entropy

Article

Geometric Characteristics of the Wasserstein Metric on SPD(n) and Its Applications on Data Processing

Yihao Luo [1], Shiqiang Zhang [2], Yueqi Cao [3] and Huafei Sun [1,*]

[1] School of Mathematics and Statistics, Beijing Institute of Technology, Beijing 100081, China; knowthingless@bit.edu.cn
[2] Department of Computing, Imperial College London, London SW7 2AZ, UK; s.zhang21@imperial.ac.uk
[3] Department of Mathematics, Imperial College London, London SW7 2AZ, UK; y.cao21@imperial.ac.uk
[*] Correspondence: huafeisun@bit.edu.cn

Abstract: The Wasserstein distance, especially among symmetric positive-definite matrices, has broad and deep influences on the development of artificial intelligence (AI) and other branches of computer science. In this paper, by involving the Wasserstein metric on $SPD(n)$, we obtain computationally feasible expressions for some geometric quantities, including geodesics, exponential maps, the Riemannian connection, Jacobi fields and curvatures, particularly the scalar curvature. Furthermore, we discuss the behavior of geodesics and prove that the manifold is globally geodesic convex. Finally, we design algorithms for point cloud denoising and edge detecting of a polluted image based on the Wasserstein curvature on $SPD(n)$. The experimental results show the efficiency and robustness of our curvature-based methods.

Keywords: symmetric positive-definite matrix; Wasserstein metric; curvature; point cloud denoising; image edge detecting

Citation: Luo, Y.; Zhang, S.; Cao, Y.; Sun, H. Geometric Characteristics of the Wasserstein Metric on SPD(n) and Its Applications on Data Processing. *Entropy* **2021**, *23*, 1214. https://doi.org/10.3390/e23091214

Academic Editor: Carlos M. Travieso-González

Received: 5 August 2021
Accepted: 10 September 2021
Published: 14 September 2021

1. Introduction

Symmetric positive-definite matrices have wide usage in many fields of information science, such as stability analysis of signal processing, linear stationary systems, optimal control strategies and imaging analysis [1–3]. Its importance is beyond words [4,5]. Instead of considering a single matrix, contemporary scientists tend to comprehend the global structure of the set consisting of all $n \times n$ symmetric positive-definite matrices. This set is known as $SPD(n)$. $SPD(n)$ can be endowed with various structures. The most traditional Euclidean metric is induced as submanifold metric from the Euclidean inner product on the space of matrices. X. Pennec, P. Fillard et al. [6] defined the affine-invariant Riemannian metric. V. Arsigny, P. Fillard et al. [7] showed the Lie group [8] structure on $SPD(n)$, which admits a bi-invariant metric called the Log-Euclidean metric.

Recently, by constructing a principle bundle, Y. Li, M. Wong et al. [9] and S. Zhang et al. [10] defined a new Riemannian metric on $SPD(n)$ whose geodesic distance is equivalent to the Wasserstein-2 [11,12] distance, the so-called Wasserstein metric. This distance rather than the metric has been widely used in artificial intelligence [13]. In geometry, encouragingly, T. Asuka [14] and E. Massart, P.-A. Absil [15] gave a series of expressions for geometric quantities theoretically. However, these expressions are too general or complicated to be applied. In this paper, we derive more computationally feasible expressions in a concrete case. Moreover, we give the Jacobi field and scalar curvature.

Along the blooming of data science, point cloud processing, especially denoising, plays an increasingly important role in data relevant researches and engineering. There are immense literature in point cloud denoising and widely used algorithms packed as inline functions of softwares such as PCL [16]. These methods share a common drawback when high density noise is added to point clouds. Utilizing the geometric structure of $SPD(n)$, we design a novel algorithm to overcome this drawback. Compared to traditional methods, our algorithm is more accurate and less dependent on artificial parameters.

In addition to that, we involve our theory for image edge detection, which is a classical problem in image processing and design a new detecting algorithm. Different from traditional gradient-based filters, such as Sobel, Prewitt and Laplacian, we present the connection between Wasserstein sectional curvature and edges. Experiments show the feasibility of our algorithm.

The paper is organized as follows. In Section 2, we introduce some basic knowledge of the Riemannian manifold $(SPD(n), g_W)$, and consider the symmetry of the $(SPD(n), g_W)$ as well. In Section 3, we describe the Wasserstein geometry of $SPD(n)$, including the geodesic, exponential map, connection and curvature. In particular, we prove the geodesic convexity and the nonexistence of cut locus and conjugate pair. In Section 4, we design an adaptive algorithm to denoise point clouds. In Section 5, we develop a curvature-based method to detect image edge. Proofs and detailed numerical results are presented in the Appendix B.

2. Preliminary

2.1. Notation

In this paper, we adopt conventional notations in algebra and geometry. Riemannian manifolds are denoted as pairs of $(\text{manifold}, \text{metric})$. For example, our mainly interesting object is $(SPD(n), g_W)$, meaning $SPD(n)$ endowed with Wasserstein metric. $\mathbb{R}^n$ is the n-dimensional Euclidean space. $M(n)$ represents the set of $n \times n$ matrices, $Sym(n)$ means the set of $n \times n$ symmetric matrices, and $O(n)$ means the set of $n \times n$ orthogonal matrices. $T_A M$ is conventionally the tangent space of M at a point A.

Λ always represents a diagonal $n \times n$ matrix. For an $n \times n$ matrix A, $\lambda(A)$ or $\lambda_i(A)$ means an eigenvalue or the i-th eigenvalue of A, respectively. The components of matrix A with the entries A_{ij} will always be noted as $[A_{ij}]$. The identity matrix is denoted as I. In this paper, we conventionally express points on manifolds as A, B and vector fields as X, Y.

Sylvester equation is one of the most classical matrix equations. The following special case of Sylvester equation plays a key role in understand the geometry of $(SPD(n), g_W)$

$$AK + KA = X, A \in SPD(n), X \in Sym(n). \tag{1}$$

We denote the solution about K of (1) as $\Gamma_A[X]$. Then, the matrix $\Gamma_A[X] \in M(n)$ satisfies

$$A\Gamma_A[X] + \Gamma_A[X]A = X. \tag{2}$$

From geometric aspects, we can ensure the existence and uniqueness of the solution in the case involved in this paper. Some properties of $\Gamma_A[X]$ can be found in Appendix A. More details about the Sylvester equation are presented in [17].

We recall an algorithm to solve this kind of Sylvester equations, which offers an explicit expression of the solution. This expression only depends on the eigenvalue decomposition. More details can be found in [10].

Algorithm 1 will be used frequently in the following passage, and it helps us to comprehend the geometry of $SPD(n)$. Note that this algorithm can also be used for general $X \in M(n)$. In particular, when X is symmetric (skew-symmetric), $\Gamma_A[X]$ is also symmetric (skew-symmetric). Moreover, this algorithm will be simplified if A is diagonal.

Algorithm 1 Solution of Sylvester Equation.

```
Input: $A \in SPD(n), X \in Sym(n)$
Output: $\Gamma_A[X]$
1: Eigenvalue decomposition, $A = Q\Lambda Q^T$, where $Q \in O(n), \Lambda := \mathrm{diag}(\lambda_1, \cdots, \lambda_n)$ are eigenvalues of $A$;
2: $C_X := [c_{ij}] = Q^T X Q$;
3: $E_X = [e_{ij}] = \left(\frac{c_{ij}}{\lambda_i + \lambda_j}\right)$;
4: $\Gamma_A[X] = QEQ^T$.
```

2.2. Wasserstein Metric

In this part, we introduce the Wasserstein metric on $SPD(n)$.

Definition 1. *For any $A \in SPD(n)$, $X, Y \in T_A SPD(n)$, define*

$$g_W|_A(X,Y) = \mathrm{tr}(\Gamma_A[X]A\Gamma_A[Y]) = \frac{1}{2}\mathrm{tr}(X\Gamma_A[Y]). \tag{3}$$

In the second equation, we use the facts that $\Gamma_A[X], \Gamma_A[Y], A$ are all symetric and that $A\Gamma_A[X] + \Gamma_A[X]A = X$. One can check that g_W is a symmetric and non-degenerated bilinear tensor fields on $SPD(n)$ [18]. We call g_W the Wasserstein metric.

Denote $g_E(\widetilde{X}, \widetilde{Y}) := \mathrm{tr}(\widetilde{X}^T\widetilde{Y}), \forall \widetilde{A} \in GL(n), \widetilde{X}, \widetilde{Y} \in T_{\widetilde{A}}GL(n)$ as Euclidean metric on $GL(n)$. Then, we have the following conclusions.

Proposition 1. *The projection*

$$\begin{aligned} \sigma : (GL(n), g_E) &\to (SPD(n), g_W) \\ \widetilde{A} &\mapsto \widetilde{A}^T\widetilde{A} \end{aligned} \tag{4}$$

is a Riemannian submersion [19], which means that $\mathrm{d}\sigma$ *is surjective and*

$$g_E(\widetilde{X}, \widetilde{Y}) = g_W(\mathrm{d}\sigma(\widetilde{X}), \mathrm{d}\sigma(\widetilde{Y})) = g_W(X, Y). \tag{5}$$

Remark 1. *The general linear group with Euclidean metric $(GL(n), g_E)$ and projection σ is a trivial principal bundles on $SPD(n)$ with orthogonal group $O(n)$ as the structure group. This bundle structure establishes two facts [10]: $SPD(n) \cong GL(n)/O(n)$, and g_E remains invariant under the group action of $O(n)$.*

Before giving more conclusions, we review some concepts. For any $\widetilde{A} \in GL(n)$, we say that the tangent vector $\widetilde{V} \in T_{\widetilde{A}}GL(n)$ is vertical if $\mathrm{d}\sigma(\widetilde{V}) = 0$, and $\widetilde{W} \in T_{\widetilde{A}}GL(n)$ is horizontal if $g_W|_{\widetilde{A}}(\widetilde{V}, \widetilde{W}) = 0$ for all vertical vecters $\widetilde{V}$. In addition to that, if $\mathrm{d}\sigma(\widetilde{X}) = X \in T_A SPD(n)$, we call $\widetilde{X}$ as a lift of X, where $A = \sigma(\widetilde{A})$. Using the notation $\Gamma_A[X]$, we can find the horizontal lift of $X \in T_A SPD(n)$.

Proposition 2. *For any $\widetilde{A} \in (GL(n), g_E)$, $A = \sigma(\widetilde{A})$ and any $X \in T_A SPD(n)$, there is a unique $\widetilde{X}$ to be the horizontal lift of X at $T_{\widetilde{A}}GL(n)$—that is,*

$$\widetilde{X} = \widetilde{A}\Gamma_A[X]. \tag{6}$$

Using Proposition 2, we can obtain the representations of horizontal and vertical vectors.

Proposition 3. *For any $\widetilde{A} \in (GL(n), g_E)$, $T_{\widetilde{A}}GL(n)$ has the following orthogonal decomposition*

$$T_{\widetilde{A}}GL(n) = H(\widetilde{A}) \oplus V(\widetilde{A}), \tag{7}$$

where $H(\widetilde{A})$ consists of all horizontal vectors, $V(\widetilde{A})$ consists of all vertical vectors, and

$$H(\widetilde{A}) = \{\widetilde{A}K \mid K^T = K\}, V(\widetilde{A}) = \{S\widetilde{A} \mid S^T = -S\}. \tag{8}$$

Proofs of Proposition 2 and 3 can be found in [10].

2.3. Symmetry

Now we study the symmetry of $(SPD(n), g_W)$. Consider orthogonal group action $\Psi : O(n) \times SPD(n) \to SPD(n)$ defined by

$$\Psi_O(A) = OAO^T, \forall O \in O(n), A \in SPD(n). \tag{9}$$

One can check that Ψ is a group action of $O(n)$ and that $\mathrm{d}\Psi_O$ are isometric for all $O \in O(n)$, which means that $O(n)$ is isomorphic to a subgroup of the isometry group $ISO(SPD(n), g_W)$ on $SPD(n)$. Precisely, we have

$$\{\Psi_O\}_{O\in O(n)} \lhd ISO(SPD(n), g_W). \tag{10}$$

According to (10), when we study local geometric characteristics, we only need to consider the sorted diagonal matrices as the representational elements under the orthogonal action rather than all general points on $SPD(n)$. Therefore, some pointwise quantities, such as the scalar curvature and the bounds of sectional curvatures, depend only on the eigenvalues.

At the end of this part, we give the symmetry degree of $(SPD(n), g_W)$, which is defined by the dimension of $ISO(SPD(n), g_W)$.

Proposition 4. *$(SPD(n)), g_W)$ has its symmetry degree controlled by*

$$\frac{1}{2}(n-1)n \leq \dim(ISO(SPD(n), g_W)) \leq \frac{1}{8}(n-1)n(n+1)(n+2) + 1. \tag{11}$$

Proof. The famous interval theorem [20] about isometric group shows the nonexistence of isometric groups with dimension between $\frac{m(m-1)}{2} + 1$ and $\frac{m(m+1)}{2}$, for any m-dimensional Riemannian manifold, except $m \neq 4$.

On one hand, For an n-dimensional Riemannian manifold, the dimension of isometry group achieves maximum if and only if it has constant sectional curvature. However, we will show later that $(SPD(n), g_W)$ has no constant sectional curvature, which means its symmetry degree is less than the highest.

On the other hand, equation (10) shows that the dimension of Wasserstein isometric group is higher than the dimension of $O(n)$. Therefore, by $\dim(SPD(n)) = \frac{n^2+n}{2} \neq 4$ and $\dim(O(n)) = \frac{n^2-n}{2}$, we obtain the desired result. □

3. Wasserstein Geometry of $SPD(n)$

3.1. Geodesic

In this part, we give the expression of geodesic on $(SPD(n), g_W)$, including the geodesic jointing of two points and the geodesic with initial values. Then, we will show that the whole Riemannian manifold $(SPD(n), g_W)$ is geodesic convex, which means that we can always find the minimal geodesic jointing any two points. To some extent, geodesic convexity may make up for the incompleteness of $(SPD(n), g_W)$.

To prove the geodesic convexity of $(SPD(n), g_W)$, we need the following theorem.

Theorem 1. *For any $A_1, A_2 \in SPD(n)$, let $\widetilde{A}_1 = A_1^{\frac{1}{2}}$ be the fixed lift of A_1, there exists a lift of A_2*

$$\widetilde{A}_2 = A_1^{-\frac{1}{2}}(A_1A_2)^{\frac{1}{2}}, \tag{12}$$

such that the line segment $\widetilde{\gamma}(t) = t\widetilde{A}_2 + (1-t)\widetilde{A}_1, t \in [0,1]$ is horizontal and non-degenerated.

Proof of Theorem 1 can be found in Appendix B. This theorem brings some geometrical and physical facts.

Corollary 1. *(geodesic convexity) $(SPD(n), g_W)$ is a geodesic convex Riemannian manifold. Between any two points $A_1, A_2 \in SPD(n)$, there exists a minimal Wasserstein geodesic*

$$\gamma(t) = (1-t)^2 A_1 + t(1-t)\Big((A_1A_2)^{\frac{1}{2}} + (A_2A_1)^{\frac{1}{2}}\Big) + t^2 A_2, \tag{13}$$

where $\gamma(t)$ lies on $SPD(n)$ strictly. Thus, $(SPD(n), g_W)$ is geodesic convex.

The similar expression of geodesic can also be found in several papers [14,15].

Theorem 2. *For any two points in $(SPD(n), g_W)$, there exists a unique geodesic jointing them. From geometric aspect, there is no cut locus on any geodesic.*

Proof. We have proved the existence of minimal geodesic jointing any two points in Corollary 1. Assume that the there exists two minimal geodesics jointing $A, B \in SPD(n)$. Fix $\widetilde{A} = A^{\frac{1}{2}}$ as the horizontal lift of A and lift horizontally these two geodesic, we will find two horizontal lifts of B. Denote these two lifts as $\widetilde{B}_1 = B^{\frac{1}{2}}Q_1, \widetilde{B}_2 = B^{\frac{1}{2}}Q_2, Q_1, Q_2 \in O(n)$. Then, Q_1 and Q_2 are both solutions of the following optimization problem

$$\underset{Q \in O(n)}{\arg\min}\, d_F(A^{\frac{1}{2}}, B^{\frac{1}{2}}Q). \tag{14}$$

Since the compactness of $O(n)$, this problem has a unique solution. Thus, we prove the uniqueness of minimal geodesic. □

Remark 2. *Due to the nonexistence of cut locus, there exists no conjugate pair on $(SPD(n), g_W)$.*

3.2. Exponential Map

Following Lemma A1, we can directly write down the Wasserstein logarithm on $SPD(n)$, for any $A_1 \in SPD(n)$, $\log_{A_1} : SPD(n) \to T_{A_1}SPD(n)$

$$\log_{A_1} A_2 = \mathrm{d}\sigma|_{A_1^{\frac{1}{2}}} \dot{\widetilde{\gamma}}(0) = (A_1A_2)^{\frac{1}{2}} + (A_2A_1)^{\frac{1}{2}} - 2A_1. \tag{15}$$

By solving the inverse problem of above equation, we gain the expression of the Wasserstein exponential.

Theorem 3. *In a small open ball $B(0,\varepsilon), \varepsilon > 0$ in $T_A SPD(n) \cong \mathbb{R}^{\frac{1}{2}n(n+1)}$, the Wasserstein exponential at A, $\exp_A : \mathcal{B}(0,\varepsilon) \to SPD(n)$ is explicitly expressed by*

$$\exp_A X = A + X + \Gamma_A[X] A \Gamma_A[X]. \tag{16}$$

Proof. By choosing the normal coordinates [21] at A, there always exist neighborhoods where $\exp_A$ is well-defined. From (15), given $\exp_A X$ as well-defined, this satisfies

$$(A \exp_A X)^{\frac{1}{2}} + (\exp_A XA)^{\frac{1}{2}} = X + 2A. \tag{17}$$

This equation can convert to the Sylvester equation, and we can express its solution as

$$\begin{aligned}
&AA^{-1}(A\exp_A X)^{\frac{1}{2}} + (\exp_A XA)^{\frac{1}{2}}A^{-1}A = X + 2A \\
\Leftrightarrow\ &A(A^{-1}(A\exp_A X)^{\frac{1}{2}}) + (A^{-1}(A\exp_A X)^{\frac{1}{2}})A = X + 2A \\
\Leftrightarrow\ &A\Gamma_A[X + 2A] = (A\exp_A X)^{\frac{1}{2}}.
\end{aligned} \tag{18}$$

Therefore, we have

$$\exp_A X = \Gamma_A[X + 2A]A\Gamma_A[X + 2A] = A + X + \Gamma_A[X]A\Gamma_A[X], \tag{19}$$

which finishes this proof. □

Remark 3. *We call the first two terms of right hand $A + X$ as the Euclidean exponential, and the last term of right hand $\Gamma_A[X]A\Gamma_A[X]$ as the Wasserstein correction for this bend manifold.*

Corollary 2. *The geodesic equations with initial conditions $\gamma(0), \dot{\gamma}(0)$ on $(SPD(n), g_W)$ have the following explicit solution*

$$\gamma(t) = \gamma(0) + t\dot{\gamma}(0) + t^2\Gamma_{\gamma(0)}[\dot{\gamma}(0)]\gamma(0)\Gamma_{\gamma(0)}[\dot{\gamma}(0)], t \in (-\varepsilon, \varepsilon). \tag{20}$$

Using an exponential map, one can directly construct Jacobi fields with the geodesic variation.

Theorem 4. *Along a geodesic $\gamma(t)$ with $\gamma(0) = A \in SPD(n), \dot{\gamma}(0) = X \in T_A SPD(n)$, there exists a unique normal Jacobi vector field $J(t)$ with initial conditions $J(0) = 0, \nabla_{\dot{\gamma}(0)} J(t) = Y \in T_A SPD(n)$, where $g_W|_A\langle X, Y\rangle = 0$. We have*

$$J(t) = tY + t^2(\Gamma_A[X]A\Gamma_A[Y] + \Gamma_A[Y]A\Gamma_A[X]). \tag{21}$$

As in [18], $J(t)$ is constructed by

$$J(t) := \frac{\partial}{\partial s}\bigg|_{s=0} \exp_A t(X + sY), \tag{22}$$

substituting (16) into (22), and Theorem 4 comes from direct computation.

Subsequently, the next natural question is what is the maximal length of the extension of a geodesic. This question is equivalent to what is the largest domain of the exponential. We still focus on diagonal matrices.

Theorem 5. *For any $A \in SPD(n)$ and $X \in T_A SPD(n)$, $\exp_A(tX) : [0, \varepsilon) \to SPD(n)$ is well-defined if and only if*

$$\varepsilon_{\max} = \begin{cases} -\frac{1}{\lambda_{\min}}, & \text{if } \lambda_{\min} < 0 \\ +\infty, & \text{if } \lambda_{\min} \geq 0 \end{cases}, \tag{23}$$

where $\lambda_{\min}$ is the minimal eigenvalue of $\Gamma_A[X]$.

Proof. Evidently, $\varepsilon_{\max} = \min\{s > 0 \mid \det(\exp_A(sX)) = 0\}$. By (19), we have

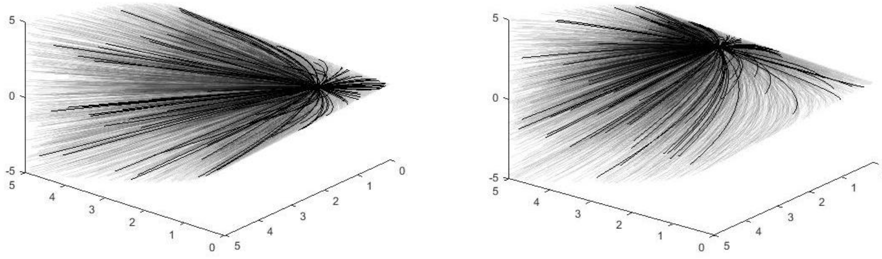
$$\det(\exp_A(sX)) = \det(A)\det^2(I + s\Gamma_A[X]) = 0 \Leftrightarrow s = -\frac{1}{\lambda(\Gamma_A[X])}, \tag{24}$$

where $\lambda(\Gamma_A[X])$ is the eigenvalue of $\Gamma_A[X]$. Thus, $\varepsilon_{\max} = \min\left\{-\frac{1}{\lambda(\Gamma_A[X])} > 0\right\}$. □

Corollary 3. *The Wasserstein metric g_W on $SPD(n)$ is incomplete.*

Corollary 3 can be directly obtained from Hopf–Rinow theorem [22]. Theorem 5 and the next theorem help us to comprehend the size of $(SPD(n), g_W)$ from sense of each point.

Figure 1 shows geodesics starting from different origins on $SPD(2)$. From this group of pictures, we can observe the outline of the manifold and some behaviors of geodesics.



(**a**) geodesics starting from [1,0;0,1]

(**b**) geodesics starting from [1,0;0,1]

Figure 1. Geodesics of g_W on $SPD(2)$.

Using $\varepsilon_{\max}$, we can obtain the injectivity radius $r(A), \forall A \in SPD(n)$. Geometrically speaking, $r(A)$ is the maximal radius of the ball in which $\exp_A$ is well-defined.

Theorem 6. *The Wasserstein radius $r(A) : SPD(n) \to (0, +\infty)$ can be given by*

$$r(A) = \sqrt{\lambda_{\min}(A)}, \tag{25}$$

and the function $r(A)$ is continuous.

Proof of Theorem 6 can be found in Appendix C. Due to the geodesic convexity, the radius actually defines the Wasserstein distance of a point on $SPD(n)$ to the 'boundary' of the manifold. It also measures the degenerated degree of a positive-definite symmetric matrix by $\sqrt{\lambda_{\min}}$.

Figure 2 shows three maximal geodisical balls with different centers on $SPD(2)$. From the viewpoint of $\mathbb{R}^3$, the three balls have different sizes in the sense of Euclidean distance, but on $(SPD(2), g_W)$, all of them have the same radius.

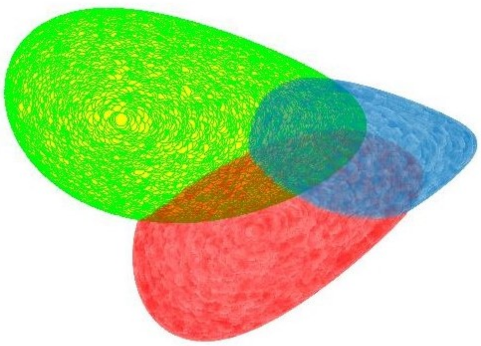


Figure 2. Three geodisical balls with the same radius.

3.3. Connection

In this section, we will study the Riemannian connection of $(SPD(n), g_W)$, called a Wasserstein connection. The flatness of $(GL(n), g_E)$ and the structure of the Riemannian submersion will take a series of convenience to our work. During computation, we denote both tensor actions of g_W on $SPD(n)$ and g_E on $GL(n)$ by $\langle\cdot,\cdot\rangle$. Then, we denote the Euclidean connection as D and the Wasserstein connection as ∇.

The main idea to express the Wasserstein connection is to compute the horizontal term of the Euclidean covariant derivative of lifted vector fields. We shall prove:

Lemma 1. *The Euclidean connection is a lift of the Wasserstein connection. For any smooth vector fields X and Y on $SPD(n)$, and $\widetilde{X}$ and $\widetilde{Y}$ are their horizontal lifts, respectively, the following equation holds*

$$\mathrm{d}\sigma|_{\widetilde{A}}(D_{\widetilde{X}}\widetilde{Y}) = \nabla_X Y, \forall \widetilde{A} \in GL(n). \tag{26}$$

Proof of Lemma 1 can be found in Appendix D. This lemma holds for general Riemannian submersion. The reason we reprove it for the case is that we will need use some middle results of the proof later. Using Lemma 1, we can find a direct corollary, which is one of the essential results in this paper.

Corollary 4. *The Wasserstein connection has an explicit expression:*

$$\nabla_X Y = \mathrm{d}Y(X) - \Gamma_A[X]A\Gamma_A[Y] - \Gamma_A[Y]A\Gamma_A[X], \tag{27}$$

where $\mathrm{d}Y(X)$ is a Euclidean directional derivative.

Proof. From Lemma 1 and (A14) (in Appendix D), we have

$$\begin{aligned}\nabla_X Y = \mathrm{d}\sigma|_{\widetilde{A}}(D_{\widetilde{X}}\widetilde{Y}) &= \widetilde{A}^T D_{\widetilde{X}}\widetilde{Y} + (D_{\widetilde{X}}\widetilde{Y})^T\widetilde{A} \\ &= \mathrm{d}Y(X) - (X\Gamma_A[Y] + \Gamma_A[Y]X) + A\Gamma_A[X]\Gamma_A[Y] + \Gamma_A[Y]\Gamma_A[X]A \\ &= \mathrm{d}Y(X) - \Gamma_A[X]A\Gamma_A[Y] - \Gamma_A[Y]A\Gamma_A[X].\end{aligned} \tag{28}$$

The linearity, Leibnitz's law and symmetry of Wasserstein connection are easily checked from the expression. □

The vertical component of lifted covariant derivative of Y along X is a vector field in $GL(n)$ whose value at $\widetilde{A}$ is defined by

$$\mathcal{T}_{\widetilde{A}}(X,Y) := D_{\widetilde{X}}\widetilde{Y} - \widetilde{\nabla_X Y}. \tag{29}$$

We say $\mathcal{T}_{\widetilde{A}}$ is an $\mathcal{A}$-tensor. The whole vector field is denoted as $\mathcal{T}(X,Y)$. By definition and previous results, we can obtain the expression of $\mathcal{T}_{\widetilde{A}}(X,Y)$.

Proposition 5. *$\mathcal{T}_{\widetilde{A}}(\cdot,\cdot)$ is a antisymmetric bilinear map: $T_A SPD(n) \otimes T_A SPD(n) \to T_{\widetilde{A}} GL(n)$, and it satisfies*

$$\mathcal{T}_{\widetilde{A}}(X,Y) = \widetilde{A}\Gamma_A[\Pi[X,Y]]A. \tag{30}$$

Proof. Using (A12) (in Appendix D), (6) and (27), we have

$$\begin{aligned}\mathcal{T}_{\widetilde{A}}(X,Y) = \ & \widetilde{A}(\Gamma_A[\mathrm{d}Y(X)] - \Gamma_A[X\Gamma_A[Y] + \Gamma_A[Y]X] + \Gamma_A[X]\Gamma_A[Y]) \\ & - \widetilde{A}\Gamma_A[\mathrm{d}Y(X) - \Gamma_A[X]A\Gamma_A[Y] - \Gamma_A[Y]A\Gamma_A[X]] \\ = \ & \widetilde{A}(-\Gamma_A[A\Gamma_A[X]\Gamma_A[Y]] - \Gamma_A[\Gamma_A[Y]\Gamma_A[X]A] + \Gamma_A[X]\Gamma_A[Y]) \\ = \ & \widetilde{A}(\Gamma_A[X]\Gamma_A[Y] - A\Gamma_A[\Gamma_A[X]\Gamma_A[Y]] - \Gamma_A[\Gamma_A[Y]\Gamma_A[X]]A) \\ = \ & \widetilde{A}\Gamma_A[\Gamma_A[X]\Gamma_A[Y] - \Gamma_A[Y]\Gamma_A[X]]A \\ = \ & \widetilde{A}\Gamma_A[\Pi[X,Y]]A,\end{aligned} \tag{31}$$

where (31) shows that $\mathcal{T}_{\widetilde{A}}(X,Y)$ depends only on $\widetilde{A}$ and the vectors on $T_A SPD(n)$. The multi-linearity and $\mathcal{T}_{\widetilde{A}}(X,Y) = -\mathcal{T}_{\widetilde{A}}(Y,X)$ are easily checked. □

Recalling (A14) in Appendix D, we also find that

$$[\widetilde{X},\widetilde{Y}] = \widetilde{[X,Y]} + 2\mathcal{T}(X,Y). \tag{32}$$

In the following parts, we will show the tensor $\mathcal{T}(X,Y)$ plays a significant role for computing curvature.

3.4. Curvature

In this part, we tend to understand the curvature of $(SPD(n), g_W)$. Although there exists some relevant results giving abstract expressions for general cases, we obtain simpler expressions and derive the scalar curvature via a special basis.

3.4.1. Riemannian Curvature Tensor

First, we derive the Riemannian curvature of $(SPD(n), g_W)$. We denote the Euclidean curvature on bundle (null entirely) as $\widetilde{R}$, and the Wasserstein (Riemannian) curvature on $(SPD(n), g_W)$ as R.

Theorem 7. *For any $A \in SPD(n)$, and X, Y are smooth vector fields on $SPD(n)$, the Wasserstein curvature tensor $R(X,Y,X,Y) := \langle R_{XY}X, Y\rangle_A$ at A has an explicit expression*

$$R(X,Y,X,Y) = 3\mathrm{tr}(\Gamma_A[X]A\Gamma_A[\Gamma_A[X]\Gamma_A[Y] - \Gamma_A[Y]\Gamma_A[X]]A\Gamma_A[Y]). \tag{33}$$

Proof of Theorem 7 can be found in Appendix E. The expression

$$R(X,Y,X,Y) = \|\mathcal{T}(X,Y)\|^2 \tag{34}$$

has been derived before by other research group in similar way. However, here we use another way to calculate curvature tensor and find a more explicit expression, which is easier than expanding $\|\mathcal{T}(X,Y)\|^2$ directly. In addition to that, from $\|\mathcal{T}(X,Y)\|^2 \geq 0$ and (A23) (in Appendix E), we can obtain the following corollary.

Corollary 5. *$(SPD(n), g_W)$ has non-negative curvatures, namely*

$$R(X,Y,X,Y) \geq 0. \tag{35}$$

By solving the Sylvester equation with Algorithm 1, we can simplify the expression. We give the sectional curvature K of the section $\mathrm{span}\{X(A), Y(A)\}$

$$K|_A(X,Y) = \frac{R(X,Y,X,Y)}{\langle X,X\rangle\langle Y,Y\rangle - \langle X,Y\rangle^2} = 12\frac{\mathrm{tr}(E_X\Lambda\Gamma_\Lambda[E_X,E_Y]\Lambda E_Y)}{\mathrm{tr}(E_XC_X)\mathrm{tr}(E_YC_Y) - \mathrm{tr}^2(E_XC_Y)}, \tag{36}$$

where we use the same donations as Algorithm 1. In particular, in diagonal cases, we obverse that the sectional curvature conforms to the inverse ratio law

$$K|_{k\Lambda}(X,Y) = \frac{1}{k}K|_{\Lambda}(X,Y), \forall k \in \mathbb{R} - \{0\}. \tag{37}$$

These results conform with our visualized views of $(SPD(n), g_W)$, as presented in Figure 1, where the manifold tends to be flat when k increases.

3.4.2. Sectional Curvature

Now, we derive more explicit expressions for sectional curvature and scalar curvature. Conventionally, we only need to consider diagonal cases. Before that, we introduce a basis on $Sym(n)$, which is the tangent space of $SPD(n)$. Define $\{S^{p,q}\}$ as

$$S^{p,q} = [S^{p,q}_{ij}], S^{p,q}_{ij} = \delta^p_i \delta^q_j + \delta^q_i \delta^p_j, \tag{38}$$

where the superscripts p, q marks the nonzero elements in $S^{p,q}$ and δ is the Kronecker delta. Apparently, $\{S^{p,q} \mid 1 \le p \le q \le n\}$ forms a basis of $Sym(n)$. For simplicity, we sometimes sign $S^{p,q}, S^{r,t}$ with S_1, S_2, respectively. In this way, we can express the curvature under this basis.

By direct calculation, we have

$$(S_1S_2)_{ij} = \sum_{k=1}^{n} S^{p,q}_{ik} S^{r,t}_{kj} = \delta^p_i \delta^{qt} \delta^r_j + \delta^p_i \delta^{qr} \delta^t_j + \delta^q_i \delta^{pt} \delta^r_j + \delta^q_i \delta^{pr} \delta^t_j. \tag{39}$$

By Algorithm 1, we know that $E_S = \left(\frac{S_{ij}}{\lambda_i + \lambda j}\right) = \Gamma_\Lambda S, \forall S \in T_\Lambda SPD(n) (Q = I$ in the decomposition of Λ). Note that the elements of Λ, S_1, S_2 are all positive; therefore, we have

$$E_{S_1}E_{S_2} \neq 0 \Longleftrightarrow S_1S_2 \neq 0 \Longleftrightarrow \{p,q\} \cap \{r,t\} \neq \emptyset. \tag{40}$$

According to the anti-symmetry of curvature tensor, the non-vanishing curvature means that $\{p,q\} \neq \{r,t\}$. Moreover, by definition we know $S^{p,q} = S^{q,p}, S^{r,t} = S^{t,r}$. Without loss of generality, we only need to consider the following particular case:

$$p = r, q \neq t. \tag{41}$$

Theorem 8. *For any diagonal matrix $\Lambda = \mathrm{diag}(\lambda_1, \cdots, \lambda_n) \in SPD(n)$, where $\lambda_1 \le \lambda_2 \le \cdots \le \lambda_n$, Wasserstein sectional curvature satisfies*

$$K|_\Lambda(S_1, S_2) = \frac{3(1+\delta^{pq})(1+\delta^{pt})\lambda_q\lambda_t}{(\lambda_p+\lambda_q)(\lambda_p+\lambda_t)(\lambda_q+\lambda_t)}, \tag{42}$$

where $S_1 = S^{p,q}, S_2 = S^{r,t}, p = r, q \neq t$.

Proof of Theorem 8 can be found in Appendix F. With the above expansion for sectional curvatures, we can easily find that sectional curvature can be controlled by the secondly minimal eigenvalue, which implies that the curvature will seldom explode even on a domain almost degenerated. Only when the matrices degenerate at over two dimensions will the curvatures be very large. This phenomenon ensures the curvature information makes sense in most applications. Some examples for this phenomenon can be observed later.

3.4.3. Scalar Curvature

In the last part of this section, we calculate the scalar curvature directly.

Theorem 9. *For any $A \in SPD(A)$, its scalar curvature $\rho(A)$ is*

$$\rho(A) = 3\mathrm{tr}(U\Lambda(U+U^T) + (U+U^T)\Lambda U + (U+U^T)\Lambda U\Lambda(U+U^T)), \tag{43}$$

where the diagonal matrix $\Lambda = \mathrm{diag}(\lambda_1, \cdots, \lambda_n)$ is orthogonal similar to A, and $U = \left(\frac{1}{\lambda_i+\lambda_j}\right)_{i<j}$.

Proof of Theorem 9 can be found in Appendix G. Figure 3 presents some examples for scalar curvatures on $(SPD(2), g_W)$, which shows our argument in the last part of Section 3.4.2.

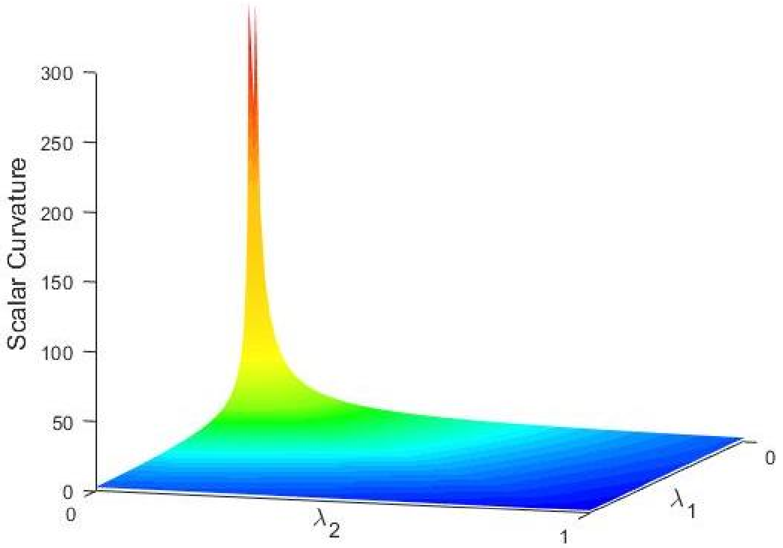


Figure 3. Scalar curvatures on *SPD*(2).

4. Point Cloud Denoising

Denoising or outlier removal is a fundamental step of point cloud preprocessing since real-world data are often polluted by noise. There are immense literature in point cloud denoising and widely used algorithms packed as inline functions of softwares. For example, PCL [16] is a popular platform for point cloud processing, which collects four denoising schemes. However, these methods fail to give satisfactory performance when point clouds are polluted by high density noise. To solve this problem, we consider both the statistical and geometrical structure of data and design a new algorithm.

The idea is that by embedding the original point cloud from Euclidean space into $SPD(n)$, the Wasserstein scalar curvature gives essential information about noise and true data. Therefore, our new algorithm mainly contains two steps: First, we give the desired embedding by fitting a Gaussian distribution locally at each point. Then, we identify noise by looking at the histogram of the Wasserstein scalar curvature. Due to the flatness of the space of noise, it is reasonable to classify points with small curvature to be noise. The threshold is set to be the first local minimum of the histogram. We call this new scheme adaptive Wasserstein curvature denoising (AWCD).

In the following, we introduce two traditional denoising methods called radius outlier removal (ROR) and statistical outlier removal (SOR). Then, we explain details about AWCD. Additionally, we carry out experiments using different datasets, with a comparison to two classical methods. From the experimental results, AWCD presents better performance regardless of the data size and the density of noise. We also give a time complexity analysis for each denoising algorithm. The results show that AWCD is as efficient as other classical methods. Thus, it is applicable in many practical tasks.

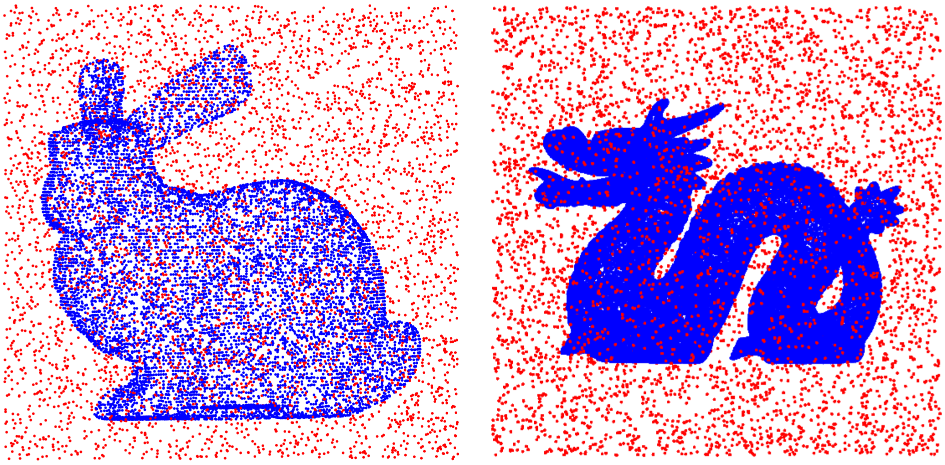
4.1. Radius Outlier Removal

In Radius Outlier Removal, called ROR (seeing Algorithm 2), points are clustered into two categories according to their local density, i.e., points with low density tend to be recognized as noise, whereas points with high density are recognized as true data. ROR requires two parameters: a preset parameter d as the radius for local neighborhoods and α as the least number of points in each neighborhood.

Algorithm 2 Radius Outlier Removal.

```
Input: initial point cloud $D_0$, parameters $d$, $\alpha$
Output: cleaned point cloud $D_1$
1: search $d$-radius neighborhood $N_i$ for each point $P_i$, where
       $N_i = \{N_{ij} \in D_0 | \|N_{ij} - P_i\| \leq d\};$
2: if number of neighbors $|N_i| \geq \alpha$ then put $P_i$ into $D_1$;
3: return $D_1$.
```

As an illustration, we add uniform noise to the Stanford Bunny with 10,000 points (see Figure 4).

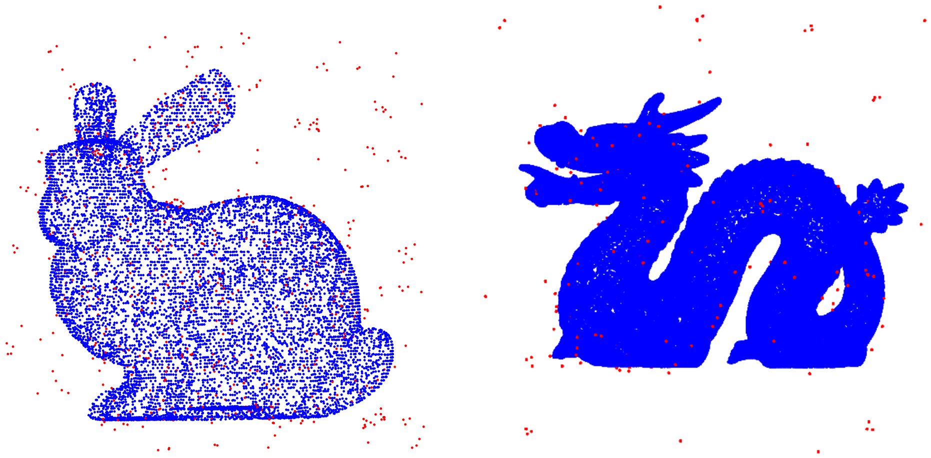


(**a**) Stanford Bunny (**b**) Duke Dragon

Figure 4. Stanford Bunny with uniform noise.

Then, we apply ROR to denoise the polluted point cloud. The result is shown in Figure 5. From a visual observation, ROR preserves almost all true points but fails to recognize a small portion of noise at any area.

In fact, from a series of repetitive experiments we find that ROR is sensitive to the choice of manual parameters. A small radius will make ROR inefficient, while a large radius will wrongly recognize true points as noise. One of the disadvantages of ROR is that there exists no universal method to determine the best parameters. Further, since ROR uses the kernel method to find the undetermined closest neighbors, the time complexity can reach to $O(n^2)$ where n is the number of points. Thus, in practice, it is often difficult to make a trade-off between efficiency and effect of ROR.



(**a**) Stanford Bunny (**b**) Duke Dragon

Figure 5. Cleaned point cloud by ROR.

4.2. Statistical Outlier Removal

Compared to ROR, Statistical Outlier Removal (SOR) considers more detailed local structures than density does. SOR showed in Algorithm 3 is one of the most popular methods to preprocess point clouds due to its efficiency when dealing with low density noise. However, SOR gives worse performance than ROR when the noise is of high density. The main idea of SOR comes from one-sigma law from classical statistics [23]. An outlier is believed to be far from the center of its k-nearest neighborhood. Conversely, a true point should lie in a confidence area of its neighborhoods. Let Φ be a d-variate Gaussian distribution with expectation μ and covariance Σ, and let P be a fixed point in $\mathbb{R}^d$. Then, Φ induces a Gaussian distribution on the line $\{\mu + tv_P | t \in \mathbb{R}\}$ where $v_P = \frac{P-\mu}{\|P-\mu\|}$. In fact, we write the eigendecomposition of Σ as $\Sigma = E\text{diag}(\sigma_1^2, \cdots, \sigma_d^2)E^T$, where $E = [e_1, \cdots, e_d]$ is an orthogonal matrix. If we write

$$v_P = \sum_{i=1}^{d} \lambda_i e_i, \tag{44}$$

the projected Gaussian distribution in direction v_P has null expectation and variance $\sum_{i=1}^{d} \lambda_i^2 \sigma_i^2$. According to one-sigma law, we say P is in the confidence area of Φ if

$$\|P-\mu\|^2 \leq \sum_{i=1}^{d} \lambda_i^2 \sigma_i^2, \tag{45}$$

which is equivalent to

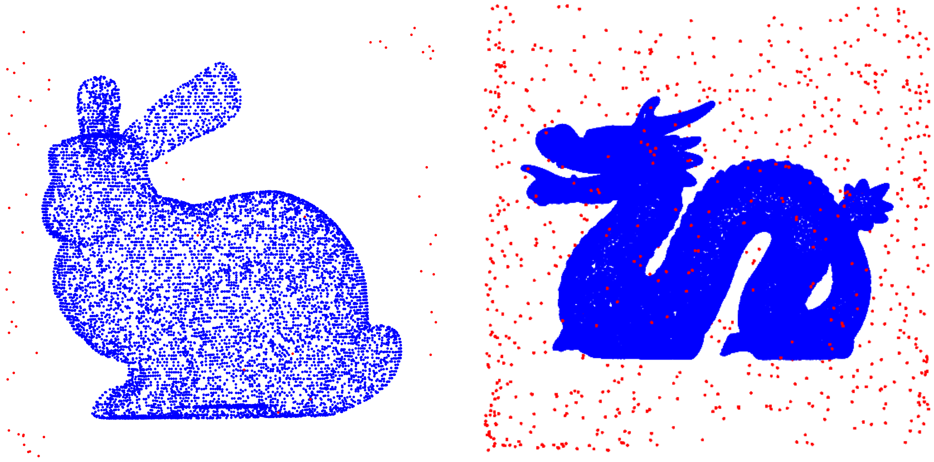
$$(P-\mu)^T \Sigma (P-\mu) \geq \|P-\mu\|^4. \tag{46}$$

This inequality is a generalization of one-sigma law in high dimensions.

Algorithm 3 Statistical Outlier Removal.

```
Input: initial point cloud $D_0$, parameter $k$
Output: cleaned point cloud $D_1$
search kNN $N_i$ for each point $P_i$;
compute local mean and local covariance
    $$\mu_i = \frac{1}{k}\sum_{j=1}^{k} N_{ij},\ \Sigma_i = \frac{1}{k-1}\sum_{j=1}^{k}(N_{ij}-\mu_i)^T(N_{ij}-\mu_i);$$
if $(P_i-\mu_i)^T\Sigma_i(P_i-\mu_i) \geq \|P_i-\mu_i\|^4$ then put $P_i$ into $D_1$;
return $D_1$.
```

Thus, SOR consists of three steps: first we search the k-nearest neighbors (kNN) for every point. Then, we compute the empirical mean and covariance under the assumption of Gaussian distribution for each neighborhood. Finally, true points are identified using (46). SOR requires a single parameter k for kNN. Again, as an illustration, we use the data in Figure 4. After SOR, the result is shown in Figure 6.



(**a**) Stanford Bunny (**b**) Duke Dragon

Figure 6. Cleaned point cloud by SOR.

We use KD-tree in kNN search. Thus, the time complexity is known as $O(kn\log n)$ where k is the number of neighbors and n is the number of points. The remaining steps are finished in $O(n)$ time. Therefore, the total time complexity is $O(kn\log n)$.

4.3. Adaptive Wasserstein Curvature Denoising

Note that the key step in SOR is to compute the local covariance, which is a positive-definite matrix. Motivated by the idea of SOR, we extract the covariance matrix at each point, which is equivalent to embed the original point cloud into $SPD(n)$. From an intuitive perspective, since the true data presents a particular pattern, the covariance matrices should have a large Wasserstein curvature. Conversely, for noise, the covariance matrices form a flat region. Hence, AWCD is based on a principal hypothesis that the Wasserstein curvature of true data is larger than noise.

Under such a hypothesis, what we need to do is to set a threshold to pick out points with a small Wasserstein curvature. To do so, we gather all information in a histogram counting the number of points of different curvature. By the continuity of curvature function, true data and noise will form two different 'hills'. Figure 7 shows an example for the histogram.

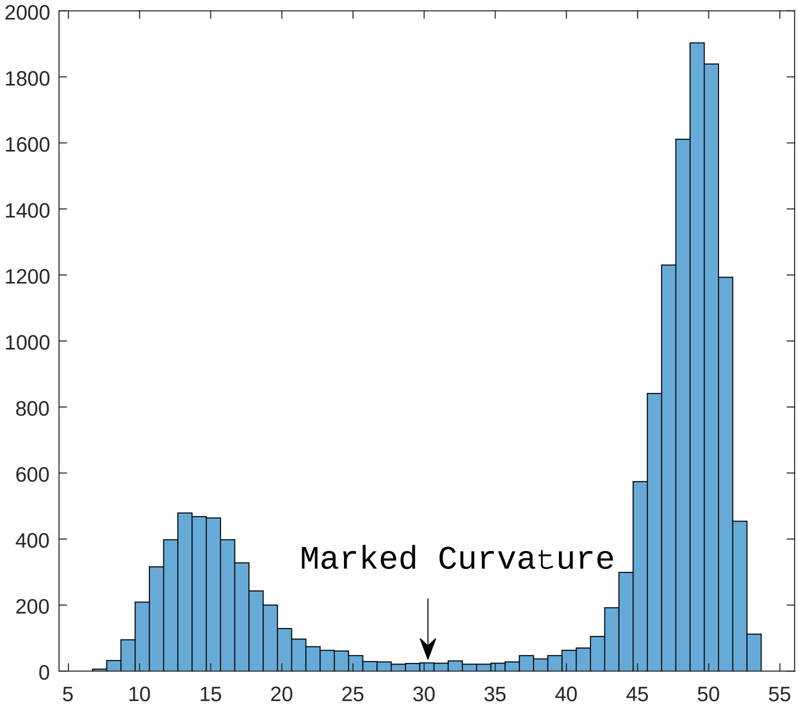


Figure 7. Histogram of the Wasserstein scalar curvature; x-axis: scalar curvature and y-axis: point number.

The phase change happens at the borderline of two hills, i.e., we seek to find the second minimal value of the histogram. In Figure 7, the critical value is annotated as 'marked curvature'. In this way, we do not need to set the threshold manually and, instead, achieve an adaptive selection process. Algorithm 4 shows the processing of this adaptive denoising via wasserstein curvature.

Algorithm 4 Adaptive Wasserstein Curvature Denoising.

```
Input: initial point cloud $D_0$, parameter $k$
Output: cleaned point cloud $D_1$
search kNN neighbors $N_i$ for each point $P_i$;
compute local mean and local covariance as before;
compute Wasserstein curvature $\rho(\Sigma)$ as (43);
construct curvature histogram and determine the marked curvature $\rho_0$;
if $\rho(\Sigma_i) \geq \rho_0$ then put $P_i$ into $D_1$;
return $D_1$.
```

We use the same example as in Figure 4. The performance of AWCD is shown in Figure 8.

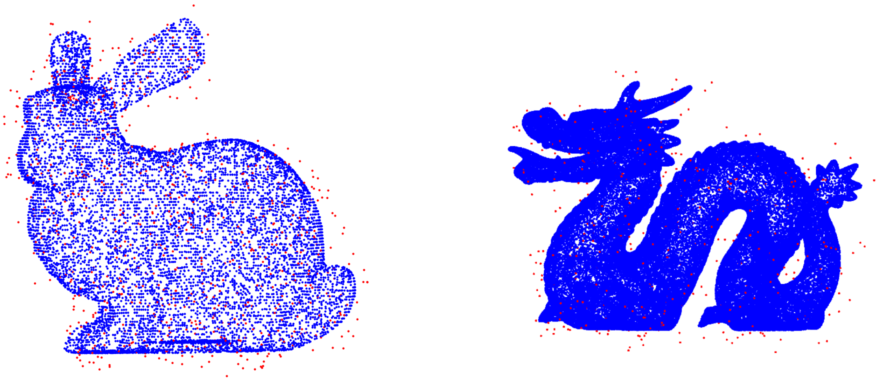

(**a**) Stanford Bunny (**b**) Duke Dragon

Figure 8. Cleaned point cloud by AWCD.

In this example, AWCD removes almost all noise far from Stanford Bunny, and remains almost all true data. The only problem is that a small portion of noise lying on the dragon cause the false positiveness and some true data located on the flat part are wrongly removed.

Since the main step in AWCD is also kNN, the time complexity is the same as SOR, which is $O(kn \log n)$. Therefore, AWCD is applicable in practice. It is remarkable that AWCD is effective for data with dense noise and robust to the unique parameter k.

4.4. Experiments

We use ROR, SOR and AWCD to denoise polluted data sets with noise of different levels of densities. The point clouds are from the Stanford 3D scanning repository, including Stanford Bunny, Duke Dragon, Armadillo, Lucy and Happy Buddha. For each data set, we add noise and record its signal-noise ratio (SNR). To show the influence of data size, we downsample the original data sets of different scales.

We adopt three criteria to measure the performance of the algorithms, including true positive rate (TPR), false positive rate (FPR) and signal-noise rate growing (SNRG). TPR describes the accuracy to preserve true points from unpolluted data sets. FPR describes the success rate to remove noisy points. SNRG explicates the promotion of SNR after processing. For any polluted point cloud $D_0 = D \cup N$, where D is the points set of true data and N is the set of noise. We obtain the cleaned point cloud D_1 after the denoising algorithms. Then, the computation of these measurements are

$$\begin{aligned} \mathrm{TPR} &= \frac{|D_1 \cap D|}{|D|}, \\ \mathrm{FPR} &= 1 - \frac{|D_1 \cap N|}{|N|}, \\ \mathrm{SNRG} &= \frac{|D_1 \cap D|}{|D_1 \cap N|} \cdot \frac{|N|}{|D|} - 1, \end{aligned} \tag{47}$$

where $|\cdot|$ denotes the cardinality or size of a finite set. Intuitively, higher TPR, SNRG and lower FPR mean better performance of an algorithm. The experimental results are shown in Table A1 in Appendix I. In each experiment, we highlight the lowest FPR, the highest TPR and the SNRG over 99%. Table A1 shows the superiority of AWCD to ROR and SOR. In general, AWCD can remove almost all noise and meanwhile preserves the true data, except for Armadillo.

5. Edge Detection

In this part, we attempt to apply the Wasserstein curvatures to detect the edges of images with noises. This application follows the idea that the edge parts contain more local information while the relatively flat parts tend to be regarded locally as white noise. Hence, the Wasserstein curvatures have natural advantages to depict the local information. This leads to the following Wasserstein sectional curvature edge detection (WSCED) of Algorithm 5.

Algorithm 5 Wasserstein sectional curvature edge detection.

```
Input: initial grayscale $F_0$ with pixels of $n \times m$, parameter $k$
Output: edge figure $F_e$
1: search kNN $N_{ij}$ for each point $P_{ij}$;
2: compute every local covariance $\Sigma_{ij}$ to obtain the covariance image $CI$, which is a $(n-2k) \times (m-2k)$ matrix constructed by matrixes $\Sigma_{ij}$;
3: determine the section $\sigma_{ij} := X_{ij} \wedge Y_{ij}$ for every point $\Sigma_{ij}$ on $CI$ by computing tangent vectors $X_{ij} = (\Sigma_{(i+1)j} - \Sigma_{(i-1)j}), Y_{ij} = (\Sigma_{i(j+1)} - \Sigma_{i(j-1)})$;
4: compute the Wasserstein sectional curvature for every $Kw|_{\Sigma_{ij}}(\sigma_{ij})$ with (36) to obtain the curvature image $F_e$, which is a $(n-2k) \times (m-2k)$ real matrix;
5: return $F_e$.
```

Similar to what we have done in the last section, the first step of WSCED is computing the local mean and variance after kNN, which can be regarded as a two-dimensional embedding from a image into $SPD(n)$. Every pixel coordinate (i, j) determines a local covariance matrix Σ_{ij}. In the second step, we compute the sectional curvature for every Σ_{ij}. The chosen section $X_{ij} \wedge Y_{ij}$ is determined by two difference vectors along x-axis and y-axis,

$$\begin{aligned} X_{ij} &:= \Sigma_{(i+1)j} - \Sigma_{(i-1)j}, \\ Y_{ij} &:= \Sigma_{i(j+1)} - \Sigma_{i(j-1)}. \end{aligned} \tag{48}$$

According to (36), we obtain the chosen curvature $KI_{ij} = K_w|_{\Sigma_{ij}}(X_{ij}, Y_{ij})$ on Σ_{ij}. Then, we obtain a curvature image KI. Finally, with some appropriate image transformation, we can detect edges on KI.

In simulations, we compare WSCED to traditional edge detecting filters, including Sobel, Prewitt and Laplacian [24]. We tend to detect edges for images with noises in high density. From Figure 9, we find that WSCED approaches the same outcome as Sobel and Prewitt filters, which implies the potential connection between Wasserstein curvature and edges. This result also shows the robustness of WSCED to noises. We present more effects of digital experiments in Figure A1 in Appendix H.

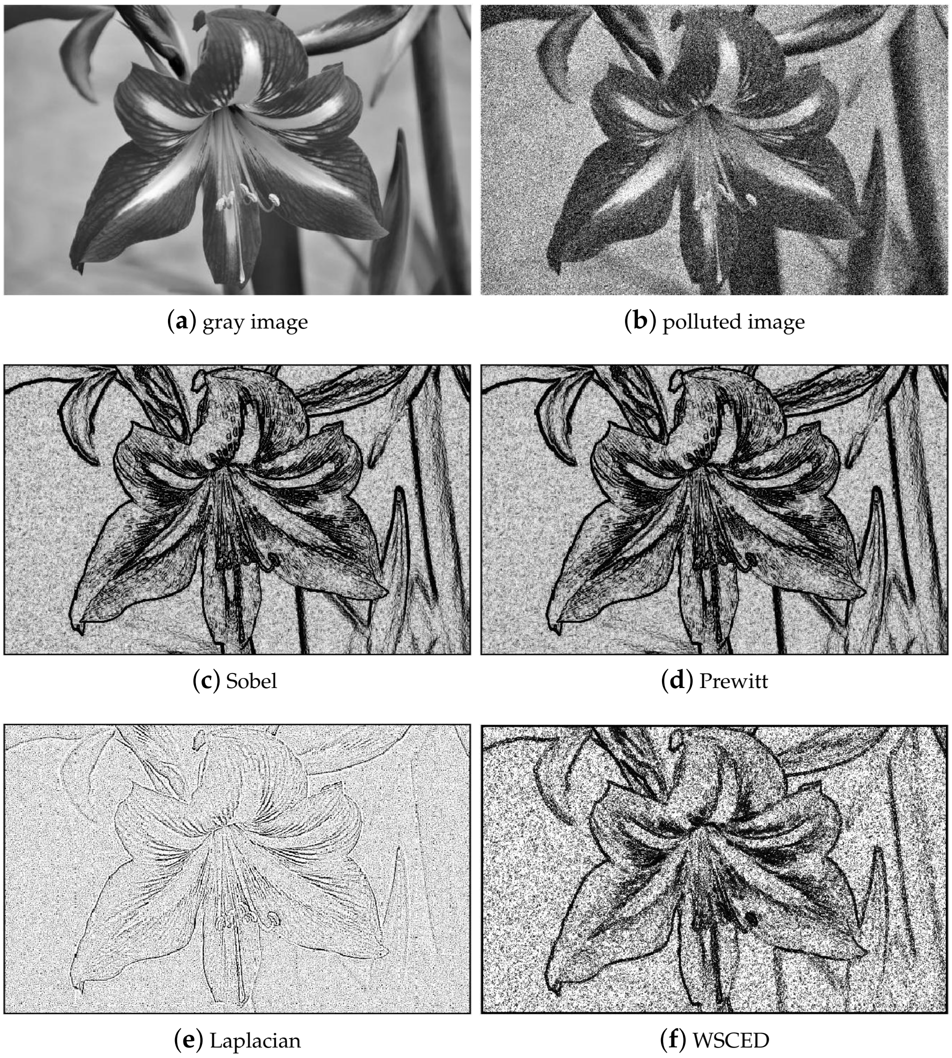

(**a**) gray image
(**b**) polluted image
(**c**) Sobel
(**d**) Prewitt
(**e**) Laplacian
(**f**) WSCED

Figure 9. An example to show different edge detection algorithms on the flower image.

6. Conclusions and Future Work

In this paper, we studied the geometric characteristics of $(SPD(n), g_W)$, including geodesics, the connection, Jacobi fields and curvatures. Compared with the existing results, our results are simpler in form and more suitable for computation. Based on these results, we designed novel algorithms for point cloud denoising and image edge detection. Numerical experiments showed that these geometry-based methods were valid for applications. From both a theoretical and practical prospective, we gained a more comprehensive understanding regarding the Wasserstein geometry on $SPD(n)$, which shows that the Wasserstein metric has both deep application potential and mathematical elegance.

In our future work, on the one hand, we aim to study Wasserstein geometry on other matrix manifolds, such as the Stiefel manifold [25], Grassman manifold [26] and some complex matrix manifolds [27]. On the other hand, we would like to generalize geometry-based methods to solve more problems in image, signal processing [28] and data science.

Author Contributions: Conceptualization, Y.L.; methodology, Y.L.; software, Y.L., Y.C.; validation, S.Z.; investigation, H.S.; resources, H.S.; writing—original draft preparation, Y.L.; writing—review and editing, S.Z., Y.C.; visualization, Y.C.; supervision, H.S.; funding acquisition, H.S. All authors have read and agreed to the published version of the manuscript.

Funding: This subject is supported partially by National Key Research and Development Plan of China (No. 2020YFC2006201).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: In this paper, we obtain all of our point clouds data from the Large Geometric Models Archive of Georgia Tech at https://search.crossref.org/funding (on 5 April 2021). The images were gained from the open resources from Google.

Acknowledgments: The authors would like to express their sincere thanks to Chao Qian for his valuable suggestions.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A. Properties of $\Gamma_A[X]$

Here, we list some basic properties, which will be used frequently in the discussions of this paper.

Proposition A1. *For any $A, B \in SPD(n), X, Y \in M(n), Q \in O(n), k \in \mathbb{R}$, the following equations hold:*

1. $\Gamma_A[X + kY] = \Gamma_A[X] + k\Gamma_A[Y]$;
2. $\Gamma_{kA}[X] = \frac{1}{k}\Gamma_A[X]$;
3. $\Gamma_{A+B}[X] = \Gamma_A[X] - \Gamma_A[B\Gamma_{A+B}[X] + \Gamma_{A+B}[X]B], A + B \in SPD(n)$;
4. $\Gamma_A[AX] = A\Gamma_A[X], \Gamma_A[XA] = \Gamma_A[X]A$;
5. $\Gamma_{A^{-1}}[X] = \Gamma_A[AXA] = A\Gamma_A[X]A$;
6. $\Gamma_{QAQ^{-1}}[QXQ^{-1}] = Q\Gamma_A[X]Q^{-1}$.

Proposition A1 will determine the geometry on $(SPD(n), g_W)$ and be involved into every calculation repetitively and alternatively. We omit the proof since these properties are easily checked.

Appendix B. Proof of Theorem 1

To prove Theorem 1, we shall give two lemmas first.

Lemma A1. *For any $\widetilde{A}_1, \widetilde{A}_2 \in GL(n)$, $l(t) = t\widetilde{A}_2 + (1-t)\widetilde{A}_1, t \in [0,1]$. $\dot{l}(t)$ is horizontal if and only if* $\det(l(t)) > 0$, *and $\dot{l}(0)$ is horizontal at $T_{\widetilde{A}_1}GL(n)$.*

Proof.

$$\begin{aligned}
&\dot{l}(0) = \widetilde{A}_2 - \widetilde{A}_1 \text{ is horizontal at } T_{\widetilde{A}_1}GL(n) \\
\Leftrightarrow\ & \widetilde{A}_1^{-1}(\widetilde{A}_2 - \widetilde{A}_1) \in Sym(n) \\
\Leftrightarrow\ & \widetilde{A}_1^{-1}\widetilde{A}_2 \in Sym(n) \\
\Leftrightarrow\ & \widetilde{A}_2^{-1}\widetilde{A}_1 \in Sym(n) \\
\Leftrightarrow\ & \widetilde{A}_2^{-1}(\widetilde{A}_2 - \widetilde{A}_1) \in Sym(n) \\
\Leftrightarrow\ & \dot{l}(1) \text{ is also horizontal at } T_{\widetilde{A}_2}GL(n).
\end{aligned} \tag{A1}$$

For any $s \in (0,1)$ that satisfies $\det(l(s)) \neq 0$, we constrict the line segment by $l_s(t) := l(st)$, and the lemma can be induced by a horizontal of $\dot{l}_s(1)$. □

Lemma A1 ensures that a line segment is horizontal if its initial vector is horizontal, which brings quite convenience for the following proofs of certain results.

Lemma A2. *For any $A_1, A_2 \in SPD(n)$, there exists $P = A_1^{-\frac{1}{2}}(A_1A_2)^{\frac{1}{2}}A_2^{-\frac{1}{2}} \in O(n)$ such that $\widetilde{\gamma}(t) = tPA_2^{\frac{1}{2}} + (1-t)A_1^{\frac{1}{2}}$. When* $\det(\widetilde{\gamma}(t)) > 0$, *$\dot{\widetilde{\gamma}}(t)$ keeps horizontal at $T_{\widetilde{\gamma}(t)}GL(n)$.*

Proof. First, we have

$$\begin{aligned} P^T P &= A_2^{-\frac{1}{2}}(A_2A_1)^{\frac{1}{2}}A_1^{-1}(A_1A_2)^{\frac{1}{2}}A_2^{-\frac{1}{2}} \\ &= A_2^{-\frac{1}{2}}A_1^{-1}(A_1A_2)^{\frac{1}{2}}(A_1A_2)^{\frac{1}{2}}A_2^{-\frac{1}{2}} \\ &= I. \end{aligned} \tag{A2}$$

Then, we only need to check that $A_1^{-\frac{1}{2}}\dot{\tilde{\gamma}}(0) \in Sym(n)$. In fact, we have

$$\begin{aligned} A_1^{-\frac{1}{2}}\dot{\tilde{\gamma}}(0) &= A_1^{-1}(A_1A_2)^{\frac{1}{2}} - I \\ &= A_1^{-\frac{1}{2}}A_1^{-\frac{1}{2}}(A_1A_2)^{\frac{1}{2}}A_1^{\frac{1}{2}}A_1^{-\frac{1}{2}} - I \\ &= A_1^{-\frac{1}{2}}(A_1^{\frac{1}{2}}A_2A_1^{\frac{1}{2}})^{\frac{1}{2}}A_1^{-\frac{1}{2}} - I \in Sym(n), \end{aligned} \tag{A3}$$

which proves the lemma. □

Remark A1. *Finding the orthogonal matrix P is equivalent to solving a Riccati equation. One can see [29] for details.*

To prove Theorem 1, the last step is to clarify the non-degeneration.

Proof. (Theorem 1) Due to the symmetry from (10), it suffices to prove the diagonal case $A_1 = \Lambda$. We shall prove $\det(\tilde{\gamma}(t)) > 0, \forall t \in (0,1)$. We have

$$\begin{aligned} \det[\tilde{\gamma}(t)] &= \det\Big[t\Lambda^{-\frac{1}{2}}(\Lambda A_2)^{\frac{1}{2}} + (1-t)\Lambda^{\frac{1}{2}}\Big] \\ &= (1-t)^n \det\Big[I + \frac{t}{(1-t)}\Lambda^{-\frac{1}{2}}(\Lambda A_2)^{-\frac{1}{2}}\Lambda^{-\frac{1}{2}}\Big]\det\Big[\Lambda^{\frac{1}{2}}\Big]. \end{aligned} \tag{A4}$$

Since $\Lambda^{-\frac{1}{2}}(\Lambda A_2)^{\frac{1}{2}}\Lambda^{-\frac{1}{2}}$ is congruent to $(\Lambda A_2)^{\frac{1}{2}}$, its eigenvalues are all positive. Thus, $\det[\tilde{\gamma}(t)] > 0$ holds for all $t \in (0,1)$. Combining with Lemmas A1 and A2, the proof is done. □

Appendix C. Proof of Theorem 6

Proof. Again, with (10), we have $r(A) = r(\Lambda)$ where $A = Q\Lambda Q^T$. Thus, we still focus on $r(\Lambda)$. Following the definition and discussions around (23),

$$\begin{aligned} r(\Lambda) &= \inf\{\epsilon(V) \mid V \in T_\Lambda SPD(n), \|V\|_{g_W} = 1\} \\ &= \inf\{\|V\|_{g_W} \mid V \in T_\Lambda SPD(n), \det(I + \Gamma_\Lambda[V]) = 0\}, \end{aligned} \tag{A5}$$

in which $\|\cdot\|_{g_W}$ is the inner product reduced by g_W. Then, we have

$$2r^2(\Lambda) = \inf\{\mathrm{tr}(V\Gamma_\Lambda[V]) \mid \Gamma_\Lambda[V] \text{ has eigenvalue } \eta_k = -1\}. \tag{A6}$$

Denote $\{\lambda_i > 0\}$ as eigenvalues of Λ, and η_j as eigenvalues of $\Gamma_\Lambda[V]$. Let x_j be eigenvectors corresponding to η_j such that $\{x_j\}$ forms an orthogonal basis of $\mathbb{R}^n$. By direct calculation, we find

$$\begin{aligned} \operatorname{tr}(V\Gamma_\Lambda[V]) = & \sum_{i=1}^{n} x_i^T V\Gamma_\Lambda[V]x_i = \sum_{i=1}^{n} \eta_i x_i^T V x_i \\ = & \sum_{i=1}^{n} \eta_i x_i^T (\Lambda\Gamma_\Lambda[V] + \Gamma_\Lambda[V]\Lambda)x_i \\ = & \sum_{i=1}^{n} (\eta_i(\Gamma_\Lambda[V]x_i)^T \Lambda x_i + \eta_i x_i^T \Lambda\Gamma_\Lambda[V]x_i) \\ = & 2\sum_{i=1}^{n} \eta_i^2 x_i^T \Lambda x_i \geq 2x_k^T \Lambda x_k, \end{aligned} \tag{A7}$$

where we used the facts that Γ is positive-definite and $\eta_k = -1$. The equality $\operatorname{tr}(V\Gamma_\Lambda[V]) = x_k^T \Lambda x_k$ holds if and only if $\eta_j = 0, \forall j \neq k$. In these cases, by Algorithm 1, we have

$$\frac{V_{ij}}{\lambda_i + \lambda_j} = (\Gamma_\Lambda[V])_{ij} = -\delta_{ik}\delta_{jk}, \tag{A8}$$

then we obtain that

$$V_{ij} = -2\delta_{ik}\delta_{ik}\lambda_k, (x_k)_i = \delta_{ki}. \tag{A9}$$

Thus,

$$\operatorname{tr}(V\Gamma_\Lambda[V]) = 2x_k^T \Lambda x_k = 2\lambda_k. \tag{A10}$$

In particular, we obtain $2\lambda_{\min}$ as $\min\{\operatorname{tr}(V\Gamma_\Lambda[V])\}$, when $\Lambda_k = \lambda_{\min}$ and $V_{ij} = -2\delta_{ik}\delta_{jk}\lambda_{\min}$. The tangent vector V is called the speed-degenerated direction. The function $\lambda_{\min}(A)$ is certainly continuous, hence $r(A)$ is also continuous. □

Appendix D. Proof of Lemma 1

Before proving Lemma 1, we shall prove a key lemma, which points the relation between the Lie-brackets on the total space and base space.

Lemma A3. *The horizontal lift of vector fields commutes with Lie-brackets,*

$$\mathrm{d}\sigma|_{\widetilde{A}}[\widetilde{X}, \widetilde{Y}] = [X, Y], \forall \widetilde{A} \in GL(n). \tag{A11}$$

Proof. (Lemma A3) On the flat manifold $(GL(n), g_E)$, the connection equals to the usual directional derivative in the Euclidean space. Therefore, for vector fields X, Y on $SPD(n)$, we have

$$\begin{aligned} D_{\widetilde{X}}\widetilde{Y} &= \lim_{t\to 0} \frac{1}{t}\left(\widetilde{Y}|_{\widetilde{A}+\widetilde{X}t} - \widetilde{Y}|_{\widetilde{A}}\right) \\ &= \lim_{t\to 0} \frac{1}{t}\left(\left(\widetilde{A} + \widetilde{X}t\right)\Gamma_{A+Xt}[Y|_{A+Xt}] - \widetilde{A}\Gamma_A[Y|_A]\right) \\ &= \lim_{t\to 0} \frac{\widetilde{A}}{t}(\Gamma_{A+Xt}[Y|_{A+Xt}] - \Gamma_A[Y|_A]) + \widetilde{X}\Gamma_A[Y] \\ &= \lim_{t\to 0} \frac{\widetilde{A}}{t}(\Gamma_{A+Xt}[Y|_{A+Xt}] - \Gamma_A[Y|_A]) + \widetilde{A}\Gamma_A[X]\Gamma_A[Y] \\ &= \widetilde{A}(\Gamma_A[\mathrm{d}Y(X)] - \Gamma_A[X\Gamma_A[Y] + \Gamma_A[Y]X] + \Gamma_A[X]\Gamma_A[Y]). \end{aligned} \tag{A12}$$

With $\Pi_A[X, Y] := \Gamma_A[X]\Gamma_A[Y] - \Gamma_A[Y]\Gamma_A[X]$ into (2), we obtain

$$\Gamma_A[X\Gamma_A[Y] + \Gamma_A[Y]X - Y\Gamma_A[X] - \Gamma_A[X]Y] = \Pi_A[X, Y] - 2\Gamma_A[\Pi_A[X, Y]]A. \tag{A13}$$

Then, we compute the Lie-bracket

$$\begin{aligned}[\widetilde{X},\widetilde{Y}] :=\ & D_{\widetilde{X}}\widetilde{Y} - D_{\widetilde{Y}}\widetilde{X} \\ =\ & \widetilde{A}\Gamma_A[\mathrm{d}Y(X) - \mathrm{d}X(Y)] + \widetilde{A}(\Gamma_A[X]\Gamma_A[Y] - \Gamma_A[Y]\Gamma_A[X]) \\ & - \widetilde{A}\Gamma_A[X\Gamma_A[Y] + \Gamma_A[Y]X - Y\Gamma_A[X] - \Gamma_A[X]Y] \\ =\ & \widetilde{[X,Y]} + 2\widetilde{A}\Gamma_A[\Pi_A[X,Y]]A, \end{aligned} \tag{A14}$$

where the last equality in (A14) comes from (A13).

Finally, we show the second term in (A14) is vertical. In fact, we have

$$2\widetilde{A}\Gamma_A[\Pi_A[X,Y]]A = (2\widetilde{A}\Gamma_A[\Pi_A[X,Y]]\widetilde{A}^T)\widetilde{A}, \tag{A15}$$

where $2\widetilde{A}\Gamma_A[\Pi_A[X,Y]]\widetilde{A}^T$ is anti-symmetric. Thus, the proof for Lemma A3 has been done. □

By Lemma A3, the proof for Lemma 1 is clarified.

Proof. (Lemma 1) For any smooth vector field Z on $SPD(n)$, and its horizontal lift $\widetilde{Z}$, we have

$$\begin{aligned} & \langle \widetilde{\nabla_X Y}, \widetilde{Z}\rangle = \langle \nabla_X Y, Z\rangle \\ =\ & \frac{1}{2}(X\langle Y,Z\rangle + Y\langle Z,X\rangle - Z\langle X,Y\rangle + \langle Z,[X,Y]\rangle + \langle Y,[Z,X]\rangle - \langle X,[Y,Z]\rangle) \\ =\ & \frac{1}{2}\Big(\widetilde{X}\langle\widetilde{Y},\widetilde{Z}\rangle + \widetilde{Y}\langle\widetilde{Z},\widetilde{X}\rangle - \widetilde{Z}\langle\widetilde{X},\widetilde{Y}\rangle + \langle\widetilde{Z},\widetilde{[X,Y]}\rangle + \langle\widetilde{Y},\widetilde{[Z,X]}\rangle - \langle\widetilde{X},\widetilde{[Y,Z]}\rangle\Big) \\ =\ & \frac{1}{2}\Big(\widetilde{X}\langle\widetilde{Y},\widetilde{Z}\rangle + \widetilde{Y}\langle\widetilde{Z},\widetilde{X}\rangle - \widetilde{Z}\langle\widetilde{X},\widetilde{Y}\rangle + \langle\widetilde{Z},[\widetilde{X},\widetilde{Y}]\rangle + \langle\widetilde{Y},[\widetilde{Z},\widetilde{X}]\rangle - \langle\widetilde{X},[\widetilde{Y},\widetilde{Z}]\rangle\Big) \\ =\ & \langle D_{\widetilde{X}}\widetilde{Y},\widetilde{Z}\rangle. \end{aligned} \tag{A16}$$

By the arbitrariness of Z, Lemma 1 is proved. □

Remark A2. *Proof of Lemma 1 implies that* $\mathrm{d}\sigma|_{\widetilde{A}}(D_{\widetilde{X}}\widetilde{Y})$ *is independent on* $\widetilde{A}$ *chosen among a fixed fiber.*

Appendix E. Proof of Theorem 7

Proof. Calculating $\widetilde{R}|_{\widetilde{A}}(\widetilde{X},\widetilde{Y},\widetilde{X},\widetilde{Y})$ as follows

$$\begin{aligned} & \widetilde{R}|_{\widetilde{A}}(\widetilde{X},\widetilde{Y},\widetilde{X},\widetilde{Y}) = \langle \widetilde{R}_{\widetilde{X}\widetilde{Y}}\widetilde{X},\widetilde{Y}\rangle \\ =\ & \langle D_{[\widetilde{X},\widetilde{Y}]}\widetilde{X},\widetilde{Y}\rangle - \langle D_{\widetilde{X}}D_{\widetilde{Y}}\widetilde{X},\widetilde{Y}\rangle + \langle D_{\widetilde{Y}}D_{\widetilde{X}}\widetilde{X},\widetilde{Y}\rangle \\ =\ & \langle D_{\widetilde{[X,Y]}}\widetilde{X},\widetilde{Y}\rangle + \langle D_{2\mathcal{T}(X,Y)}\widetilde{X},\widetilde{Y}\rangle - \langle D_{\widetilde{X}}\widetilde{\nabla_Y X},\widetilde{Y}\rangle \\ & - \langle D_{\widetilde{X}}\mathcal{T}(Y,X),\widetilde{Y}\rangle + \langle D_{\widetilde{Y}}\widetilde{\nabla_X X},\widetilde{Y}\rangle + \langle D_{\widetilde{Y}}\mathcal{T}(X,X),\widetilde{Y}\rangle \\ =\ & \langle \nabla_{[X,Y]}X,Y\rangle + 2\langle D_{\mathcal{T}(X,Y)}\widetilde{X},\widetilde{Y}\rangle - \langle\nabla_X\nabla_Y X,Y\rangle \\ & - \langle\mathcal{T}(X,Y),\mathcal{T}(X,Y)\rangle + \langle\nabla_Y\nabla_X X,Y\rangle \\ =\ & R|_A(X,Y,X,Y) + 2\langle D_{\mathcal{T}(X,Y)}\widetilde{X},\widetilde{Y}\rangle - \langle\mathcal{T}(X,Y),\mathcal{T}(X,Y)\rangle, \end{aligned} \tag{A17}$$

where we used (29), (32), $\mathcal{T}_{\widetilde{A}}(X,Y) \in V(\widetilde{A})$, $\mathcal{T}_{\widetilde{A}}(X,Y) = -\mathcal{T}_{\widetilde{A}}(Y,X)$ and the following equation

$$\begin{aligned} \langle D_{\widetilde{X}}\mathcal{T}(Y,X),\widetilde{Y}\rangle &= \widetilde{X}\langle\mathcal{T}(Y,X),\widetilde{Y}\rangle - \langle\mathcal{T}(Y,X),D_{\widetilde{X}}\widetilde{Y}\rangle \\ &= \langle\mathcal{T}(X,Y),\mathcal{T}(X,Y) + \widetilde{\nabla_X Y}\rangle \\ &= \langle\mathcal{T}(X,Y),\mathcal{T}(X,Y)\rangle. \end{aligned} \tag{A18}$$

Since $\mathcal{T}(X,Y)$ and $\widetilde{X}, \widetilde{Y}$ are both vertical, we have

$$\langle [\mathcal{T}(X,Y), \widetilde{X}], \widetilde{Y} \rangle = 0. \tag{A19}$$

Then, using $D_{\mathcal{T}(X,Y)}\widetilde{X} - D_{\widetilde{X}}\mathcal{T}(X,Y) = [\mathcal{T}(X,Y), \widetilde{X}]$ we can obtain that

$$\langle D_{\mathcal{T}(X,Y)}\widetilde{X}, \widetilde{Y} \rangle = \langle D_{\widetilde{X}}\mathcal{T}(X,Y), \widetilde{Y} \rangle = -\langle \mathcal{T}(X,Y), \mathcal{T}(X,Y) \rangle. \tag{A20}$$

By the definition of the directional derivative and $\sigma(\widetilde{A} + \mathcal{T}(X,Y)) = A$, we have

$$\begin{aligned} D_{\mathcal{T}(X,Y)}\widetilde{X} &= \lim_{t \to 0} \frac{1}{t}\Big(\Big(\widetilde{A} + \mathcal{T}(X,Y)t\Big)\Gamma_A[X] - \widetilde{A}\Gamma_A[X]\Big) \\ &= \mathcal{T}(X,Y)\Gamma_A[X]. \end{aligned} \tag{A21}$$

Combining (A20) and (A21), we obtain the following equation

$$\langle \mathcal{T}(X,Y), \mathcal{T}(X,Y) \rangle = -\langle \mathcal{T}(X,Y)\Gamma_A[X], \widetilde{Y} \rangle, \tag{A22}$$

and

$$\begin{aligned} \widetilde{R}(\widetilde{X}, \widetilde{Y}, \widetilde{X}, \widetilde{Y}) &= R(X,Y,X,Y) + 3\langle \mathcal{T}(X,Y)\Gamma_A[X], \widetilde{A}\Gamma_A[Y] \rangle \\ &= R(X,Y,X,Y) - 3\langle \mathcal{T}(X,Y), \mathcal{T}(X,Y) \rangle. \end{aligned} \tag{A23}$$

Due to $\widetilde{R} \equiv 0$, we obtain the explicit expression for the Wasserstein curvature

$$\begin{aligned} R(X,Y,X,Y) = &\ -3\langle \mathcal{T}(X,Y)\Gamma_A[X], \widetilde{A}\Gamma_A[Y] \rangle \\ = &\ 3\mathrm{tr}(\Gamma_A[X]\mathcal{T}(X,Y)\widetilde{A}\Gamma_A[Y]) \\ = &\ 3\mathrm{tr}(\Gamma_A[X]A\Gamma_A[\Pi_A[X,Y]]A\Gamma_A[Y]). \end{aligned} \tag{A24}$$

□

Appendix F. Proof of Theorem 8

Proof. Using $p = r, q \neq t$, we have

$$(S_1S_2)_{ij} = (1+\delta^{pq})(1+\delta^{pt})\delta_i^q\delta_j^t = \begin{cases} 2\delta_i^q\delta_j^t, & q = p \text{ or } t = p \\ \delta_i^q\delta_j^t, & q \neq p \text{ and } t \neq p \end{cases}, \tag{A25}$$

and

$$(E_{S_1}E_{S_2})_{ij} = \frac{(1+\delta^{pq})(1+\delta^{pt})}{(\lambda_p+\lambda_q)(\lambda_p+\lambda_t)}\delta_i^q\delta_j^t. \tag{A26}$$

Then, we obtain

$$\begin{aligned} (\Gamma_\Lambda[E_{S_1}, E_{S_2}])_{ij} = &\ \frac{[E_{S_1}, E_{S_2}]_{ij}}{\lambda_i + \lambda_j} \\ = &\ \frac{(1+\delta^{pq})(1+\delta^{pt})(\delta_i^q\delta_j^t - \delta_j^q\delta_i^t)}{(\lambda_p+\lambda_q)(\lambda_p+\lambda_t)(\lambda_q+\lambda_t)}, \end{aligned} \tag{A27}$$

and

$$\begin{aligned} &\sum_{k=1}^{n}(\Gamma_\Lambda[E_{S_1}, E_{S_2}])_{ik}(\Lambda E_{S_2}E_{S_1}\Lambda)_{kj} \\ = &\ \frac{(1+\delta^{pq})(1+\delta^{pt})(\delta_i^q\delta_k^t - \delta_k^q\delta_i^t)}{(\lambda_p+\lambda_q)(\lambda_p+\lambda_t)(\lambda_q+\lambda_t)}\frac{(1+\delta^{pq})(1+\delta^{pt})\lambda_q\lambda_t}{(\lambda_p+\lambda_q)(\lambda_p+\lambda_t)}\delta_k^t\delta_j^q \\ = &\ \frac{(1+\delta^{pq})^2(1+\delta^{pt})^2\lambda_q\lambda_t}{(\lambda_p+\lambda_q)^2(\lambda_p+\lambda_t)^2(\lambda_q+\lambda_t)}\delta_i^q\delta_j^q. \end{aligned} \tag{A28}$$

Thus, we obtain the expression for curvature as follows

$$\begin{aligned} R|_{\Lambda}(S_1, S_2, S_1, S_2) &= 3\mathrm{tr}(E_{S_1}\Lambda\Gamma_{\Lambda}[E_{S_1}, E_{S_2}]\Lambda E_{S_2}) \\ &= \frac{3(1+\delta^{pq})^2(1+\delta^{pt})^2\lambda_q\lambda_t}{(\lambda_p+\lambda_q)^2(\lambda_p+\lambda_t)^2(\lambda_q+\lambda_t)}. \end{aligned} \tag{A29}$$

On the other hand, by (A26), we have

$$S_1 \neq S_2 \Longrightarrow \langle S_1, S_2\rangle = \mathrm{tr}(E_{S_2}\Lambda E_{S_1}) = \mathrm{tr}(\Lambda E_{S_1}E_{S_2}) = 0, \tag{A30}$$

and

$$\begin{aligned} \langle S_1, S_1\rangle &= \frac{1}{2}\mathrm{tr}(S_1E_{S_1}) = \frac{1+\delta^{pq}}{\lambda_p+\lambda_q}, \\ \langle S_2, S_2\rangle &= \frac{1}{2}\mathrm{tr}(S_2E_{S_2}) = \frac{1+\delta^{pt}}{\lambda_p+\lambda_t}. \end{aligned} \tag{A31}$$

Then, we can find the area of the sectional parallelogram as

$$\langle S_1, S_1\rangle\langle S_2, S_2\rangle - \langle S_1, S_2\rangle^2 = \frac{(1+\delta^{pq})(1+\delta^{pt})}{(\lambda_p+\lambda_q)(\lambda_p+\lambda_t)}. \tag{A32}$$

From (A29) and (A32), the proof is done consequently. □

Appendix G. Proof of Theorem 9

Before proving this theorem, we need to do some preparation. For any $A \in SPD(n)$ with eigenvalues $\lambda_i, i = 1, \cdots, n$, construct diagonal matrix $\Lambda = \mathrm{diag}(\lambda_1, \cdots, \lambda_n)$ and upper triangle matrix $U = \left(\frac{1}{\lambda_i+\lambda_j}\right)_{i<j}$. In $p = r, q < t$ cases, by Theorem 8, we have

$$\begin{aligned} K|_A(S^{p,q}, S^{r,t}) = &\ \frac{3(1+\delta^{pq})(1+\delta^{pt})\lambda_q\lambda_t}{(\lambda_p+\lambda_q)(\lambda_p+\lambda_t)(\lambda_q+\lambda_t)} \\ = &\ 3U_{qt}((U+U^T)_{pq}\Lambda_{qq}+\delta_{pq})((U+U^T)_{pt}\Lambda_{tt}+\delta_{pt}) \\ = &\ 3U_{qt}\Lambda_{tt}(U+U^T)_{tp}\delta_{pq} + 3(U+U^T)_{pq}\Lambda_{qq}U_{qt}\delta_{pt} \\ &+ 3(U+U^T)_{pt}\Lambda_{tt}U_{tq}\Lambda_{qq}(U+U^T)_{qp}, \end{aligned} \tag{A33}$$

where we used

$$\frac{1-\delta^{pq}}{\lambda_p+\lambda_q} = (U+U^T)_{pq}, \tag{A34}$$

and

$$\frac{(1+\delta^{pq})\lambda_q}{\lambda_p+\lambda_q} = \frac{(1-\delta^{pq})\lambda_q}{\lambda_p+\lambda_q}\lambda_q + \frac{2\delta^{pq}\lambda_q}{\lambda_p+\lambda_q} = (U+U^T)_{pq}\lambda_q + \delta^{pq}. \tag{A35}$$

Then, we can obtain the scalar curvature by summing (A33).

Proof. Note that $U_{qt} = 0, q \geq t$, we have

$$\begin{aligned}
\rho(\Lambda) = & \sum_{p,q<t} K|_{\Lambda}(S^{p,q}, S^{p,t}) \\
= & \sum_{p,q<t} \Big(3U_{qt}\Lambda_{tt}(U+U^T)_{tp}\delta_{pq} + 3(U+U^T)_{pq}\Lambda_{qq}U_{qt}\delta_{pt} \Big) \\
& + \sum_{p,q<t} \Big(3(U+U^T)_{pt}\Lambda_{tt}U_{tq}\Lambda_{qq}(U+U^T)_{qp} \Big) \\
= & \sum_{p,q,t} \Big(3U_{qt}\Lambda_{tt}(U+U^T)_{tp}\delta_{pq} + 3(U+U^T)_{pq}\Lambda_{qq}U_{qt}\delta_{pt} \Big) \\
& + \sum_{p,q,t} \Big(3(U+U^T)_{pt}\Lambda_{tt}U_{tq}\Lambda_{qq}(U+U^T)_{qp} \Big) \\
= & \; 3\mathrm{tr}(U\Lambda(U+U^T) + (U+U^T)\Lambda U + (U+U^T)\Lambda U\Lambda(U+U^T)).
\end{aligned} \tag{A36}$$

□

Appendix H. Effects of WSCED

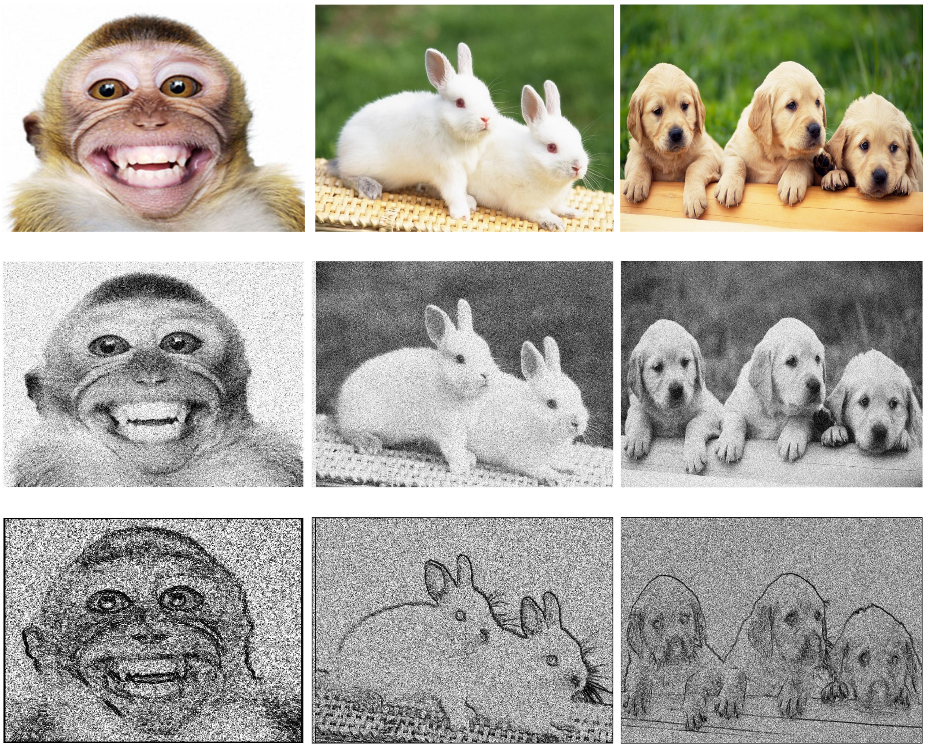


Figure A1. WSCED on different polluted images. The first row shows the original RGB images. The second row shows images with uniform noises. The third row shows edge images after WSCED.

Appendix I

Table A1. Comparison of three denosing algorithms.

Datasource (Size)	Original SNR	ROR			SOR			AWCD		
		TPR	FPR	SNRG	TPR	FPR	SNRG	TPR	FPR	SNRG
Stanford Bunny	10	**100.00%**	18.10%	452.5%	87.78%	49.00%	79.1%	**99.28%**	**4.80%**	**1968.3%**
(10,000)	0.1	**100.00%**	100.00%	0.0%	84.44%	**71.61%**	**17.9%**	**99.93%**	90.53%	10.4%
Duke Dragon	100	**100.00%**	9.60%	941.7%	82.29%	51.90%	58.6%	**100.00%**	**2.90%**	**3348.3%**
(100,000)	1	**100.00%**	100.00%	0.0%	82.12%	70.16%	17.1%	**99.88%**	**2.40%**	**4054.8%**
Armadillo	10	**62.26%**	0.54%	114.3	88.06%	63.46%	38.8%	34.84%	**0.01%**	**347.38**
(50,000)	1	**62.66%**	0.41%	153.34	87.99%	69.62%	26.4%	34.84%	**0.09%**	**404.14**
Lucy	100	**100.00%**	6.00%	1566.6%	80.71%	51.80%	55.8%	**99.99%**	**0.80%**	**123.99**
(50,000)	10	**100.00%**	35.50%	181.7%	80.65%	66.44%	21.4%	**99.99%**	**2.86%**	**3396.1%**
Happy Buddha	100	**100.00%**	6.80%	1370.6%	77.82%	56.60%	37.5%	**99.92%**	**2.40%**	**4063.3%**
(50,000)	10	**100.00%**	13.86%	621.5%	78.24%	64.76%	20.8%	**99.93%**	**1.88%**	**5215.6%**

References

1. Hosseini, R.; Sra, S. Matrix Manifold Optimization for Gaussian Mixtures. *Advances in Neural Information Processing Systems*; MIT Press: Cambridge, MA, USA, 2015.
2. Manton, J.H. A Primer on Stochastic Differential Geometry for Signal Processing. *IEEE J. Sel. Top. Signal Process.* **2013**, *4*, 681–699. [CrossRef]
3. Rubner, Y.; Tomasi, C.; Guibas, L.J. A Metric for Distributions with Applications to Image Databases. In Proceedings of the International Conference on Computer Vision, Bombay, India, 7 January 1998.
4. R. Bhatia. *Matrix Ananlysis. Graduate Texts in Mathematics*; Springer: New York, NY, USA, 1997.
5. Bhatia, R. *Positive Definite Matrices*; Princeton University Press: Princeton, NJ, USA, 2009.
6. Pennec, X.; Fillard, P.; Ayache, N. A Riemannian Framework for Tensor Computing. *Int. J. Comput. Vis.* **2006**, *1*, 41–66. [CrossRef]
7. Arsigny, V.; Fillard, P.; Pennec, X.; Ayache, N. Log-Euclidean Metrics for Fast and Simple Calculus on Diffusion Tensors. *Magn. Reson. Med.* **2010**, *2*, 411–421. [CrossRef] [PubMed]
8. Hall, B.C. *Lie Groups, Lie Algebras, and Representations*; Springer: Berlin, Germany, 2015.
9. Li, Y.; Wong, K. Riemannian Distances for Signal Classification by Power Spectral Density. *IEEE J. Sel. Top. Signal Process.* **2013**, *7*, 655–669. [CrossRef]
10. Zhang, S.; Cao, Y.; Li, W.; Yan, F.; Luo, Y.; Sun, H. A New Riemannian Structure in SPD(n). In Proceedings of the IEEE International Conference on Signal, Information and Data Processing (ICSIDP), Chongqing, China, 11–13 December 2019.
11. Gelbrich, M. On a Formula for the L2 Wasserstein Metric between Measures on Euclidean and Hilbert Spaces. *Math. Nachrichten* **2015**, *1*, 185–203. [CrossRef]
12. Givens, C.R.; Shortt, R.M. A Class of Wasserstein Metrics for Probability Distributions. *Mich. Math. J.* **1984**, *2*, 231–240. [CrossRef]
13. Martin, A.; Soumith, C.; Leon, B. Wasserstein GAN. *arXiv* **2017**, arXiv:1701.07875.
14. Asuka, T. On Wasserstein Geometry of Gaussian Measures. In *Probability Approach to Geometry*; Mathematical Society of Japan: Tokyo, Japan, 2010; pp. 463–472.
15. Massart, E.; Absil, P.-A. Quotient Geometry with Simple Geodesics for the Manifold of Fixed-Rank Positive-Semidefinite Matrices. *Siam J. Matrix Anal. Appl.* **2020**, *1*, 171–198. [CrossRef]
16. Rusu, R.B.; Cousins, S. 3D is Here: Point Cloud Library (PCL). In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Shanghai, China, 9–13 May 2011.
17. Ward, A.J.B. A General Analysis of Sylvester's Matrix Equation. *Int. J. Math. Educ. Sci. Technol.* **1991**, *22*, 615–620. [CrossRef]
18. do Carmo, M.P. *Riemannian Geometry*; Springer: Boston, MA, USA 1992.
19. Lee, J.M. *Introduction to Smooth Manifolds*; Springer: Berlin, Germany, 2003.
20. Kobayashi, S. Transformation Groups in Differential Geometry. In *Classics in Mathematics*; Springer: Berlin, Germany, 1932.
21. Lee, J.M. *Riemannian Manifolds An Introduction to Curvature*; Springer: New York, NY, USA, 1997.
22. Boothby, W.M. *An Introduction to Differentiable Manifolds and Riemannian Geometry*, 2nd ed.; Academic Press: Cambridge, MA, USA, 1986.
23. Masry, E. Multivariate Probability Density Estimation for Associated Processes: Strong Consistency and Rates. *Stat. Probab. Lett.* **2002**, *2*, 205–219. [CrossRef]
24. Nadernejad, E.; Sharifzadeh, S.; Hassanpour, H. Edge Detection Techniques: Evaluations and Comparisons. *Appl. Math. Sci.* **2008**, *2*, 1507–1520.

25. Nishimori, Y.; Akaho, S. Learning Algorithms Utilizing Quasi-geodesic Flows on the Stiefel Manifold. *Neurocomputing* **2005**, *67*, 106–135. [CrossRef]
26. Fiori, S. A Theory for Learning by Weight Flow on Stiefel-Grassman Manifold. *Neural Comput.* **2001**, *13*, 1625–1647. [CrossRef]
27. Hua, X.; Ono, Y.; Peng, L.; Cheng, Y.; Wang, H. Target Detection Within Nonhomogeneous Clutter Via Total Bregman Divergence-Based Matrix Information Geometry Detectors. *IEEE Trans. Signal Process.* **2021**, *69*, 4326–4340. [CrossRef]
28. Hua, X.; Peng, L. MIG Median Detectors with Manifold Filter. *Signal Process.* **2021**, *188*, 108176. [CrossRef]
29. Malag, L.; Montrucchio, L.; Pistone, G. Wasserstein Riemannian Geometry of Gaussian densities. In *Information Geometry*; Springer: Berlin/Heidelberger, Germany, 2018; pp. 137–179.