Grassmannian Dimensionality Reduction for Optimized Universal Manifold Embedding Representation of 3D Point Clouds

Yuval Haitman
Ben-Gurion University
Beer-Sheva, Israel
haitman@post.bgu.ac.il

Joseph M. Francos
Ben-Gurion University
Beer-Sheva, Israel
francos@ee.bgu.ac.il

Louis L. Scharf
Colorado State University
Fort Collins, Colorado USA
louis.scharf@colostate.edu

Abstract

Consider a 3-D object and the orbit of equivalent objects turned out by the rigid transformation group. The set of possible observations on these equivalent objects is generally a manifold in the ambient space of observations. It has been shown that the rigid transformation universal manifold embedding (RTUME) provides a mapping from the orbit of observations on some object to a single low dimensional linear subspace of Euclidean space. This linear subspace is invariant to the geometric transformations and hence is a representative of the orbit. In the classification set-up the RTUME subspace extracted from an experimental observation is tested against a set of subspaces representing the different object manifolds, in search for the nearest class. We clarify the way in which level-set functions, computed at each quantization level in an observation, serve as a basis for the invariant subspaces in RTUME. In the presence of observation noise and random sampling patterns of the point clouds, the observations do not lie strictly on the manifold and the resulting RTUME subspaces are noisy. Inspired by the ideas of Locality Preserving Projections and Grassmannian dimensionality reduction, we derive an optimal companding of the level-set functions yielding the Grassmannian dimensionality reduction universal manifold embedding (GDRUME). We evaluate the proposed method in a classification task on a noisy version of the ModelNet40 dataset and compare its performance to that of PointNet classification DNN. We show that in the presence of noise, GDRUME provides highly accurate classification results, while the performance of PointNet is poor.

This research is supported by NSF-BSF Computing and Communication Foundations (CCF) grants, CCF-2016667, CCF-1712788 and BSF-2016667.

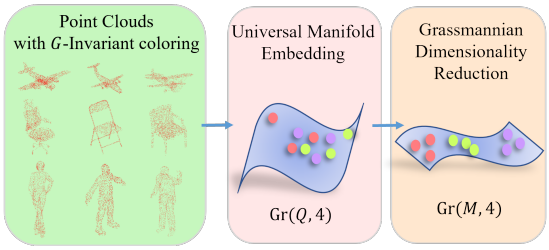

Figure 1. Illustration of our proposed method- 3D point clouds of different objects colored by a transformation invariant coloring function are embedded in a high dimensional Grassmannian manifold Gr$(Q,4)$, by the Rigid Transformation Universal Manifold Embedding (RTUME). Grassmannian dimensionality reduction (GDR) is used to find a lower ambient space Grassmannian manifold Gr$(M,4)$ where $M < Q$, to preform better separation between different objects.

1. Introduction

Detection and classification of whole objects or parts thereof are elementary building blocks in solving 3-D vision problems, from object classification, to part segmentation, to keypoint detection and matching for point cloud registration. However, an object to be detected may present itself subject to *a-priori* unknown geometric transformations. Hence an understanding of the set of all possible observations of that single object is essential. As a result of the action of geometric deformations, the set of observations of an object is generally a manifold in the observations space. Thus, although the data may be sampled and presented in a high-dimensional space, in fact the intrinsic complexity and dimensionality of the observed physical phenomenon are low.

Universal manifold embedding (UME) [6, 15], is a methodology for constructing a matrix representation of an observation such that it is covariant with the transformation, and then using this representation to identify a linear subspace that is invariant to affine coordinate transformations

of the observation. The covariant UME matrix representation obtained by this procedure may be inverted for the parameters of the geometric transformation. This framework has been recently expanded to address the registration of 3D point clouds related by a rigid transformation, [4]. Practical application of the method requires a high-quality estimate of the invariant subspace for each of K objects, and each of these subspaces must be estimated from one or more versions of an object, imperfectly imaged in one or more of its representative poses.

Since the classifier performance highly depends on the choice of the set of functions composing the UME operator, we wish to find the optimal set of functions, that best separates the UME matrix representation of each class (object) from those of the other classes, while minimizing the distance between observations from the same class. In this paper we address these issues by extending the theory of universal manifold embedding and Grassmannian dimensionality reduction:

1. We clarify the way in which *level-set functions*, computed at each quantization level in an observation, serve as a basis for the invariant subspaces in RTUME.

2. Building upon the ideas of Locality Preserving Projections, [8], and Grassmannian dimensionality reduction, [19], we derive an optimal companding of the level-set functions yielding the Grassmannian dimensionality reduction universal manifold embedding (GDRUME).

3. In the presence of observation noise, and random sampling patterns, the observations do not lie strictly on the manifold and the resulting RTUME subspaces are noisy. The derived analytic solution for designing the GDRUME operators is equivalent to *simultaneous* denoising of all the object manifolds.

The structure of this paper is as follows: In Section 2 we provide the basic definitions and properties of the rigid transformation universal manifold embedding. Then, in Section 3 we define the problem of detection and classification on the Grassmannian manifold. In Section 4 we consider the problem of finding the optimal UME as an optimal selection of weight coefficients for the Fundamental UME representation. In Section 5 we present our proposed method for Grassmannian optimization - GDRUME. Finally, Section 6 includes experimental results.

2. Problem Formulation

Consider a 3-D object $s \in \{s_1, \cdots, s_K\}$, and an *orbit* of equivalent objects formed by the action of the transformation group $G = SE(3)$. An observation $X(s_k)$ on object s_k will be denoted X_k and the set $\psi_k = \{\alpha \circ s_k, \alpha \in G\}$ will denote the orbit of possible appearances of s_k turned out by the group G. There exists one such orbit for each object s_k. Our aim is to nonlinearly map each observation X_k, taken from the orbit ψ_k, to a matrix representation $\mathbf{T}(X_k)$. This matrix is to be linearly covariant with the parametrization of G; Its column space, which we denote by $\langle \mathbf{T}(X_k) \rangle$ is to be G-invariant. In other words, the orbit ψ_k is mapped into a linear subspace $\langle \mathbf{T}(X_k) \rangle$, such that the mapping is G-invariant.

It has been shown [15] that in the case where the observations on an object are determined by an affine geometric transformation of coordinates the UME operator returns a basis $\mathbf{T}(X)$ that is covariant with the coordinate transformation, and a subspace $\langle \mathbf{T}(X) \rangle$ that is G-invariant. That is, the set of *all* possible observations on an object under group action G is mapped by the UME operator into a *single* linear subspace which is invariant to the geometric transformation. Since the Special Euclidean group is a subgroup of the Affine group, in this paper we adapt this general framework for classification and detection of 3-D point clouds.

2.1. The RTUME Descriptor

The *universal manifold embedding* (UME) maps every observation X from the orbit of s to a matrix $\mathbf{T}(X) \in \mathcal{T}(M, n+1)$, such that $\mathbf{T}(X)$ is covariant with the geometric transformation, and where $\mathcal{T}(M, n+1)$ is the space of $M \times (n+1)$ real-valued matrices, and M the dimension of the embedding Euclidean space. The map $\mathcal{Q} : \mathcal{T}(M, n+1) \to \mathrm{Gr}(M, n+1)$, where $\mathrm{Gr}(M, n+1)$ is the Grassmann manifold of $n+1$-dimensional linear subspaces of M-dimensional Euclidean space, maps $\mathbf{T}(X)$ to its column space $\langle \mathbf{T}(X) \rangle$. Thus, the UME maps the orbit of s into the G-invariant subspace $\langle \mathbf{T}(X) \rangle \in \mathrm{Gr}(M, n+1)$.

Considering the special case of rigid transformations of 3-D objects, the Rigid Transformation UME (RTUME) [4] is a mapping of functions to matrices. It is covariant with rigid transformations of coordinates, *i.e.* the RTUME matrices of functions on $\mathbb{R}^3$ related by a rigid transformation of coordinates are related by the same rigid transformation: Given a function $h : \mathbb{R}^3 \mapsto \mathbb{R}$, the RTUME matrix representation of $h(\mathbf{x})$ is given by

$$\mathbf{T}(h) = \begin{bmatrix} \int\limits_{\mathbb{R}^3} w_1 \circ h(\mathbf{x}) d\mathbf{x} & \int\limits_{\mathbb{R}^3} x_1 w_1 \circ h(\mathbf{x}) d\mathbf{x} & \dots & \int\limits_{\mathbb{R}^3} x_3 w_1 \circ h(\mathbf{x}) d\mathbf{x} \\ & \vdots & & \\ \int\limits_{\mathbb{R}^3} w_M \circ h(\mathbf{x}) d\mathbf{x} & \int\limits_{\mathbb{R}^3} x_1 w_M \circ h(\mathbf{x}) d\mathbf{x} & \dots & \int\limits_{\mathbb{R}^3} x_3 w_M \circ h(\mathbf{x}) d\mathbf{x} \end{bmatrix} \tag{1}$$

where w_i, $i = 1, ..., M$ are measurable functions aimed at generating many compandings of the observation.

Let $h(\mathbf{x})$ and $g(\mathbf{x})$ be two functions related by a rigid transformation of coordinates such that $h(\mathbf{x}) = g(\mathbf{R}\mathbf{x} +$

$\mathbf{t}$). The RTUME matrices $\mathbf{T}(h)$ and $\mathbf{T}(g)$ constructed from $h(\mathbf{x})$ and $g(\mathbf{x})$ as in (1) are related by the relation

$$\mathbf{T}(h) = \mathbf{T}(g)\mathbf{D}^{-1}(\mathbf{R}, \mathbf{t}) \tag{2}$$

where $\mathbf{D}(\mathbf{R}, \mathbf{t})$ is given by:

$$\mathbf{D}(\mathbf{R}, \mathbf{t}) = \begin{bmatrix} 1 & \mathbf{t}^T \\ \mathbf{0} & \mathbf{R}^{-1} \end{bmatrix} \tag{3}$$

Since $\mathbf{T}(h)$ and $\mathbf{T}(g)$ are related by an invertible transformation that is a re-expression of the rigid transformation relating the observations, we say that the basis $\mathbf{T}(g)\mathbf{D}^{-1}(\mathbf{R}, \mathbf{t})$ is *covariant* with the rigid transformation. Hence it provides a method for estimating the transformation that relates any two observations. Furthermore, since $\mathbf{T}(h)$ and $\mathbf{T}(g)$ are related by a right invertible linear transformation, the column space of $\mathbf{T}(g)$ and the column space of $\mathbf{T}(h)$ are identical. Their bases are different, but their range spaces are identical.

Note that unlike the classical moment invariant methods that use high-order moments and their nonlinear invariant functions for classification, the RTUME representation uses only zeroth- and first-order moments of many compandings of the observation. These produce a subspace representation of the observation orbit, a subspace that may be used for invariant detection-and-classification.

3. The Detection-Classification Problem and The Distance Between Equivalence Classes

The RTUME uses the operator $\mathbf{T}$ to universally map a manifold, generated by the set all rigid coordinate transformations of a 3-D object, into a G-invariant linear subspace. That is, the RTUME operator maps every observation X taken from the orbit $\{\alpha \circ s, \alpha \in G\}$, to a point $\langle \mathbf{T}(X) \rangle$ on the Grassmannian $\mathrm{Gr}(M, 4)$.

In the RTUME framework the problem of detection and classification of geometrically deformed objects is formalized as follows: Given an observation Z, in the form of a sampled point cloud, where its geometric deformation is unknown, the problem is to determine whether $Z = \alpha \circ X$ or $Z = \beta \circ Y$, for some $\alpha, \beta \in G$, and X, Y some reference observations of known objects.

Since the detection and classification are to be G-invariant, we propose to compute $\mathbf{T}(Z)$ using (1) and measure the distance between the subspace $\langle \mathbf{T}(Z) \rangle$ and the subspaces $\langle \mathbf{T}(X) \rangle$ and $\langle \mathbf{T}(Y) \rangle$. That is, the observation Z is determined to belong to the orbit ψ_s if the distance from $\langle \mathbf{T}(Z) \rangle$ to $\langle \mathbf{T}(X) \rangle$, where X is some representative observation on object s, is smaller than its distance to $\langle \mathbf{T}(Y) \rangle$, where Y is some representative observation on a different object (and is small enough to be considered a detection).

Following [5], [3] we compute the distance between a pair of subspaces as the extrinsic distance, evaluated using the projection Frobenius-norm

$$d_{pF}(\langle \mathbf{T}(Z) \rangle, \langle \mathbf{T}(X) \rangle) = \frac{1}{\sqrt{2}} ||\mathbf{P}_X - \mathbf{P}_Z||_F = ||\sin \boldsymbol{\theta}||_2 \tag{4}$$

where $\sin \boldsymbol{\theta}$ is a vector of sines of principal angles between the subspaces. The matrix $\mathbf{P}_X$ denotes the orthogonal projection matrix onto the subspace $\langle \mathbf{T}(X) \rangle$.

Since the classifier performance highly depends on the choice of the set of w-functions composing the operator $\mathbf{T}$, we wish to find the optimal set of w-functions, that best separates the UME matrix representation of each class (object) from those of the other classes.

4. Design of the RTUME Operator

4.1. Defining an $SE(3)$-invariant Function

As point clouds are sets of coordinates in 3-D with no functional relation imposed on them, a necessary step in adapting the UME framework for point cloud processing is to define a function that assigns each point in the cloud with a value, invariant to the action of the transformation group. In the ideal case where finite support objects are considered, sampling is dense and uniform, and there is no observation noise - such functions can be defined in a relatively robust and simple manner, *e.g.*, distance from the point cloud center of mass. However, when sampling of the point clouds is sparse, non uniform, and noisy, which is the case in practice, evaluation of an $SE(3)$-invariant function from the observed sampled point cloud, can be achieved only approximately. The design of the RTUME operator presented in the following sections, is aimed at handling these approximations.

4.2. Representation by Level-Sets

Assume we are given an observation $X(\mathbf{u})$, $\mathbf{u} \in \mathbb{R}^3$, where, in general, $X(\mathbf{u})$ is evaluated from the raw point cloud measurements using an SE(3)-invariant function such as the distance of $\mathbf{u}$ from the object center of mass, or alternatively is provided by the measurement device, for example as an RGB measurement at $\mathbf{u}$. Further assume that the values of X are uniformly quantized at levels $\{q_i\}_{i=1}^{Q}$, so that it may be written as

$$X(\mathbf{u}) = \sum_{i=1}^{Q} q_i I_i^X(\mathbf{u}) \tag{5}$$

where $I_i^X(\mathbf{u})$ is the indicator function that equals 1 on the level-set of $\mathbf{u}$ where $q_{i-1} < X(\mathbf{u}) \leq q_i$, and zero elsewhere. Since $X(\mathbf{u})$ is SE(3)-invariant, the support of $I_i^X(\mathbf{u})$ for every i should be on the same surface points regardless of the rigid transformation the object has undergone.

The w operators must be designed such that the result of their application is covariant with the geometric transformation, and hence they are not functions of the coordinates. The action of w_m on X is simply to map the levels q_i into levels $w_m(q_i)$, leaving the indicator functions I_i^X unchanged. Then, each term in the matrix $\mathbf{T}(X)$ may be written as

$$\begin{aligned}\mathbf{T}_{m,j} &= \int_{\mathbb{R}^3} w_m \circ X(\mathbf{u}) u_j d\mathbf{u} \\ &= \sum_{i=1}^{Q} w_m(q_i) \int_{\mathbb{R}^3} I_i^X(\mathbf{u}) u_j d\mathbf{u} = \sum_{i=1}^{Q} w_{m,i} F_{ij}^X \quad (6)\end{aligned}$$

where $w_{m,i} = w_m(q_i)$. This makes the moments $F_{ij}^X = \int_{\mathbb{R}^3} I_i^X(\mathbf{u}) u_j d\mathbf{u}$, the point cloud features of fundamental interest. Moreover, we can now write the moment matrix $\mathbf{T}(X)$ as

$$\mathbf{T}(X) = \mathbf{W}^T \mathbf{F}^X;\ \mathbf{W}^T = \{w_{m,i}\} \in \mathbb{R}^{M \times Q} \quad (7)$$

where $\mathbf{F}^X = \{F_{ij}^X\} \in \mathbb{R}^{Q \times 4}$ may be called the *Fundamental RTUME* representation matrix (FUME) for point cloud X. Since $M \leq Q$, the role of $\mathbf{W}$ is to transform the subspace $\langle \mathbf{F}^X \rangle \subset \text{Gr}(Q, 4)$ to the subspace $\langle \mathbf{G}^X \rangle \subset \text{Gr}(M, 4)$. A single $\mathbf{W}$ has to serve for all the orbits $\psi_1, \ldots, \psi_K$.

The FUME representation uses zeroth-order and first-order moments of many level-sets of the observation. These produce a matrix representation of the observation orbit that may be used for covariant estimation and invariant classification.

However when differently sampled and noisy observations on the point clouds are considered (6) no longer holds due to the presence of noise and $X(\mathbf{u})$ evaluated from the point cloud is noisy as well. Hence, letting $\widetilde{\mathbf{u}} = \mathbf{u} + \mathbf{n}$, (6) should be rewritten as

$$\mathbf{T}_{m,j} = \int_{\mathbb{R}^3} w_m \circ \widetilde{X}(\widetilde{\mathbf{u}}) \widetilde{\mathbf{u}} d\widetilde{\mathbf{u}} = \sum_{i=1}^{Q} w_{m,i} \widetilde{F}_{ij}^X \quad (8)$$

We have thus reduced the problem of finding an optimal set of RTUME representations for the K objects, to a problem of finding $\mathbf{W}$.

4.3. The Optimal RTUME Operators

We next wish to find the optimal set of w-functions, that best separates the RTUME matrix representation of each class (object) from those of the other classes. It is assumed that we have a set of K objects, such that for each object N observations are available. Applying the RTUME operator (1), using a set of functions, $\mathbf{W}$, chosen to be the indictor functions on the level-sets of the quantization levels, *i.e.*, $\mathbf{W} = \mathbf{I}_Q$, we find the fundamental RTUME representation for each of the observations. Next, we wish to find an improved alternative for this choice of $\mathbf{W}$.

4.4. Locality Preserving Projections

In the language of a generic problem of linear dimensionality reduction, the problem of finding the optimal w-functions is the following: Given a set of N observations from each of K different orbits, and the FUME matrices $\{\mathbf{F}^{k,j}\}_{k=1,j=1}^{k=K,j=N}$ evaluated from these observations, where $\langle \mathbf{F}^{k,j} \rangle \in \text{Gr}(Q, 4)$, find a transformation matrix $\mathbf{W}^T \in \mathbb{R}^{M \times Q}$ that maps these points on the Grassmann to a set of points $\{\langle \mathbf{Y}^{k,j} \rangle\}_{k=1,j=1}^{k=K,j=N} \in \text{Gr}(M, 4)$ where $\text{Gr}(M, 4)$ is a Grassmann of a smaller ambient space such that $\mathbf{Y}^{k,j}$ "represents" $\mathbf{F}^{k,j}$, where $\mathbf{Y}^{k,j} = \mathbf{W}^T \mathbf{F}^{k,j}$ and the mapping $\mathbf{W}$ is designed such that observations from the same orbit generate close together subspaces and observations from different orbits generate subspaces that are as far as possible from each other. Note that a sufficient condition that guarantees that all $\langle \mathbf{Y}^{k,j} \rangle$ are indeed points on $\text{Gr}(M, 4)$, *i.e.*, that the dimension of the subspace remains 4 despite the reduction in the dimensionality of the ambient space is that $\mathbf{W}$ has a full column rank, or alternatively, to choose the columns of $\mathbf{W}$ to be orthonormal.

4.5. Related Work

Grassmannian dimensionality reduction (GDR) is often employed in the framework of Grassmannian discriminant analysis (GDA), [19]. There are two common paradigms for discriminant analysis using GDR: kernel based and Grassmannian optimization. Kernel based method first embed the Grassmann manifold into a high dimensional Hilbert space by using a kernel function, followed by a learned mapping into a lower dimensional space. In [7], the discriminant analysis on the lower dimension Hilbert space is done by using LDA and can be followed by a K-NN for classification. However, kernel-based methods do not fully utilize the geometric structure of the manifold. This led to an alternative paradigm, the Grassmannian optimization approach, which is adopted in this work as well. This approach exploits the Riemannian geometry of the Grassmann manifold by using gradient based methods defined on the Riemannian manifold.

A method that employs Grassmannian optimization for GDR is the Projection Metric Learning (PML) on the Grassmann manifold [9]. PML learns a mapping $f : \text{Gr}(D, q) \to \text{Gr}(d, q)$ where $D > d$ by optimizing a discriminant function, designed to minimize the Grassmannian projection distances of within-class subspace pairs while maximizing the projection distances of between-class subspace pairs. The optimization problem is eventually reduced to the search for a symmetric PSD matrix $\mathbf{P}$ that minimizes the cost function. A similar approach is the Joint Normalization and Dimensionality Reduction on Grassmannian [13], that employs Grassmannian optimization to learn a mapping from a Grassmann manifold with high ambient space

to a Grassmann with a lower ambient space. The procedure employs a discriminant function with an affinity matrix for encoding the within-class and between-class information. This method is eventually reduced into finding a matrix $\mathbf{W}$ on the Stifel manifold. Similarly, to the PML the input data is represented on the Grassmann manifold by applying the SVD.

In [10], the framework of GDR and the use of Grassmannian optimization are expanded to construct GRNet, a deep neural network that works directly on the Grassmann manifold.

5. RTUME Design Using Grassmannian Dimensionality Reduction

We are given a training set of N labeled point clouds (observations) $\{X_i, k_i\}_{i=1}^N$, $k_i \in \{1, \dots K\}$ for each one of K different objects (classes), where observations of the same object differ by a rigid coordinate deformation, sampling pattern, and white additive Gaussian noise. FUME matrix is generated for each observation $\{\mathbf{F}_i\}_{i=1}^N$. The column space of each FUME matrix is a point in $\mathrm{Gr}(Q, 4)$. Inspired by [13], we want to solve an optimization problem that will map FUME matrices belonging to the same orbit (class) to close points on a Grassmannian of a reduced dimension ambient space, and those from different classes to be far apart. Therefore we formulate the next optimization problem:

$$\begin{aligned} \min_{\mathbf{W}\in\mathbb{R}^{Q\times M}} \quad & L(\mathbf{W}) = \sum_{i,j=1}^{N} \mathbf{G}_{ij} d_{pF}^2 \left(\langle \mathbf{Q}_i(\mathbf{W})\rangle, \langle \mathbf{Q}_j(\mathbf{W})\rangle\right) \\ \text{subject to} \quad & \mathbf{W}^{\mathbf{T}}\mathbf{W} = \mathbf{I}_M \end{aligned} \tag{9}$$

where $\mathbf{G}_{ij}$ is an affinity matrix created from the dataset and $\{\mathbf{Q}_i(\mathbf{W})\}_{i=1}^N$ are orthogonal matrices given by the *QR-decomposition* of $\mathbf{W}^T\mathbf{F}_i$. Due to the constraint on $\mathbf{W}$ this optimization problem is solved over the Stiefel manifold $\mathrm{St}(Q, M)$. Enforcing this constraint guarantees that we avoid degeneration of the solution to the trivial one.

5.1. The Affinity Matrix

We would like to encode the structure of the data by an affinity function. Because our problem is supervised, by making use of the data labels we create an affinity matrix that favours observations from the same class, so that the choice of $\mathbf{W}$ by the cost function makes them "closer" and penalizes observations from different class so that the choice of $\mathbf{W}$ by the cost function tends to separate them.

$$[\mathbf{G}]_{ij} = g_w(\mathbf{X}_i, \mathbf{X}_j) - \alpha g_b(\mathbf{X}_i, \mathbf{X}_j) \tag{10}$$

where $\alpha > 0$ is a trade-off factor between the within-class information and the out-of-class information, $g_w(.,.), g_b(.,.)$ are binary functions defined as follows:

$$g_w(\mathbf{X}_i, \mathbf{X}_j) = \begin{cases} 1, & \mathbf{X}_i \in N_w(\mathbf{X}_j) \text{or} \mathbf{X}_j \in N_w(\mathbf{X}_i) \\ 0, & \text{otherwise} \end{cases} \tag{11}$$

$$g_b(\mathbf{X}_i, \mathbf{X}_j) = \begin{cases} 1, & \mathbf{X}_i \in N_b(\mathbf{X}_j) \text{or} \mathbf{X}_j \in N_b(\mathbf{X}_i) \\ 0, & \text{otherwise} \end{cases} \tag{12}$$

where $N_w(\mathbf{X}_i)$ is the set of k_w nearest neighbors of $\mathbf{X}_i$ that share the same label as $\mathbf{X}_i$ which we name within-class neighborhood, and $N_b(\mathbf{X}_i)$ is the set of k_b nearest neighbors of $\mathbf{X}_i$ that have different label from $\mathbf{X}_i$ which we name between-class neighborhood. The neighborhood is computed according to the distance (4) and its size is determined by cross validation, where usually k_w is the size of the smallest class in the train data set ($k_w = N$ if training data has N samples for each object) and $k_b \leq k_w$.

5.2. Solving The Optimization Problem

In order to solve (9) we used the framework of Riemannian Conjugate Gradient (RCG) [3, 16, 17, 1], this is a generalization of the Euclidean Conjugate Gradient on a Riemannian manifold. The method was implemented using Manopt [2]. In our case we solve the optimization problem over the Stiefel manifold.

5.2.1 Finding The Gradient

We first find the cost function's gradient on the Stiefel manifold. In this case the gradient is defined as follows [3]:

$$\nabla L = L_{\mathbf{W}} - \mathbf{W} L_{\mathbf{W}}^T \mathbf{W}^T \tag{13}$$

$$[L_{\mathbf{W}}]_{i,j} = \frac{\partial L}{\partial \mathbf{W}_{i,j}} \tag{14}$$

(14) describes the Euclidean gradient of the cost function $L(\mathbf{W})$, given by (see Appendix A for the derivation):

$$L_{\mathbf{W}} = -4 \sum_{i=1}^{N} \mathbf{F}_i \mathbf{R}_i^{-1} \mathbf{Q}_i^T \mathbf{H}_i \mathbf{S}_i \tag{15}$$

where $(\mathbf{Q}_i, \mathbf{R}_i)$ is the *QR-decomposition* of $\mathbf{Y}_i = \mathbf{W}^T\mathbf{F}_i$, $\mathbf{S}_i = \mathbf{I}_M - \mathbf{Q}_i\mathbf{Q}_i^T$ and $\mathbf{H}_i = \sum_{j=1}^N [\mathbf{G}]_{ij}\mathbf{Q}_j\mathbf{Q}_j^T$.

6. Experimental Results

We demonstrate the use of FUME and GDRUME in the context of point cloud classification. We show that by using GDRUME we can achieve better separation on the Grassmann manifold between deformed observations on objects from different classes. We examine the two methods under different noise conditions and varying number of points in the point cloud. We compare our methods with the well known PointNet deep neural network [14].

Algorithm 1 GDRUME

Input: $\{\mathbf{F}_i\}_{i=1}^N \subseteq \mathcal{T}(Q, n+1)$, $\{k_i\}_{i=1}^N \subseteq \{1, \dots K\}$, $M \in \mathbb{N}$

Output: $\mathbf{W} \in \mathrm{St}(Q, M)$

1: Apply *QR Decomposition* to $\{\mathbf{F}_i\}_{i=1}^N \to \{\mathbf{Q}_i, \mathbf{R}_i\}_{i=1}^N$
2: $\{\mathbf{F}_i\}_{i=1}^N \leftarrow \{\mathbf{Q}_i\}_{i=1}^N$
3: Create Affinity matrix $\mathbf{G}$ from $\{\mathbf{F}_i, k_i\}_{i=1}^N$ using (10)
4: Initialize $\mathbf{W} \leftarrow \mathbf{W}_0 \in \mathrm{St}(Q, M)$
5: Compute $L(\mathbf{W})$ according to (9)
6: Solve (9) using RCG [1]

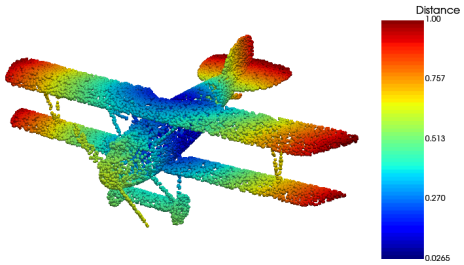

Figure 2. Example of a point cloud from ModelNet40 colored by distance from center of mass.

6.1. Data Set and Prepossessing

We evaluate our methods for the task of object separation and classification on the ModelNet40 dataset. ModelNet40 contains CAD models of objects from 40 different classes given as triangular meshes. We used a point cloud representation of each object by sampling its mesh at $20,000$ uniformly distributed points. We normalized each point cloud to the unit sphere to ignore potential scale differences between the models. In the following experiment we used a subset of ModelNet40, *i.e.*, we took 10 different models from each class, for each, we generated 150 different observations (100 for the training phase and 50 for testing), each with different rigid transformation, random sub-sampling and additive Gaussian noise. The rigid transformations are composed of a random roll and yaw in the range of $[0, 180]$ degrees and random pitch in the range of $[0, 90]$ degrees. As for the noise, two noise levels are investigated: medium noise with std of 0.5 mesh resolution and stronger noise with std of 0.8 mesh resolution. The mesh resolution is estimated from the point cloud by sampling its most dense regions. In order to employ the RTUME framework, we need to define on each point cloud an $SE(3)$ - invariant function. However, since the two point clouds are differently sampled and noisy every function which is $SE(3)$ - invariant in the noise-free case, will be noisy, as well. In the experiments we employ the Euclidean distance from center of mass.

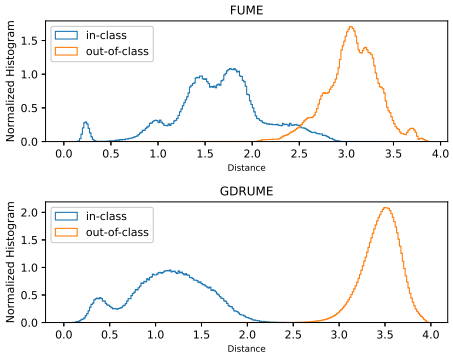

Figure 3. Distance histogram on the Grassmann manifold evaluated using (4) between each pair of samples from the training set of the medium level noise. Following the use of GDRUME, observations from the same class (in-class) are closer while observations from different classes (out-of-class) are further apart. Setting the decision threshold at the intersection point between the in-class and the out-of-class histograms yields for the initial representation by the FUME $\hat{P}(out|in) = 0.044$ and $\hat{P}(in|out) = 0.037$, while after the GDRUME optimization both $\hat{P}(out|in)$ and $\hat{P}(out|in)$ are null.

Method	Accuracy [%]
PointNet [14] (Noisy train set)	1.28
FUME	90.03
GDRUME	**91.87**

Table 1. Accuracy comparison of PointNet, FUME, GDRUME on deformed ModelNet40 dataset in the presence of the stronger noise level, tested on point clouds with 2048 points.

6.2. Training

For every point cloud, the FUME matrix is evaluated using (1) with $Q = 128$. The desired reduced dimension is $d = 32$ (The values of Q and d were experimentally set). The training procedure is detailed in Algorithm 1. In Figure 3 we depict the measured distanced on the Grassmann manifold between every pair of samples in the training set - before and after the optimization. We repeated the training procedure for the two data sets: the medium and the stronger noise.

6.3. Results Evaluation

Our evaluation has two parts: We examine the separation performance on the Grassmann manifold generated from the test dataset. We then examine the classification performance of the FUME and GDRUME.

We repeat the same procedure for different numbers of points in the point cloud and for different noise statistics. We also compared our classification results to those of PointNet on point clouds with 2048 (the maximal size

PointNet enables).

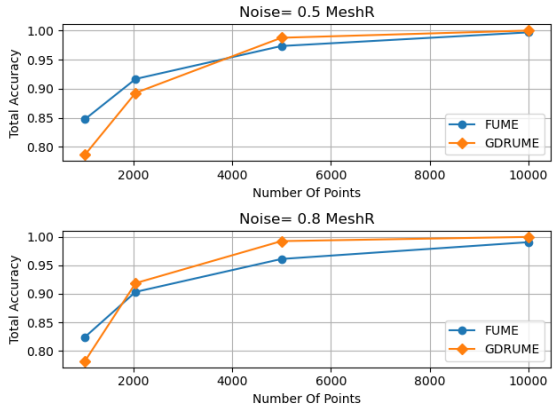

Figure 4. Classification accuracy as a function of number of points in the point cloud. Top: Medium noise level (0.5 meshR). Bottom: Stronger noise (0.8meshR). Tested point cloud sizes are $[1024, 2048, 5000, 10000]$

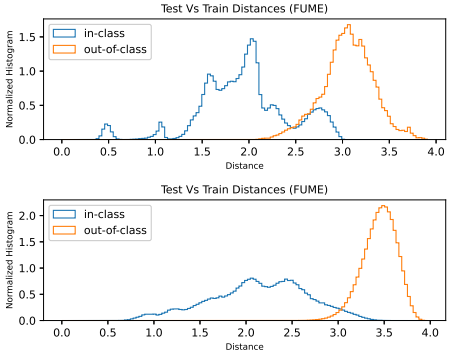

Figure 5. Distance histograms between training samples and test samples from the medium noise dataset. Upper histogram: Setting the decision threshold at the intersection point between the in-class and the out-of-class histograms yields for the initial representation by the FUME $\hat{P}(out|in) = 0.1534$ and $\hat{P}(in|out) = 0.0274$. Lower histogram: Using GDRUME optimization $\hat{P}(out|in) = 0.0355$ and $\hat{P}(in|out) = 0.0241$.

6.3.1 Separation Performance

As explained in Section 5, our aim is to map samples from the same orbit to RTUME matrices of small distance on the lower dimension Grassmann manifold while separating samples from different objects as much as possible. As shown in Figure 3, GDRUME achieves perfect separation between the in-class and out-of-class samples on the training set.

In order to evaluate the generalization capabilities of our methods, we compute the distance histograms between the observations in the test set to those in the training set. We observe from Figure 5 that compared to the FUME , the GDRUME achieves better separation between the in-class and out-of-class samples.

6.3.2 Classification Performance

In order to translate the separation methodology on the Grassmann manifold to a classification procedure, we used a nearest neighbor classifier between the test set and the training set, *i.e.*, the class of each observation in the test is determined to be the label of its nearest neighbor in the training set. We tested the classification performance under two different noise statistics, for each we also tested the effect of changing the number of points in the point cloud. The results are evaluated using the estimated accuracy computed as the ratio of the number of correct decisions to the total number of trials. The results are depicted in Figure 4.

It can be seen that both the FUME and GDRUME methods achieve high accuracy. On smaller point clouds the FUME preforms slightly better, but when the noise level increases GDRUME achieves better results. Also, for a larger number of points, the GDRUME outperforms the FUME. We also conduct a comparison between our classifiers and the PointNet classifier in the presence of stronger noise level and rigid deformations. We used a trained PointNet model and tested its performance on observations from the ModelNet40 training set after applying the mentioned deformations. Although PointNet was evaluated on samples used for its training, we conclude from Table 1 that it failed. PointNet performance is known to deteriorate in the presence of observation noise, [14], and where the observations are subject to arbitrary $SO(3)$ transformations, [18]. In our experiment we test the classification performance in the presence of both noise and arbitrary transformations, which may explain the relatively poor performance of PointNet. On the other hand GDRUME employs integral operators that are less sensitive to sampling differences and noise.

7. Conclusions

We have presented a novel approach for designing the rigid transformation universal manifold embedding of 3D point clouds towards optimizing its performance for detection and classification tasks. It has been shown that the RTUME provides a mapping from the orbit of observations on some object to a single low dimensional linear subspace of Euclidean space. This linear subspace is invariant to the geometric transformations. In the classification set-up the RTUME subspace extracted from an experimental observation is tested against a set of subspaces representing the different object manifolds, in search for the nearest class. In the presence of observation noise and random sampling patterns of the point clouds, the observations do not lie strictly on the manifold and the resulting RTUME subspaces are

noisy. The proposed method employs Grassmannian dimensionality reduction to derive an optimal structure for the universal manifold embedding, which we name GDRUME. We showed that in the presence of noise, GDRUME provides highly accurate classification results.

Appendix A - The Cost Function Euclidean Gradient

Proof. In the following we use the abbreviated notation $\mathbf{Q}_i \triangleq \mathbf{Q}_i(\mathbf{W})$, where it is understood that $\mathbf{Q}_i$ is a function of $\mathbf{W}$. Recall that $L(\mathbf{W})$ is given by (9), $\mathbf{Q}_i, \mathbf{R}_i$ are obtained by the *QR-Decomposition* of $\mathbf{Y}_i = \mathbf{W}^T\mathbf{F}_i$ and $\mathbf{G}$ is given by (10). We will use the differential notations: $dy = \mathrm{tr}(\mathbf{A}d\mathbf{W}) \iff \frac{dy}{d\mathbf{W}} = \mathbf{A}^T$, [12, 11]. Let

$$L_{i,j}(\mathbf{W}) = \frac{1}{2}\left\|\mathbf{Q}_i\mathbf{Q}_i^T - \mathbf{Q}_j\mathbf{Q}_j^T\right\|_F^2 \tag{16}$$

The differential of $L_{i,j}(\mathbf{W})$, after using the trace properties, can be written as:

$$dL_{i,j}(\mathbf{W}) = -2\mathrm{tr}\left(\mathbf{Q}_i^T\mathbf{Q}_j\mathbf{Q}_j^T d\mathbf{Q}_i + \mathbf{Q}_j^T\mathbf{Q}_i\mathbf{Q}_i^T d\mathbf{Q}_j\right) \tag{17}$$

Note that the two summands on the RHS are identical up to a change of roles between i and j, therefore we will simplify only the first one and the solution for the second is similar. The differential of $\mathbf{Y}_i$ is given by:

$$d\mathbf{Y}_i = d(\mathbf{Q}_i\mathbf{R}_i) = d\mathbf{Q}_i\mathbf{R}_i + \mathbf{Q}_i d\mathbf{R}_i \tag{18}$$

Since $\mathbf{Q}_i^T\mathbf{Q}_i = \mathbf{I}$ we have that $d(\mathbf{Q}_i^T\mathbf{Q}_i) = \mathbf{0}$, and hence

$$d\mathbf{Q}_i^T\mathbf{Q}_i = -\mathbf{Q}_i^T d\mathbf{Q}_i \tag{19}$$

i.e., $d\mathbf{Q}_i^T\mathbf{Q}_i$ is skew-symmetric. Left multiplying (18) by $\mathbf{Q}_i^T$ and right multiplying by $\mathbf{R}_i^{-1}$ we have:

$$\mathbf{Q}_i^T d\mathbf{Y}_i\mathbf{R}_i^{-1} = \mathbf{Q}_i^T d\mathbf{Q}_i + d\mathbf{R}_i\mathbf{R}_i^{-1} \tag{20}$$

Multiplying (20) by $\mathbf{Q}_i$ on the left we have

$$d\mathbf{Q}_i = d\mathbf{Y}_i\mathbf{R}_i^{-1} - \mathbf{Q}_i d\mathbf{R}_i\mathbf{R}_i^{-1} \tag{21}$$

Similarly, multiplying (20) by $\mathbf{R}_i$ on the right we have

$$d\mathbf{R}_i = \mathbf{Q}_i^T d\mathbf{Y}_i - \mathbf{Q}_i^T d\mathbf{Q}_i\mathbf{R}_i \tag{22}$$

$d\mathbf{R}_i\mathbf{R}_i^{-1}$ is upper triangular matrix as a product of two upper triangular matrices. Hence, applying the tril operator on (20), which returns the lower triangular part of a matrix, and since $d\mathbf{R}_i\mathbf{R}_i^{-1}$ is upper-triangular we get:

$$\mathrm{tril}(\mathbf{Q}_i^T d\mathbf{Y}_i\mathbf{R}_i^{-1}) = \mathrm{tril}(\mathbf{Q}_i^T d\mathbf{Q}_i) \tag{23}$$

Because $\mathbf{Q}_i^T d\mathbf{Q}_i$ is skew-symmetric and by substituting $d\mathbf{Y}_i = d\mathbf{W}^T\mathbf{F}_i$ and (21), we obtain

$$\mathbf{Q}_i^T d\mathbf{Q}_i = \left(\mathbf{Q}_i^T d\mathbf{W}^T\mathbf{F}_i\mathbf{R}_i^{-1}\right)_{\text{asym}} \tag{24}$$

where we define $(\mathbf{A})_{\text{asym}} \triangleq (\mathbf{A})_{\text{tril}} - (\mathbf{A})_{\text{tril}}^T$. Substituting (24) to (20) and left multiplying by $\mathbf{Q}_i^T$ yields:

$$d\mathbf{R}_i = \mathbf{Q}_i^T d\mathbf{W}^T\mathbf{F}_i - \left(\mathbf{Q}_i^T d\mathbf{W}^T\mathbf{F}_i\mathbf{R}_i^{-1}\right)_{\text{asym}}\mathbf{R}_i \tag{25}$$

Substituting $d\mathbf{R}_i$ into (18) and right multiplying by $\mathbf{R}_i^{-1}$ we get:

$$d\mathbf{Q}_i = \mathbf{S}_i d\mathbf{W}^T\mathbf{F}_i\mathbf{R}_i^{-1} + \mathbf{Q}_i\left(\mathbf{Q}_i^T d\mathbf{W}^T\mathbf{F}_i\mathbf{R}_i^{-1}\right)_{\text{asym}} \tag{26}$$

where $\mathbf{S}_i \triangleq \mathbf{I} - \mathbf{Q}_i\mathbf{Q}_i^T$. Substituting (26) into the first summand of (17) RHS yields

$$\begin{aligned} &2\mathrm{tr}\left(\mathbf{Q}_i^T\mathbf{Q}_j\mathbf{Q}_j^T d\mathbf{Q}_i\right) = 2\mathrm{tr}\left(\mathbf{Q}_i^T\mathbf{Q}_j\mathbf{Q}_j^T\mathbf{S}_i d\mathbf{W}^T\mathbf{F}_i\mathbf{R}_i^{-1}\right) \\ &+ 2\mathrm{tr}\left(\mathbf{Q}_i^T\mathbf{Q}_j\mathbf{Q}_j^T\mathbf{Q}_i\left(\mathbf{Q}_i^T d\mathbf{W}^T\mathbf{F}_i\mathbf{R}_i^{-1}\right)_{\text{asym}}\right) \end{aligned} \tag{27}$$

Next we will introduce some useful identities:

$$\begin{aligned} &(\mathbf{A})_{\text{bsym}} \triangleq (\mathbf{A})_{\text{tril}} - (\mathbf{A}^T)_{\text{tril}} \\ &\mathrm{tr}\left(\mathbf{A}^T(\mathbf{B})_{\text{asym}}\right) = \mathrm{tr}\left(\mathbf{A}_{\text{bsym}}^T(\mathbf{B})\right) \end{aligned} \tag{28}$$

Applying (28) to the second summand of (27) RHS we get:

$$\begin{aligned} &\mathrm{tr}\left(\mathbf{Q}_i^T\mathbf{Q}_j\mathbf{Q}_j^T\mathbf{Q}_i\left(\mathbf{Q}_i^T d\mathbf{W}^T\mathbf{F}_i\mathbf{R}_i^{-1}\right)_{\text{asym}}\right) \\ &= \mathrm{tr}\left(\left(\mathbf{Q}_i^T\mathbf{Q}_j\mathbf{Q}_j^T\mathbf{Q}_i\right)_{\text{bsym}}\mathbf{Q}_i^T d\mathbf{W}^T\mathbf{F}_i\mathbf{R}_i^{-1}\right) \end{aligned} \tag{29}$$

Since $\mathbf{Q}_i^T\mathbf{Q}_j\mathbf{Q}_j^T\mathbf{Q}_i$ is a symmetric matrix we have $\left(\mathbf{Q}_i^T\mathbf{Q}_j\mathbf{Q}_j^T\mathbf{Q}_i\right)_{\text{bsym}} = \mathbf{0}$, and hence the second summand of (27) is zero. Finally, using the last result and the trace properties, we re-evaluate (27) and substitute it into (17) to obtain

$$\begin{aligned} dL_{i,j} = &-2\mathrm{tr}\left(\mathbf{S}_i\mathbf{Q}_j\mathbf{Q}_j^T\mathbf{Q}_i\mathbf{R}_i^{-T}\mathbf{F}_i^T d\mathbf{W}\right) \\ &-2\mathrm{tr}\left(\mathbf{S}_j\mathbf{Q}_i\mathbf{Q}_i^T\mathbf{Q}_i\mathbf{R}_j^{-T}\mathbf{F}_j^T d\mathbf{W}\right) \end{aligned} \tag{30}$$

Writing $dL_{i,j}$ using derivative notations we have:

$$\frac{dL_{i,j}}{d\mathbf{W}} = -2\left[\mathbf{F}_i\mathbf{R}_i^{-1}\mathbf{Q}_i^T\mathbf{Q}_j\mathbf{Q}_j^T\mathbf{S}_i + \mathbf{F}_j\mathbf{R}_j^{-1}\mathbf{Q}_j^T\mathbf{Q}_i\mathbf{Q}_i^T\mathbf{S}_j\right]$$

The Euclidean gradient of the cost function is the weighted sum over i, j:

$$\begin{aligned} \frac{dL}{d\mathbf{W}} = &-2\sum_{i=1}^N \mathbf{F}_i\mathbf{R}_i^{-1}\mathbf{Q}_i^T\underbrace{\left(\sum_{j=1}^N \mathbf{G}_{i,j}\mathbf{Q}_j\mathbf{Q}_j^T\right)}_{\triangleq\mathbf{H}_i}\mathbf{S}_i \\ &-2\sum_{j=1}^N \mathbf{F}_j\mathbf{R}_j^{-1}\mathbf{Q}_j^T\underbrace{\left(\sum_{i=1}^N \mathbf{G}_{i,j}\mathbf{Q}_i\mathbf{Q}_i^T\right)}_{\triangleq\mathbf{B}_j}\mathbf{S}_j \end{aligned}$$

Since $\mathbf{G}$ is a symmetric matrix we have that for every i, $\mathbf{H}_i = \mathbf{B}_i$, therefore the left sum and right sum are equal, and we obtain (15). □

References

[1] Nicolas Boumal. An introduction to optimization on smooth manifolds. *Available online, Aug*, 2020. 5, 6

[2] Nicolas Boumal, Bamdev Mishra, P-A Absil, and Rodolphe Sepulchre. Manopt, a matlab toolbox for optimization on manifolds. *The Journal of Machine Learning Research*, 15(1):1455–1459, 2014. 5

[3] Alan Edelman, Tomás A Arias, and Steven T Smith. The geometry of algorithms with orthogonality constraints. *SIAM journal on Matrix Analysis and Applications*, 20(2):303–353, 1998. 3, 5

[4] Amit Efraim and Joseph M Francos. The universal manifold embedding for estimating rigid transformations of point clouds. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5157–5161. IEEE, 2019. 2

[5] Gene H Golub and Charles F Van Loan. Matrix computations. edition, 1996. 3

[6] Rami R Hagege and Joseph M Francos. Universal manifold embedding for geometrically deformed functions. *IEEE Transactions on Information Theory*, 62(6):3676–3684, 2016. 1

[7] Jihun Hamm and Daniel D Lee. Grassmann discriminant analysis: a unifying view on subspace-based learning. In *Proceedings of the 25th international conference on Machine learning*, pages 376–383, 2008. 4

[8] Xiaofei He and Partha Niyogi. Locality preserving projections. *Advances in neural information processing systems*, 16(16):153–160, 2004. 2

[9] Zhiwu Huang, Ruiping Wang, Shiguang Shan, and Xilin Chen. Projection metric learning on grassmann manifold with application to video based face recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 140–149, 2015. 4

[10] Zhiwu Huang, Jiqing Wu, and Luc Van Gool. Building deep networks on grassmann manifolds. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018. 5

[11] Catalin Ionescu, Orestis Vantzos, and Cristian Sminchisescu. Matrix backpropagation for deep networks with structured layers. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2965–2973, 2015. 8

[12] Catalin Ionescu, Orestis Vantzos, and Cristian Sminchisescu. Training deep networks with structured layers by matrix backpropagation. *arXiv preprint arXiv:1509.07838*, 2015. 8

[13] Tianci Liu, Zelin Shi, and Yunpeng Liu. Joint normalization and dimensionality reduction on grassmannian: a generalized perspective. *IEEE Signal Processing Letters*, 25(6):858–862, 2018. 4, 5

[14] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017. 5, 6, 7

[15] Ran Sharon, Joseph M Francos, and Rami R Hagege. Geometry and radiometry invariant matched manifold detection. *IEEE Transactions on Image Processing*, 26(9):4363–4377, 2017. 1, 2

[16] Steven Thomas Smith. *Geometric optimization methods for adaptive filtering*. Harvard University, 1993. 5

[17] Steven T Smith. Optimization techniques on riemannian manifolds. *Fields institute communications*, 3(3):113–135, 1994. 5

[18] Chenxi Xiao and Juan Wachs. Triangle-net: Towards robustness in point cloud learning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 826–835, 2021. 7

[19] Jiayao Zhang, Guangxu Zhu, Robert W Heath Jr, and Kaibin Huang. Grassmannian learning: Embedding geometry awareness in shallow and deep learning. *arXiv preprint arXiv:1808.02229*, 2018. 2, 4