Geodesic Learning With Uniform Interpolation on Data Manifold

CONG GENG, JIA WANG, LI CHEN, (Member, IEEE), AND ZHIYONG GAO, (Member, IEEE)
Institute of Image Communication and Network Engineering, Shanghai Jiao Tong University, Shanghai 200240, China
Corresponding author: Jia Wang (jiawang@sjtu.edu.cn)

ABSTRACT Recently with the development of deep learning on data representation and generation, how to sampling on a data manifold becomes a crucial problem for research. In this paper, we propose a method to learn a minimizing geodesic within a data manifold. Along the learned geodesic, our method is able to generate high-quality uniform interpolations with the shortest path between two given data samples. Specifically, we use an autoencoder network to map data samples into the latent space and perform interpolation in the latent space via an interpolation network. We add prior geometric information to regularize our autoencoder for a flat latent embedding. The Riemannian metric on the data manifold is induced by the canonical metric in the Euclidean space in which the data manifold is isometrically immersed. Based on this defined Riemannian metric, we introduce a constant-speed loss and a minimizing geodesic loss to regularize the interpolation network to generate uniform interpolations along the learned geodesic on the manifold. We provide a theoretical analysis of our model and use image interpolation as an example to demonstrate the effectiveness of our method.

INDEX TERMS Geodesic learning, uniform interpolation, flat embedding, autoencoder, constant-speed.

I. INTRODUCTION

With the success of deep learning on approximating complex non-linear functions, some notable generative models such as Autoencoders (AE) [1], generative adversarial network (GAN) [2] and their combinations [3], [4] have received considerable interest and been studied from multiple perspectives on representing data manifolds. However, they usually fail to generate a flat latent representation even when confronting with simple curved manifolds such as Swiss-roll and S-curve, resulting in possible meaningless interpolation. Besides, after obtaining a latent representation, interpolation trajectory between samples on the data manifold has attracted wide attention because of its extensive applications, such as image editing [5], [6] and trajectory prediction [7], [8].

However, little works of literature focus on the interpolation along geodesics which can be very helpful to some downstream tasks such as clustering [9], classification [10] and segmentation [11]. For geodesic learning, Arvanitidis et al. [9], [10] and Chen et al. [12], [13] both present a magnification factor [14] to help find the shortest path that follows the regions of high data density in the latent space as the learned geodesic. But these methods explore geodesics on latent space rather than on data space and lack strict guarantees for geodesics in mathematics.

In this paper, we propose a method to capture the geometric structure of a data manifold and to find smooth geodesics between data samples. First, motivated by DIMAL [15], we introduce a framework combining an autoencoder with traditional manifold learning methods to explore the geometric structure of a data manifold. In contrast to existing combinations of autoencoders and GANs, we resort to the classical manifold learning algorithms to obtain an approximation of geodesic distance as a constraint for an autoencoder. Using this method, our encoder can unfold the data manifold to obtain a flat latent representation.

Second, we propose a geodesic learning method by interpolating on the data manifold to establish smooth geodesics between data samples. The Riemannian metric on the data manifold can be induced by the canonical metric in the Euclidean space in which the manifold is isometrically immersed. The interpolation between data samples has
attracted wide attention because it can achieve a smooth transition from one sample to another, such as intermediate face generation and path planning. We parameterize the generated curve by a parameter \( t \) and propose a constant-speed loss to achieve a uniform interpolation by making \( t \in [0, 1] \) be an arc-length parameter of the interpolation curve. According to Riemannian geometry [16], we propose a geodesic loss to force the interpolation network to generate points along a geodesic. Another key factor is a minimizing loss to make the geodesic be the minimal one. Our contributions are summarized as follows:

- We propose a framework in which an autoencoder and an interpolation network are introduced to explore the manifold structure. The autoencoder network promotes flat latent representations achieved by adding some prior geometric information to it.
- With the Riemannian metric induced from the canonical metric in the Euclidean space, we propose a constant-speed loss and a minimizing geodesic loss for the interpolation network to generate the minimizing geodesic on the manifold given two endpoints. We parameterize each geodesic by an arc-length parameter \( t \) to fulfill a uniform interpolation.

II. METHOD

A. MANIFOLD RECONSTRUCTION

Autoencoders can generate high-quality samples from specific latent distributions on high-dimensional data manifold. But they fail to get flat embeddings on some curved surfaces with changing curvature such as Swiss-roll or S-curve. Some other works [5], [17], [18] were proposed to generate interpolations within the distribution of real data by distinguishing interpolations from real samples. But they generate unsatisfying samples on the above-mentioned low-dimensional manifolds. The encoding results can be seen in Fig. 1. The reason for this problem may be the insufficient ability of GANs and the decoder of AE. The discriminator of GANs can only distinguish the similarities of distributions between generated samples and real ones and the autoencoders are prone to put the curvature information into latent representations to make the network as simple as possible, resulting in non-flat representations.

For our method, we add some prior geometric information obtained by traditional manifold learning approaches to encode a flat latent representation. Traditional non-linear manifold learning approaches such as Isomap [19], LLE [20] and LTSA [21] are classical algorithms to get a flat embedding by unfolding some curved surfaces. We apply them to our method by adding a regularizer to the autoencoder. The loss function of the autoencoder can be written as:

\[
L_{AE} = \|D(E(x)) - x\|^2 + \left(\|E(x_i) - E(x_j)\|_2^2 - DKS_{ij}\right)^2, \tag{1}
\]

where \( x \) is the input sampled from the data manifold. \( DKS_{ij} \) represents the approximated geodesic distance between two latent embeddings \( E(x_i) \) and \( E(x_j) \). For more details we refer the interested reader to [19], [20], [21].

Encoder \( E \) and Decoder \( D \) of an autoencoder are trained to minimize the above loss \( L_{AE} \). With \( DKS_{ij} \) as an expected approximation, the encoder is forced to train towards obtaining a flat latent representation while the decoder is forced to learn the lost curvature information from latent embeddings. Behaviors induced by the \( L_{AE} \) loss and other four autoencoder-based methods can be observed in Fig. 1: only by using our \( L_{AE} \) loss, the swiss-roll can be flattened on 2-dimensional latent space. For our experiments, we choose LTSA to compute the approximated geodesic distance because its learned local geometry views the neighborhood of a data point as a tangent space to flat the manifold. The parameters setting for those compared methods are the same as their original papers.

B. GEODESIC LEARNING

In our model, we denote \( X \) as a data manifold. The Riemannian metric on \( X \) can be induced from the canonical metric on the Euclidean space \( \mathbb{R}^N \) to guarantee the immersion is an isometric immersion. Thus to obtain a geodesic on manifold \( X \), we can use the Riemannian geometry on \( \mathbb{R}^N \) and the characteristics of isometric immersion.

1) INTERPOLATION NETWORK

We produce geodesics on manifold \( X \) by interpolating in the latent space and decoding them into data space. The simplest interpolation is linear interpolation as \( z = (1 - t) \cdot z_1 + t \cdot z_2 \). For geodesic learning, linear interpolation is not applicable in most situations. Yang et al. [9] propose to use the restricted class of quadratic functions and Chen et al. [12] employ a neural network to parameterize the geodesic curves. We use polynomial functions similar to Yang’s approach as our interpolation network. The difference is that we employ cubic functions to parameterize interpolants considering the diversity of latent representations, i.e., \( ct(t) = at^3 + bt^2 + ct + d \). Therefore, a curve generated by our interpolation network has four \( m \)-dimensional free parametric vectors \( a, b, c \) and \( d \), where \( m \) is the dimension of latent coordinates. In practice,
we train a geodesic curve \( c(t) \) that connects two pre-specified points \( z_0 \) and \( z_1 \), so the function should be constrained to satisfy \( c(0) = z_0 \) and \( c(1) = z_1 \). We initialize our interpolation network by setting \( a = 0 \) and \( b = 0 \) to make the initial interpolation be a linear interpolation and perform the optimization using gradient descent. More details can be referred to in Yang’s paper [9].

2) CONSTANT-SPEED LOSS
We can produce interpolations along a curve on manifold \( \mathcal{X} \) by decoding the output \( c(t) \) of the interpolation network as \( t \) from 0 to 1. We expect the parameter \( t \) to be an arc-length parameter which means the parameter \( t \) is proportional to the arc length of the curve \( \gamma(t) \). To realize this, the following theorem can provide theoretical support:

**Theorem 1:** Suppose \( \mathcal{X} \subset \mathbb{R}^N \) is a Riemannian differentiable manifold. The Riemannian metric on \( \mathcal{X} \) is induced from the canonical metric on \( \mathbb{R}^N \). If \( \gamma : I \rightarrow \mathcal{X} \) is a geodesic on \( \mathcal{X} \), \( \{x_1(t), x_2(t), \cdots, x_N(t)\} \) is the Cartesian coordinate of \( \gamma(t) \) in \( \mathbb{R}^N \), then \( \sqrt{\sum_i \|x_i'(t)\|^2} \) is a constant, for \( \forall t \in I \).

As stated in Theorem 1, the length of the tangent vector along a geodesic is a constant. Let \( G(t) = Dc(t) \) denote the output of the decoder taking the interpolation curve as input, we design the following constant-speed loss as:

\[
L_{\text{const-speed}} = \frac{1}{n} \sum_{i=1}^{n} \left( \frac{\|G'(t)\|_2}{\text{mean}_i(\|G'(t)\|_2)} - 1 \right)^2,
\]

where \( n \) denotes the number of sampling points and \( \text{mean}_i(\|G'(t)\|_2) = \frac{1}{n} \sum_{i=1}^{n} \|G'(t)\|_2 \). \( G'(t) \) denotes the derivative of the output \( G(t) \) with respect to \( t \). It can be viewed as the velocity at \( t \) along \( G(t) \) and \( \|G'(t)\|_2 \) represents the magnitude of this velocity corresponding to the length of the tangent vector. Therefore, from a certain point of view, constant-speed loss makes \( G(t) \) have a constant speed with \( t \) moving from 0 to 1.

3) MINIMIZING GEODESIC LOSS
After guaranteeing the output curve \( G(t) \) of our decoder has a constant speed, we need to let the curve \( G(t) \) be a geodesic. We have the following theorem to ensure a curve is a geodesic.

**Theorem 2:** Suppose \( \gamma : I \rightarrow \mathcal{X} \) is a curve on \( \mathcal{X} \), \( (U, h) \) is a system of coordinates of \( \mathcal{X} \) with \( \gamma(t) \subset h(U) \), \( z = [z^1, z^2, \cdots, z^m] \) is the local coordinate of \( h(U) \), \( x(t) = [x_1(t), x_2(t), \cdots, x_N(t)] \) is the Cartesian coordinate of \( \gamma(t) \) in \( \mathbb{R}^N \), then \( \gamma(t) \) is a geodesic on \( \mathcal{X} \), if and only if:

\[
\frac{\partial^2 x(t)}{\partial t^2} - \frac{\partial h}{\partial z} \gamma(t) = 0, \quad \forall t \in I.
\]

Theorem 2 demonstrates a curve is a geodesic if and only if its second derivative with respect to parameter \( t \) is orthogonal to the tangent space. In practice, we can assume our encoder maps a point on the manifold \( \mathcal{X} \) to its local coordinates. So based on Theorem 2, we are able to optimize the following problem as the geodesic loss:

\[
L_{\text{geo}} = \frac{1}{n} \sum_{i=1}^{n} \|G''(t_i) \cdot D'(c(t_i))\|_2,
\]

where \( D \) is the function of decoder and \( D'(c(t_i)) \) denotes the derivative of \( D \) at \( c(t_i) \). \( G''(t) \) is a \( N \)-dimensional vector corresponding to \( \frac{\partial^2 x(t)}{\partial t^2} \) and \( D'(c(t)) \) is an \( N \times m \) matrix corresponding to \( \frac{\partial h}{\partial z} \gamma(t) \) in Theorem 2. Geodesic loss and constant-speed loss jointly force curve \( G(t) \) to have zero acceleration. That is, \( G(t) \) is a geodesic on data manifold. But geodesic connecting two points may not be unique, such as the geodesic on a sphere. The minimizing geodesic is the curve with minimal length connecting two points. Thus we add a minimizing length constraint to ensure \( G(t) \) is a minimizing geodesic. We approximate the curve length using the summation of velocity at \( t_i \). The minimizing loss is proposed to minimize curve length as:

\[
L_{\text{min}} = \sum_{i=1}^{n} \|G'(t_i)\|_2.
\]

For implementation, we use the following difference approximation to reduce the computational burden:

\[
G'(t) \approx \frac{G(t + \Delta t) + G(t - \Delta t)}{2\Delta t},
\]

\[
G''(t) \approx \frac{G(t + \Delta t) + G(t - \Delta t) - 2G(t)}{\Delta t^2}.
\]

For \( D'(c(t)) \), we can use the Jacobian of the decoder as implemented by Pytorch or difference approximation that is similar to \( G'(t) \). To summarize this part, the overall loss function of our interpolation network is:

\[
L_{\text{total}} = \lambda_1 L_{\text{const-speed}} + \lambda_2 L_{\text{geo}} + \lambda_3 L_{\text{min}},
\]

where \( \lambda_1, \lambda_2 \) and \( \lambda_3 \) are the weights to balance these three losses. Under this loss constraint, we can generate interpolations moving along the minimizing geodesic with constant speed and thus fulfill a uniform interpolation.

The overall geodesic learning algorithm is given in Algorithm 1.

III. EXPERIMENTS
In this section, we present experiments on the geodesic generation and image interpolation to demonstrate the effectiveness of our method.

A. GEODESIC GENERATION
First, we do experiments on 3-dimensional datasets since their geodesics can be better visualized. We choose the semi-sphere and swiss-roll as our data manifolds.

1) SEMI-SHPE DATASET
We randomly sample 4,956 points subjecting to the uniform distribution on semi-sphere. In Fig. 2, we compare our approach with other interpolation methods, i.e.,
Algorithm 1 Geodesic Learning

Require: Two random points $x_0, x_1$ on dataset; $N$ points of time $t_i = \frac{i-1}{N-1} (1 \leq i \leq N)$; The number of iterations of the interpolation network $n_{iter}$; The weight parameters $\lambda_1, \lambda_2, \lambda_3$; Delta-time $\Delta t$.
1: Train an autoencoder by optimizing Eq. (1);
2: Obtain latent embeddings with encoder $E$ of the trained autoencoder: $z_0 = E(x_0), z_1 = E(x_1)$;
3: Initialize interpolation network by setting $a = 0, b = 0, c = z_1 - z_0, d = z_0$;
4: for $j = 1$ to $n_{iter}$ do
5: Obtain latent representations of interpolations $c(t_i)$ using interpolation network.
6: Decode $c(t_i)$ using decoder $D$ to get $G(t_i) = D(c(t_i))(1 \leq i \leq N)$.
7: Based on Eq. (6), compute $G'(t_i)$ and $G''(t_i)$ with $\Delta t$.
8: Compute the overall loss $L_{total}$ using Eq. (7).
9: Update interpolation network’s parameters $a, b, c, d$ with RMSProp optimizer.
10: end for
11: return Interpolations $G(t_i)$

AAE [2], A&H [22] and Chen’s method [12]. For AAE and Chen’s methods, we reproduce them with the same autoencoder-based network architecture with our model. For A&H, we use its released official code. Parameters are chosen as in the original papers. From Riemannian geometry, we know the geodesic of a sphere under our defined Riemannian metric is a circular arc or part of it. The center of this circular arc is the center of this sphere. AAE cannot guarantee that the curve connecting two points is a geodesic. A&H can find the shortest path connecting two corresponding reconstructed endpoints. But the endpoints are inconsistent with the original inputs due to the uncertainty of their VAE network and their stochastic Riemannian metric. Chen’s method can generate interpolations along a geodesic, but they cannot fulfill a uniform interpolation. In Fig. 2, we observe that our method can generate uniform interpolation along a fairly accurate geodesic on semi-sphere based on the defined Riemannian metric.

We also do ablation study on the semi-sphere dataset to investigate the effect of different losses proposed in our method, including the constant-speed loss, geodesic loss, and minimizing loss. The estimated curve length shown in Table 1 presents the results obtained by the combinations of different losses.
C. Geng et al.: Geodesic Learning With Uniform Interpolation on Data Manifold

losses. For the weights of different loss terms, we empirically set $\lambda_1 = 1$, $\lambda_2 = 0.001$ and $\lambda_3 = 10$ in Eq. 7 as we observe from the experiments that this setting can ensure a stable training and a robust performance. For our ablation study in Table 1, we just set the corresponding weight of the removed loss term as 0 to compare the estimated geodesic length. From Table 1 we can see that even though without the geodesic loss, our network can generate a shorter path that is close to real geodesic compared with linear interpolation. Because just constant-speed loss and minimizing loss can constrain the interpolation curve to have small velocities at the interpolation points. But it’s not accurate enough since the interpolation curve is discretized by time interval $\Delta t$. When incorporated with the geodesic loss, the interpolated points are fine-tuned to move along an accurate geodesic. The generated curve trained with all three losses contributes the best result and the estimated curve length 1.1028 is the closest to the length of real geodesic, namely 1.0776.

2) SWISS-ROLL DATASET
We choose Swiss-roll dataset to demonstrate the effectiveness of our method on manifolds with large curvature variations. We randomly sample 5,000 points subjecting to the uniform distribution on the Swiss-roll manifold. We compare our approach with AAE [3], ACAI [17], GAIA [18] and Chen’s method [5]. All the methods are reproduced using the same autoencoder-based network architecture with ours. Parameters are chosen as in the original papers. Fig. 3 shows the experimental results. We observe that except our approach, other methods fail to generate interpolations within the data manifold.

AAE just forces the latent distribution to be a Gaussian distribution, even though there is a trade-off between latent distribution matching and data reconstruction [23]. So there is no guarantee for a useful latent interpolation, let alone along a geodesic. ACAI, GAIA and Chen’s method try to make decoded points from linear latent interpolations remain...
within data manifold, but from Fig. 3 we can see that for
manifold with large curvature, it’s hard to obtain a flat
unfolded latent embedding without a strong prior geometric
information, which can be further verified through Fig. 1.
Besides, the interpolation curve with these three methods are
not related to the geodesics, which means we can’t find a
meaningful interpolation to depict geometric characteristics
of data manifold. Compared with those methods, our method
can get a flat unfolded latent embedding to facilitate some
following manipulations on latent space. With this useful
representation, we can interpolate along a geodesic through
a specific interpolation network given two samples.

B. IMAGE INTERPOLATION
To further demonstrate our model’s effectiveness on
image interpolation, we choose MNIST [24] and Fashion-
MNIST [25] datasets. They both consist of a training set
of 60,000 samples and a test set of 10,000 samples asso-
ciated with a label from 10 classes. We don’t employ our
manifold reconstruction method to train the autoencoder
because the concept of distance-based nearest neighbors is
no longer meaningful when the dimension goes sufficiently
high [26], [27]. Shao et al. [28] argue that high-dimensional
manifold has some curvature, but it is close to zero. Therefore,
we can directly employ variational autoencoder (VAE) [1] or
adversarial autoencoder (AAE) [3] to obtain latent embed-
dings. For each dataset, we randomly select two images from
the test set as the start-point and endpoint in the data manifold.

For MNIST dataset, we provide results of image interpo-
lation in Fig. 4 to verify the invariance of geodesic interpola-
tion for both VAE and AAE being the autoencoder respectively.
It worth mentioning that geodesics do not depend on the
distributions of latent space because they only depend on
the geometry structure of data manifold. Thus we should
theoretically get the same interpolation results for different
autoencoders if the interpolations are along geodesics. From
Fig. 4, we can observe for linear interpolation, different
autoencoders result in very different results. For geodesic
interpolation, although there is a slight difference between
different autoencoders, the general transition tendency is con-
sistent which means the interpolated curve using our method
learns some geodesic’s characteristics.

For Fashion-MNIST dataset, we show the interpolation
results trained with VAE and AAE in Fig. 5. We use the
velocity defined in Section II-B2 to evaluate the smoothness
of different interpolation methods. Our geodesic interpolation
can always obtain recognizable images with a smoother speed
while linear interpolation may generate some ghosting when
transiting over different classes resulting in drastic velocity
changes. This demonstrates our geodesic learning method can
fulfill a uniform interpolation along a geodesic.

We further verify the characteristic of the geodesic’s short-
est path for our method on both datasets above. In Fig. 6, we
present a comparison of the interpolant trajectory’s length with linear and geodesic interpolation for both VAE
and AAE. We randomly choose 250 pairs of endpoints on
data manifold for each evaluation and approximate the tra-
jectory’s length using the summation of velocity at $t_i$ which
is described in Section II-B3. Fig. 6 shows that our geodesic
interpolation has smaller average length and variance on both
MNIST and Fashion-MNIST datasets. This demonstrates that
compared with linear interpolation, our interpolation method
can make the interpolation curve traverse along a shorter path
which is the main characteristic of a geodesic.

IV. CONCLUSION
We explore the geometric structure of the data manifold by
proposing a geodesic learning algorithm with uniform inter-
polation. We add prior geometric information to regularize
our autoencoder to generate a flat unfolded latent embedding.
We also propose a constant-speed loss and a minimizing
geodesic loss to interpolate along geodesics on the underlying
data manifold given two endpoints. Different from existing
methods in which geodesic is defined as the shortest path on
3-D curved manifolds and high-dimensional image space.

APPENDIX A

Lemma 1: If $\gamma : I \rightarrow M$ is a geodesic on Riemannian
manifold $M$, then the length of the tangent vector $\frac{d\gamma}{dt}$ is constant.

Proof: Suppose $D$ is the Riemannian connection of $M$, then we have

$$\frac{d}{dt} \left( \frac{d\gamma}{dt} \cdot \frac{d\gamma}{dt} \right) = 2 \left[ D \frac{d\gamma}{dt}, \frac{d\gamma}{dt} \right] = 0,$$

Thus,

$$|\gamma'(t)| = C \text{ (constant)} \quad (9)$$

Proof of Theorem 1. Proof: Suppose the Riemannian metric on $\mathcal{X}$ is induced from $\mathbb{R}^N$ by identity mapping $i$.

$[x_1, x_2, \ldots, x_N]$ is the Cartesian coordinate system of $\mathbb{R}^N$. 

$[x_1(t), x_2(t), \ldots, x_N(t)]$ is the Cartesian coordinate of $\gamma(t)$.

Based on the characteristic of $di$, we obtain the corresponding
tangent vector of $\gamma'(t)$ in $T_{\gamma(t)}\mathbb{R}^N$:

$$di_{\gamma(t)}(\gamma'(t)) = \frac{dx_i(t)}{dt} \frac{\partial}{\partial x^i} \quad (10)$$

According to the canonical metric on $\mathbb{R}^N$, we have

$$|di_{\gamma(t)}(\gamma'(t))| = \sqrt{\left( \frac{dx_i(t)}{dt} \frac{\partial}{\partial x^i}, \frac{dx_i(t)}{dt} \frac{\partial}{\partial x^i} \right)}$$

$$= \sqrt{\sum_{i=1}^{N} ||x_i'(t)||^2}. \quad (11)$$

Because $\gamma'(t)$ is a geodesic on $\mathcal{X}$, from Lemma 1, we know
$|\gamma'(t)|$ is a constant. Considering identity mapping $i$ is a
isometric immersion, we have

\[ \left| \frac{d}{dt} y'(t) \right| = |y'(t)|. \tag{12} \]

Thus, \( \sum \| x'(t) \|^2 \) is a constant, for \( \forall t \in I \).

Proof of Theorem 2.

Suppose the Riemannian metric on \( \mathcal{X} \) is induced from \( \mathbb{R}^N \) by identity mapping \( i : [x_1, x_2, \ldots, x_N] \) is the Cartesian coordinate system of \( \mathbb{R}^N \). From the definition of \( h \), we can get \( h(z_1, z_2, \ldots, z_m) = [x_1, x_2, \ldots, x_N] \). Suppose \( \frac{dh}{dt} \) is the \( j \)-th component of \( \frac{dh}{dt} \), according to the characteristic of \( di \), we can deduce the corresponding tangent vector of \( \frac{d}{dt} \) on \( T_{y(t)} \mathbb{R}^N \) as follows:

\[ di_y \left( \frac{\partial}{\partial z^i} \right) = \frac{\partial h'}{\partial z^i} \frac{\partial}{\partial x^j} y(t), \tag{13} \]

\[ \{ di_y \left( \frac{\partial}{\partial z^i} \right), 1 \leq i \leq \mathcal{N} \} \text{ spans a tangent space} \]

\( T_{y(t)}(\mathcal{X}) \) \( \forall t \). Suppose \( D \) is the Riemannian connection on \( \mathcal{X} \). According to the definition of \( D \)(\( y'(t) \)) (refer to Chapter 2, Eq. (3.1), in [29]), we have

\[ dD \left( di_y \left( \frac{\partial}{\partial z^i} \right) y'(t) \right) = D \left( di_y \left( \frac{\partial}{\partial z^i} \right) y'(t) \right) \]

\[ = \left( \frac{d\left( di_y \left( \frac{\partial}{\partial z^i} \right) y'(t) \right)}{dt} \right)^T. \tag{14} \]

The first equality holds by Remark 3.2 of Chapter 2 in [29]. \( \top \) denotes the orthogonal projection from \( \mathbb{R}^N \) to the tangent space on \( i(\mathcal{X}) \). Because Riemannian connection is invariant under isometric condition (see Theorem 4.6 of Chapter 2 in [29]), we have

\[ di_y \left( \frac{\partial}{\partial x^i} \right) = \left( \frac{d\left( di_y \left( \frac{\partial}{\partial z^i} \right) y'(t) \right)}{dt} \right)^T. \tag{15} \]

Combined with Eq. (10), we can deduce

\[ \left( \frac{d\left( di_y \left( \frac{\partial}{\partial z^i} \right) y'(t) \right)}{dt} \right)^T = \left( \frac{\partial^2 x_i}{dt^2} \frac{\partial}{\partial x^i} \right)^T. \tag{16} \]

Therefore, based on the definition of geodesic, \( \gamma : I \rightarrow \mathcal{X} \) is a geodesic on \( \mathcal{X} \), if and only if:

\[ \left( \frac{\partial^2 x_i}{dt^2} \frac{\partial}{\partial x^i} \right)^T = 0. \tag{17} \]

It means \( \frac{\partial^2 x_i}{dt^2} \frac{\partial}{\partial x^i} \) is orthogonal to tangent space \( T_{y(t)} \mathcal{X} \).

That is,

\[ \left( \frac{\partial^2 x_i}{dt^2} \frac{\partial}{\partial x^i}, \frac{\partial h'}{\partial x^j} \right) = 0, \quad 1 \leq i \leq m. \tag{18} \]

Eq. (18) is equivalent to

\[ \sum_{j=1}^N \frac{\partial^2 x_i}{dt^2} \frac{\partial h'}{\partial x^j} = 0, \quad 1 \leq i \leq m. \tag{19} \]

We write it in the form of matrix multiplication,

\[ \frac{\partial^2 x_i}{dt^2} \frac{\partial h'}{\partial x} y(t) = 0, \quad \forall t \in I. \tag{20} \]
C. Geng et al.: Geodesic Learning With Uniform Interpolation on Data Manifold

[27] K. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft, “When is ‘nearest neighbor’ meaningful?” in Proc. Int. Conf. Database Theory. Cham, Switzerland: Springer, 1999, pp. 217–235.

[28] H. Shao, A. Kumar, and P. T. Fletcher, “The Riemannian geometry of deep generative models,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW), Jun. 2018, pp. 428–4288.

[29] W. Chen and X. Li, Introduction to Riemann Geometry, vol. 1. Beijing, China: Peking University Press, 2004.

CONG GENG received the B.S. degree from the University of Electronic Science and Technology of China, China, in 2016. She is currently pursuing the Ph.D. degree in electrical engineering with the Institute of Image Communication and Network Engineering, Shanghai Jiao Tong University, Shanghai, China. Her research interests include manifold learning, generative model, and image processing.

JIA WANG received the B.Sc. degree in electronic engineering, the M.S. degree in pattern recognition and intelligence control, and the Ph.D. degree in electronic engineering from Shanghai Jiao Tong University, China, in 1997, 1999, and 2002, respectively. He is currently a Professor with the Department of Electronic Engineering, Shanghai Jiao Tong University, and also a member of the Shanghai Key Laboratory of Digital Media Processing and Transmission. His research interests include multiuser information theory and mathematics in artificial intelligence. He is an Associate Editor of Digital Signal Processing.

LI CHEN (Member, IEEE) received the B.S. and M.S. degrees in electrical engineering from Northwestern Polytechnical University, Xi’an, China, in 1998 and 2000, respectively, and the Ph.D. degree in electrical engineering from Shanghai Jiao Tong University, China, in 2006. He is currently an Associate Professor with the Department of Electronic Engineering, Shanghai Jiao Tong University. His current research interests include image and video processing, DSP, and VLSI for image and video processing.

ZHIYONG GAO (Member, IEEE) received the B.S. and M.S. degrees from the Changsha Institute of Technology, China, in 1981 and 1984, respectively, and the Ph.D. degree from Tsinghua University, Beijing, China, in 1989. From 1994 to 2010, he held several senior technical positions in U.K., including a Video Architect with 3DLabs, a Consultant Engineer with Sony European Semiconductor Design Center, and a Digital Video Architect with Imagination Technologies. Since 2010, he has been a Professor with Shanghai Jiao Tong University. His research interests include video processing, image and video coding, and computer vision.

---

VOLUME 10, 2022

98669