\{\boldsymbol{S}_1, \ldots, \boldsymbol{S}_\ell\}$ is a full spanning collection if and only if $\mathsf{null}(\Phi) = \mathsf{span}([e, T^T])$.

Lemma 3.6. Let $\boldsymbol{S} = \{\tau_1, \ldots, \tau_N\} \subset \mathbb{R}^d$ be an ordered set of dimension d and let $\{\boldsymbol{S}_i, 1 \leq i \leq \ell\}$ be a collection of subsets with $\boldsymbol{S} = \bigcup_{i=1}^{\ell} \boldsymbol{S}_i$. If $\{\boldsymbol{S}_i, i = 1, \ldots, \ell-1\}$ is a full spanning collection and $\boldsymbol{S}_\ell$ is rigidly connected to some subset $\boldsymbol{S}_j$ (with $1 \leq j \leq \ell-1$), then $\{\boldsymbol{S}_1, \boldsymbol{S}_2, \ldots, \boldsymbol{S}_\ell\}$ is a full spanning collection .

Proof: Let $\boldsymbol{S}_i = \{\tau_{i_1}, \ldots \tau_{i_{k_i}}\}$ and write $T_i = [\tau_{i_1}, \ldots \tau_{i_{k_i}}]$ and $T = [\tau_1, \ldots \tau_N]$. Let $H_i \in \mathbb{R}^{q \times k_i}$ $(1 \leq i \leq \ell)$ be the discrete Hessian operator for $\boldsymbol{S}_i$. Let E_i be the selection matrix defined in (18) such that $T_i = TE_i$ and let $\widetilde{H}_i = H_i E_i^T$ be the embedding of H_i into $\mathbb{R}^{q \times N}$. Set

$$H = \begin{pmatrix} \widetilde{H}_1 \\ \vdots \\ \widetilde{H}_\ell \end{pmatrix}.$$

We have $\mathsf{null}(H) = \bigcap_{i=1}^{\ell} \mathsf{null}(\widetilde{H}_i)$. It follows that $\mathsf{null}(H) = \mathsf{null}(\Phi)$ by Lemma 3.5.

Let $0 \leq k < N$ be such that there are $N-k$ vectors in $\hat{\boldsymbol{S}}_1 = \bigcup_{i=1}^{\ell-1} \boldsymbol{S}_i$ and there are k vectors in $\hat{\boldsymbol{S}}_2 = \bigcup_{i=1}^{\ell} \boldsymbol{S}_i \setminus \hat{\boldsymbol{S}}_1$. Without loss of generality, we assume that $\hat{\boldsymbol{S}}_1 = \{\tau_1, \tau_2, \ldots, \tau_{N-k}\}$ and $\hat{\boldsymbol{S}}_2 = \{\tau_{N-k+1}, \ldots, \tau_N\}$. Set $\hat{T}_1 = [\tau_1, \tau_2, \ldots, \tau_{N-k}]$ and $\hat{T}_2 = [\tau_{N-k+1}, \ldots, \tau_N]$. Embedding H_i $(1 \leq i \leq \ell - 1)$ into $\mathbb{R}^{q \times (N-k)}$ according to the embedding of T_i into $\hat{T}_1$, we have $\hat{H}_i = H_i \hat{E}_i^T$, where $\hat{E}_i$ is the selection matrix such that $T_i = \hat{T}_1 \hat{E}_i$. Then $\hat{\Phi}_1 = \sum_{i=1}^{\ell-1} \hat{E}_i P_i \hat{E}_i^T$ is the Hessian alignment matrix for the collection $\{\boldsymbol{S}_i, 1 \leq i \leq \ell - 1\}$, where $P_i = H_i^\dagger H_i$. Since $\{\boldsymbol{S}_i, 1 \leq i \leq \ell - 1\}$ is a full spanning collection, we have $\mathsf{null}(\hat{\Phi}_1) = \mathsf{span}([e, \hat{T}_1^T])$. Let

$$\bar{H}_1 = \begin{pmatrix} \hat{H}_1 \\ \vdots \\ \hat{H}_{\ell-1} \end{pmatrix}.$$

We have

$$\mathsf{null}(\bar{H}_1) = \bigcap_{i=1}^{\ell-1} \mathsf{null}(\hat{H}_i) = \mathsf{null}(\hat{\Phi}_1). \tag{20}$$

We first consider the case $k = 0$, i.e. $\hat{\boldsymbol{S}}_1 = \{\tau_1, \ldots, \tau_N\}$. In this case, we have $\hat{E}_i = E_i$ and $\hat{H}_i = \tilde{H}_i$. Then $\mathsf{null}(\Phi) = \bigcap_{i=1}^{\ell} \mathsf{null}(\tilde{H}_i) \subset \bigcap_{i=1}^{\ell-1} \mathsf{null}(\tilde{H}_i) = \mathsf{null}(\hat{\Phi}_1) = \mathsf{span}([e, T^T])$. Noting also $\mathsf{null}(\Phi) \supset \mathsf{span}([e, T^T])$ by Lemma 3.5, we have $\mathsf{null}(\Phi) = \mathsf{span}([e, T^T])$ and hence $\{\boldsymbol{S}_1, \boldsymbol{S}_2, \ldots, \boldsymbol{S}_\ell\}$ is a full spanning collection.

Next, we consider the case $k > 0$. Since $\{\boldsymbol{S}_i, i = 1, \ldots, \ell - 1\}$ is a full spanning collection, we have $\mathsf{rank}(\hat{\Phi}_1) = (N - k) - (d + 1)$. It follows from (20) that

$$\mathsf{rank}(\bar{H}_1) = (N - k) - (d + 1).$$

Let

$$\widetilde{H}_\ell = \begin{array}{c} \begin{array}{cc} {\scriptstyle N-k} & {\scriptstyle k} \end{array} \\ \left(\widetilde{H}_{\ell,1}, \quad \widetilde{H}_{\ell,2} \right) \end{array}.$$

Then we can rewrite H as follows.

$$H = \begin{array}{c} \begin{array}{cc} {\scriptstyle N-k} & \quad {\scriptstyle k} \end{array} \\ \begin{pmatrix} \bar{H}_1 & 0 \\ \widetilde{H}_{\ell,1} & \widetilde{H}_{\ell,2} \end{pmatrix} \end{array}.$$

Since $\boldsymbol{S}_\ell$ is rigidly connected to some subset $\boldsymbol{S}_j$ (with $1 \le j \le \ell - 1$), the submatrix of $\widetilde{H}_\ell$ consisting of the columns corresponding to the vectors in $\boldsymbol{S}_\ell \setminus \boldsymbol{S}_j$, of which $\widetilde{H}_{\ell,2}$ is a submatrix, has full column rank. Therefore $\widetilde{H}_{\ell,2}$ has full column rank. It follows that $\mathsf{rank}(H) \ge N - (d + 1)$. Equivalently, $\dim \mathsf{null}(H) \le d + 1$. Noticing $\mathsf{span}([e, T^T]) \subset \mathsf{null}(H)$ by Lemma 3.5, we have $\mathsf{null}(H) = \mathsf{null}(\Phi) = \mathsf{span}([e, T^T])$. Hence, the collection $\{\boldsymbol{S}_i, 1 \le i \le \ell\}$ is also a full spanning collection. ∎

We now proceed to prove our main theorem, which generalizes Theorem 2.1 of Donoho and Grimes.

Theorem 3.3. Let $\boldsymbol{S} = \{\tau_1, \ldots, \tau_N\} \subset \mathbb{R}^d$ be an ordered set of dimension d and let $\{\boldsymbol{S}_i, 1 \le i \le \ell\}$ be a collection of subsets with $\boldsymbol{S} = \bigcup_{i=1}^{\ell} \boldsymbol{S}_i$. Let Φ be the Hessian alignment matrix for $\{\boldsymbol{S}_1, \boldsymbol{S}_2, \ldots, \boldsymbol{S}_\ell\}$. Assume that there are two nonempty collections, say, $\{\boldsymbol{S}_i, i = 1, \ldots, p\}$ and $\{\boldsymbol{S}_i, i = (p+1), \ldots, \ell\}$, such that $\{\boldsymbol{S}_i, i = 1, \ldots, p\}$ is a full spanning collection and for each $\boldsymbol{S}_j$ (with $(p + 1) \le j \le \ell$) there is a path from $\boldsymbol{S}_j$ to some $\boldsymbol{S}_i$ (with $1 \le i \le p$) in the associated graph for $\{\boldsymbol{S}_1, \boldsymbol{S}_2, \ldots, \boldsymbol{S}_\ell\}$. Then we have $\mathsf{null}(\Phi) = \mathsf{span}([e, T^T])$, where $T = [\tau_1, \ldots, \tau_N]$.

Proof: By Lemma 3.6, we can expand $\{\boldsymbol{S}_i, i = 1, \ldots, p\}$ to obtain a new full spanning collection by including every subset that is rigidly connected to some $\boldsymbol{S}_j$ (with $1 \le j \le p$). Repeatedly expanding this full spanning collection, the maximum expansion will include all sets that are connected to some $\boldsymbol{S}_i$ (with $1 \le i \le p$) through a path in the associated graph and hence all $\boldsymbol{S}_j$ (with $(p+1) \le j \le \ell$). Therefore, $\{\boldsymbol{S}_i, i = 1, \ldots, \ell\}$ is a full spanning collection and thus $\mathsf{null}(\Phi) = \mathsf{span}([e, T^T])$. ∎

The following corollary follows immediately from the theorem.

Corollary 3.1. Let $\boldsymbol{S} = \{\tau_1, \ldots, \tau_N\} \subset \mathbb{R}^d$ be an ordered set of dimension d and let $\{\boldsymbol{S}_i, 1 \le i \le \ell\}$ be a collection of subsets with $\boldsymbol{S} = \bigcup_{i=1}^{\ell} \boldsymbol{S}_i$. Let Φ be the Hessian alignment matrix for $\{\boldsymbol{S}_1, \boldsymbol{S}_2, \ldots, \boldsymbol{S}_\ell\}$. Assume that the associate graph for $\{\boldsymbol{S}_i, 1 \le i \le \ell\}$ is connected (i.e. there is a path from any node to any other node) and a subset of $\{\boldsymbol{S}_i, 1 \le i \le \ell\}$ is a full spanning collection. Then we have $\mathsf{null}(\Phi) = \mathsf{span}([e, T^T])$, where $T = [\tau_1, \ldots, \tau_N]$.

The last two results identify conditions under which T can be recovered from the Hessian alignment matrix. These are not necessary conditions but if they are not satisfied, we can easily find examples such that the collection is not full spanning collection; see the examples in Section 4.

The Hessian alignment matrix Φ discussed in this section is constructed from subsets $\boldsymbol{S}_i$ of the original coordinates τ_i and this is a generalization of the $\mathcal{H}$-functional in the isometric coordinates $\mathcal{H}^{iso}$. In the next section, we consider a more general definition of the Hessian alignment matrix that generalizes the $\mathcal{H}$-functional in the tangent coordinates $\mathcal{H}(\cdot)$.

3.3. Hessian LLE

We now discuss the problem of how to reconstruct the global coordinates τ_i's for a given data set from their local coordinates as outlined at the beginning of the section using the Hessian alignment matrix. We first state the procedure as the following algorithm.

Algorithm 3.1. Hessian LLE Method

Given $\boldsymbol{M} = \{m_1, \cdots, m_N\} \subset \mathbb{R}^n$.

1. Construct $\{\boldsymbol{M}_i, i = 1, \ldots, s\}$ with $\boldsymbol{M}_i = \{m_{i_1}, \ldots, m_{i_{k_i}}\}$ consisting of points in a small neighborhood and $\bigcup_{i=1}^{s} \boldsymbol{M}_i = \boldsymbol{M}$.
2. For each $\boldsymbol{M}_i$, construct its local coordinates $\boldsymbol{\Theta}_i = \{\theta_1^{(i)}, \ldots, \theta_{k_i}^{(i)}\} \in \mathbb{R}^d$. Approximate local coordinates can be computed by projections onto an approximate local tangent space as in (5)
3. Construct $P_i = G_i^{\dagger} G_i$ where G_i $(1 \leq i \leq s)$ is the discrete Hessian operator for $\boldsymbol{\Theta}_i$ $(1 \leq i \leq s)$; see Theorem 3.2 for practical computations of P_i.
4. Construct
$$\Psi = \sum_{i=1}^{s} E_i P_i E_i^T.$$
We call Ψ the Hessian alignment matrix for the collection $\{\boldsymbol{\Theta}_i\}$.
5. Compute $[e/\sqrt{N}, Z^T]$ as an orthonormal basis of the spectral subspace of Ψ corresponding to the smallest $d+1$ eigenvalues, where $Z^T \in \mathbb{R}^{N \times d}$. The columns of Z are used as the coordinate set for $\boldsymbol{M}$.

The above algorithm is theoretically equivalent to the Hessian LLE introduced in [1] and described in Section 2 with the only difference being step 3 where P_i is constructed from the Hessian operator G_i while, in [1], it is constructed from the Hessian estimator Q_0. By Theorem 3.2, the Hessian operator G_i and the Hessian estimator Q_0 result in the same projection matrix P_i. We have introduced the Hessian operator G_i for the purpose of theoretical analysis.

In the algorithm, the Hessian alignment matrix Ψ now is defined through local coordinates $\boldsymbol{\Theta}_i = \{\theta_1^{(i)}, \ldots, \theta_{k_i}^{(i)}\} \subset \mathbb{R}^d$ without the knowledge of the original coordinates τ_i's. This generalizes the $\mathcal{H}$-functional in the tangent coordinate in the continuous case. Noting that in the continuous case, the Hessian is also defined through approximate tangent coordinates in a small neighborhood but the error caused by the projection on the tangent plane disappears in the limit. The discrete Hessian operator, on the other hand, is defined from some fixed points, for which any error in the computed local coordinates $\{\theta_1^{(i)}, \ldots, \theta_{k_i}^{(i)}\}$ will carry over to the definition of the Hessian. For this reason, we need to make a theoretical assumption in our analysis that the local coordinates $\{\theta_1^{(i)}, \ldots, \theta_{k_i}^{(i)}\}$ are computed correctly, i.e. they are isometric to $\{\tau_{i_1}, \ldots, \tau_{i_{k_i}}\}$ in the Euclidean distance. Under this assumption, the following theorem generalizes $\mathcal{H}(f) = \mathcal{H}^{iso}(f)$ of Theorem 2.1 of [1].

Theorem 3.4. Given $\boldsymbol{M} = \{m_1, \cdots, m_N\} \subset \mathbb{R}^n$ with $m_i = \psi(\tau_i)$ for some $\tau_i \in \mathbb{R}^d$, let $\boldsymbol{M}_i$, $\boldsymbol{\Theta}_i$ $(1 \leq i \leq s)$ and Ψ be obtained from Hessian LLE

(Algorithm 3.1). For $1 \le i \le s$, let $\boldsymbol{S}_i = \{\tau_{i_1}, \ldots, \tau_{i_{k_i}}\}$ and let Φ be the Hessian alignment matrix for the collection $\{\boldsymbol{S}_i, i = 1, \ldots, s\}$. Assume that $\{\theta_1^{(i)}, \ldots, \theta_{k_i}^{(i)}\}$ is isometric to $\{\tau_{i_1}, \ldots, \tau_{i_{k_i}}\}$ in the Euclidean distance, i.e. $\|\theta_p^{(i)} - \theta_q^{(i)}\|_2 = \|\tau_{i_p} - \tau_{i_q}\|_2$ for any $1 \le p, q \le k_i$. Then we have $\Psi = \Phi$.

Proof: Since $\{\theta_1^{(i)}, \ldots, \theta_{k_i}^{(i)}\}$ is isometric to $\{\tau_{i_1}, \ldots, \tau_{i_{k_i}}\}$ in the Euclidean distance, it follows from the proof of Theorem 2.4 of [16] that there exist an orthogonal matrix $\hat{V}_i$ and a vector $c \in \mathbb{R}^d$ such that

$$\Theta_i = \hat{V}_i T_i + c e^T,$$

where $T_i = [\tau_{i_1}, \ldots, \tau_{i_{k_i}}]$. By Lemma 3.3, we have $\mathsf{span}(G_i^T) = \mathsf{span}(H_i^T)$, where H_i is the discrete Hessian operator for $\boldsymbol{S}_i$. Since $G_i^\dagger G_i$ and $H_i^\dagger H_i$ are the orthogonal projections onto $\mathsf{span}(G_i^T)$ and $\mathsf{span}(H_i^T)$ respectively, we have $G_i^\dagger G_i = H_i^\dagger H_i$. Thus

$$\Psi = \sum_{i=1}^{s} E_i G_i^\dagger G_i E_i^T = \sum_{i=1}^{s} E_i H_i^\dagger H_i E_i^T = \Phi.$$

∎

With Theorem 3.4, if the local neighborhoods satisfy the assumptions of Theorem 3.3, we have $\mathsf{null}(\Psi) = \mathsf{span}\left([e, T^T]\right)$. In this case, we also say that $\{\boldsymbol{\Theta}_i, i = 1, \ldots, s\}$ is a full spanning collection.

4. Construction of Neighborhoods

Our analysis in Section 3 shows that the collection of local neighborhoods needs to satisfy certain conditions in order for the Hessian alignment matrix to recover the original global coordinates. Indeed, for Hessian LLE, some straightforward constructions of local neighborhoods do not result in a full spanning collection; see the examples below. In this section, we discuss how to construct local neighborhoods using the theoretical results of the previous section.

A typical construction of local neighborhoods for dimensionality reduction methods is by taking k-nearest points (or ϵ-neighborhood) of each point for some fixed k (or $\epsilon > 0$ resp.). This works with most methods under some mild conditions such as fully overlap in LTSA [16], which can be satisfied by increasing the size of neighborhoods, i.e. increasing k (or ϵ). For the Hessian

Eigenmaps method, however, a simple construction of k-nearest point neighborhood may lead to a collection that is not full spanning. We first present two examples to illustrate this.

Example 4.1. Consider $\boldsymbol{S} = \{t_1, \ldots, t_6\}$, where $t_1 = 1, t_2 = 2, t_3 = 3, t_4 = 4, t_5 = 5, t_6 = 6$. Constructing 4-nearest point neighborhood for each point results in 3 different subsets $\boldsymbol{S}_1 = \{t_1, t_2, t_3, t_4\}$, $\boldsymbol{S}_2 = \{t_2, t_3, t_4, t_5\}$ and $\boldsymbol{S}_3 = \{t_3, t_4, t_5, t_6\}$. With only three neighborhoods, we have $\mathsf{rank}(\Phi) = 3$, where Φ is the Hessian alignment matrix for $\{\boldsymbol{S}_1, \boldsymbol{S}_2, \boldsymbol{S}_3\}$. Then $\dim \mathsf{null}(\Phi) = 3$ and $\mathsf{null}(\Phi) \neq \mathsf{span}([e, T^T])$, where $T = [t_1, \cdots, t_6]$.

Inspecting the neighborhoods, we see that the graph associated with $\{\boldsymbol{S}_1, \boldsymbol{S}_2, \boldsymbol{S}_3\}$ is connected, but we do not have a subset or a collection of subsets that is full spanning collection to begin with.

The situation in this example is typical of k-nearest point neighborhoods in 1-dimensional problems when $k \geq 4$. With $k \geq 4$, any single $\boldsymbol{S}_i$ is not full spanning. Then, even if the collection is graph connected, we may not have a full spanning collection. This can not be remedied by increasing k for example. It is perhaps for this reason that a peculiar construction of local neighborhoods was proposed in the implementation of Hessian LLE [21]. There, a local neighborhood is constructed for each point by finding k nearest points but excluding the point itself. In this way, possible repetition of same local neighborhoods is avoided. While it is not clear whether this construction leads to a full spanning collection, it works in some cases.

Another situation leading to not full spanning collection of local neighborhoods is when it is not graph connected. The following examples illustrate this.

Example 4.2. Consider $\boldsymbol{S} = \{t_1, \ldots, t_6\}$, where $t_1 = 1, t_2 = 2, t_3 = 3, t_4 = 6, t_5 = 7, t_6 = 8$. Constructing local neighborhoods for each point by taking 3-nearest points, we obtain two different subsets $\boldsymbol{S}_1 = \{t_1, t_2, t_3\}$ and $\boldsymbol{S}_2 = \{t_4, t_5, t_6\}$. In this case, each $\boldsymbol{S}_i$ is a full spanning collection, but it can be easily checked that $\mathsf{rank}(\Phi) = 2$ and then $\mathsf{null}(\Phi) \neq \mathsf{span}([e, T^T])$. The reason is clear because neither $\boldsymbol{S}_1$ is rigidly connected to $\boldsymbol{S}_2$ nor $\boldsymbol{S}_2$ is rigidly connected to $\boldsymbol{S}_1$.

Example 4.3. Consider $\boldsymbol{S} = \{t_1, \ldots, t_8\}$, where $t_1 = 0, t_2 = 1, t_3 = 10, t_4 = 15, t_5 = 16, t_6 = 17, t_7 = 18, t_8 = 19$. Constructing local neighborhoods for each point by taking 4-nearest points but excluding the point itself, we obtain eight different subsets $\boldsymbol{S}_1 = \{t_2, t_3, t_4, t_5\}$, $\boldsymbol{S}_2 = \{t_1, t_3, t_4, t_5\}$, $\boldsymbol{S}_3 = \{t_4, t_5, t_6, t_7\}$, $\boldsymbol{S}_4 = \{t_5, t_6, t_7, t_8\}$, $\boldsymbol{S}_5 = \{t_4, t_6, t_7, t_8\}$, $\boldsymbol{S}_6 = \{t_4, t_5, t_7, t_8\}$, $\boldsymbol{S}_7 = \{t_4, t_5, t_6, t_8\}$ and $\boldsymbol{S}_8 = \{t_4, t_5, t_6, t_7\}$. It can be checked that $\mathsf{rank}(\Phi) = 5$. Then $\dim \mathsf{null}(\Phi) = 3$ and $\mathsf{null}(\Phi) \neq \mathsf{span}([e, T^T])$. We notice that the graph associated with $\{\boldsymbol{S}_1, \ldots, \boldsymbol{S}_8\}$ has no path between $\{\boldsymbol{S}_1, \boldsymbol{S}_2\}$ and the remaining subsets.

We note that the rigid connectivity problem encountered in the above examples can not be remedied by increasing k. Generally, for the 1-dimensional problem ($d = 1$), simple construction methods of local neighborhoods that we have discussed can easily cause the Hessian Eigenmaps method to fail; see numerical examples in the next section. We now discuss a scheme to address this problem by adding some nested neighborhoods (subsets) to the collection so that

1. the collection of nested neighborhoods added is a full spanning collection;
2. the entire collection is graph connected.

For the ease of notation, we consider the method of constructing local k-nearest point neighborhoods, although our scheme can be applied to other constructions such as the one proposed in [21].

First, we show that any local neighborhood of $d+2$ points is full spanning.

Proposition 4.1. Let $\boldsymbol{S} \subset \mathbb{R}^d$ be a subset of dimension d and consist of $d + 2$ distinct points. Then $\boldsymbol{S}$ is full spanning.

Proof: Let H be the discrete Hessian operator for the subset $\boldsymbol{S}$. We have $\mathsf{rank}(H) \geq 1$ by Lemma 3.4. Then $\dim \mathsf{null}(H) \leq d + 1$. However, since $\mathsf{span}([e, T^T]) \subset \mathsf{null}(H)$ from Lemma 3.2 where T is the matrix corresponding to $\boldsymbol{S}$, we have $\dim \mathsf{null}(H) \geq \dim \mathsf{span}([e, T^T]) = d + 1$. Thus $\dim \mathsf{null}(H) = d + 1$ and $\mathsf{null}(H) = \mathsf{span}([e, T^T])$, which means $\{\boldsymbol{S}\}$ is full spanning. ∎

Given a coordinate set $\boldsymbol{S}_j$ of dimension d consisting of $k > d + 2$ points, if it is not full spanning, we can extend it into a full spanning collection by constructing a sequence of nested subsets $\{\boldsymbol{S}_j^{(1)}, \boldsymbol{S}_j^{(2)}, \ldots, \boldsymbol{S}_j^{(k-(d+1))}\}$, i.e.

$$\boldsymbol{S}_j^{(k-(d+1))} \subset \boldsymbol{S}_j^{(k-(d+2))} \subset \cdots \subset \boldsymbol{S}_j^{(1)} = \boldsymbol{S}_j, \tag{21}$$

with $\boldsymbol{S}_j^{(k-(d+1))}$ consisting of exactly $d+2$ points. The sequence is constructed by removing one vector at a time starting from $\boldsymbol{S}_j^{(1)} = \boldsymbol{S}_j$ as follows. Let

$\boldsymbol{S}_j^{(1)} = \{\tau_1^{(j)}, \ldots, \tau_k^{(j)}\} \subset \mathbb{R}^d$ and let $G_j^{(1)}$ be the discrete Hessian operator for $\boldsymbol{S}_j^{(1)}$. First, note that $G_j^{(1)}$ has at least one nonzero column vector by Lemma 3.4. Assume that the ℓ-th column of $G_j^{(1)}$ is nonzero, i.e. $G_j^{(1)} e_\ell \neq 0$. Let $\boldsymbol{S}_j^{(2)}$ be the subset obtained from $\boldsymbol{S}_j^{(1)}$ by removing $\tau_\ell^{(j)}$. Then $\boldsymbol{S}_j^{(1)}$ is rigidly connected to $\boldsymbol{S}_j^{(2)}$ and vice versa. In the same way, we can construct $\boldsymbol{S}_j^{(3)}$ from $\boldsymbol{S}_j^{(2)}$ and so on until $\boldsymbol{S}_j^{(k-(d+1))}$ is constructed. The selection of the vectors for deletion ensures that $\boldsymbol{S}_j^{(i)}$ is rigidly connected to $\boldsymbol{S}_j^{(i+1)}$. Since $\boldsymbol{S}_j^{(k-(d+1))}$ contains $d+2$ points, assuming that it is of dimension d, it is full spanning. Hence, the collection of subsets so constructed is a full spanning collection by Theorem 3.3.

We now discuss the second issue, i.e. how to add some neighborhoods to yield a graph connected collection. For any neighborhood (subset) $\boldsymbol{S}_i$, let Ω_i be the set of neighborhoods that are rigidly connected to $\boldsymbol{S}_i$ and vice versa, and Γ_i be the set of neighborhoods $\boldsymbol{S}_j$ that are fully overlapped with $\boldsymbol{S}_i$ (i.e. $\dim(\boldsymbol{S}_j \cap \boldsymbol{S}_i) = d$) but are not rigidly connected to $\boldsymbol{S}_i$ or vice versa. For a subset $\boldsymbol{S}_j \in \Gamma_i$, we can construct a collection of nested subsets of $\boldsymbol{S}_j$ with $\boldsymbol{S}_j \bigcap \boldsymbol{S}_i$ being the smallest one as in (21); see (22) below for details. We also do the same for $\boldsymbol{S}_i$. Then, $\boldsymbol{S}_j$ will be rigidly connected to $\boldsymbol{S}_j \bigcap \boldsymbol{S}_i$ and hence to $\boldsymbol{S}_i$. If we do this for each $\boldsymbol{S}_j \in \Gamma_i$, then $\boldsymbol{S}_i$ will be rigidly connected to all neighborhoods that it is fully overlapped with. However, this may be expensive and is not necessary.

In practice, for a subset $\boldsymbol{S}_j \in \Gamma_i$, its rigid connectivity with $\boldsymbol{S}_i$ may or may not be critical. Heuristically, if $\boldsymbol{S}_j$ is rigidly connected to a neighborhood already in Ω_i, then its rigid connectivity to $\boldsymbol{S}_i$ is not needed. We therefore propose the following simplified procedure that forces rigid connectivity between $\boldsymbol{S}_i$ and $\boldsymbol{S}_j$ only if $\boldsymbol{S}_j$ is not rigidly connected to any subset that is already rigidly connected to $\boldsymbol{S}_i$ and vice versa.

Given a subset $\boldsymbol{S}_j \in \Gamma_i$, if $\boldsymbol{S}_j \notin \left(\bigcup_{\boldsymbol{S}_\ell \in \Omega_i} \Omega_\ell\right)$, we construct from $\boldsymbol{S}_j$ a collection of nested subsets $\{\hat{\boldsymbol{S}}_j^{(1)}, \hat{\boldsymbol{S}}_j^{(2)}, \ldots, \hat{\boldsymbol{S}}_j^{(k-n_{ij}+1)}\}$, as in (21) with

$$(\boldsymbol{S}_j \bigcap \boldsymbol{S}_i) = \hat{\boldsymbol{S}}_j^{(k-n_{ij}+1)} \subset \hat{\boldsymbol{S}}_j^{(k-n_{ij})} \subset \hat{\boldsymbol{S}}_j^{(k-n_{ij}-1)} \subset \cdots \subset \hat{\boldsymbol{S}}_j^{(1)} = \boldsymbol{S}_j, \quad (22)$$

where n_{ij} is the number of vectors in $\boldsymbol{S}_j \bigcap \boldsymbol{S}_i$. Assume that the construction succeeds. Then $\hat{\boldsymbol{S}}_j^{(\ell)}$ is rigidly connected to $\hat{\boldsymbol{S}}_j^{(\ell+1)}$ for $1 \leq \ell \leq (k - n_{ij})$. In the same way, we also construct from $\boldsymbol{S}_i$ another collection of nested subsets

$\{\hat{\boldsymbol{S}}_i^{(1)}, \hat{\boldsymbol{S}}_i^{(2)}, \ldots, \hat{\boldsymbol{S}}_i^{(k-n_{ij}+1)}\}$, with

$$(\boldsymbol{S}_j \bigcap \boldsymbol{S}_i) = \hat{\boldsymbol{S}}_i^{(k-n_{ij}+1)} \subset \hat{\boldsymbol{S}}_i^{(k-n_{ij})} \subset \hat{\boldsymbol{S}}_i^{(k-n_{ij}-1)} \subset \cdots \subset \hat{\boldsymbol{S}}_i^{(1)} = \boldsymbol{S}_i. \quad (23)$$

Then we set

$$\Omega_i = \Omega_i \bigcup \{\boldsymbol{S}_j\} \text{ and } \Omega_j = \Omega_j \bigcup \{\boldsymbol{S}_i\},$$

and repeat the process for every set in Γ_i. At the end of this process, each $\boldsymbol{S}_j$ that is fully overlapped with $\boldsymbol{S}_i$ will be rigidly connected to $\boldsymbol{S}_i$ and vice versa. We summarize the process as the following algorithm.

Algorithm 4.1. Construction of a full spanning collection of neighborhoods

Input $\boldsymbol{M} = \{m_1, \cdots, m_N\} \subset \mathbb{R}^n$;
For each m_i, construct $\boldsymbol{M}_i = \{m_{i_1}, \ldots, m_{i_k}\}$ consisting of k nearest points of m_i;
$\Omega = \{\boldsymbol{S}_1, \boldsymbol{S}_2, \cdots, \boldsymbol{S}_N\}$ where $\boldsymbol{S}_i = \{\tau_{i_1}, \ldots, \tau_{i_k}\}$;
For $i = 1$ to N,
    $\Omega_i = \{\boldsymbol{S}_j : \boldsymbol{S}_j$ is rigidly connected to $\boldsymbol{S}_i$ and vice versa $\}$
    $\Gamma_i = \{\boldsymbol{S}_j : \dim(\boldsymbol{S}_j \cap \boldsymbol{S}_i) = d$ and $\boldsymbol{S}_j \notin \Omega_i\}$;
End
For $i = 1$ to N,
    For each $\boldsymbol{S}_j \in \Gamma_i$, if $\boldsymbol{S}_j \notin \left(\bigcup_{\boldsymbol{S}_\ell \in \Omega_i} \Omega_\ell\right)$,
        Construct $\{\hat{\boldsymbol{S}}_j^{(1)}, \hat{\boldsymbol{S}}_j^{(2)}, \ldots, \hat{\boldsymbol{S}}_j^{(k-n_{ij}+1)}\}$ according to (22);
        Construct $\{\hat{\boldsymbol{S}}_i^{(1)}, \hat{\boldsymbol{S}}_i^{(2)}, \ldots, \hat{\boldsymbol{S}}_i^{(k-n_{ij}+1)}\}$ according to (23);
        $\Omega = \Omega \bigcup \{\hat{\boldsymbol{S}}_j^{(1)}, \hat{\boldsymbol{S}}_j^{(2)}, \ldots, \hat{\boldsymbol{S}}_j^{(k-n_{ij}+1)}\} \bigcup \{\hat{\boldsymbol{S}}_i^{(1)}, \hat{\boldsymbol{S}}_i^{(2)}, \ldots, \hat{\boldsymbol{S}}_i^{(k-n_{ij}+1)}\}$;
        $\Omega_i = \Omega_i \bigcup \{\boldsymbol{S}_j\}$ and $\Omega_j = \Omega_j \bigcup \{\boldsymbol{S}_i\}$;
    End
End
Pick any $\boldsymbol{S}_i$ and construct $\{\boldsymbol{S}_i^{(1)}, \boldsymbol{S}_i^{(2)}, \ldots, \boldsymbol{S}_i^{(k-(d+1))}\}$ according to (21);
$\Omega = \Omega \bigcup \{\boldsymbol{S}_i^{(1)}, \boldsymbol{S}_i^{(2)}, \ldots, \boldsymbol{S}_i^{(k-(d+1))}\}$.

Remark 4.1. We have stated Algorithm 4.1 in terms of the original coordinate sets $\boldsymbol{S}_i$ for the ease of notation. In practice, we only have local coordinate sets $\boldsymbol{\Theta}_i$, but all the operations of the algorithm can be carried out equivalently in terms of $\boldsymbol{\Theta}_i$. For example, to test whether $\boldsymbol{S}_j$ is rigidly connected to $\boldsymbol{S}_i$, we need to check the rank of $H_{j\backslash i} = H_j E_{j\backslash i}$ for the selection matrix $E_{j\backslash i}$ (see Definition 3.4). Let G_j be the discrete Hessian operators for $\boldsymbol{\Theta}_j$ and let $G_{j\setminus i} = G_j E_{j\setminus i}$. Since $G_j = V_j H_j$ for some nonsingular matrix V_j by Lemma 3.3, we have $\mathsf{rank}(G_{j\setminus i}) = \mathsf{rank}(H_{j\setminus i})$. Therefore, we actually check $\mathsf{rank}(G_{j\setminus i})$ to determine whether $\boldsymbol{S}_j$ is rigidly connected to $\boldsymbol{S}_i$.

By adding some neighborhoods to the collection Ω, Algorithm 4.1 is expected to result in a collection that is graph connected and has at least one full spanning set. Then, it will be a full spanning collection. In the next section, we will present numerical examples to demonstrate the benefits of this heuristic strategy. We also note that the standard construction of neighborhoods without the extension of this algorithm seems to work fine in practice for problems with $d > 2$. In the 1-dimensional case where the above strategy which is needed, Algorithm 4.1 typically involves moderately extra cost for constructing nested subsets.

5. Numerical Examples

There are a number of numerical examples presented in [1] demonstrating the effectiveness of the Hessian LLE method. The main result of this paper shows that its success depends on a proper construction of the local neighborhoods. In particular, a simple construction of k-nearest neighborhoods may fail. In this section, we present a numerical example to demonstrate the benefit of the expanded construction of neighborhoods proposed in Algorithm 4.1. We compare it with two standard construction methods. The first constructs the k-nearest neighborhood of each point and the second constructs k-nearest neighborhood of each point with the center point itself excluded as in the implementation of [21].

We test whether the constructed collection of neighborhoods is full spanning collection, i.e. whether $\dim \mathsf{null}(\Psi) = d + 1$, by examining the eigenvalues $\lambda_{d+1}(\Psi)$ and $\lambda_{d+2}(\Psi)$, where Ψ is the Hessian alignment matrix for a given collection of neighborhoods and $\lambda_i(\Psi)$ is its i-th smallest eigenvalue. More specifically, we use the ratio $\lambda_{d+2}(\Psi)/\lambda_{d+1}(\Psi)$ as a measure of the full spanning property. We also compare them on the recovered parametrization.

Example 5.1. We consider 4000 random sample points on a 1-d parametric curve in $\mathbb{R}^3$ defined by

$$m_i = [cos(s_i), s_i, sin(s_i)]^T,$$

where s_i is uniformly distributed random numbers between 0 and 0.05. We generate local neighborhoods by constructing, for each point, its k-nearest

Table 1: Eigenvalues for three construction methods of neighborhoods in one dimension.

Methods	Eigenvalues	k=12	k=16	k=20
k-nearest neighbors including center	λ_2	2.1×10^{-20}	3.0×10^{-19}	7.3×10^{-20}
	λ_3	3.7×10^{-20}	3.5×10^{-19}	3.3×10^{-19}
	λ_3/λ_2	1.8	1.2	4.6
k-nearest neighbors excluding center	λ_2	5.7×10^{-18}	2.8×10^{-18}	1.3×10^{-17}
	λ_3	1.1×10^{-17}	4.7×10^{-18}	2.8×10^{-17}
	λ_3/λ_2	1.9	4.2	2.1
Algorithm 4.1	λ_2	2.1×10^{-15}	3.2×10^{-15}	1.0×10^{-14}
	λ_3	1.4×10^{-9}	2.7×10^{-8}	1.3×10^{-7}
	λ_3/λ_2	6.6×10^{5}	8.4×10^{6}	1.2×10^{7}

neighborhood, its k-nearest neighborhood with the point itself removed [21], or by Algorithm 4.1. We then implement the Hessian LLE algorithm with the local neighborhoods constructed. We present relevant spectral properties of the Hessian alignment matrix Ψ in Table 1 with $k = 12$, $k = 16$ and $k = 20$. With $d = 1$, we list the second smallest eigenvalue λ_2 and the third smallest eigenvalue λ_3 as well as the ratio λ_3/λ_2 for each method. We further present the parametrization obtained by each of the methods by plotting the recovered coordinates (in the x-axis) against the original ones s_i (in the y-axis) in Figure 5.1.

We see that the first two methods result in both λ_2 and λ_3 nearly zero and fail to detect the null space. In particular there is no gap between the two eigenvalues. Algorithm 4.1 results in a much larger λ_3 and a significant gap between the two eigenvalues. With respect to the parametrization recovered using the second eigenvector, Algorithm 4.1 correctly recovers all the original coordinates as a linear transformation (the third row of Figure 5.1), and the construction of k-nearest neighborhoods excluding the center point only correctly recovers a portion of the original coordinates (the second row of Figure 5.1), while the coordinates obtained by k-nearest neighborhoods appears to be random noise (the first row of Figure 5.1).

We point out that the result of this example is typical in our experiments for 1-dimensional problems where extension of neighborhoods using Algorithm 4.1 is necessary for the Hessian LLE to work. We have tested the three constructions for problems with $d \geq 2$ and they all produce significant gaps between the two eigenvalues. It appears that for $d \geq 2$, the standard constructions of neighborhoods are practically sufficient.

6. Conclusion

In this paper, we have analyzed a discrete version of the Hessian Eigenmaps method by investigating the null space of Hessian alignment matrix defined from the local coordinates and from isometric coordinates. We prove that Hessian Eigenmaps can recover the original coordinates up to a rigid motion under certain conditions on local neighborhoods. We also propose a heuristic algorithm to construct the neighborhoods for Hessian Eigenmaps, which significantly improves standard constructions in 1-dimensional problems.

For the future works, it will be interesting to study and compare several dimensionality reduction methods from the point of view of spectral analysis. For example, methods such as LLE, LTSA, and Hessian Eigenmaps all share a common framework of reconstruction through alignment matrices. Spectral analysis can play an important role in better understanding of the performance of these algorithms.

Acknowledgement

This research was supported in part by National Science Foundation under grants DMS-1317424 and DMS-1318633.

References

[1] D. Donoho, C. Grimes, Hessian Eigenmaps: Locally linear embedding techniques for high-dimensional data, Proceedings of National Academy of Sciences 100 (2003) 5591–5596.

[2] S. T. Roweis, L. K. Saul, Nonlinear dimensionality reduction by locally linear embedding, Science 290 (2000) 2323–2326.

[3] J. B. Tenenbaum, V. de Silva, J. C. Langford, A global geometric framework for nonlinear dimensionality reduction, Science 290 (2000) 2319–2323.

[4] M. Belkin, P. Niyogi, Laplacian eigenmaps for dimensionality reduction and data representation, Neural Computation 15 (2003) 1373–1396.

[5] Z. Zhang, H. Zha, Principal manifolds and nonlinear dimension reduction via local tangent space alignment, SIAM J. Sci. Comput. 26 (2004) 313–338.

[6] M. Brand, Charting a manifold, in: S. Becker, S. Thrun, K. Obermayer (Eds.), Advances in Neural Information Processing Systems 15, MIT Press, 2003, pp. 961–968.

[7] K. Q. Weinberger, B. D. Packer, L. K. Saul, Nonlinear dimensionality reduction by semidefinite programming and kernel matrix factorization, in: Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics, 2005, pp. 381–388.

[8] K. Q. Weinberger, L. K. Saul, An introduction to nonlinear dimensionality reduction by maximum variance unfolding, in: Proceedings of the Twenty First National Conference on Artificial Intelligence, AAAI, Boston, MA, 2006, pp. 1683–1686.

[9] M. Wang, F. Sha, M. I. Jordan, Unsupervised kernel dimension reduction, in: Proceedings of Neural Information Processing Systems, 2010, pp. 2379–2387.

[10] L. K. Saul, K. Q. Weinberger, J. Ham, F. Sha, D. Lee, Spectral methods for dimensionality reduction, in: O. Chapelle, B. Schoelkopf, A. Zien (Eds.), Semisupervised Learning, 2006.

[11] L. K. Saul, S. T. Roweis, Think globally, fit locally: Unsupervised learning of low dimensional manifolds, Journal of Machine Learning Research 4 (2003) 119–155.

[12] H. Zha, Z. Zhang, Spectral analysis of alignment in manifold learning, in: Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '05), Vol. 5, 2005, pp. 1069–1072.

[13] F. Sha, L. K. Saul, Analysis and extension of spectral methods for nonlinear dimensionality reduction, in: Proceedings of the Twenty Second International Conference on Machine Learning, 2005, pp. 784–791.

[14] C.-K. Li, R.-C. Li, Q. Ye, Eigenvalues of an alignment matrix in nonlinear manifold learning, Comm. Math. Sciences 5 (2007) 313–329.

[15] X. Huo, X. S. Ni, A. K. Smith, A survey of manifold-based learning methods, in: T. W. Liao, E. Triantaphyllou (Eds.), Recent Advances in Data Mining of Enterprise Data, World Scientific, Singapore, 2007, pp. 691–745.

[16] Q. Ye, H. Zha, R.-C. Li, Analysis of an alignment algorithm for nonlinear dimensionality reduction, BIT - Numerical Mathematics 47 (2007) 873–885.

[17] A. K. Smith, X. Huo, H. Zha, Convergence and rate of convergence of a manifold-based dimension reduction algorithm, in: Proceedings of Neural Information Processing Systems, 2008, pp. 1529–1536.

[18] A. K. Smith, New results in dimension reduction and model selection, Ph.D. Thesis.

[19] X. Huo, A. K. Smith, Matrix perturbation analysis of local tangent space alignment, Linear Algebra and Its Applications 430 (2009) 732–746.

[20] H. Zha, Z. Zhang, Spectral properties of the alignment matrices in manifold learning, SIAM Review 51 (2009) 545–566.

[21] D. Donoho, C. Grimes, MATLAB program for HLLE, http://basis.stanford.edu/HLLE.

[22] J. Demmel, Applied Numerical Linear Algebra, SIAM, Philadelphia, PA, 1997.

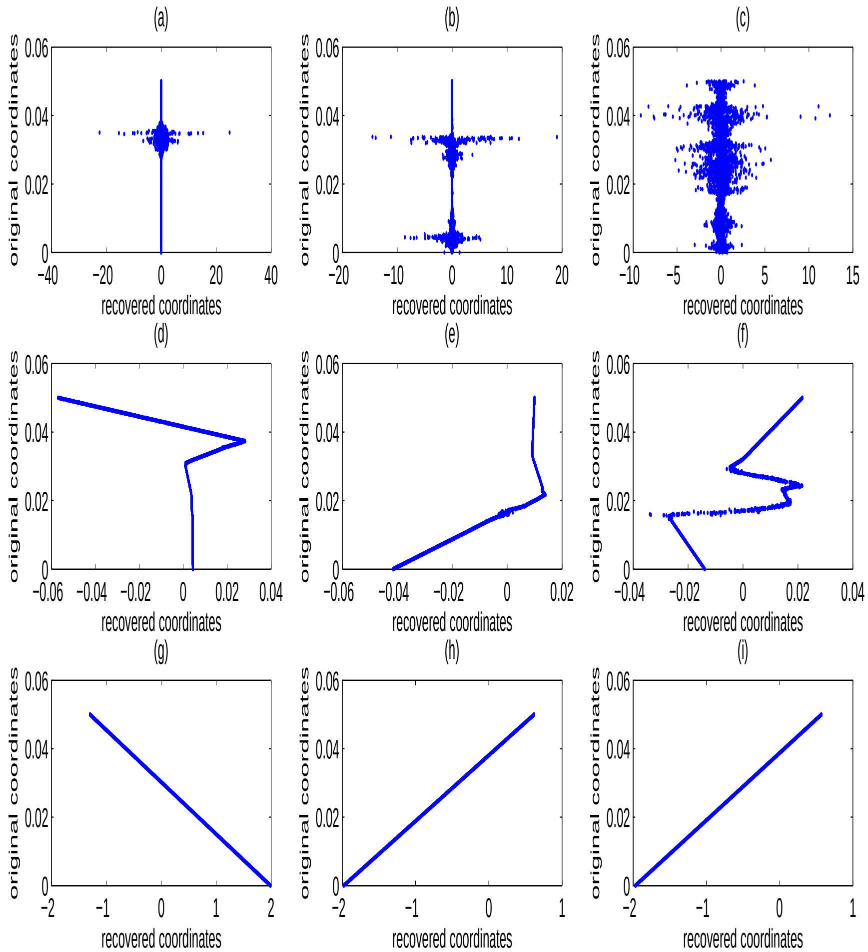

Figure 5.1: The recovered coordinates vs. the original coordinates. First row: k-nearest neighborhoods with $k = 12$ (in (a)), $k = 16$ (in (b)) and $k = 20$ (in (c)); Second row: k-nearest neighbors excluding the center point with $k = 12$ (in (d)), $k = 16$ (in (e)) and $k = 20$ (in (f)); Third row: Algorithm 4.1 with $k = 12$ (in (g)), $k = 16$ (in (h)) and $k = 20$ (in (i))